

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DO CONHECIMENTO**

Douglas Dyllon Jeronimo de Macedo

**UM MODELO DISTRIBUÍDO DE ARMAZENAMENTO
HIERÁRQUICO DE CONHECIMENTO MÉDICO**

Florianópolis (SC)

2014

Douglas Dyllon Jeronimo de Macedo

**UM MODELO DISTRIBUÍDO DE ARMAZENAMENTO
HIERÁRQUICO DE CONHECIMENTO MÉDICO**

Tese submetida ao Programa de Pós-Graduação
em Engenharia e Gestão do Conhecimento
para a obtenção do Grau de Doutor em En-
genharia do Conhecimento.

Orientador: Prof. Dr. Mário Antonio Ri-
beiro Dantas

Florianópolis (SC)

2014

Catálogo na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:

<http://www.bu.ufsc.br/design/Catalogacao.html>

Douglas Dyllon Jeronimo de Macedo

**UM MODELO DISTRIBUÍDO DE ARMAZENAMENTO
HIERÁRQUICO DE CONHECIMENTO MÉDICO**

Esta Tese foi julgada aprovada para a obtenção do Título de “Doutor em Engenharia do Conhecimento”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento.

Florianópolis (SC), 07 de fevereiro 2014.

Prof. Dr. Gregório Varvakis
Coordenador

Prof. Dr. Mário Antonio Ribeiro Dantas
Orientador

Banca Examinadora:

Prof. Dr. Mário Antonio Ribeiro Dantas
Presidente

Prof. Dr. rer. nat. Eros Comunello

Prof. Dr. João Bosco da Mota Alves

Prof. Dr. José Leomar Todesco

Prof. Dr. Nelson Francisco Favilla Ebecken

Profa. Dra. Aletéia Patrícia Favacho de Araújo

Para minha amada Mãe, Dona Graça
Para meu amado Pai, Seu Salvandir
Isto é por e para vocês.

AGRADECIMENTOS

Eu gostaria de agradecer primeiramente a Deus.

De antemão, gostaria de agradecer pelas preciosas considerações da banca examinadora desta defesa de tese, representada pelos Professores Doutores, Mario Antônio Ribeiro Dantas, José Leomar Todesco, Eros Comunello, João Bosco da Mota Alves, Nelson Francisco Favilla Ebecken e Aletéia Patrícia Favacho de Araújo.

Gostaria de deixar aqui gravado o meu mais sincero e respeitoso agradecimento ao meu amigo, Prof. Dr. Mario Antônio Ribeiro Dantas, pelas inúmeras reuniões de orientação, apoio e discussão, que sem elas não conseguiria ter dado andamento a este trabalho. Mas principalmente gostaria de agradecê-lo pela amizade e confiança que sempre fez questão de demonstrar.

Ainda, gostaria de agradecer ao Prof. Dr. José Leomar Todesco (o Tite), principalmente pelo apoio e confiança que sempre demonstrou. Lembrome bem, que em uma fase difícil de escolhas que tive que fazer em minha vida, tivemos uma longa conversa sobre as pesquisas que gostaria de fazer e de possíveis oportunidades que poderiam vir a surgir no futuro, e sinceramente, sem aquela conversa, hoje não estaria aqui.

Gostaria de agradecer a *The University of Western Ontario* (UWO), representada pela Profa. Dr. Miriam A. M. Capretz, pelo apoio dado no período que estive em London/ON. Foi um tempo de intenso aprendizado e de muito crescimento acadêmico e profissional. Hoje, me considero uma pessoa e um pesquisador melhor, por conta desta experiência. Ainda, gostaria de lembrar aqui do Prof. Dr. Michael Bauer, grande pessoa, a quem devo agradecimentos também pelo apoio.

Aos alunos e pesquisadores Thiago Coelho Prado (o Coeio), Marcus Vinicius Florêncio o Souza (o Marcão), Tiago Steinmetz Soares (o Tiago), Marcone Magnus (o Cone), Elias Amaral (o Elias) e Hilton William Ganzo Perantunes (o Paris) por terem auxiliado direta e indiretamente em várias fases desta pesquisa, onde compartilhamos angústias, expectativas, conquistas

e felicidades. Meus amigos, sem o apoio de vocês eu não teria conseguido chegar até aqui.

Ao Laboratório de Experimentação Remota (RExLab) da Universidade Federal de Santa Catarina, representado pelo Prof. Dr. João Bosco da Mota Alves, grande pessoa, que me ajudou imensamente na minha chegada em Florianópolis e sempre quando solicitado, me deu apoio sem restrições. Ao Prof. Dr. Luiz Fernando Jacinto Maia, um amigo, meu orientador do mestrado, cujo o tenho em alto estima, pois a confiança que depositou em mim, veio a me inspirar nesta longa jornada.

Ao Instituto Nacional de Convergência Digital (INCoD), ao Laboratório de Telemedicina (LabTelemed) e ao Grupo Cyclops, representados na pessoa do Prof. Dr. rer. nat. Aldo von Wangenheim, pelo auxílio em todas as etapas desta pesquisa, onde pude contar com o seu apoio incondicional, tanto financeiramente, profissionalmente e pessoalmente. Ainda, gostaria de agradecer o Prof. Dr. rer. nat. Eros Comunello pela amizade, profissionalismo e às preciosas discussões que tivemos em vários âmbitos, que sempre acrescentaram em minha formação. Aos amigos Harley Miguel Wagner, Rafael Andrade e Cleidson Cavalcante, pela confiança, apoio e suporte.

A Universidade Federal de Santa Catarina (UFSC) e ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento (PPGEGC), representados por todos os professores, por compartilharem seus conhecimentos e experiências conosco, sempre gerando ricos debates e profundos momentos de aquisição de conhecimento. Ainda gostaria de agradecer aos funcionários da secretaria, representados pelo Airton, Nathana e equipe, obrigado pelo apoio nesta longa caminhada.

À minha família, por sempre me proporcionarem o apoio necessário em todas as etapas de minha vida. A minha irmã Helen, pelo exemplo de vida, sempre focado na seriedade e compromisso, cujo me espelho todos os dias. A minha irmã Hemily, pela alegria, companheirismo, confiança, que me inspira sempre a ser uma pessoa melhor. E ao meu sobrinho, Guilherme, pela felicidade que me trás nos poucos momentos que podemos estar juntos. E aos meus amados pais, Seu Salvandir e Dona Graça, este trabalho é para vocês e por vocês. À minha amada namorada Veridiana, por seu apoio, confiança, compreensão, dedicação e por seu amor irrestrito. Seu suporte foi vital para chegar até aqui e serei eternamente grato por isto.

Deixo a todos aqui, o meu mais humilde, OBRIGADO!

“A day without laughter is a day wasted.”

Charles Chaplin

RESUMO

MACEDO, D. D. J. **UM MODELO DISTRIBUÍDO DE ARMAZENAMENTO HIERÁRQUICO DE CONHECIMENTO MÉDICO**, 2014. 214 f. Defesa de Tese (Doutorado em Engenharia e Gestão do Conhecimento). Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Departamento de Engenharia do Conhecimento, Universidade Federal de Santa Catarina (UFSC), Florianópolis, Brasil, 2014.

O advento da disseminação e consolidação dos sistemas computacionais, como ferramentas de apoio aos mais variados tipos de negócios, criou uma dependência dos usuários em geral, em armazenar mais e mais seus dados. Atualmente é inquestionável a necessidade de registros anteriores de transações dos usuários em sistemas computacionais. Esta necessidade vai, por exemplo, desde a recuperação de um e-mail de anos anteriores, passando por registros de transações bancárias legadas, indo até prontuários eletrônicos. Com esta crescente demanda de armazenamento de dados, informações e conhecimento, para os mais variados nichos da Indústria e da Ciência em geral, criou um interessante espaço para pesquisa e desenvolvimento de formas alternativas para a persistência de longo prazo em sistemas computacionais. Na área médica, esta premissa é ainda mais verdadeira, pois todos os dias, centenas de milhares de pacientes ao redor do mundo, realizam exames médicos baseados em imagens, buscando auxílio para o diagnóstico dos mais variados tipos de doenças. As formas de persistência de dados que atualmente são utilizadas na área médica, em sua grande maioria, se apoiam no paradigma relacional, que geralmente suportam os sistemas de informação e conhecimento que estão nos hospitais e clínicas médicas. Neste sentido, esta tese propôs um novo modelo de armazenamento para sistemas médicos baseados em imagens. Este modelo foi concebido a partir de um paradigma hierárquico, seguindo o padrão para imagens médicas internacionalmente reconhecido (DICOM), o que além de representar os dados de uma forma mais natural, simplifica o processo de busca e de recuperação. Ao final dos experimentos e das etapas de análises dos resultados, o modelo proposto mostrou ter um desempenho mais eficiente no geral, se comparado ao modelo atual, utilizando bancos de dados relacionais.

Palavras-chave: Armazenamento, HDF5, DICOM, Telemedicina.

ABSTRACT

MACEDO, D. D. J. **A Distributed Model of Hierarchical Storage Medical Knowledge**, 2014. 214 f. Defesa de Tese (Doutorado em Engenharia e Gestão do Conhecimento). Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Departamento de Engenharia do Conhecimento, Universidade Federal de Santa Catarina (UFSC), Florianópolis, Brasil, 2014.

The advent of the spread and consolidation of computer systems as tools supporting various kinds of business, created a dependence on general users to store more and more data. Currently there is an unquestionable need for previous records of user transactions in computer systems. This will need, for example, from the restore of an e-mail in previous years, going through records of bank transactions legacy, going to electronic records. With this increasing demand for data storage, information and knowledge for various niches of Science and Industry in general has created an interesting space for research and development of alternative forms for the persistence of long-term computer systems. In the medical field, this premise is even more true because every day, hundreds of thousands of patients around the world, conduct medical examinations based on images, seeking aid for the diagnosis of various types of diseases. Forms of data persistence that are currently used in the medical field, mostly, rely on the relational paradigm, which generally support information systems and knowledge that are in hospitals and medical clinics. In this sense, this thesis proposes a new storage model for systems-based medical images. This model was designed from a hierarchical paradigm, following the standard for internationally recognized medical images (DICOM), which also represents the data in a more natural, simplifies the process of search and retrieval. At the end of experiments and after the stages of analysis of the results, the proposed model has shown a better performance when compared to the current model, using relational databases.

Keywords: Storage, HDF5, DICOM, Telemedicine.

LISTA DE FIGURAS

Figura 1	Tela Inicial do STT da RCTM.	31
Figura 2	Exemplo de uma Imagem DICOM.	32
Figura 3	Modelo da Informação - DICOM.	33
Figura 4	Evolução da base de dados da RCTM 2006-2012.	35
Figura 5	Tabela com a quantidade de exames do STT (ANDRADE; WAGNER; WANGENHEIM, 2012).	50
Figura 6	Exemplo de uma Imagem DICOM.	52
Figura 7	Representação de um Elemento de Dados (NEMA, 2004b). . .	53
Figura 8	Representação Clássica de um SAD.	55
Figura 9	Representação Clássica de um SAD com <i>Middleware</i>	57
Figura 10	Arquitetura do ROMIO - Baseado em (THAKUR; GROPP; LUSK, 1999).	60
Figura 11	Arquitetura do CEPH (WEIL et al., 2006a).	62
Figura 12	Arquitetura do PVFS2 (KUNKEL; LUDWIG, 2007).	65
Figura 13	Arquitetura do Lustre (BRAAM et al., 2004).	66
Figura 14	Componentes do Lustre (WHANCLOUD, 2013).	67
Figura 15	Arquitetura do FhGFS (FRAUNHOFERFS, 2013).	68
Figura 16	Arquitetura do HDFS (BORTHAKUR, 2007)	70
Figura 17	Representação de um SAD com Bibliotecas de Altos Nível. . .	73
Figura 18	Representação de um <i>HDF5 Dataset</i> (HDFGROUP, 2013). . .	77
Figura 19	Representação de um <i>HDF5 Group</i> (HDFGROUP, 2013). . .	78
Figura 20	Arquitetura do HDF5 (FOLK et al., 2011).	80
Figura 21	Abstração da Arquitetura com a Camada de Indexação.	85
Figura 22	Arquitetura proposta em (MACEDO et al., 2009).	95
Figura 23	Arquitetura proposta em (SOARES et al., 2011).	96
Figura 24	Modelo Conceitual Atual do STT/RCTM.	100
Figura 25	Arquitetura Atual do STT/RCTM	102
Figura 26	Modelo Conceitual de Armazenamento Proposto.	105
Figura 27	Camada Virtual de Arquivos	107
Figura 28	Modelo de Dados Hierárquicos.	108
Figura 29	Extração dos Metadados e Geração do Índice.	109
Figura 30	Arquitetura Proposta.	111

Figura 31	WorkFlow da Arquitetura Proposta para o Modelo.....	114
Figura 32	Exemplo de uma Imagem DICOM Hierarquizada	116
Figura 33	Arquitetura Computacional do Cluster utilizado nos Testes...	119
Figura 34	Gráficos do Tempo Médio de Armazenamento Serial.	125
Figura 35	Gráficos do Tempo Total de Armazenamento Serial.....	126
Figura 36	Gráficos do Tempo Médio de Armazenamento Paralelo.	127
Figura 37	Gráficos do Tempo Total de Armazenamento Paralelo.	128
Figura 38	Gráficos do Tempo Médio de Recuperação Serial.....	129
Figura 39	Gráficos do Tempo Total de Recuperação Serial.	130
Figura 40	Gráficos do Tempo Médio de Recuperação Paralela.....	131
Figura 41	Gráficos do Tempo Total de Recuperação Paralela.....	132
Figura 42	Gráficos da Análise das Avaliações de Armazenamento de Dados.....	136
Figura 43	Gráficos Evolutivos das Avaliações de Armazenamento de Dados.....	138
Figura 44	Análise dos Experimentos de Armazenamento de Dados.	139
Figura 45	Gráficos da Análise das Avaliações de Recuperação de Dados.141	
Figura 46	Gráficos Evolutivos das Avaliações de Recuperação de Dados.143	
Figura 47	Análise dos Experimentos de Recuperação de Dados.	144
Figura 48	Análise Consolidada dos Experimentos.....	145

LISTA DE TABELAS

Tabela 1	Quantidade de Imagens e Tamanhos.	33
Tabela 2	Exemplos de Metadados DICOM.	54
Tabela 3	Comparação dos SAD Selecionados.	62
Tabela 4	Tipos de <i>Links</i> - HDF5.	79
Tabela 5	RSL – Artigos Encontrados nas Bases de Conhecimento.	91
Tabela 6	Configurações dos Computadores utilizados nos Testes.	120
Tabela 7	Composição da Base de Dados.	122
Tabela 8	Tamanho da Base de Dados.	123
Tabela 9	Resumo dos Experimentos.	124

LISTA DE ABREVIATURAS E SIGLAS

HIS	Hospital Information System	28
RIS	Radiology Information System	28
PACS	Picture Archiving and Communication System	28
STT/SC	Sistema Integrado de Telemedicina e Telessaúde.	29
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico	29
HL7	Health Level Seven	30
DICOM	Digital Imaging and Communications in Medicine.	30
SGBC	Sistema Gerenciador de Bancos de Dados.	33
CFM	Conselho Federal de Medicina	33
GPL	General Public License	39
PPGEGC	Pós-Graduação em Engenharia e Gestão do Conhecimento . .	40
PPGGEP	Programa de Pós-Graduação em Engenharia de Produção . . .	40
UFSC	Universidade Federal de Santa Catarina	40
HDF5	Hierarchical Data Format	44
HDF5	Hierarchical Data Format	47
STT/SC	Sistema Integrado de Telemedicina e Telessaúde.	49
ACR	American College of Radiology	51
NEMA	National Electrical Manufacturers Association	51
IOD	Information Object Definition	52
SAD	Sistema de Arquivos Distribuído	55
API	Application Programming Interface	58
PAD	Parallel File Systems	58
MPI	Message Passing Interface	58
PVM	Parallel Virtual Machine	58
ADIO	Abstract-Device Interface IO	59
MDS	Metadata Servers	62
ODS	Object Storage Devices	62
CRUSH	Controlled Replication Under Scalable Hashing	63
PVFS	Parallel Virtual File System	64
PVM	Parallel Virtual Machine	64
MPI-IO	Message Passing Interface – Input/Output.	64
FhGFS	The Fraunhofer Parallel Cluster File System	67

TCP	Transmission Control Protocol.....	69
HDFS	Hadoop File System.....	69
RAID	Redundant Array of Independent Disks.....	70
ODBC	Open Database Connectivity.....	80
RSL	Revisão Sistemática da Literatura.....	87
SLR	Systematic Literature Review.....	87
NFS	Network File System.....	96
RCTM	Rede Catarinense de Telemedicina.....	165

SUMÁRIO

1 INTRODUÇÃO	27
1.1 CONTEXTUALIZAÇÃO	27
1.2 PROBLEMA DE PESQUISA	34
1.2.1 Pergunta de Pesquisa	36
1.3 OBJETIVOS	36
1.3.1 Objetivo Geral	36
1.3.2 Objetivos Específicos	36
1.4 JUSTIFICATIVA E RELEVÂNCIA	37
1.5 DELIMITAÇÕES DA PESQUISA	38
1.6 INEDITISMO DA PROPOSTA	39
1.7 ADERÊNCIA DO TEMA AO PPEGC	40
1.8 METODOLOGIA	43
1.9 ESTRUTURA DO TRABALHO	44
2 FUNDAMENTAÇÃO TEÓRICA	47
2.1 TELEMEDICINA	47
2.1.1 STT	49
2.2 DICOM	50
2.3 SISTEMAS DE ARQUIVOS DISTRIBUÍDOS	55
2.3.1 Messaging Passing Interface	58
2.3.2 Análise dos Sistemas de Arquivos Distribuídos	60
2.3.3 CEPH	62
2.3.4 PVFS	64
2.3.5 Lustre	65
2.3.6 FhGFS - The Fraunhofer Parallel Cluster File System	67
2.3.7 HDFS - Hadoop File System	69
2.4 FORMATOS DE DADOS	71
2.4.1 Hierarchical Data Format - HDF5	75
2.4.2 Estrutura do HDF5	76
2.4.2.1 HDF5 Datasets	76
2.4.2.2 HDF5 Groups	77
2.4.2.3 HDF5 Datatypes	78
2.4.2.4 HDF5 Links	79
2.4.2.5 HDF5 Attributes	79
2.4.3 Arquitetura do HDF5	80
2.4.4 Desempenho	82
2.4.5 Segurança e Consistência	83
2.5 INDEXAÇÃO	83

2.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO	86
3	ESTADO DA ARTE	87
3.1	REVISÃO SISTEMÁTICA DA LITERATURA	87
3.1.1	Contextualização da Pesquisa	87
3.1.2	Questões Específicas da Pesquisa	88
3.1.3	Definições de Bases de Dados e Termos de Pesquisa	88
3.1.4	Procedimentos e Critérios de Seleção dos Estudos	89
3.1.5	Procedimentos para Avaliação da Qualidade dos Estudos	89
3.1.6	Estratégia para Extração e Sintetização dos Dados	90
3.1.7	Identificação dos Estudos Relevantes	90
3.1.8	Estudos Relevantes Sintetizados	90
3.2	TRABALHOS RELACIONADOS	92
3.3	TRABALHOS REALIZADOS	95
3.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO	97
4	MODELO PROPOSTO	99
4.1	MODELO CONCEITUAL DO STT/RCTM	99
4.2	ARQUITETURA DO STT/RCTM	101
4.3	MODELO PROPOSTO	104
4.3.1	Procedimentos Metodológicos	104
4.3.2	Modelo Conceitual	104
4.3.3	Modelo de Dados e Indexação	107
4.4	ARQUITETURA PROPOSTA	110
4.4.1	Workflow da Arquitetura Proposta	114
4.4.2	Design do Arquivo no Formato HDF5	115
4.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO	115
5	AMBIENTE E RESULTADOS EXPERIMENTAIS	119
5.1	ARQUITETURA COMPUTACIONAL	119
5.2	METODOLOGIA DOS EXPERIMENTOS	121
5.3	EXPERIMENTOS REALIZADOS	123
5.3.1	Experimento 01: Armazenamento Serial	125
5.3.2	Experimento 02: Armazenamento Paralelo	127
5.3.3	Experimento 03: Recuperação Serial	129
5.3.4	Experimento 04: Recuperação Paralela	131
5.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO	133
6	ANÁLISES E DISCUSSÕES DOS EXPERIMENTOS	135
6.1	ANÁLISE	135
6.2	ANÁLISE DAS AVALIAÇÕES DE ARMAZENAMENTO DE DADOS	136
6.2.1	Análise do Armazenamento	136
6.2.2	Tabela da Análise do Armazenamento	138
6.3	ANÁLISE DAS AVALIAÇÕES DE RECUPERAÇÃO DE DADOS	140

6.3.1	Análise da Recuperação	140
6.3.2	Tabela da Análise da Recuperação	143
6.4	DISCUSSÕES FINAIS DOS EXPERIMENTOS	145
6.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO	148
7	CONCLUSÕES E TRABALHOS FUTUROS	149
7.1	TRABALHOS FUTUROS	150
	Referências Bibliográficas	153
	APÊNDICE A – Motivação Pessoal	165
	APÊNDICE B – Publicações Relacionadas	169
	APÊNDICE C – Tabelas - Armazenamento Paralelo	175
	APÊNDICE D – Tabelas - Armazenamento Serial	185
	APÊNDICE E – Tabelas - Recuperação Paralela	195
	APÊNDICE F – Tabelas - Recuperação Serial	205

1 INTRODUÇÃO

Este capítulo está estruturado em 8 seções. Na primeira seção será apresentada uma contextualização, que tem a função de embasar os preceitos técnicos e teóricos da pesquisa. Na segunda seção são apresentados o problema de pesquisa e as perguntas de pesquisas que guiarão o desenvolvimento desta tese. Após, na seção três (3), o objetivo geral e os objetivos específicos desta tese serão descritos. Na seção quatro (4), será mostrada a justificativa para o desenvolvimento deste modelo e a relevância científica do tema de armazenamento de dados neste contexto. Na sequência, na seção cinco (5) serão descritas as fronteiras do tema, falando das delimitações da pesquisa. Na seção seis (6), é dissertado sobre o ineditismo da proposta. Na seção sete (7), o alinhamento do tema da tese, com seu programa de Pós-Graduação é traçado. Por fim, na seção oito (8), a estrutura deste trabalho é descrita, buscando antecipar o que será visto em cada capítulo.

1.1 CONTEXTUALIZAÇÃO

Em 1895, o físico alemão Wilhelm Conrad Röntgen (W.C. Röntgen) descobriu a possibilidade da criação de radiação eletromagnética nos comprimentos de onda, atualmente conhecido como Raios-X, e desde então, a área de medicina sofreu uma reviravolta. Em 1901, em reconhecimento aos extraordinários serviços que a descoberta dos raios possibilitaram, ele ganhou o primeiro Prêmio Nobel de Física da história.

Desde então, as imagens médicas tem contribuído significativamente com o progresso da medicina. Várias modalidades de exames médicos baseados em imagens foram desenvolvidas nos últimos 50 anos, incluindo tomografia computadorizada, cintilografia, ultrassonografia, ressonância magnética e radiografia digital. Desta forma, o diagnóstico médico baseado em imagens cresceu da infância até os mais elevados níveis de maturidade (DOI, 2006).

Entretanto, os modernos sistemas de diagnóstico baseados em imagens, não se resumem somente a equipamentos de última geração, que de fato, produzem a imagem. Há-se ainda a necessidade de componentes que façam o processamento, a visualização, a gravação, o **armazenamento**, a **recuperação** e a transmissão das imagens produzidas pelos equipamentos (DOI, 2006). Dito isto, é possível inferir que a “produção” da imagem é somente um dos muitos componentes que devem ser pesquisados, para o desenvolvimento de ciência e tecnologia no desenvolvimento dos diagnósticos médicos baseados em imagens (WETERING; BATENBURG, 2009).

Para que as imagens produzidas sejam usadas nas instituições de saúde, atualmente é natural utilizar arquiteturas compostas de *hardware* e *software* para auxiliar nas tarefas dos componentes já explanados. Estas arquiteturas têm a função, em suma, de facilitar a aquisição, o processamento e o armazenamento das imagens e dos metadados gerados pelos equipamentos. Estas arquiteturas, podem ser resumidas em: *Hospital Information System (HIS)*, *Radiology Information System (RIS)* e *Picture Archiving and Communication System (PACS)*.

Os conceitos históricos imputados aos HIS surgiram no final da década de 60 (BAKKER, 1991). As motivações para sua criação foram originadas das aplicações hospitalares que eram desenvolvidas na época, que eram totalmente isoladas, não havendo uma interação entre os setores dos hospitais. De acordo com Collen (1988), o objetivo de um HIS é usar computadores e equipamentos de comunicação para coletar, armazenar, recuperar, processar e comunicar dados de pacientes e informações administrativas para todos os setores do hospitais, satisfazendo todos os requisitos funcionais para os usuários que forem autorizados a usar às informações (COLLEN, 1988).

O conceito de RIS surgiu, de acordo com Collen (1988), talvez por conta do baixo suporte da parte dos HIS, aos setores de radiologia dos hospitais. Ele ainda cita, que não há uma diferença principal que difira os dois conceitos, apenas há um limiar que os RIS são específicos para os setores de radiologia, enquanto os HIS procuram cobrir todos os setores do hospitais, inclusive o de radiologia (COLLEN, 1988). Desta forma, é usual que os RIS tenham integração com os HIS, para fins de interoperabilidade. Ainda, é comum que dentro dos HIS, aja subsistemas complexos para tratarem os problemas particulares dos setores de radiologia.

O conceito de PACS surgiu há mais de três décadas, no anos 80 (WETERING; BATENBURG, 2009), e de acordo com Lodwick (1986) pode ser descrito como um componente de software que é capaz de armazenar imagens digitais produzidas por diferentes modalidades em um banco de dados de imagens centralizado. Ainda, um PACS deve ser capaz de proporcionar facilidades de processamento de imagens e visualização para os usuários. Também, deve ser possível uma rápida transmissão entre o aparelho que esta adquirindo a imagem e a base de dados de imagens (LODWICK, 1986).

Dito isto, é possível inferir que a relação entre os HIS/RIS e os PACS dá-se enquanto os HIS/RIS, são responsáveis pelas informações textuais e gráficas da instituição de saúde, que vão desde dados de pacientes, médicos e enfermeiros, até informações administrativas de toda a instituição de saúde, seja ela um grande hospital, ou até menores clinicas. Enquanto os PACS, são responsáveis pela organização, aquisição, armazenamento e recuperação de imagens, advindas de equipamentos médicos hospitalares.

Uma outra classificação/modelo de sistemas de apoio ao diagnóstico médico é a Telemedicina. Esta pode ser definida como o uso de telecomunicações e tecnologias de informação, para compartilhar e manter informações de saúde do paciente e para prestar cuidados clínicos e de educação em saúde para pacientes e profissionais, quando a distância separa os participantes do processo (BASHSHUR, 1995).

A Telemedicina vem sendo utilizada desde 1959 (CABRAL J.E.; KIM, 1996) quando experimentos envolvendo telepsiquiatria (WITTON; AFFLECK; JOHNSON, 1961) e telefluoroscopia (JUTRAS, 1959) e novos procedimentos médicos foram mostrados em transmissões ao vivo em conferências médicas. Na década de 1970 e 1980, experimentos em telemedicina foram focados na transmissão de imagens médicas usando a televisão, via *broadcast* ou usando fitas gravadas. Nos anos 90, ocorreram experimentos voltados a telemedicina via vídeo-conferência, para realização de consultas médicas a distância (CABRAL J.E.; KIM, 1996).

A partir dos anos 90, até os dias de hoje, os sistemas de telemedicina se popularizaram pelo mundo. Os fatores que culminaram nesta disseminação foram a maturidade que as redes de comunicação ao redor do mundo atingiram e ao poder computacional que atualmente pode-se adquirir. Estes dois adventos, somados ao fato da modernidade das linguagens de programação, e a alta tecnologia aplicada em equipamentos médico hospitalares, resultaram em uma equação de sucesso para a implantação deste tipo de modelo.

Atualmente, é possível encontrar projetos consolidados da aplicação de Telemedicina, isoladamente, ou unido a sistemas PACS, em várias partes do mundo, como por exemplo na Inglaterra (MAY et al., 2000), Alemanha (THIELSCHER; DOARN, 2008), (DUENNEBEIL et al., 2010), China (WANG; GU, 2009), (XUE; LIANG, 2007), (HSIEH et al., 2001), Paquistão (MALIK, 2007), Índia (PAL et al., 2005), entre outros países.

Na América Latina, a principal iniciativa neste sentido é o Sistema Integrado de Telemedicina e Telessaúde (STT), integrante da Rede Catarinense de Telemedicina (WANGENHEIM et al., 2009), (WALLAUER et al., 2008), (MAIA; WANGENHEIM; NOBRE, 2006). Esta infraestrutura de serviços de Telemedicina e Telessaúde, aplicada inicialmente em Santa Catarina, Brasil, teve seu marco inicial em 1999, com um projeto de aplicação de algumas tecnologias que estavam naquele momento sendo desenvolvidas pelo Cyclops Group (GROUP, 2013), da UFSC. Estas tecnologias estavam sendo desenvolvidas em parceria com instituições da Alemanha e ainda contava com financiamento do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e da Fundação Volkswagen (Wolfsburg/German).

Depois da etapa de análise e desenvolvimento, no ano de 2002 o projeto piloto desta rede iniciou a operação, fazendo a ligação em pequenas

clínicas médicas dos municípios de Florianópolis e São José. Após isto, no ano de 2004, baseados no sucesso da aplicação da tecnologia nas clínicas, o Governo do Estado de Santa Catarina, juntamente com o Projeto Cyclops, iniciaram a análise para implementação de um piloto em nível estadual, para ser implementado no primeiro semestre do 2005. Este piloto tinha o intuito de demonstrar em números, o quão positivo era a aplicação de serviços de Telemedicina Assíncrona, no contexto do Sistema Único de Saúde (SUS).

Esse projeto piloto então teve início em dois municípios, sendo eles, Quilombo e São Lourenço do Oeste, e eles formaram o marco inicial da RCTM (Rede Catarinense de Telemedicina). Devido aos bons resultados obtidos, logo após, mais 3 pontos de presença foram implantados, sendo oferecido primeiramente laudos à distância, remessa eletrônica de resultados e armazenamento de imagens nas modalidades de Eletrocardiograma, Tomografia Computadorizada e Ultrassonografia. Deste então, esta rede vem sendo maximizada, já ultrapassando mais de 2.500.000 (dois milhões e quinhentos mil) de exames realizados. No Capítulo 2, sobre a Fundamentação Teórica, a arquitetura da RCTM será descrita em maiores detalhes, juntamente com as modalidades atendidas e os serviços oferecidos. Uma imagem do sistema pode ser vista na Figura 1.

A organização das informações, tanto texto quanto imagens, dentro destes modelos, é feita por padrões reconhecidos e amplamente utilizados. Para os HIS/RIS, o padrão usado é o *Heath Level Seven* (HL7), que é caracterizado por ser composto por um conjunto de padrões para troca e integração de informações médicas (DOLIN et al., 2001) (DOLIN et al., 2006). No caso da Telemedicina e dos PACS, o padrão utilizado é o *Digital Imaging and Communications in Medicine* (DICOM), que visa reduzir custos (HUANG, 2011) e organizar o *workflow* de uma instituição de saúde com base em suas imagens digitais, visando sempre a interoperabilidade, quando haja atividades de aquisição, armazenamento, visualização, processamento, recebimento, envio, busca e impressão de exames (ASSOCIATION, 1993) (MILDENBERGER; EICHELBERG; MARTIN, 2002). Alguns exemplos de uma imagens DICOM podem ser vistas a seguir, na Figura 2.

Entretanto, como o nome diz, uma “imagem” DICOM, não é composta apenas pela “figura” explicitada. Uma imagem DICOM pode ser composta por mais de 2000 metadados, que vão desde a identificação do paciente (*Patient ID*), até por exemplo o laudo dado para tal imagem. Desta forma, a complexidade do gerenciamento destas imagens aumenta significativamente, pois o que será armazenamento não é somente uma figura, mas sim um conjunto de informações (metadados) de identificação e composição da imagem, juntamente com um *pixeldata* (a figura).

Para efeitos práticos, quando um paciente realiza um exame médico



Figura 1: Tela Inicial do STT da RCTM.

baseado em imagem no formato DICOM, este exame é segmentado hierarquicamente, onde é realizado um estudo (*study*), que contém uma ou mais séries (*series*), com uma ou mais imagens (*images*) embutidas. Na Figura 3 é possível visualizar um exemplo de operação e o modelo de informação, nativamente hierárquico do DICOM.

Em relação a forma com que os dados são armazenados, isoladamente, estes dados não significam nada. Porém, automaticamente quando estes dados são hierarquizados gera-se uma informação acerca do paciente. Neste caso, pode-se inferir, por exemplo, que o paciente “X”, com “Y” endereço, realizou um estudo, com uma ou mais séries, que contém uma ou mais imagens. Isto é só informação, sem nenhum conhecimento relacionado. Entretanto, há uma atividade intensiva em conhecimento, por parte do médico, quando este efetua a análise destes estudos, séries e imagens, para e realização de um laudo. Este laudo é um dos metadados do DICOM e também esta contido dentro das imagens. Desta forma, é intrínseco que há um conhecimento embutido em cada uma dessas imagens.

Em relação ao tamanho e a quantidade de imagens por estudo e a

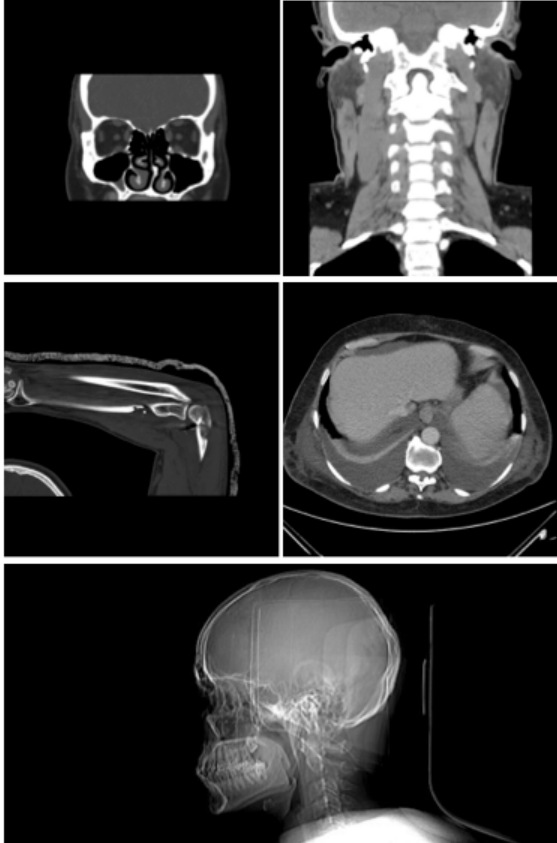


Figura 2: Exemplo de uma Imagem DICOM.

quantidade de espaço utilizada por estas imagens, na Tabela 1 é possível visualizar esta informação. São dados os exemplos em relação à Radiologia Computadorizada, Ressonância Magnética, Tomografia Computadorizada e Ultrassom. A informação foi extraída das bases de dados de produção do STT/RCTM e foi aplicada média aritmética para ilustrar os dados.

Com relação ao cerne desta pesquisa – o armazenamento e recuperação, o protocolo DICOM define apenas as diretrizes para formatos de arquivos, perfis para aplicação, modelos de dados e mídias físicas (NEMA, 2011). A especificação das formas de armazenamento não são definidas, ficando a cargo de cada desenvolvedor escolher a melhor forma de persistência para sua aplicação. A falta deste padrão, culminou no surgimento de várias alternati-

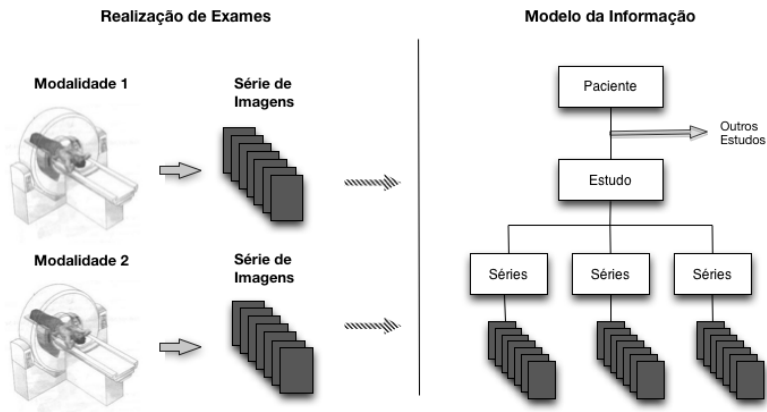


Figura 3: Modelo da Informação - DICOM.

Modalidade	Quant. de Imagens	Tamanho (Mb)
Radiologia Computadorizada	1.6	16
Ressonância Magnética	325.6	5
Tomografia Computadorizada	163.3	87
Ultrassom	7.2	6

Tabela 1: Quantidade de Imagens e Tamanhos.

vas de armazenamento, que vão desde armazenamento simples em sistemas de arquivos, passando pelos tradicionais Sistemas de Gerenciadores de Bancos de Dados (SGBD), até complexos sistemas de armazenamento baseados em outros paradigmas.

Entretanto, a definição da plataforma de armazenamento deve ser feita com extremo cuidado, pois este tipo de banco de dados ultrapassam facilmente a barreira dos centenas de *terabytes*. Há casos, não muito raros, onde estas bases de dados ultrapassam a barreira dos *petabytes* (JEPSEN, 2003). Isto acontece porque além do volume natural na quantidade de exames que são realizados, ainda há regulações ao redor do mundo para a persistência destes exames.

No Brasil, por exemplo, o Conselho Federal de Medicina (CFM) determina que o tempo que um exame deve ser armazenado, independente da modalidade, é de 20 anos (CFM: Conselho Federal de Medicina, 2007). Somado ao fato de que estas imagens não devem possuir nenhum tipo de perda de qualidade, visto que isto poderia induzir a um diagnóstico equivocado. No

exterior, por exemplo na Inglaterra, o período de tempo que um exame deve ser armazenado é de 9 a 15 anos, dependendo da modalidade (RADIOLOGISTS, 2008). Na Alemanha, é de 30 anos para radiografia (BMJ, 1987) e nos Estados Unidos, varia de 5 a 10 anos, sendo que este tempo pode ser modificado, se houver um requerimento do governo (SICKLES; WOLVERTON; DEE, 2002).

À primeira vista, estas regulações podem parecer um pouco “exageradas”, pelo fato dos recursos computacionais e diretamente, os custos a eles relacionados, que devem ser aplicados para suportá-las. Entretanto, a possibilidade de acesso a exames que fazem parte do histórico de um paciente é vital para o acompanhamento de um tratamento, para checar a evolução do caso e ainda, isso pode influenciar diretamente no pedido de novos exames. Um exemplo que pode ser dado é se um paciente já realizou exames que envolvem a irradiação para aquisição de uma imagem, ou seja, o paciente foi exposto a radiação com Raios-X, é vital que novos pedidos de exames, desta natureza, não sejam autorizados a fim de evitar exposições desnecessárias aos pacientes (RON, 2003).

Por fim, dentro dos sistemas médicos, sejam eles HIS, RIS, PACS ou de Telemedicina há vários dados e informações armazenadas, que vão deste registros de data de nascimento de pacientes, passando por informações diversas sobre os equipamentos e até informações sobre a natureza do exame. Entretanto, um fator de extrema importância, que raramente é citado é o conhecimento que há embutido neste tipo de sistema. Todas às vezes que exames são realizados e que médicos especialistas efetuam análises – baseadas em sua experiência, e traçam diagnósticos, dentro do sistema fica gravado o resultado de uma tarefa intensiva em conhecimento e que pode ser usada no futuro em vários cenários, como por exemplo *Knowledge Discovery Textual Databases*, *Knowledge Discovery in Databases* ou *Data Mining*, para diferentes propósitos, como por exemplo para busca de estudo de casos correlatos ou ainda para fins educacionais, entre outros.

1.2 PROBLEMA DE PESQUISA

A partir do contexto apresentado, nota-se um interessante problema de pesquisa, visto que dia-a-dia os sistemas médicos recebem mais e mais exames, isto deve-se ao crescimento natural da população ou ainda, pelo fato de que à acessibilidade a cuidados médicos têm aumentado nas últimas décadas, com novos equipamentos, tratamentos e metodologias. Consequentemente, as bases de dados desse tipo de serviço crescem exponencialmente, podendo ultrapassar as barreiras dos *petabytes*. Este crescimento esta diretamente re-

lacionado com a manutenção e a gestão do sistema, na qualidade do serviço e na experiência que o usuário terá na utilização de tais plataformas.

Em relação à manutenção e à gestão de sistemas hospitalares, este fator está ligado diretamente na escalabilidade de tais serviços, pois a medida que as bases crescem, a complexidade inerente a este evento está intrínseca. Ainda sobre manutenção, há um processo de *backup* dos dados, que é tão complexo quanto os quesitos de escalabilidade. Ainda, há um desafio relacionado a este advento, que são as taxas de *uptime* destes serviços, pois por lhe darem diretamente com apoio a vida, sua não disponibilidade pode causar a morte de pacientes. Quanto a qualidade do serviço, isto está ligado diretamente ao desempenho do mesmo, pois sistemas com uma baixa *performance* são difíceis de serem aceitos pelos usuários, pois prejudica a experiência do mesmo na utilização do serviço.

Na Figura 4 pode-se ver uma evolução no crescimento da base de dados DICOM do STT/RCTM do ano de 2006 até 2012¹. Como já foi salientado, os exames suportados por esta rede, já passaram da barreira dos 2.500.000 (dois milhões e quinhentos mil) exames e a base de dados DICOM já ultrapassou a barreira dos 10.000.000 (dez milhões de imagens) e isso só tende a crescer.

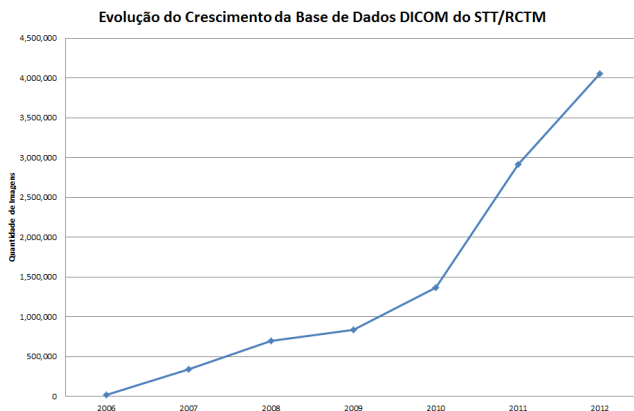


Figura 4: Evolução da base de dados da RCTM 2006-2012.

Desta maneira, a persistência de dados de longo prazo tem se tornado um problema real a ser enfrentado, pois com as tecnologias existentes

¹Os dados relativos ao ano de 2013 ainda não foram completamente consolidados, desta forma, pode haver uma divergência nos números. O dado em questão é relativo até o mês de Dezembro de 2012. Isto será atualizado assim que os dados forem sintetizados

atualmente, o desempenho e a escalabilidade desta Rede tem sofrido perdas contundentes. Uma importante informação é que esta pesquisa não tem a intenção de apontar que as tecnologias atuais não são boas o suficiente. Afirma-se com este trabalho que a atividade de pesquisa sobre o tema é vital e o desenvolvimento de modelos alternativos de armazenamento são necessários. O exemplo dado desta Rede em específico é só um, entretanto, na literatura pode-se ver claramente os esforços de pesquisadores pelo Mundo, buscando alternativas viáveis, que vão deste *tunning* dos SGBDs tradicionais, passando por otimizações em sistemas de arquivos, usando índices dos mais variados tipos, até aplicações usando o paradigma NoSQL.

1.2.1 Pergunta de Pesquisa

Apresentados os problemas de pesquisa citados na seção anterior, a seguinte pergunta de pesquisa deve ser levantada:

“A aplicação de uma camada de persistência hierárquica e distribuída é viável para sistemas médicos baseados em imagens?”

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Propor um modelo hierárquico de armazenamento e recuperação de dados, informações e conhecimentos médicos, de forma paralela e distribuída, para Sistemas de Telemedicina.

1.3.2 Objetivos Específicos

1. Identificar os requisitos para elaboração de um modelo hierárquico de armazenamento de dados para imagens médicas DICOM.
2. Identificar plataformas de sistemas de arquivos distribuídos e paralelos que se adequem ao modelo proposto, visando a manutenção de longo prazo de forma distribuída e que seja de fácil acesso.
3. Propor um modelo hierárquico de armazenamento e recuperação empregando o formato de dados HDF5, integrado com um servidor de imagens médicas DICOM.
4. Demonstrar a viabilidade do modelo proposto por meio de imple-

mentação de um protótipo integrado com a infraestrutura do estudo de caso, neste caso a Rede Catarinense de Telemedicina, substituindo assim o SGBD.

5. Realizar uma análise comparativa usando o modelo proposto nas plataformas de sistemas de arquivos distribuídos identificadas, por meio de uma análise de desempenho buscando identificar qual se adapta melhor para o modelo proposto.

1.4 JUSTIFICATIVA E RELEVÂNCIA

Atualmente, grande parte do que é visto, lido, ouvido, escrito e medido, pode ser coletado e disponibilizado em sistemas computacionais. Para obter efetividade e eficiência neste sistemas, é fundamental criar soluções escaláveis, que possam responder às demandas de desenvolvimento de aplicações com esses dados. Tais aplicações visam desenvolver soluções para o tratamento, a recuperação e a disseminação de informação e/ou conhecimento, de natureza tanto narrativa quanto descritiva, a partir de volumes exponencialmente crescentes de dados multimídia (CARVALHO; LEON et al., 2006).

Desta forma, a justificativa para o desenvolvimento desta pesquisa se dá pela importância do tema citado para as mais variadas áreas, inclusive a medicina. O tema armazenamento de dados é visto como um dos pilares para o crescimento e o desenvolvimento de diversos setores básicos, como por exemplo educação, saúde e finanças. Setores como os citados, necessitam de ferramentas que auxiliem a boa gestão dos dados, informações e conhecimentos que seus sistemas visem armazenar. Neste caso, estes modelos são criados a partir de necessidades que venham a surgir, geralmente por demandas relacionadas ao desempenho, a escalabilidade e a disponibilidade. Neste sentido, para que aja a criação de novos modelos, que derivarão ferramentas, que irão compor soluções em geral, é necessário um esforço em pesquisa e no desenvolvimento de novos modelos, métodos e metodologias que busquem sanar as demandas do mercado.

A relevância do tema armazenamento de dados, ou tópicos relacionados, são citados em vários importantes relatórios técnicos de comunidades científicas pelo mundo, como por exemplo *The Grand Challenges in Global Health* (HEALTH, 2013), *Grand Research Challenges in Computing*, “*Grandes Desafios da Pesquisa em Computação no Brasil*” (CARVALHO; LEON et al., 2006) (MEDEIROS, 2008).

No caso da iniciativa desenvolvida no Brasil em 2006 (CARVALHO; LEON et al., 2006) (MEDEIROS, 2008), foram propostos cinco (5) grandes desafios a serem enfrentados nos próximos 10 anos, a partir da data de publicação do relatório, no caso 2006–2016. Para a projeção destes desafios,

uma comissão analisou 47 propostas, que foram enviadas de todo o Brasil e consolidaram os resultados nos seguintes tópicos:

1. Gestão da Informação em grandes volumes de dados multimídia distribuídos;
2. Modelagem computacional de sistemas complexos artificiais, naturais e sócio- culturais e da interação homem-natureza;
3. Impactos para a área da computação da transição do silício para novas tecnologias;
4. Acesso participativo e universal do cidadão brasileiro ao conhecimento;
5. Desenvolvimento tecnológico de qualidade: sistemas disponíveis, corretos, seguros, escaláveis, persistentes e ubíquos.

Como primeiro item da lista, a gestão da informação (ou Gestão do Conhecimento, dependendo do contexto) em grandes volumes de dados multimídia distribuídos foi citada. Este primeiro desafio vai diretamente ao encontro do cerne desta pesquisa, a qual visa desenvolver um modelo de armazenamento distribuído para grandes volumes de dados multimídia, voltados para sistemas de Telemedicina.

Ainda, como justificado para o desenvolvimento desta pesquisa, agora no enfoque da medicina, no ano de 2007 o CFM (Conselho Federal de Medicina) determinou que a disponibilidade de acesso de um exame médico é de no mínimo 20 anos (CFM: Conselho Federal de Medicina, 2007). Esta determinação por parte do CFM tem impacto direto sobre armazenamento dos dados. Resumidamente, um hospital, clínica ou instituições de saúde em geral, que realizam exames médicos baseados em imagens, são responsáveis por mantê-lo disponível por um período de, no mínimo, 20 anos.

Dito isto, a relevância do tema é vital para o bom desenvolvimento do setor de saúde no Brasil e no Mundo, visto que se escalado isto, para dezenas ou centenas de pacientes, que realizam anualmente centenas ou milhares de exames, que são compostos por milhares ou milhões de imagens, tem-se um cenário altamente complexo, onde o crescimento exponencial das bases de dados deve ser um dos principais problemas na arquitetura computacional para sistemas desta natureza.

1.5 DELIMITAÇÕES DA PESQUISA

A proposta de modelo de armazenamento exposto nesta tese, não tem a intenção de explorar questões relacionadas ao *design* de interação entre o

usuário e os sistemas, visto que no modelo não há interação direta com o usuário final, e sim com os engenheiros e os administradores que tem a função de implantar e manter o sistema de persistência. Dessa forma, este trabalho não contemplará visões qualitativas de sua implementação em larga escala, devido ao tempo que seria necessário para esta avaliação.

O modelo a ser apresentado, pressupõe que aja uma infraestrutura computacional e de comunicação que seja confiável, pois neste trabalho, não serão tratados temas como tolerância a faltas, ou controle de congestionamento, tanto dos sistemas, como dos meios de comunicações envolvidos. Esta função será deixada para os provedores de serviços e para os sistemas de arquivos distribuídos que serão integrados e avaliados, juntamente com o modelo proposto.

Nesta pesquisa, não há o objetivo de desenvolver uma plataforma completa em sua versão final, desenhada na forma de produto para distribuição de propósito geral. Neste sentido, será desenvolvido um protótipo que terá a função de provar a hipótese proposta, respondendo aos objetivos traçados. Este protótipo será disponibilizado sob a licença General Public License (GPL) para toda a comunidade de software, para apreciação, discussão, melhorias, para a partir disto, gerar uma versão final, na forma de produto.

Devido a grande disseminação e uso de sistemas gerenciadores de bancos de dados relacionais, e sua aplicação em massa em sistemas de medicina em geral, esta tese não tem a pretensão de substituir imediatamente estes sistemas de persistência. O que se espera, é que quando engenheiros e desenvolvedores, deste tipo de sistema, se deparem com os problemas de pesquisa acima relacionados, eles busquem este modelo como uma das alternativas possíveis para melhoria dos tempos de resposta dos sistemas.

1.6 INEDITISMO DA PROPOSTA

Para verificar a originalidade do modelo, ou ainda, o ineditismo do mesmo, no sentido de averiguar trabalhos relacionados ou similares ao proposto, foi realizada uma Revisão Sistemática da Literatura (RSL), onde foram realizadas pesquisas bibliográficas que abordassem o tema sistemas médicos, vinculados com sistemas de persistência de dados, nas seguintes bases de conhecimento:

- ACM Digital Library
- IEEEExplore
- Springer Link

- PubMed
- Science Direct

As pesquisas realizadas foram limitadas aos trabalhos publicados de 1993 até 2013. Esta definição foi traçada devido a data de lançamento do padrão DICOM. As buscas foram realizadas por meio da utilização de palavras-chave nos mecanismos de busca das bases de dados consultadas, sempre adequando as *strings* solicitadas aos mecanismos de cada base. As palavras-chave foram divididas em três (3) classes, relacionadas ao objeto, as operações realizadas, as bases tecnológicas, visando assim fornecer uma maior acuracidade ao levantamento.

O objeto relacionado a esta tese é o *DICOM*, o qual necessita de operações de *‘indexing’*, *‘storage’*, *‘store’*, *‘index’*, *‘query’*, *‘retrival’*, em bases tecnológicas de armazenamento hierárquico *“HDF”* e *“Hierarchical Data Format”*. Como informação adicional, os contextos (PACS, HIS, RIS e Telemedicina) foram retirados das buscas, pois independente do contexto e da forma com que ele se aplica, o objeto de pesquisa ainda permanece o mesmo. Ainda, de acordo com a RSL, na primeira fase da pesquisa foi limitada somente ao título, *abstract* e as palavras-chave dos artigos.

Com este método, não foram encontrados trabalhos que contemplem a persistência de dados de forma hierárquica, usando técnicas de armazenamento de forma distribuída e paralela para sistemas médicos baseados em imagens DICOM. No Capítulo 3, será demonstrado o resultado deste estudo, onde é feita uma análise dos trabalhos relacionados que foram recuperados. Por fim, após a conclusão da RSL, baseando-se na metodologia proposta e nas bases de conhecimento pesquisadas, pôde-se confirmar o ineditismo desta proposta de modelo.

1.7 ADERÊNCIA DO TEMA AO PPEGC

Devido a natureza interdisciplinar do Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento (PPGEGC), há-se a necessidade de alinhar os temas das pesquisas de mestrado e doutorado, com a gênese central do programa. Entretanto, para se fazer este alinhamento, é necessário discutir um pouco sobre como esta gênese foi criada e esta sendo desenvolvida.

O PPGEGC surgiu em 2004, devido à reformulação realizada no Programa de Pós-Graduação em Engenharia de Produção (PPGGEP) da Universidade Federal de Santa Catarina (UFSC) e tem como cerne o desenvolvimento de pesquisas acerca de macroprocessos de explicitação, gestão e disseminação do conhecimento que, foram divididas em três áreas de concen-

tração: Engenharia do Conhecimento, Gestão do Conhecimento e Mídia do Conhecimento. Estas áreas foram divididas nas seguintes linhas de pesquisa (EGC, 2013):

- Engenharia do Conhecimento Aplicada a Governo Eletrônico;
- Engenharia do Conhecimento Aplicada às Organizações;
- Teoria e Prática em Engenharia do Conhecimento;
- Gestão do Conhecimento da Sustentabilidade;
- Gestão do Conhecimento, Empreendedorismo e Inovação Tecnológica;
- Teoria e Prática em Gestão do Conhecimento;
- Mídia e Conhecimento na Educação;
- Mídia e Disseminação do Conhecimento;
- Teoria e Prática em Mídia e Conhecimento.

Desde então, o PPGEGC tem desenvolvido e otimizado suas linhas de pesquisa, e tem focado suas pesquisas e sua formação no conhecimento, percebido como produto, processo e resultado de interações sociais e tecnológicas entre agentes humanos e tecnológicos, que desenvolvem os seguintes processos de conhecimento (EGC, 2013):

- criação;
- descoberta;
- aquisição;
- formalização/codificação;
- armazenamento;
- uso;
- compartilhamento;
- transferência;
- evolução.

Dito isto, esta tese está alinhada principalmente com a área de Engenharia do Conhecimento, com o intuito de auxiliar na Gestão do Conhecimento no âmbito médico, ainda auxiliando na Mídia do Conhecimento em aspectos diversos. Além disto, esta pesquisa tem aderência direta com as linhas de pesquisa de Engenharia do Conhecimento Aplicada às Organizações, que visa estudar a concepção, desenvolvimento e implantação de soluções da Engenharia do Conhecimento nas organizações. No caso desta pesquisa, em particular, as organizações vão desde grandes hospitais, até pequenas clínicas.

Ainda, há um alinhamento direto com a linha de pesquisa de Teoria e Prática em Gestão do Conhecimento, que tem o objetivo estudar a teoria e a prática da gestão do conhecimento nas organizações e suas relações com a engenharia e com a mídia e conhecimento. Feito o alinhamento com as áreas de pesquisa e com as linhas de pesquisa do PPGEHC, ainda há uma ligação direta nos processos de conhecimento de “armazenamento de conhecimento”, que é o cerne da pesquisa, entretanto ainda há aderência com outros processos, como o de “aquisição de conhecimento” e “transferência de conhecimento”.

Sobre os processos restantes, esta tese ainda tende a auxiliar ou servir como base para a execução dos processos. No caso da criação, descoberta, codificação, uso, compartilhamento, transferência e evolução de conhecimento, este trabalho visa proporcionar uma camada básica de armazenamento e recuperação que pode ser usada como *backend* para o desenvolvimento de outros trabalhos, utilizando de outros métodos e técnicas da engenharia do conhecimento, que tenham como diretos os processos de conhecimento supracitados.

O PPGEHC tem como missão promover o ensino, pesquisa e extensão, de forma interdisciplinar, sobre o conhecimento como elemento agregador de valor para a sociedade (EGC, 2013). No caso desta pesquisa em específico, ela tem como objetivo fim uma melhoria dos tempos de resposta e do atendimento em geral no âmbito da saúde. Com esta maximização do parque de equipamentos médicos hospitalares, mais pacientes poderão ser atendidos. Ainda, com uma construção de uma base de conhecimento de casos passados, a sociedade como um todo ganhará, por intermédio da prática médica, melhores tratamentos, baseados em experiências passadas.

Entretanto, somente o alinhamento do tema desta pesquisa, com as linhas de pesquisa que são desenvolvidas no PPGEHC, e com os processos de conhecimentos nele desenvolvidos, é necessário que a pesquisa desenvolvida tenha caráter interdisciplinar e é vital que a tese esteja vinculada ao “tripé” engenharia, gestão e mídias do conhecimento, que formam a base conceitual do PPGEHC. Desta forma, em suma, o projeto de pesquisa deve pertencer obrigatoriamente a umas das três áreas, porém deve estar atrelada

na fusão de duas ou até das três áreas do PPGEGC.

Sobre a interdisciplinaridade, Repko (2008) cita que ela atua na fronteira do conhecimento humano e segue um padrão que ocorre em três etapas (REPKO, 2008):

1. os pesquisadores selecionam um assunto ou objeto de um *framework* disciplinar;
2. preenchem as lacunas do conhecimento derivadas da falta de atenção a essa tópico pela prática disciplinar;
3. se a pesquisa atrai uma massa crítica, os pesquisadores expandem fronteiras, criando um espaço de novos conhecimentos e novos papéis profissionais.

1.8 METODOLOGIA

Para o desenvolvimento desta tese, do ponto de vista da natureza desta pesquisa, esta será *aplicada*. Este tipo de pesquisa tem o intuito de criar conhecimentos, com objetivo de aplicação prática para solucionar problemas do mundo real, neste caso, o problema de armazenamento de imagens médicas de forma hierárquica e distribuída. Do ponto de vista da abordagem para o desenvolvimento desta pesquisa, ela será de natureza *quantitativa*.

Para efeitos elucidativos, a abordagem qualitativa não foi levada em consideração nesta tese, visto o grau de complexidade em avaliar esta abordagem em cenários reais de aplicação, e ainda, visto que esta tese não pretende gerar um produto final, mas sim, um conjunto de protótipos, que têm o objetivo de provar o modelo proposto. Ainda, com base neste modelo, há-se a possibilidade de construir outros protótipos com tecnologias diferentes, desta forma, a pesquisa qualitativa, focada somente em um protótipo, poderia se tornar deficitária.

Do ponto de vista dos objetivos desta, esta será uma pesquisa experimental e exploratória, onde têm-se o objetivo de elaborar hipóteses e explicitar o problema de pesquisa através de experimentos que serão baseados no modelo que será apresentado. Sobre os procedimentos técnicos para a elaboração desta tese, serão desenvolvidos levantamentos bibliográficos acerca dos temas relacionados, visando embasar a construção do modelo proposto.

Para o desenvolvimento do modelo teórico do modelo proposto, serão desenvolvidos um conjunto de protótipos, utilizando-se de tecnologias de vanguarda com integrações entre os múltiplos componentes e por fim, será realizada uma pesquisa experimental com os protótipos, onde os mesmos serão

aplicados em um cenário real, derivando assim o primeiro estudo de caso do modelo. Por fim, ainda haverá duas etapas em que metodologias serão descritas, porém estas relacionadas aos procedimentos metodológicos da proposta e ainda ao conjunto de testes do protótipo no estudo de caso, sendo que estes procedimentos serão apresentadas na Seção 5.2.

1.9 ESTRUTURA DO TRABALHO

Esta tese esta subdividida em 7 capítulos, apresentados abaixo.

1. Capítulo 1 - Introdução: Neste capítulo, foi dada uma contextualização geral sobre o tema de pesquisa, culminando no problema de pesquisa que esta tese se propõe a abordar através das perguntas de pesquisa elaboradas. Após isto foram dissertados os objetivos gerais e específicos que formam esta tese. Na sequência a relevância e a justificativa para a elaboração da pesquisa foram levantados, buscando sensibilizar da importância do tema para a comunidade científica. Logo após foram traçadas a delimitações do escopo que esta tese irá abordar, bem como a aderência ao PPEGC foi determinada. Por fim, foi falado sobre a estrutura do trabalho que rege esta tese.
2. Capítulo 2 - Fundamentação Teórica: neste capítulo será dada a fundamentação teórica desta pesquisa. Os assuntos relacionados tem a função de embasar o leitor acerca dos temas abordados no modelo proposto. Será dissertado sobre os conceitos de Telemedicina, passando pelo objeto de aplicação desta tese, o Sistema Integrado de Telemedicina e Telessaúde (STT), pois será nele que os testes serão realizados. Após isto, será explanado sobre o DICOM, e sobre a aplicação que desenvolve este padrão no STT, o CyclopsDCMServer. Em seguida, será dissertado sobre formatos de dados, passando pelo *Hierarchical Data Format* (HDF5) . Após isto serão abordados os sistemas de indexação de dados, onde será falado dos seus fundamentos e sua importância para sistemas de persistência de dados. Ainda, serão abordados os Sistemas de Arquivos Distribuídos, falando de sua importância, modelos, ferramentas e aplicações. Serão abordados os seguintes tipos de sistemas CEPH, PVFS, Lustre, FhGFS e o HDFS, pois estes servirão de *backend* no modelo. Por fim, serão dadas as considerações finais do capítulo.
3. Capítulo 3 - Estado da Arte: para verificar o ineditismo, ou ainda, a originalidade do modelo proposto, nesta capítulo será discutido o estado

da arte do tema desta tese, buscando averiguar a existência de trabalhos relacionados ou similares. Para realização deste levantamento, será utilizado o método Revisão Sistemática da Literatura, proposto e disseminado por Kitchenham (KITCHENHAM, 2004). Ainda, serão discutidos trabalhos relacionados e trabalhos realizados, relativos ao tema central do modelo proposto. Por fim, serão dadas as considerações finais do capítulo.

4. Capítulo 4 - Modelo Proposto: neste capítulo, dado o contexto apresentado sobre o tema e baseando-se no levantamento bibliográfico e posteriormente a revisão sistemática da literatura, pode-se desenvolver um modelo de armazenamento para imagens médicas DICOM de forma hierárquica e distribuída. Neste capítulo, será discutido o modelo atual que é usado no estudo de caso (STT/RCTM) e sua respectiva arquitetura. Ainda, será apresentado o modelo proposto para esta tese e a arquitetura proposta para implementar este modelo. Serão traçadas observações sobre todos os componentes da arquitetura, suas interligações, fundamentos e funções dentro do contexto que será apresentado. Por fim, serão dadas as considerações finais do capítulo.
5. Capítulo 5 - Ambiente e Resultados Experimentais: Neste capítulo serão apresentados os experimentos baseados no modelo proposto e em sua arquitetura. Será explicada a metodologia procedural de testes para avaliação do modelo, bem como a arquitetura computacional utilizada nos testes. Em outras palavras, serão mostradas as configurações dos computadores e os pacotes de software utilizados no protótipo. Ainda, serão apresentados dados preliminares dos resultados dos quatro (4) experimentos, apresentando observações iniciais. Finaliza-se o capítulo com considerações gerais relacionadas a esta fase da pesquisa.
6. Capítulo 6 - Análises e Discussões dos Experimentos: Neste capítulo será efetuada a análise dos experimentos realizados no Capítulo 5. Busca-se traçar observações sobre os resultados obtidos e ainda, elaborar comparações entre o modelo proposto, utilizando formatos de dados hierárquicos de forma distribuída e paralela, para armazenamento e recuperação de imagens médicas DICOM *versus* o modelo atual, utilizando o sistema gerenciador de bancos de dados PostgreSQL. O capítulo está dividido em duas partes, sendo a primeira delas a análise dos resultados, que será dividida na análise do processo de armazenamento e posteriormente, a análise do processo de recuperação de dados. Na segunda parte, será proposta uma discussão sobre os resultados e análises obtidas. Por fim, serão dadas as considerações finais do capítulo.

7. Capítulo 7 - Conclusões e Trabalhos Futuros: Neste capítulo serão dadas as considerações finais desta tese, apontando às contribuições alcançadas, fazendo a ligação entre os objetivos almejados e o fechamento dos mesmos. Ainda, será respondida a pergunta de pesquisa e por fim, serão dados direcionamentos para o desenvolvimento de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será dada a fundamentação teórica desta pesquisa. Os assuntos aqui relacionados têm a função de prover embasamento acerca dos temas abordados no modelo proposto. Será dissertado sobre os conceitos de Telemedicina, passando pelo objeto de aplicação desta tese, o Sistema Integrado de Telemedicina e Telessaúde (STT), pois será nele que os testes serão realizados. Após isto, será explanado sobre o padrão de imagens médicas DICOM, pois ele é o objeto central desta tese. Ainda, serão abordados os Sistemas de Arquivos Distribuídos (SAD), falando de sua importância, modelos, ferramentas e aplicações. Serão abordados os seguintes tipos de sistemas CEPH, PVFS, Lustre, FhGFS e o HDFS, pois estes servirão de *backend* no modelo. Em seguida, será dissertado sobre formatos de dados, abordando o *Hierarchical Data Format* (HDF5), pilar central desta tese. A seguir, serão abordados os indexação de dados, onde será falado dos seus fundamentos e sua importância para sistemas de persistência de dados e será dado o exemplo de mecanismo de indexação escolhido para esta tese, o Clucene. Por fim, serão dadas as considerações finais deste capítulo.

2.1 TELEMEDICINA

A Telemedicina pode ser definida, de acordo com Bashshur (1995) como o uso de telecomunicações e tecnologias de informação para compartilhar e manter informações de saúde do paciente e para prestar cuidados clínicos e de educação em saúde para pacientes e profissionais, quando a distância separa os participantes envolvidos no processo (BASHSHUR, 1995). Moore (1999) ainda afirma que Telemedicina é a entrega de serviços de saúde, de forma remota, através de meio de telecomunicações, em atividades de consulta interativa e serviços de diagnóstico (MOORE, 1999).

No passado, durante a década de 80 e 90, os serviços de Telemedicina eram exclusivamente, ou quase que totalmente, endereçados aqueles que estavam excluídos dos grandes centros médicos ou ainda, para localizações distantes não assistidas por cuidados médicos, em especial as zonas rurais. Entretanto, isto mudou nos últimos anos, pois atualmente as aplicações de Telemedicina têm crescido e se espalhado por várias áreas, como por exemplo em medicina militar, na prisão ou mesmo nas casas dos pacientes (MOORE, 1999).

A primeira aparição deste tipo de abordagem foi em 1959 (CABRAL J.E.; KIM, 1996), quando experimentos envolvendo tele-psiquiatria (WITT-

SON; AFFLECK; JOHNSON, 1961) e tele-fluoroscopia (JUTRAS, 1959) onde novos procedimentos médicos foram mostrados em uma transmissão ao vivo em conferências médicas. Na década de 1970 e 1980, experimentos em Telemedicina foram focados na transmissão de imagens médicas usando a televisão, via *broadcast* ou usando fitas gravadas, buscando levantar diagnósticos com especialistas, ou ainda, visando segunda opinião médica. Nos anos 90, o foco principal foram os experimentos voltados a Telemedicina via vídeo-conferência, para realização de consultas médicas a distância (CABRAL J.E.; KIM, 1996).

Desde então, os sistemas de Telemedicina tem se popularizado exponencialmente. Os principais fatores que culminaram nesta disseminação foram a maturidade que as redes de comunicação ao redor do mundo chegaram e também ao poder computacional que atualmente pode-se adquirir. Estes dois adventos, somados ao fato da modernidade das linguagens de programação, e a alta tecnologia aplicada em equipamentos médico hospitalares, resultaram em uma equação de sucesso para implantação deste tipo de modelo.

Atualmente, as Redes de Telemedicina (*Telemedicine Networks*) também são bastante utilizadas para fins educacionais – buscando formar melhor profissionais na área médica, para decisões clínicas, inteligência artificial, para registro de prontuário de pacientes, para suporte administrativo, organização de fluxo de trabalho, entre outras tarefas (MOORE, 1999).

Hoje em dia, é possível encontrar projetos consolidados da aplicação de Telemedicina, isoladamente, ou unido a sistemas PACS, em várias partes do mundo, como por exemplo na Inglaterra (MAY et al., 2000), Alemanha (THIELSCHER; DOARN, 2008), (DUENNEBEIL et al., 2010), China (WANG; GU, 2009), (XUE; LIANG, 2007), (HSIEH et al., 2001), Paquistão (MALIK, 2007), Índia (PAL et al., 2005), entre outros Países. Estes ainda, podem se encontrar totalmente integrados com os RIS/HIS, visando uma melhor gerência do hospital como um todo.

De acordo com Moore (1999), a Telemedicina pode ser dividida em duas categorias, sendo elas a assíncrona e a síncrona. O serviço síncrono deve ocorrer em tempo real e, primariamente, inclui interações com áudio, vídeo e imagens. Este serviço é usado comumente para consultas remotas, discussões de casos entre médicos ou para fins educacionais (MOORE, 1999). Já o serviço assíncrono (WILSON, 2003), ocorre em diferentes tempos e geralmente é caracterizado pelo uso de imagens, e-mail e algumas vezes vídeos. Este tipo de serviço geralmente é usado na tele-radiologia ou tele-patologia. O serviço assíncrono algumas vezes podem ser baseados somente em software, algumas vezes ela pode ser portátil para outras arquitetura e estes sistemas podem incluir *scanners* de filmes, sistemas DICOM, sistemas de gerenciamento de imagens, PACS, dermatoscópios, entre outros (MOORE,

1999).

2.1.1 STT

Na âmbito da América Latina, a principal iniciativa neste sentido é o Sistema Integrado de Telemedicina e Telessaúde (STT), integrante da Rede Catarinense de Telemedicina (RCTM) (WANGENHEIM et al., 2009), (WALLAUER et al., 2008), (MAIA; WANGENHEIM; NOBRE, 2006). Esta infraestrutura de serviços de Telemedicina e Telessaúde, aplicada inicialmente em Santa Catarina, Brasil, teve seu marco inicial em 1999, como um projeto de aplicação de algumas tecnologias que estavam naquele momento sendo desenvolvidas pelo Cyclops Group (GROUP, 2013), da UFSC. Estas tecnologias estavam sendo desenvolvidas em parceria com instituições da Alemanha e ainda contava com financiamento do CNPq e da Fundação Volkswagen (Wolfsburg/German).

Depois da etapa de análise e de desenvolvimento, no ano de 2002 o piloto desta rede iniciou a operação, fazendo a ligação em pequenas clínicas médicas dos municípios de Florianópolis e São José. Após isto, no ano de 2004, baseados no sucesso da aplicação da tecnologia nas clínicas, o Governo do Estado de Santa Catarina, juntamente com o Projeto Cyclops, iniciaram a análise para implementação de um piloto a nível estadual, para ser implementado no primeiro semestre do 2005. Este piloto tinha o intuito de demonstrar em números o quão positivo era a aplicação de serviços de Telemedicina Assíncrona, no contexto do Sistema Único de Saúde (SUS).

Este projeto piloto teve início em dois municípios, sendo eles Quilombo e São Lourenço do Oeste, e eles formaram o marco inicial da RCTM. Devido aos bons resultados obtidos, logo após mais três (3) pontos de presença foram implantados, sendo oferecido primeiramente laudos a distância, remessa eletrônica de resultados e armazenamento de imagens nas modalidades de Eletrocardiograma, Tomografia Computadorizada e Ultra-Sonografia. Deste então, esta rede vem sendo maximizada, já ultrapassando mais de dois milhões e quinhentos mil (2.500.00) exames realizados.

Até o mês de abril de 2012, o STT já estava presente em 287 dos 293 municípios catarinenses, sendo que mais de 193 já tem a capacidade de enviar exames a distância. O volume de exames armazenados no sistema ultrapassou os 2 milhões exames e engloba, dentre outros, eletrocardiogramas, exames dermatológicos, análises clínicas, tomografias computadorizadas, ressonâncias magnéticas, ultrassonografias e raios-X, conforme mostra a Figura 5. Até abril de 2012, a base era composta por 2.118.709 exames, sendo os exames mais expressivos são Análises Clínicas, com 41,05%, os exames

de Radiologia Computadorizada, com 27,21% e os Eletrocardiogramas, com 21,63% (ANDRADE; WAGNER; WANGENHEIM, 2012).

Exames por modalidades no STT			
Abrev.	Descrição	Qtde	%
AC	Análises Clínicas	869.765	41,052%
RC	Radiologia Computadorizada	576.562	27,213%
ECG	Eletrocardiografia	458.389	21,635%
TC	Tomografia Computadorizada	141.766	6,691%
US	Ultrassonografia	49.387	2,331%
EN	Endoscopia	10131	0,478%
NP	Nutrição Parenteral	3.700	0,175%
RM	Ressonância Magnética	2819	0,133%
CIN	Cintilografia	2.806	0,132%
HEM	Hemodinâmica	1.130	0,053%
COL	Colonoscopia	987	0,047%
PB	Patologia Bucal	756	0,036%
DER	Dermatoscopia	284	0,013%
OTO	Otorrinolaringologia	126	0,006%
DO	Densitometria Óssea	51	0,002%
BRO	Broncoscopia	33	0,002%
VE	Vídeo- Eletroencefalografia	17	0,001%
Total		2.118.709	100%

Figura 5: Tabela com a quantidade de exames do STT (ANDRADE; WAGNER; WANGENHEIM, 2012).

O STT/RCTM servirá como base de testes no estudo de caso que será proposto para avaliação do modelo de armazenamento proposto nesta tese. A proposta é alterar apenas o *backend* de armazenamento de dados, deixando assim a aplicação intocada. Este é um requisito primordial para o sucesso do modelo que esta sendo proposto, visto que a adaptabilidade à sistemas de Telemedicina, PACS, HIS e RIS, já existentes deve ser feito de forma simples, não causando altos custos com adaptações de códigos para suportar a nova arquitetura que irá persistir os exames.

2.2 DICOM

Nos últimos 40 anos ocorreu um crescimento exponencial no uso de sistemas digitais na área da medicina, principalmente figurados na forma de modernos equipamentos médicos e sistemas de computação em geral. Isto

aconteceu devido a maturidade em que a computação e a engenharia atingiram, proporcionando para os mais variados meios, facilidades nunca vistas até então. No âmbito da Medicina, atualmente, hospitais, centros médicos e clínicas podem contar com o apoio de equipamentos médicos baseados em imagens para auxiliar no diagnóstico de seus pacientes. Entre estes equipamentos, pode-se citar como exemplos, as modalidades de Tomografia Computadorizada, Ressonância Magnética, Raio-X, entre outros.

Entretanto, com este crescimento e posterior popularização, muitos fabricantes iniciaram a construção de seus próprios equipamentos, para determinadas modalidades, o que culminou em uma inexistência de formato de interoperabilidade entre equipamentos de fabricantes diferentes, o que rapidamente tornou-se um problema, pois a falta de integração e sistemas, prejudicava a atividade médica.

Devido a este fato, no final da década de 70, o Colégio Americano de Radiologia (*American College of Radiology - ACR*) e a Associação Nacional dos Fabricantes de Produtos Elétricos (*National Electrical Manufacturers Association - NEMA*) se uniram para criação de uma padronização para transferência de imagens médicas e para troca de informações entre diferentes fabricantes, buscando assim, uma melhor interoperabilidade entre múltiplos equipamentos (BIDGOOD et al., 1997).

Desta forma, em 1985 surgiu o padrão denominado DICOM (*Digital Imaging and Communication in Medicine*). Para manutenção e melhorias no formato, um comitê foi criado, sendo este composto por diversos grupos de trabalho, que foram divididos em áreas de concentração. Houve ainda uma versão lançada em 1988 com significantes melhorias, principalmente relacionadas a estrutura de dados, terminologias e codificação. Atualmente este formato esta em sua terceira versão (1993), e foi optado que fosse a última. Desta forma não havendo mais lançamento de versões, mas sim atualizações regulares no formato (HUSSEIN et al., 2004).

Como já mencionado, o objetivo do padrão DICOM é possibilitar comunicação e interoperabilidade entre múltiplos equipamentos, visando reduzir custos e organizar o *workflow* de instituições de saúde com base em suas imagens digitais, sempre quando houver atividades de aquisição, armazenamento, visualização, processamento, recebimento, envio, busca e impressão de exames (ASSOCIATION, 1993) (MILDENBERGER; EICHELBERG; MARTIN, 2002). Na Figura 6 pode-se visualizar mais um exemplo de uma imagem DICOM.

Para assegurar o padrão, foi estabelecido uma série de documentos, divididos em partes, que tratam de vários aspectos do DICOM. Estes documentos são de suma importância para este trabalho, visto que a conformidade com o padrão é um dos requisitos primordiais para a validação científica



Figura 6: Exemplo de uma Imagem DICOM.

desde trabalho. Por exemplo, pode-se citar a padronização de quais serviços que um cliente ou servidor pode fornecer (NEMA, 2004a), o mapeamento de objetos do mundo para a estrutura de dados do padrão (NEMA, 2003) e de como o formato digital das imagens deve ser armazenada binariamente em sistemas de arquivos (NEMA, 2011).

O processo de digitalização dos objetos do mundo real para o padrão é feito pelo emprego de modelos entidade-relação, com mapeamentos pré-definidos e formalizados no documento que trata desta parte do padrão (NEMA, 2003). Assim como a parte de abstração destes objetos DICOM, que contém as informações que descrevem como este objeto foi encapsulado. Esta abstração é chamada na literatura pela sigla IOD (*Information Object Definition*).

Cada um dos IODs é constituído de elementos de dados (*data elements*) obrigatórios e opcionais que são identificados unicamente interna-

mente no padrão em um dicionário de dados (NEMA, 2004b) e agrupados de acordo com similaridade. Dessa maneira, o identificador, chamado de *tag*, é composto pelo grupo (G) ao qual pertence e o elemento dentro desse grupo (E), resultando assim em um identificador seguindo o formato hexadecimal (GGGG, EEEE), por exemplo, o identificador (0010, 0010) é mapeado para o nome do paciente (PRADO, 2012). Na Figura 7 pode-se visualizar uma representação da estrutura de um elemento de dados.

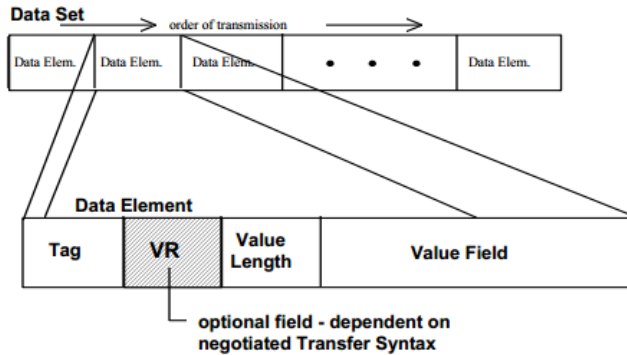


Figura 7: Representação de um Elemento de Dados (NEMA, 2004b).

Os IODs podem ser objetos de dois tipos: normalizado ou composto. Os objetos normalizados são elementos específicos do mundo real, isto é, pode existir um elemento que agrupe todas as informações referentes ao paciente, como nome, idade, sexo, altura peso, dentre outros dados e, de maneira similar, um elemento composto pelos atributos do estudo, como dados temporais sobre quando o estudo foi realizado, identificador do estudo. A combinação dos tipos de dados normalizados resulta nos IODs do tipo composto, que por sua vez são constituídos de todos os dados necessários para contextualização da informação de um exame. Cada IOD contém um identificador único global de até 64 caracteres, chamado de UID (*Unique Identifier*) em que o prefixo deste identificador deve estar registrado em alguma organização reguladora, como a ANSI nos Estados Unidos ou DIN na Alemanha (PRADO, 2012).

Para ilustrar um IOD do tipo composto, pode-se pensar em um corte de uma tomografia computadorizada, que contém centenas de cortes. Toda informação pertencente ao corte é mapeado para um IOD. Assim, individualmente um IOD composto sempre é completo de forma que contenha informações redundantes de todos os dados sobre um exame (PRADO, 2012). Por exemplo, dados do paciente examinado, dado do estudo realizado, dados

Tag	Tipo	Descrição
0002,0102	OB	Informação Privada
0008,0008	CS	Tipo da Imagem
0008,0020	DA	Data do Estudo
0008,0021	DA	Data da Série
0008,0060	CS	Modalidade
0008,0070	LO	Fabricante
0008,0080	LO	Nome da Instituição
0010,0020	LO	Identificador do Paciente
0010,21B0	LT	Histórico do Paciente Adicional

Tabela 2: Exemplos de Metadados DICOM.

da séries oriundas do estudo, dentre outros.

Desta forma, como já mencionado, uma “imagem” DICOM, não é composta apenas pela “figura” explicitada (*Pixel Data*). Uma imagem DICOM pode ser composta por mais de 2000 metadados, que vão desde do identificador do paciente (*Patient ID*), até por exemplo o laudo dado para tal imagem. Nas referências da NEMA, o termo metadado é comumente usado como *data element*, sendo assim, sinônimos para fins técnicos. Um *data element* armazena os dados, informações e conhecimentos acerca do conteúdo da imagem médica, visto que, mesmo isoladamente, cada imagem DICOM deve se auto-descrever, tendo assim um significado próprio.

Dito isto, a complexidade do gerenciamento destas imagens aumenta significativamente, pois o que será armazenamento não é somente uma figura, mas sim um conjunto de informações (metadados) de identificação e composição da imagem, juntamente com um *pixel data* (a figura). Na Tabela 2 é possível visualizar alguns exemplos de metadados que podem ser encontrados em implementações do padrão. Lembrando ainda que é somente para fins ilustrativos, visto que há mais de 2000 metadados possíveis no padrão.

Em relação aos serviços, no DICOM há dois tipos, sendo eles do tipo C ou N. Apenas serviços do tipo C serão vistos aqui. Esta delimitação foi proposta porque são eles os serviços que tem relação direta com o objeto da tese, pois tratam de armazenamento, recuperação e busca. Estes serviços serão implementados pelo sistema de indexação, para localização dos dados dentro do modelo proposto. Dentro dos serviços do tipo C, três (3) deles serão abordados e estudados, sendo eles:

- Armazenamento: **C-Store** – *Composite Store*;
- Busca: **C-Find** – *Composite Find*;

- Recuperação: **C-Get e C-Move** – *Composite Get/Move*.

No âmbito do Sistema Integrado de Telemedicina e Telessaúde (STT), foi desenvolvida uma aplicação, ou ainda, um servidor de imagens médicas DICOM, chamado de CyclopsDCMServer. Este servidor é responsável em receber as imagens enviadas pelos equipamentos médico hospitalares, segmentá-las e armazená-las no banco de dados do sistema. É neste componente de software que esta tese fará experimentos do modelo.

2.3 SISTEMAS DE ARQUIVOS DISTRIBUÍDOS

Um Sistema de Arquivos Distribuído (SAD) pode ser entendido como uma implementação distribuída do modelo clássico de *time-sharing* de um sistema de arquivos, onde múltiplos usuários compartilham recursos de compartilhamento de arquivos e de armazenamento (LEVY; SILBERSCHATZ, 1990). O propósito de um SAD é permitir aos clientes, sendo eles usuários ou aplicações, armazenarem e acessarem arquivos remotos exatamente como se fossem locais, possibilitando assim que os usuários acessem a camada de distribuição a partir de qualquer computador (TANENBAUM; STEEN, 2002) (COULOURIS, 2009).

Em relação às características dos SAD, eles são responsáveis pela organização, armazenamento, recuperação, atribuição de nomes, compartilhamento e proteção de arquivos. Além disso, eles fornecem uma interface de programação que caracteriza a abstração de arquivo, liberando os programadores da preocupação com os detalhes da alocação e *layout* do armazenamento físico nos discos (COULOURIS, 2009). Um modelo clássico de SAD pode ser visto na Figura 8.

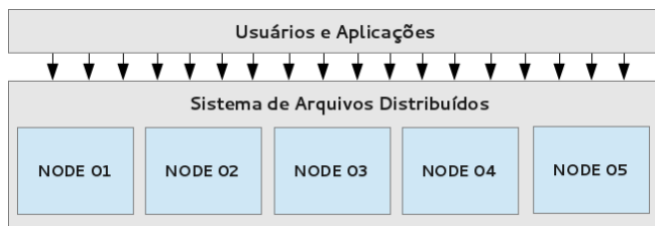


Figura 8: Representação Clássica de um SAD.

De acordo com (COULOURIS, 2009), para atingir as características pré-definidas na concepção, no projeto e no desenvolvimento de um SAD, existem muitos requisitos e armadilhas no projeto, para a implantação des-

ses serviços distribuídos que são baseados em SAD. Dentre estes requisitos, pode-se citar:

- **Transparência:** um SAD deve suportar um grau de transparência satisfatório para o usuário final. Ele deve contrabalancear a flexibilidade e escalabilidade derivadas da transparência, com a complexidade e desempenho do software. Os tipos de transparência encontrados em SAD podem ser de acesso, localização, mobilidade, desempenho e mudança de escala.
- **Atualizações Concorrentes de Arquivos:** as modificações realizadas em um arquivo, por um determinado usuário não devem interferir na operação de outros clientes que estejam acessando, ou alterando, o mesmo arquivo simultaneamente. Este é o conhecido “controle de concorrência”, sempre discutido na literatura.
- **Replicação de Arquivos:** em um SAD que suporta operações de replicação de dados, um arquivo pode ser representado por várias cópias idênticas em vários locais diferentes do sistema. Do ponto de vista técnico, isto tem duas vantagens: compartilhamento de carga, melhorando o desempenho e escalabilidade do serviço e a melhora nos níveis de tolerância a falhas, permitindo assim que se uma falha ocorrer, o arquivo ainda será acessível em outro ponto do SAD.
- **Heterogeneidade do Hardware e do Sistema Operacional:** este é um importante quesito no projeto de um sistema de arquivo distribuído, visto que a heterogeneidade pode ser um fator limitador na implantação do mesmo. Desta forma, as interfaces de serviço devem ser definidas de forma que o software, tanto do cliente, como do servidor, possam ser implantados para diferentes plataformas e sistemas operacionais.
- **Tolerância a Falhas:** um SAD deve ser capaz de continuar seu funcionamento em casa de falhas de hardware, software, no cliente ou no servidor. Em vários casos esta tolerância pode ser encontrada em abordagens *stateless* ou um servidor ou cliente pode ser totalmente restaurado, sem a necessidade de restauração de *backups*. Ainda, para este requisito, a replicação de arquivos é fundamental.
- **Consistência:** é um requisito que esta cruzado com o controle de concorrência, visto que com a implantação de múltiplas cópias de um mesmo arquivo, em várias partes do SAD, somada ao fato de que vários usuários podem acessar e modificar o arquivo, simultaneamente, o grau de complexidade aumenta significativamente. Ainda, há o fato de que,

o tempo de replicação de cópias e alterações entre múltiplos *datacenters* pode ser um pouco demorada, dependendo da rede usada.

- **Segurança:** o controle de acesso aos arquivos armazenados no SAD é um requisito primordial na concepção e projeto de um serviço. Todas as requisições, tanto de leitura, como de escrita, necessitam de autenticação, evitando assim acessos não-autorizados em parte dos arquivos. Ainda, há a possibilidade de inserir criptografia nos dados armazenados, o que aumenta os níveis de segurança e, conseqüentemente, a sobrecarga nos servidores, devido as operações de criptografia e descriptografia de dados em tempo real.
- **Eficiência:** um SAD deve oferecer os recursos que tenham pelo menos o mesmo poder e flexibilidade de sistemas de arquivos locais. Ainda, deve obter um desempenho comparável ou superior, independente da carga a qual eles estão expostos. Este é um requisito complexo e altamente desafiados, visto que acessos locais não dependem de intercomunicação entre múltiplos computadores. Por último, este é um requisito fundamental para uma boa experiência do usuários lidando com o sistema.

Para atender alguns dos requisitos acima citados, como por exemplo “eficiência” e “tolerância a falhas”, comumente em projetos de SAD, são usados “*middlewares*”. Estes *middlewares* têm a função de intermediar às requisições de entrada e saída e dar-lhes o tratamento adequado, de acordo com a solicitação realizada. Na Figura 9 é possível visualizar como estes componentes de software se encaixam em uma arquitetura básica de um SAD.

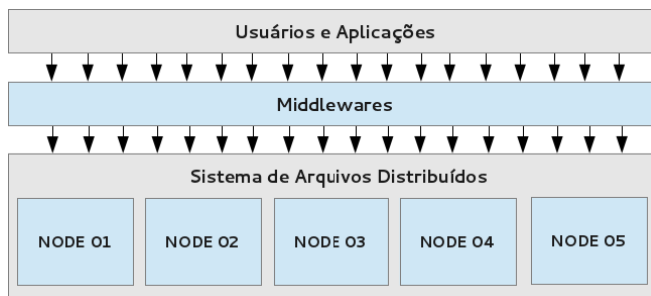


Figura 9: Representação Clássica de um SAD com *Middleware*.

Um “*middleware*” pode ser definido como um componente de software que fornece serviços para outros softwares, além das já fornecidas pelo

sistema operacional. Os *middlewares* também têm a função de facilitar o processo de desenvolvimento de software, visto que eles abstraem às tarefas de comunicação para os desenvolvedores. No contexto dos SAD, os *middlewares* são usados para auxiliar nas tarefas de comunicação, gerenciamento de dados ou ainda, proporcionando melhores níveis de abstração para os desenvolvedores, por meio do uso de API (*Application Programming Interface*) de alto nível .

Quanto à comunicação e o processamento, para proporcionar melhores níveis de eficiência e escalabilidade, comumente sistemas de arquivos distribuídos implementam *middlewares* específicos para proporcionar escrita e leitura paralela. Na literatura há referências para este tipo de abordagem como *Parallel File Systems* (PAD) . Como o próprio nome diz, esta abordagem pode receber “n” solicitações de escrita e leitura, de “n” usuários e processá-las simultaneamente, dando mais vazão ao sistema como um todo.

Para o desenvolvimento e a aplicação destas abordagens na concepção e no projeto de aplicações para sistemas distribuídos, as duas iniciativas mais disseminadas e suportadas são o *Messaging Passing Interface* (MPI) e o *Parallel Virtual Machine* (PVM) . Sendo que o último (PVM) tem caído em desuso nos últimos anos, visto a popularidade do MPI. Neste trabalho, será usado como base o paradigma do MPI, devido ao suporte dele para os mais variados tipos de SAD, e ainda somado ao fato de sua integração com as bibliotecas de alto nível. Na Seção 2.3.1 será dissertado mais sobre este paradigma.

Baseados nestes requisitos, características e funcionalidades, muitos SAD foram desenvolvidos nos últimos anos, para suportar um grandes número de naturezas de aplicações ou ainda para propósito geral. Exemplos destes SAD podem ser: AFS (HOWARD et al., 1988), Lustre (BRAAM et al., 2004), CEPH (WEIL et al., 2006a), PVFS (LIGON W.B.; ROSS, 1996), FhGFS (FRAUNHOFERFS, 2013), HDFS (BORTHAKUR, 2007), CODA (SATYANARAYANAN et al., 1990), SPRITE (BAKER; OUSTERHOUT, 1991), ECHO (BIRRELL et al., 1993), entre outros.

2.3.1 *Messaging Passing Interface*

O *Message Passing Interface* (MPI) pode ser definido como uma biblioteca padrão e portátil de sub-rotinas de comunicação para programação paralela, que foi projetado para funcionar em uma ampla variedade de computadores paralelos. Ele é útil em computadores paralelos, tais como SP2 da IBM, o Cray ResearchT3D, bem como em redes de estações de trabalho (*workstations*) (SNIR et al., 1995). Atualmente, o MPI se tornou de fato um

padrão em termos de computação paralela para alto-desempenho.

O MPI baseou-se no modelo de passagem de mensagem (*message passing* que surgiu como um expressivo paradigma para programação paralela (GROPP et al., 1996). Até então, a sintaxe e semântica de cada implementação deste modelo eram diferentes e notou-se um espaço para o desenvolvimento e implementação de uma padronização para fins de interoperabilidade entre sistemas. Desta forma, em abril de 1992 o processo de criação do padrão teve início no *Message Passing Interface Forum*, na conferência *Supercomputing'92* (GROPP et al., 1996). Após 18 meses de discussões, em maio de 1994 a versão 1.0 do padrão foi disponibilizada a comunidade e desde então o padrão vem se aperfeiçoando.

A principal característica do MPI é a troca de mensagens entre processos. Entretanto, além da função de enviar e receber mensagens e coordenar estas operações, o MPI também disponibiliza funções primitivas, como *Broadcast* e *Gather*. Neste caso o *Broadcast* é responsável em enviar uma mensagem para todos os processos envolvidos e o *Gather* é responsável em receber as mensagens de retorno e tratá-las. Com a definição do padrão, vários grupos de pesquisa pelo mundo iniciaram atividades de desenvolvimento de suas próprias versões do padrão. Dentre estas iniciativas pode-se citar as alternativas *opensource* MPICH (GROPP et al., 1996), OpenMPI (GABRIEL et al., 2004), CHIMP-MPI (ALASDAIR; BRUCE; MILLS, 1994), entre outros. Enquanto isto, há ainda alternativa proprietárias, como por exemplo o Intel MPI, HP MPI, MATHLAB MPI, entre outras. Todas estas versões tem suas particularidades e propósitos, mas seguem estritamente o padrão definido, atingindo assim interoperabilidade entre múltiplos sistemas.

O MPI além de fornecer um aumento no desempenho de processamento de transações, ele também utiliza paralelização de escrita e leitura de dados, via interface MPI-IO (CORBETT et al., 1996). O MPI-IO foi proposto como uma extensão do padrão MPI para preencher as lacunas relativas a atividades de escrita e leitura. O MPI-IO provê uma interface de alto nível para descrever o particionamento de arquivos entre processos, para transferência de estruturas globais entre processos e arquivos, atividades de escrita e leitura de modo assíncrono e otimização no armazenamento de arquivos em sistema de armazenamento (CORBETT et al., 1996).

Dessa forma, o Laboratório Nacional de Argonne, nos EUA, desenvolveu uma implementação da interface do MPI-IO, chamada ROMIO. O objetivo do projeto foi desenvolver uma interface de aplicação de alto desempenho para leitura e escrita paralela. O ROMIO foi baseado na versão 2.0 do MPI e usa um componente chamado de *ADIO Abstract-Device Interface IO* (THAKUR; GROPP; LUSK, 1996), para atividades de integração de sistemas de arquivos e tarefas de escrita e leitura. Na Figura 10 é possível visua-

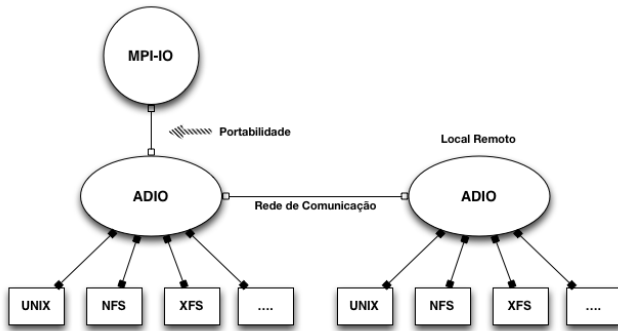


Figura 10: Arquitetura do ROMIO - Baseado em (THAKUR; GROPP; LUSK, 1999).

lizar a arquitetura do ROMIO. Nesta figura é possível que a portabilidade foi implementada no topo da arquitetura, e que as atividades de integrações entre diferentes sistemas de arquivos foi entregue ao ADIO (THAKUR; GROPP; LUSK, 1999).

Desde sua concepção até os dias de hoje, o ROMIO se disseminou e tornou-se uma das implementações do MPI-IO mais utilizadas. Fato é que algumas das interfaces, que implementam o MPI já vem com o ROMIO embutido em seus códigos-fonte. Exemplos das interfaces podem ser MPICH, LAM, HP MPI, SGI MPI e o NEC MPI. Em outras interfaces, como o OpenMPI, que não vem com o ROMIO embutido, mas pode facilmente ser estendida baixando o código-fonte e integrando as interfaces.

2.3.2 Análise dos Sistemas de Arquivos Distribuídos

De acordo com (TANENBAUM; STEEN, 2002) os SAD podem ser divididos em: cliente-servidor, baseados em *clusters* e sistemas simétricos. O tipo cliente-servidor, tem como componente um servidor que tem a função de disponibilizar o acesso aos dados armazenados, por qualquer cliente que esteja autorizado a usar o serviço. Neste modo, os clientes acessam os dados da mesma forma que um sistema de arquivos comum, assim não importando o local físico em que o dado se encontra.

O tipo de SAD baseado em *clusters* (agregados), são arquiteturas computacionais direcionadas para as aplicações que operam de forma paralela, que são a base para o surgimento da técnica de *file striping*. Esta técnica tem

a função de particionar os arquivos em partes e dividi-los em vários elementos do *cluster*, possibilitando assim que estas partes sejam acessadas de forma paralela, agilizando o desempenho do sistema como um todo (TANENBAUM; STEEN, 2002). Os tipos de SAD que suportam estas técnicas de múltiplos acessos são comumente chamados de Sistemas de Arquivos Paralelos (SAP). Uma importante informação é que mesmo eles suportando trabalhar em paralelo, eles também trabalham somente com distribuição dos dados e gravação em métodos seriais.

Os sistemas simétricos, também são considerados um tipo de SAD e estes são baseados nos sistemas *peer-to-peer*. Neste tipo de sistema é usado uma *Distributed Hash Table* para distribuição dos arquivos nos nodos participante e um mecanismo de busca e recuperação baseados em chaves. Ainda, é possível a concepção de uma arquitetura desta natureza, em cima de um outro SAD, onde cada nodo local armazena dados (TANENBAUM; STEEN, 2002). Assim, este tipo de SAD é caracterizado por ter alto poder de vazão de dados.

Essa taxonomia objetiva normalizar os três (3) tipos básicos de SAD e como são os seus comportamentos esperados para cada categoria definida. Entretanto, as implementações modernas de SAD se baseiam, em parte, nos modelos pré-estabelecidos, mas vão além em alguns aspectos. Alguns exemplos podem ser o *Distributed Object Storage* ou *Large-Scale File Systems*. Estes tipos de abordagens diferenciam-se no modo em que os dados são tratados e armazenados, porém a natureza do SAD clássico permanece.

Neste trabalho, os SAD escolhidos para serem avaliados como plataforma de armazenamento e distribuição de dados para o modelo proposto foram o CEPH, FhGFS, PVFS e o Lustre. Os critérios que definiram a utilização desses, em lugar de outros, foram a maturidade desses sistemas, a quantidade de documentação de suas API, por terem a API aberta, a possibilidade de implementação de escrita e leitura paralela por meio do MPI-IO, a popularidade dos mesmos na comunidade científica, por serem softwares de uso livre e, por fim, por serem para computação de alto desempenho.

Além disto, posteriormente, foi inserido um contra-exemplo, o HDFS, pois se trata de um popular exemplo de um SAD para infraestruturas de *Cloud Computing*, pois a intenção também é avaliar o modelo em um sistema com esta natureza, cruzando os resultados com outros sistemas. Na Tabela 3 pode-se visualizar os cinco (5) sistemas de arquivos distribuídos selecionados para integração e avaliação deste trabalho. Eles serão avaliados para fins de verificação de qual deles se adapta melhor ao modelo proposto, fornecendo assim uma quantidade maior de opções e ainda, fornecendo resultados sólidos de que SAD usar para este cenário proposto.

SAD	Linguagem	API	I/O Paralelo	License	Sistema Operacional
CEPH	C	C, C++, Python, Ruby, FUSE	MPI-IO	GPL	Linux
HDFS	Java	Java, C, HTTP	MPI-IO	Apache License 2.0	Cross-plataform
Lustre	C	POSIX, liblustre, FUSE	MPI-IO	GPLv2	Linux
FhGFS	C e C++	POSIX	MPI-IO	FhGFS EULA	Linux
PVFS	C	C, libpvfs, POSIX	MPI-IO	GPL	Linux

Tabela 3: Comparação dos SAD Selecionados.

2.3.3 CEPH

O CEPH¹ (WEIL et al., 2006a) é um sistema de arquivos que foi desenvolvido pela Universidade da Califórnia em 2006. A motivação para a sua concepção foi ter um sistema de arquivos mais escalável, confiável e eficiente. Um dos pontos importantes do CEPH, de acordo com Weil *et al.* (2006), é que os sistemas de arquivos modernos que adotam armazenamento de dados orientados a objetos, sofrem sérios problemas com as limitações em torno da escalabilidade do serviço (AZAGURY et al., 2003), principalmente devido a distribuição dos servidores de metadados (*Metadata Servers - MDS*) e de seus gerenciadores.

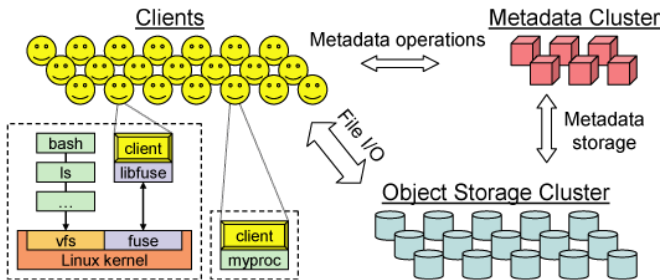


Figura 11: Arquitetura do CEPH (WEIL et al., 2006a).

Em relação a arquitetura do CEPH, que pode ser vista na Figura 11, ela foi construída visando uma escalabilidade mais satisfatória de acesso aos metadados. Assim, foi proposto a separação máxima dos MDS dos servidores de dados OSD (*Object Storage Devices*), utilizando da substituição das

¹<http://www.ceph.com>

tabelas de alocações por uma função aleatória de distribuição de dados, que foi desenhada para utilização em *clusters* heterogêneos e dinâmicos (WEIL et al., 2006a). Além disto, sua arquitetura foi desenvolvida visando sistemas largamente distribuídos, onde as falhas e as inconsistências dos nodos do *cluster* podem ser frequentes. Para contornar estes eventos, o CEPH desenvolveu três características:

1. Desassociação de dados e metadados: tem a função de maximizar a separação dos dados armazenados a partir dos metadados. As operações de metadados são gerenciadas pelo servidor de metadados (MDS), assim enquanto os clientes interagem diretamente com os servidores de dados (OSD) desenvolvem as operações de E/S (Entrada e Saída) (WEIL et al., 2006a). Desta forma, foi introduzido a função de distribuição de dados chamada CRUSH (*Controlled Replication Under Scalable Hashing*) (WEIL et al., 2006b) que é responsável em unir os dados que estão espalhados em determinados objetos e fazer a associação com o dispositivo de armazenamento.
2. Gerenciador dinâmico de metadados distribuídos: o gerenciamento efetivo dos metadados é um fator crítico para todo o desempenho do sistema, por isto, o CEPH utiliza uma arquitetura diferenciada de *cluster* baseada no *Dynamic Subtree Partitioning* que a responsabilidade do gerenciamento do sistema de arquivos entre dezenas ou até centenas de servidores de metadados (MDS) (WEIL et al., 2006a). Este particionamento é realizado de forma hierárquica e tem a função de preservar a localização dos dados, facilitando o processo de atualização e consequentemente no desempenho do sistema.
3. Armazenamento de objetos distribuídos atômicos e confiáveis: sistemas distribuídos de larga escala compostos por milhares de dispositivos são inerentemente dinâmicos. Na maioria dos casos, eles são construídos incrementalmente, crescendo em termos de espaço de armazenamento, ao mesmo tempo que dispositivos são substituídos e desta forma, um grande volume de dados são criados, movidos e deletados. O CEPH deleta a responsabilidade da migração de dados, replicação e a detecção e recuperação de falhas do *cluster* ao OSD que armazena o dado, enquanto em um nível mais alto, o OSD prove um único objeto lógico para os clientes e servidores de metadados (MDS)(WEIL et al., 2006a).

Um dos grandes atrativos para a adoção do CEPH está no seu mapa de hierarquia de objetos. No seu modo de operação, quando um cliente tem a necessidade de realizar um acesso a um determinado arquivo, primeiramente

ele deve fazer uma solicitação ao servidor de metadados. O MDS vai localizar às partes do arquivos nos OSD no mapa de hierarquia de objetos para traduzir o nome do arquivo em um *inode*², que retornará todas as informações sobre a forma de distribuição do arquivo solicitado. Diferentemente de outros sistemas de arquivos distribuídos, no CEPH não há a necessidade de fazer a alocação dos metadados, pois os objetos no sistema de arquivos são construídos a partir de um número *inode* e após isto são distribuídos com a função CRUSH, simplificando o processo e aliviando a carga de trabalho para com os metadados, permitindo assim o MDS gerenciar com grande eficiência, um grande conjuntos de objetos, teoricamente de qualquer tamanho.

2.3.4 PVFS

O *Parallel Virtual File System*, mais conhecido como PVFS³ (LIGON W.B.; ROSS, 1996) é um sistema de arquivos paralelo que teve seu desenvolvimento iniciado em 1993, através de um financiamento da NASA (*National Aeronautics and Space Administration*) que tinha o objetivo de estudar padrões de E/S em aplicações paralelas. No início a motivação era desenvolver um sistema de arquivos paralelos para o PVM (*Parallel Virtual Machine*) e após esta etapa, focou-se na integração com o paradigma MPI.

O *design* do PVFS, em princípio era prover particionamento de dados e arquivos em ambientes distribuídos, focando em alto desempenho, fornecendo uma interface mais próxima possível dos padrões *UNIX-Like*. Isso permitiria que as visões físicas e lógicas do arquivos fossem independentes, facilitando assim o acesso aos dados. O desenvolvimento do PVFS teve como foco principal o escalonamento da E/S dos discos, para quando vários clientes acessam o mesmo arquivo, simultaneamente (LIGON W.B.; ROSS, 1996).

Em 1999, foi proposta uma reformulação no código do PVFS, que culminou primeiramente no PVFS2000 e logo após foi renomeado para PVFS2. Ele tinha como propósito realizar melhorias no PVFS, e tinha como foco abordar os problemas de modularidade de código, flexibilidade de configuração e um melhoramento significativo na integração com o MPI-IO (*Message Passing Interface – Input/Output*) (KUNKEL; LUDWIG, 2007). Na Figura 12 é possível visualizar a arquitetura do PVFS2.

Um importante aspecto do PVFS2 é a flexibilidade quanto a implementação de diferentes configurações na sua arquitetura. É possível realizar

²Inode pode ser definido como uma estrutura de dados de um sistema de arquivos *UNIX-Like*, sendo este responsável pelo armazenamento de metadados um arquivo, como por exemplo localização, permissões, etc.

³<http://www.pvfs.org>

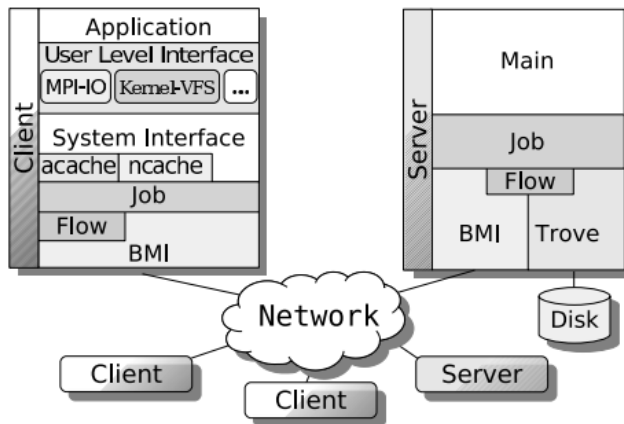


Figura 12: Arquitetura do PVFS2 (KUNKEL; LUDWIG, 2007).

definições de quantos nodos o sistema terá, até definições de quantos blocos o arquivo será dividido. O núcleo de sua arquitetura é composto por componentes clássicos de um sistema de arquivos distribuído, sendo um servidor de metadados, um servidor de dados e um cliente. Em seu modo de operação, o servidor de metadados é consultado e o mesmo retorna uma tabela de localização dos blocos, nos diferentes nodos, que estão persistindo o dado procurado. Após isto o servidor de metadados acessa paralelamente os nodos identificados na tabela para resgatar o arquivo solicitado.

2.3.5 Lustre

O Lustre File System⁴ é um sistema de arquivos distribuído e paralelo que teve o início de seu desenvolvimento dado pelo pesquisador Peter Braam, na Universidade de Carnegie Mellon, em 1999. O seu primeiro *release* foi apresentado em 2003, em sua versão 1.0 foi lançada pela empresa “Cluster File Systems”, de propriedade do pesquisador. O grau de sofisticação e inovação que o Lustre trouxe em sua arquitetura (BRAAM et al., 2004), despertou interesse da Sun Microsystems, que adquiriu a empresa em 2007. Entretanto, devido a compra da Sun Microsystems pela Oracle, o desenvolvimento do Lustre ficou parado por algum tempo e no final de 2010, os principais engenheiros e desenvolvedores do Lustre foram contratados pela Wham-

⁴<http://www.lustre.org>

Cloud⁵, que atualmente tem um contrato com a Oracle para a continuação do desenvolvimento do Lustre. Na Figura 13 é possível visualizar a arquitetura do Lustre File System em um *cluster*.

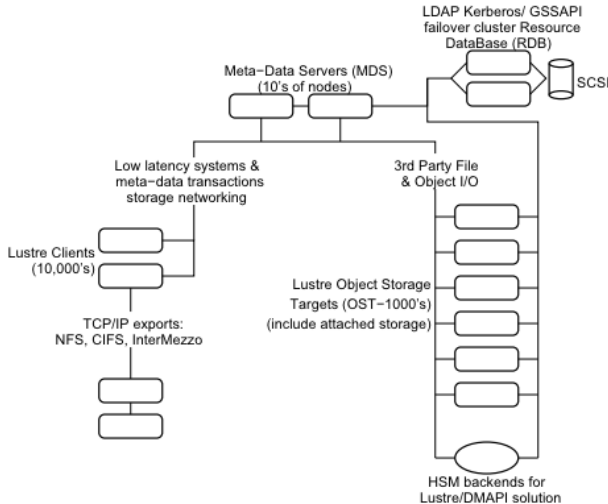


Figura 13: Arquitetura do Lustre (BRAAM et al., 2004).

Uma importante característica do Lustre é a *Management Server* – MGS, que tem a função de armazenar todas as informações sobre configurações dos sistemas de arquivos da infraestrutura e ainda, provê estas informações para ou outros componentes do Lustre. Sobre os componentes que formam a arquitetura, podem-se citar o *Metadata Server* – MDS, o *Metadata Target* – MDT, o *Object Storage Servers* – OSS, o *Object Storage Target* – OST e os clientes de fato. Na Figura 14 pode-se visualizar a arquitetura de componentes do Lustre e as relações entre estes componentes.

O MDS tem a função de armazenar os metadados em um ou mais MDTs disponíveis para os clientes do Lustre. Cada um destes MDS, tem o papel de gerenciar os nomes e diretórios no sistema de arquivos distribuídos, e ainda, prover o gerenciamento da rede para um ou mais MDTs. Por outro lado, o MDT é responsável por armazenar metadados, como por exemplo nomes de arquivos, diretórios, permissões e *layouts* para um MDS. Cada um dos sistemas tem pelo menos um MDT, entretanto, podem haver configurações compartilhadas onde um MDT pode ser compartilhado entre vários MDS,

⁵www.whancloud.com

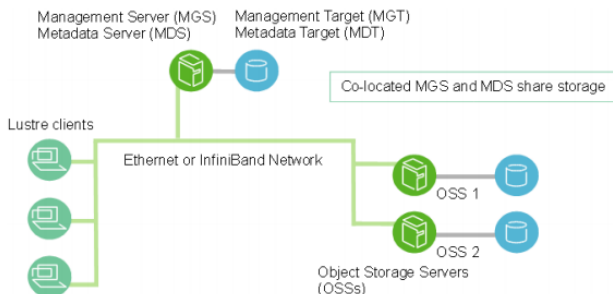


Figura 14: Componentes do Lustre (WHANCLOUD, 2013).

porém só pode haver um acesso por vez ao MDS, gerando desta forma um “gargalo” no sistema de arquivos (WHANCLOUD, 2013).

Em relação aos objetos de um sistema de arquivos Lustre, o OSS tem o papel de prover o serviço de E/S e gerenciamento de rede para um ou mais OSTs. Normalmente em um arquitetura Lustre, os servidores OSS podem oferecer o serviço para até oito OSTs, com até 16 *terabytes* cada. Em arquiteturas típicas, a configuração de um MDT dedicado, dois ou mais OSTs em cada nodo OSS e um cliente para cada. Já o OST é onde os arquivos dos usuários são armazenados em um ou mais objetos, sendo que cada um destes objetos estão em um OST separado no sistema de arquivos (WHANCLOUD, 2013). Um importante aspecto é a flexibilidade pois a quantidade de objetos por arquivo pode ser configurado, maximizando assim o desempenho. Em relação aos clientes, estes tem a função natural em uma arquitetura de sistema de arquivos distribuídos, ou seja, eles usam a infraestrutura.

2.3.6 FhGFS - *The Fraunhofer Parallel Cluster File System*

O *The Fraunhofer Parallel Cluster File System*, ou FraunhoferFS ou ainda, como é mais conhecido o FhGFS⁶ é um sistema de arquivos paralelo, que foi desenvolvido pelo Centro de Competências Fraunhofer para *High Performance Computing* (FRAUNHOFERFS, 2013). A arquitetura de metadados distribuídos que foi desenvolvida para suportá-lo foi desenhada para proporcionar uma alta escalabilidade e fácil flexibilidade (FRAUNHOFERFS, 2013). O FhGFS é distribuído livre de taxas, e ainda conta com um suporte comercial que é feito diretamente com o Instituto Fraunhofer, ou ainda, por

⁶<http://www.fhgfs.com>

algum parceiro internacional.

A documentação desenvolvida para o sistema é voltada praticamente para cientistas, desta forma ficando limitada a projetos de pesquisa que são voltados para este sistema. Em termos de configuração e implementação do sistema, ela é de fato mais simples, se comparada aos sistemas de arquivos distribuídos convencionais, que exigem um alto grau de conhecimento sobre a arquitetura, diferente disto, o FhGFS abstrai isto para o usuário. Na Figura 15 é possível visualizar a arquitetura do FhGFS, que se assemelha muito com uma arquitetura clássica de um sistema de arquivo distribuído e paralelo, sendo composto por um servidor de metadados, um servidores de dados e um ou mais clientes realizando acessos.

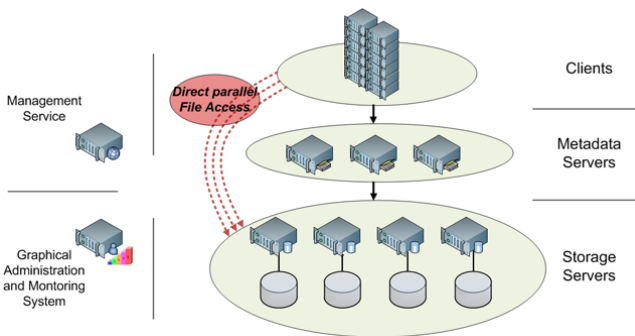


Figura 15: Arquitetura do FhGFS (FRAUNHOFERFS, 2013).

Na arquitetura citada, existem dois componentes que se destacam, sendo o servidor para gerenciamento da infraestrutura e um serviço de administração e monitoramento, que provê uma interface via Web para instalação, configuração e monitoramento do sistema. O *core* do sistema foi desenvolvido na linguagem de programação Java⁷, tornando o sistema multiplataforma. Como adição, o FhGFS não tem seu código fonte disponível, somente foi disponibilizado o cliente de acesso sob a licença GPL. O FhGFS tem alguns interessantes componentes chave, entre eles pode-se citar (FRAUNHOFERFS, 2013):

1. Distribuição dos Conteúdos dos Arquivos e dos Metadados: de acordo com seus idealizadores, o conceito fundamental da arquitetura do FhGFS é evitar fortemente “gargalos”. Desta forma, a divisão dos arquivos e metadados em múltiplos servidores de armazenamento é somente um dos conceitos, pois uma outra importante característica é a distribuição

⁷<http://www.java.com>

dos metadados no sistema de arquivos, em múltiplos servidores de armazenamento. Desta maneira, os gargalos são evitados, pois haverá múltiplos pontos de acesso ao sistema, o que melhora significativamente os tempos de resposta quando há casos de centenas ou milhares de arquivos concorrendo ao mesmo metadado.

2. Tecnologias de *High Performance Computing*: devido a sua natureza, o seu núcleo foi construído de um esquema escalável e *multi-thread*, com suporte nativo a Infiniband⁸. Além de possuir suporte nativo a Ethernet, ou qualquer outro tipo de rede baseado em TCP (*Transmission Control Protocol*), ele pode funcionar simultaneamente com o Infiniband, gerenciando a troca de conexões entre as tecnologias em caso de alguma falha.
3. Clientes e Servidores em Qualquer Máquina: não há uma especificação de distribuição de sistema operacional Linux, pois no caso, tanto o cliente, quando o servidor do FhGFS podem rodar ao mesmo tempo na mesma máquina, desta forma, incrementando um maior desempenho para pequenos *clusters* ou pequenas redes. Ainda, não há a necessidade de uma partição dedicada no servidor, pois ele usa as partições existentes. Um ponto importante diz respeito com sua compatibilidade com qualquer tipo de sistema de arquivos nativo para o sistema operacional Linux.

2.3.7 HDFS - Hadoop File System

O *Hadoop File System* – HDFS (BORTHAKUR, 2007) é um sistema de arquivos distribuído e faz parte do conjunto de aplicações do Hadoop⁹, servindo assim como *backend* para as informações. Ele foi organizado e concebido pela Apache Foundation¹⁰ que buscou desenvolver um sistema, que fosse ao mesmo tempo escalável e confiável, e tinha como objetivo fim suprir a necessidade de sistemas largamente distribuídos com grande vazão, sempre priorizando a baixa latência (SHVACHKO et al., 2010).

Em sua arquitetura, o HDFS armazena os dados e metadados em servidores distintos, assim como o PVFS, Lustre e FhGFS. O seu servidor dedicado de metadados é chamado de *NameNodes* e os seus servidores de dados são chamados de *DataNodes*. Um importante aspecto da arquitetura do HDFS

⁸InfiniBand é um tipo de barramento de comunicação ou dispositivo de comunicação que pode chegar até 2.5GBps.

⁹<http://hadoop.apache.org/>

¹⁰<http://www.apache.org>

é que diferentemente do Lustre e do PVFS, o HDFS não usa um esquema de proteção de dados, como por exemplo RAID (*Redundant Array of Independent Disks*), para proporcionar durabilidade aos dados. Ao invés disto, ele usa um esquema de replicação de dados para proporcionar a segurança necessária para o sistema de arquivos (SHVACHKO et al., 2010). Na Figura 16 é possível visualizar a arquitetura do HDFS.

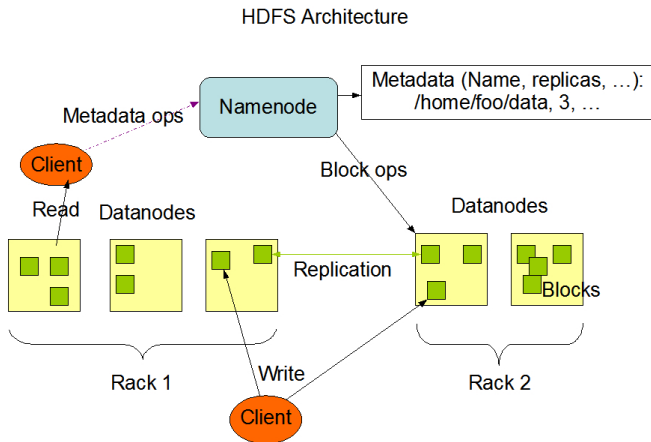


Figura 16: Arquitetura do HDFS (BORTHAKUR, 2007)

Em seu modo de operação, o acesso aos dados no nodos foi construído sob o paradigma *write-once-read-many* (BORTHAKUR, 2007), ou seja, somente um cliente poderá escrever um arquivo. Desta forma não há simultaneidade de escrita, o que proporciona um desempenho mais satisfatório, já que os mecanismos de consistência de dados são simplificados. Entretanto, quanto a leitura dos dados, eles podem ser lidos “N” vezes, concorrentemente, já que os dados são replicados pra outro nodos, evitando assim “gargalos”.

Um importante componente do HDFS é o mecanismo de MapReduce (DEAN; GHEMAWAT, 2008). Este componente da arquitetura tem a função de processar e gerar grandes volumes de dados, visando a redução dos espaços de busca e por fim, otimizando o desempenho do sistema. Ele é dividido em dois passos, onde há o “mapa”, onde o *NameNode* recebe um arquivo, divide-o em pequenos pedaços e os envia para os *DataNodes*, que repetem a mesma operação formando assim uma estrutura hierárquica de múltiplos níveis e um mapa a ela relacionada.

2.4 FORMATOS DE DADOS

Um formato de dados pode ser definido como o formato de um arquivo de dados e como ele está codificado. Geralmente esta terminologia está associada a outros conceitos, como por exemplo: tipo de dado, formato de arquivos, *container* de dados, formato de conteúdo, modelos de dados, entre outros. O conceito fundamental aponta para “como” um dado será explicitado para as aplicações.

Estes dados, podem ser de muitas naturezas e a partir destas, empresas e desenvolvedores em geral determinam os formatos de dados para tal conteúdo. Por exemplo, um arquivo multimídia de vídeo pode ser codificado de várias formas, e cada uma destas formas, tem seu próprio modelo de dados e um propósito específico de existir. Neste contexto, pode-se desejar mais resolução para um determinado conteúdo de vídeo, desde forma podendo-se usar o formato MP4 ou MKV. Este é somente um exemplo de como dados podem ser embutidos dentro de formatos.

Na literatura há uma diversidade de formatos de dados, para os mais variados fins, propósitos e aplicações. Já foram produzidos uma quantidade significativa de estudos acerca de formatos de dados, sendo focados no meio científico ou comercial. Entretanto, no contexto comercial, as empresas de software definem seus próprios formatos ou usam formatos já estabelecidos. Alguns exemplos destes formatos comerciais podem ser:

- **Texto:** DOC, DOCX, ODT, TEX, etc.
- **Imagens:** PNG, JPG, JPEG, EPS, etc.
- **Vídeo:** AVI, MKV, MP4, WMV, etc.
- **Som:** WAV, MP3, WMA, FLAC, etc.

No contexto científico, há também uma extensa variedade de formatos já foram criados. Exemplos destes podem ser visualizados no trabalho (GRAY et al., 2005) e (GEORGIEVA; GANCHEVA; GORANOVA, 2009) onde são vistas uma variedade de formatos científicos relevantes para a comunidade científica. Estes formatos são XML (*Extensible Markup Language*), XDF (*Extensible Data Format*), CGNS (*General Notation System*), XSIL (*Extensible Scientific Interchange Language*), RDF (*Resource Description Framework*), CDF (*Common Data Format*), NetCDF (*Network Common Data Format*) e o HDF (*Hierarchical Data Format*). De acordo com (GEORGIEVA; GANCHEVA; GORANOVA, 2009), estes formatos podem ser baseados em XML (XML Schema, RDF, XSIL, CGNS e o XDF) ou podem ser para fins científicos (CDF, HDF e o NetCDF).

Cada um desses formatos tem suas próprias particularidades, modos de operação diferentes, representações de dados distintas, modelos de dados diversos, entre outras características que os diferenciam e até algumas delas que se interpõem. Entretanto, como propósito fim, todos eles se convergem em tarefas intensivas de armazenamento, representação e gerenciamento de dados.

Para os formatos de dados que são baseados no XML, há uma quantidade de estudos relacionados à persistência de dados médicos, de várias naturezas. Exemplos destes estudos podem ser vistos em (HU; LEE, 2004), (BLAZONA; KONCAR, 2007), (YOU; YAO, 2005), (LEE; HU, 2003), (LEIF; LEIF, 2011), entre outros. Estes exemplos foram citados somente para ilustrarem a variedade e a maturidade dos estudos que foram e são realizados baseados em tecnologias que tem o XML como base e imagens médicas DICOM como foco de estudos.

Quanto à essas tecnologias, é consenso na comunidade acadêmica que o XML pode não ser tão eficiente para gerenciar grandes volumes de dados, porque a sobrecarga de processamento e armazenamento podem tornar-se proibitivos. Uma solução muito eficiente para contornar este problema é implementar o XML para representar metadados, que tem natureza mais complexa e deixar a tarefa de armazenamento para uma outra tecnologia que use formato binário e seja mais eficiente (GEORGIEVA; GANCHEVA; GORANOVA, 2009). Há outra quantidade significativa de estudos, como por exemplo (TIAN et al., 2002), que tentam anular ou contornar esta ineficiência para determinados tipos de aplicações. Devido ao fato de uma abrangência significativa de estudos acerca de tecnologias baseadas em XML no âmbito da medicina, mas especificamente para imagens médicas e sistemas em geral, avaliações destas tecnologias não são o foco deste trabalho.

Para os formatos de dados científicos, em formato binário (CDF, NetCDF e HDF), poucos estudos acadêmicos sérios, com metodologias, avaliações de modelos e protótipos foram desenvolvidos, visando modelos de persistência de dados de longo prazo para imagens médicas. Exemplos destes podem ser vistos em (MACEDO et al., 2009), (MAGNUS et al., 2012), (SOARES et al., 2012), (MACEDO et al., 2011) e (SOARES et al., 2013). Estes trabalhos, em sua totalidade, tratam cenários de armazenamento distribuídos de imagens DICOM, usando NetCDF ou o HDF5, foco principal desta tese.

Esses tipo de formato de dados funcionam com esquemas bem elaborados de bibliotecas para suporte aos desenvolvedores, juntamente com um formato de dados auto-descritivo (*self-describing*). Eles formam uma camada auxiliar aos *middlewares* e integrado a eles para formar uma camada de abstração que tem a função de proporcionar aos desenvolvedores níveis mais elevados de abstração no processo de desenvolvimento de software, pro-

porcionando um melhor gerenciamento dos dados e do *workflow* como um todo. Este tipo de esquema também é chamado de biblioteca de alto nível ou *high-level libraries*. Na Figura 17 é possível visualizar a localização destas bibliotecas na arquitetura.

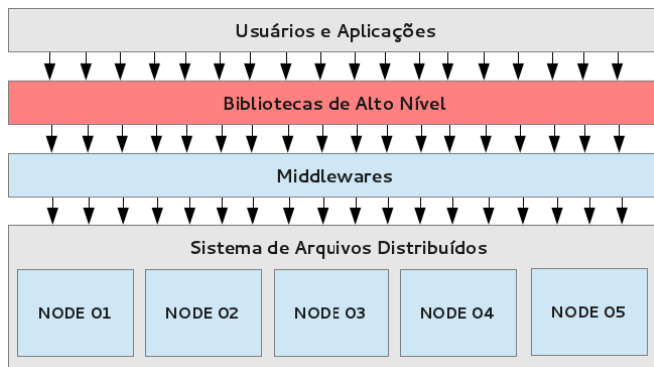


Figura 17: Representação de um SAD com Bibliotecas de Altos Nível.

O *Hierarchical Data Format* e o *Network Common Data Format* são comumente usados em múltiplas comunidades científicas, incluindo astronomia, física de fusão e medicina (SHASHARINA et al., 2007). Em relação ao *Common Data Format*, da NASA, não foram encontradas referências de trabalhos científicos que se propusessem a persistir imagens médicas DICOM. Este fato pode ser notado devido a natureza do NetCDF, que foi inspirado no CDF e conta com um suporte mais elaborado para um grande número de propósitos.

No caso do NetCDF, no trabalho de (MAGNUS et al., 2012), foi realizada uma análise do formato de dados, como camada distribuída de persistência para imagens médicas DICOM. Além disso, foi realizada uma modelagem de dados e todo o processo de construção de uma arquitetura para persistir os mesmos. Foram comparadas a abordagem proposta com uma abordagem usando HDF5 e um SGBD convencional (*PostgreSQL*). Como resultado, foi constatado que o NetCDF tem um tempo de latência maior, se comparado com os outros dois sistemas. Desta forma, após esta avaliação sobre este formato de dados específico, nesta tese, não serão abordados aspectos relacionados ao NetCDF.

No trabalho de (DOUGHERTY et al., 2009) intitulado *Unifying Biological Image Formats with HDF5* é proposta uma discussão sobre a adoção do formato HDF5 como plataforma básica para imagens biológicas. Este trabalho teórico foi desenvolvido por vários pesquisadores de centros de ex-

celência, que concluíram que o assunto não é se as ciências biomédicas vão adotar o HDF5 como um *framework* para suporte a imagens, pois isto aos poucos já esta acontecendo, mas quanto tempo isso vai demorar e quão difícil será integrar os muitos tipos de dados médicos sobre o formato HDF5.

Dentro desse contexto, nesta tese, o formato de dados HDF5 foi definido como formato padrão para este modelo de armazenamento hierárquico proposto. Esta decisão foi tomada devido a alguns fatores, entre eles podem-se citar: à representação hierárquica natural de um exame médico baseados no padrão DICOM; devido a trabalhos científicos anteriores realizados que atestam e/ou indicam que o HDF5 pode persistir imagens médicas DICOM (MACEDO et al., 2009), (SOARES et al., 2012), (MACEDO et al., 2011) e (SOARES et al., 2013). O HDF5 é uma tecnologia única que torna possível o gerenciamento de grandes e complexas coleções de dados (GEORGIEVA; GANCHEVA; GORANOVA, 2009), que é exatamente o caso de bases de dados de imagens médicas.

O HDF5 também é amplamente suportado por uma quantidade significativa de plataformas, que vão desde pequenos computadores até máquinas para computação de alto desempenho. Ele pode ser acessível via uma grande quantidade de linguagens de programação, como por exemplo, Fortran, Java, C e C++, sendo estas duas últimas a linguagem de programação que o servidor de imagens DICOM do estudo de caso (CyclopsDCMServer) é escrito. O HDF5 é altamente escalável, sem limitação alguma relacionada a este aspecto.

Esse formato ainda é regido por uma licença BSD (*Berkley Software Distribution*), o que torna possível, por exemplo, fechar o código-fonte e comercializar a aplicação desenvolvida. Por fim, o HDF5 tem um grande grupo de pesquisadores e cientistas que fazem parte do *HDF Group* (HDFGROUP, 2013), o que o torna constantemente atualizado e melhorado. Outros formatos de dados, como por exemplo o CDF ou o NetCDF também satisfazem estes critérios, entretanto, o HDF5 é o que apresenta o melhor equilíbrio entre eles (JACKSON; SIMMONS; GRAEF, 2010). Além disto, há outras importantes características do formato de dados HDF5, que são importantes para a persistência de dados desta natureza. Entre estas podem ser citadas:

- **Acesso Randômico:** aponta para a possibilidade de recuperação de dados específicos, dentro de um conjunto de dados complexo. Um exemplo deste acesso pode ser dado através da recuperação de somente a imagem (*pixel data*) ou ainda seus metadados.
- **Portabilidade e Multiplataforma:** indica que os arquivos HDF5 podem ser utilizados em uma grande quantidade de plataformas computacionais e arquiteturas diferentes, sendo que os dados podem ser por-

tados entre estas múltiplas plataformas, naturalmente, sem alterações.

- **Armazenamento de Dados:** aponta para a possibilidade de se armazenar qualquer quantidade de dados, sem limitação alguma. Certamente, as tarefas de gerencia destes dados devem ser otimizadas, mas não há limitação para tamanhos de conjuntos de dados que podem ser persistidos. No caso de bases de dados de imagens médicas, este é um importante requisito.
- **Preservação de Dados:** indica que o HDF5 foi concebido para persistência de dados de longo prazo (*long-term data preservation*). Este é um requisito primário, visto que como já mencionado, o histórico de um paciente no Brasil deve ser armazenado por, no mínimo, 20 anos.

2.4.1 Hierarchical Data Format - HDF5

O *Hierarchical Data Format* (HDF) foi criado pelo NCSA (*National Center for Supercomputing Applications*), na University of Illinois em Urbana-Champaign, em conjunto com a NASA (*National Aeronautics and Space Administration*) e atualmente ele é aplicado em diversas áreas, como por exemplo: simulação computacional em grids computacionais (SHASHARINA et al., 2007), computação paralela (YU et al., 2006), gerenciamento de dados geográficos (LEE; SPENCE, 2002), dentre outros.

O HDF é composto por um modelo de dados, uma biblioteca e um formato de dados para armazenamento e gerenciamento de dados. O HDF suporta uma variedade ilimitada de tipos de dados e é projetado para processos de leitura e escrita flexível e eficiente para grandes volumes de dados. O HDF atualmente encontra-se na versão de número 5, chamada de HDF5. O HDF5 é portátil e extensível, permitindo às aplicações se integrarem de forma simples. A tecnologia HDF5 também inclui ferramentas e aplicações para gerenciamento, manipulação, visão e análise de dados no formato HDF5 (FOLK et al., 2011).

O objetivo do HDF é prover armazenamento eficiente de grandes volumes de dados científicos expressos de maneira hierárquica para acesso por longo período de tempo (FOLK; POURMAL, 2010). As operações de criação, manipulação e modificação dos dados, são realizadas por meio de uma API nativa em linguagem Fortran ou "C". Há duas versões, conceitualmente diferentes, que recebem atualizações regularmente, o HDF4, para compatibilidade de sistemas legados, e o HDF5, atual versão. O uso do HDF5 é encorajado para novos projetos uma vez que soluciona uma série de limitações do HDF4, principalmente o tamanho máximo de 2GB de um arquivo no formato

HDF4 e aperfeiçoamento do código para ser mais eficiente, dentre outras características (PRADO, 2012). Nas próximas subseções as características do HDF5 serão expostas, para um melhor entendimento deste formato de dados.

2.4.2 Estrutura do HDF5

A estrutura de um arquivo HDF5 possui cinco (5) estruturas simples, porém essenciais que formam o seu modelo de dados e a base de seu funcionamento interno via biblioteca. Estes cinco componentes são: *HDF5 Dataset* (conjunto de dados), *HDF5 Groups* (grupos), *HDF5 Datatypes* (tipos de dados), *HDF5 Links* (atalhos) e o *HDF5 Attributes* (atributos) (FOLK et al., 2011). A seguir será apresentado em detalhes cada um destes componentes que formam a arquitetura do HDF5.

2.4.2.1 HDF5 Datasets

Os *HDF5 Datasets* podem ser entendido como uma matriz de variáveis, cujos elementos são dados logicamente representados como uma matriz multidimensional (FOLK et al., 2011). Ainda, ele também pode ser entendido como um objeto composto de uma coleção de elementos de dados. Ele é composto por um cabeçalho uma matriz de dados, que pode ser unidimensional ou multidimensional, dependendo de como o conjunto de dados foi definido. O cabeçalho é composto por um nome (*name*), um tipo de dados (*datatype*), um espaço de dados (*dataspace*) e um esquema (*layout*). Na Figura 18 é possível visualizar uma representação lógica de um *dataset*.

O componente “nome” (*name*) é usado para identificação de cada um dos conjunto de dados. O componente “tipo de dados”(*datatype*) pode ser do tipo atômico ou composto, sendo que o primeiro pode ser representado por caracteres ou números e o segundo tem a função de encapsulá-los. O componente “espaço de dados” (*dataspace*) pode ser definido como um metadado que tem a função de representar a dimensão dos conjuntos de dados, os quais podem suportar tamanhos infinitos, dependendo do esquema de armazenamento definido (FOLK et al., 2011). Por último, o “esquema” (*layout*) pode ser definido como a forma que os dados serão armazenados dentro do *container* de dados. Por padrão, este esquema usa contínuo, mas pode ser alterado para armazenar por tamanho de blocos específicos, que podem ser pré-definidos pelos desenvolvedores.

Um conjunto de dados HDF5 pode crescer com ou sem limites máximos de extensão. Isto vai depender diretamente do esquema de armazenamento

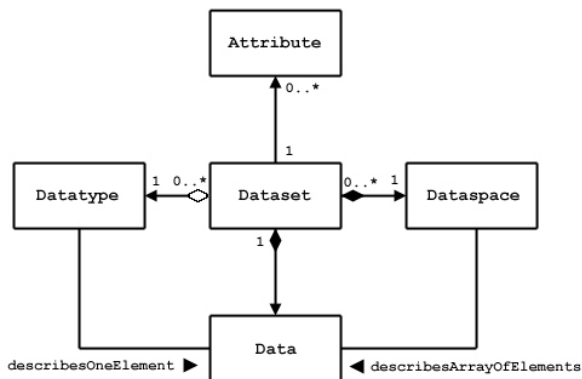


Figura 18: Representação de um *HDF5 Dataset* (HDFGROUP, 2013).

que foi pré-definido. Assim, dependendo do esquema de armazenamento definido, é possível armazenar uma quantidade ilimitada de dados dentro de um *container* HDF5 (FOLK et al., 2011). Atualmente, as seguintes opções de esquema são suportadas:

- **Contínuo (*Contiguos*):** define que os elementos da matriz de dados estão representados com uma única sequência em um conjunto de matrizes HDF5.
- **Fragmentada (*Chunked*):** define que os elementos da matriz estão representados como sub-matrizes normais de tamanho fixo, ou ainda, fragmentadas.
- **Compacto (*Compact*):** define que os pequenos conjuntos de dados HDF5 (até 64KBs) são representados de uma forma que todos os elementos da matriz podem ser lidos como parte de uma matriz de metadados ou um cabeçalho.

2.4.2.2 HDF5 Groups

Os grupos HDF5 podem ser entendidos como representações de associações entre zero ou mais itens HDF5, como por exemplo em um conjunto de dados HDF5 (FOLK et al., 2011). Estas representações na prática são formadas por *containers* que contém outros grupos ou ainda, outros conjuntos

de dados. Uma analogia para esta representação pode ser dada através de um sistema de arquivos clássico do UNIX, onde pode-se encontrar os diretórios. Estes diretórios podem estar vazios, ou ainda, podem conter mais diretórios ou arquivos, formando assim uma árvore de diretórios e arquivos.

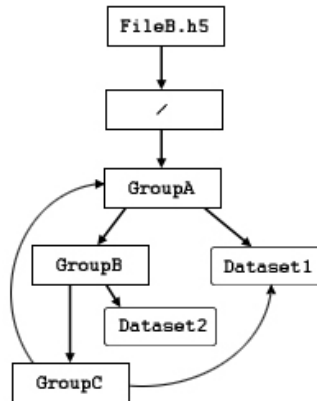


Figura 19: Representação de um HDF5 *Group* (HDFGROUP, 2013).

Na Figura 19 é possível visualizar um grupo HDF5 (HDFGROUP, 2013). A diferença da analogia de sistemas de arquivos UNIX dos próprios grupos HDF5, é que estes últimos são interligados através de um grafo direcionado, permitindo assim, traçar referências circulares entre objetos. Os grupos HDF5 são de grande importância para arquitetura como um todo, visto que eles têm função de auxiliar na organização da estrutura em formato de árvore e assim, hierárquica de um arquivo HDF5. Como informação complementar, e para melhor expor os grupos HDF5, todo arquivo em formato HDF5 tem um grupo inicial denominado, grupo raiz ou ainda, (*root group*) (“/”).

2.4.2.3 HDF5 *Datatypes*

Um tipo de data (*datatype*) HDF5 implementa um mecanismo flexível, extensível e portátil para armazenamento de elementos de dados, determinando assim, a forma como os elementos serão interpretados, e ainda, para transferência de dados entre esquemas compatíveis (HDFGROUP, 2013). Um conjunto de dados tem um único tipo de dados do qual aplica todo o elemento de dados. No momento da criação de um conjunto de dados, também ocorre a definição do tipo de dados a ser armazenado e após o processo de criação,

o tipo de dados não pode ser modificado. Assim, o tipo de dados (*datatype*) de um conjunto de dados (*dataset*) é imutável (HDFGROUP, 2013).

O HDF5 provê um sistema de tipos de dados de ilimitada flexibilidade. Uma matriz HDF5 tem dois pontos-chave: um espaço de dados (*dataspace*) que descreve a forma da matriz e um tipo de dados (*datatype*) que descreve o tipo do elemento de dados. Dezenas de famílias ou classes de tipos de dados HDF5 atualmente são suportados, sendo eles: inteiro, ponto flutuante, *string*, *bitfield*, *opaque*, *compound*, *reference*, *enum*, *variable-length* e *array* (FOLK et al., 2011) (HDFGROUP, 2013).

2.4.2.4 HDF5 Links

Atualmente o HDF5 suporta somente links (atalhos) unidirecionais e *links* com única fonte e destino (*single source/single destination*). Além disto, a fonte deve ser um grupo HDF5. Na Tabela 4 é fornecida uma visão geral dos tipos de *links* proporcionados pelo HDF5. O *link* “*soft*” e “*external*” são chamados de *links* simbólicos devido ao fato de que eles referenciam os seus destinos via um caminho (*path*) ou pela combinação de nome e caminho HDF5. Diferentemente, os “*hard*” *links* não modificam o estado de seus destinos (FOLK et al., 2011). Ainda há o *links* que são definidos pelo usuário (*user-defined*) que são utilizados comumente para dar mais flexibilidade na construção da árvore HDF5.

Tipo de Link HDF5	Destino
Hard Link	Endereço / Identificador
Soft Link	Caminho HDF5
External Link	Arquivo / Caminho HDF5
User-Defined Link	User-Defined

Tabela 4: Tipos de *Links* - HDF5.

2.4.2.5 HDF5 Attributes

O modelo de dados do HDF5 define um mecanismo chamado HDF5 *Attributes* (atributos HDF5) que podem ser utilizados para descrever objetos HDF5, sendo eles os conjuntos de dados (*datasets*), grupos (*groups*) e tipos de dados (*datatypes*) (FOLK et al., 2011). Estes atributos tem como característica o fato de serem análogos aos tipos primitivos de linguagens de programação. Por questões de implementação e lógica, o nome de um atri-

buto HDF5 deve ser único no escopo de um arquivo HDF5. Um atributo HDF5 não suporta tipos de armazenamento com compressão ou fragmentados e ainda, não há acesso a atributos de forma parcial e não podem ser compartilhados entre diferentes estruturas HDF5.

2.4.3 Arquitetura do HDF5

Na Figura 20 é possível visualizar um esquema da arquitetura de software que o HDF5 implementa. Tipicamente, as aplicações, ferramentas e APIs de alto nível acessam arquivos HDF5 por meio da API da biblioteca do HDF5 (FOLK et al., 2011). Uma analogia para este tipo de abordagem, para os usuários de sistemas gerenciadores de bancos de dados relacionais, é que estas camadas seriam como interfaces do estilo ODBC (*Open Database Connectivity*).

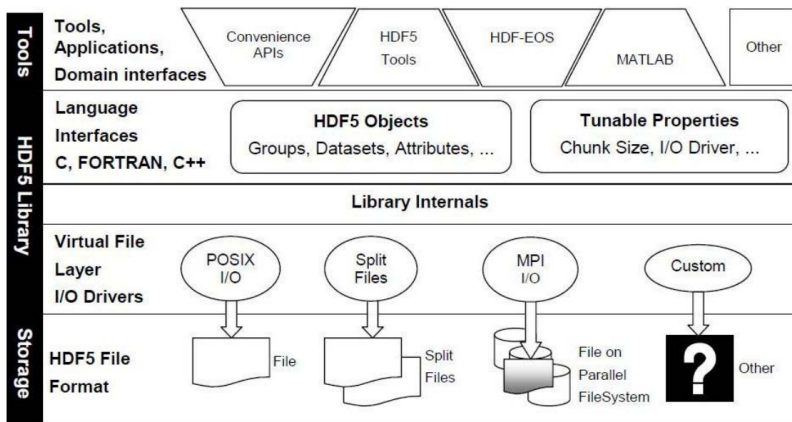


Figura 20: Arquitetura do HDF5 (FOLK et al., 2011).

Como definido nas seções sobre os componentes do HDF5, na criação de um conjunto de dados, os seguintes parâmetros devem ser definidos: tipo de elemento de dados e um tipo de dados específico do HDF4, a forma da matriz multidimensional, inclusive com seu tamanho máximo e um espaço de dados do HDF5. Ainda, deve ser definido um esquema de armazenamento de dados, que pode ser do tipo contínuo, fragmentado ou compacto, que vai depender diretamente do tipo de dados que esta sendo armazenado. E por fim, devem ser definidos os filtros e o esquema externo de acesso ao conteúdo (HDFGROUP, 2013).

Sobre a Figura 20, é possível visualizar três camadas horizontais, que compõem toda a arquitetura do HDF5. As camadas são: ferramentas (*tools*), biblioteca HDF5 (*HDF5 Library*) e a camada de armazenamento (*storage*). Cada uma destas camadas, tem funções bem definidas dentro da arquitetura, auxiliando em todo o processo de modelagem, construção dos arquivos, armazenamento e recuperação de dados.

A primeira camada, a de ferramentas, é responsável pela integração com as camadas superiores de uma arquitetura de software ou até aplicações que estão nela contidas. Nesta camada, ainda podem estar contidas as aplicações e as interfaces para domínios mais específicos que podem ser desenvolvidas e acopladas livremente. Como exemplo, podemos citar a aplicação *HDFView* (HDFGROUP, 2013) que auxilia no processo de visualização de arquivos HDF5. Outro exemplo que pode ser citado é a interface com a aplicação MATLAB, para armazenamento de dados de simulações, em sua maioria numéricas, que podem gerar *terabytes* de dados.

A segunda camada, a da biblioteca HDF5, é composta por três (3) subcamadas, sendo elas a camada de interface com linguagens de programação (C++, Python, Fortran, etc), as bibliotecas internas e a camada virtual de arquivos. Na primeira delas, a de interface com as linguagens de programação suportadas, elas são responsáveis pela integração da biblioteca com aplicações que serão acopladas na arquitetura. Nela está contida uma API completa para interação com o restante da arquitetura. A segunda camada, biblioteca interna, esta o núcleo (*core*) da arquitetura do HDF5. Na terceira camada, estão as camadas virtuais de arquivos. Esta camada é responsável em prover as interfaces necessárias de escrita e leitura para a biblioteca do HDF5 e para as camadas inferiores da arquitetura. Por exemplo, a uma interface virtual de arquivos para o MPI-IO, sendo nesta interface podem ser acopladas sistemas de armazenamento que suportem procedimentos de escrita e de leitura em paralelo.

Por fim, a terceira camada, a de armazenamento, podem estar de fato os arquivos no formato HDF5. Esta suposição é levantada, pois do ponto de vista arquitetural, ainda podem haver algumas camadas abaixo desta, entretanto estas, estão fora do escopo do formato de dados HDF5. Como primeiro exemplo, no processo de criação de um arquivo HDF5, se este for ser armazenado localmente, em um sistema de arquivos convencional em um sistema operacional que suporte POSIX-IO, tipo um EXT4 do Linux, este processo é finalizado neste momento. Como segundo exemplo, ainda há outras camadas virtuais de arquivos que podem ser utilizadas para direcionar o armazenamento dos arquivos, como por exemplo o MPI-IO ou outra qualquer que venha a ser desenvolvida e acoplada a arquitetura.

2.4.4 Desempenho

Em relação ao desempenho do HDF5, ele tem como característica primária ser um formato de dados de alto desempenho, devido ao seu projeto (*design*) interno, seu modelo de dados e de acesso e ao seu modo de operação. Sua biblioteca permite ainda, que desenvolvedores utilizem técnicas de paralelização de processamento (CPU) e sistemas de arquivos para atingir um melhor desempenho de leitura e escrita. Sobre esta paralelização, a nível de armazenamento, aplicações paralelas baseadas em MPI podem acessar arquivos HDF5 de forma paralela, permitindo assim acesso rápido para grandes conjuntos de dados HDF5 (FOLK et al., 2011). Ainda, podem ser acoplados sistemas de arquivos distribuídos que funcionam de forma paralela, como por exemplo Lustre, PVFS, FhGFS, etc.

De acordo com (FOLK et al., 2011) alguns desenvolvedores tem tido sucesso implementando esquemas de indexação de conjuntos de dados HDF5. Ainda, de acordo com (PRADO, 2012), internamente o formato de dados HDF5 utiliza uma implementação de árvore B para acelerar a recuperação dos dados, mas apesar disso não há meio de acesso externo para essa facilidade, sendo considerada uma desvantagem para os usuários que necessitam de algum meio de manipulação similar aos SGBDR. Como consequência é encontrada pesquisas para superar o problema de consultas e gerenciamento dos dados armazenado em formato HDF5 em (NAM; SUSSMAN, 2003), (GOSINK et al., 2006), (ALTED; VILATA, 2007).

É importante salientar que isto não é uma falta de interesse da comunidade que utiliza o HDF5 e do HDF Group, mas sim, um problema de domínios de aplicação tão diversos, que a tarefa de desenvolver um mecanismo de indexação para propósito geral se torna praticamente inviável. De acordo com (FOLK et al., 2011), não há uma solução definitiva para indexação de conjuntos de dados HDF5 que sejam efetivos, tanto para esquemas multidimensionais ou unidimensionais, para todos os tipos de *queries*.

Do ponto de vista dos usuários e desenvolvedores, o ideal seria um esquema de indexação de propósito geral, porém para o HDF Group (HDF-GROUP, 2013) é um desafio atingir um denominador comum, devido como já citado, a natureza dinâmica dos múltiplos domínios de aplicação que usam o formato de dados HDF5. Desta forma, fica a cargo de cada um dos usuários, desenvolverem seus próprios mecanismos de indexação de dados, que atendem a um domínio específico e disponibilizá-los para a comunidade como um todo. Nesta tese, será proposto um modelo de indexação de dados para imagens médicas DICOM.

2.4.5 Segurança e Consistência

A biblioteca HDF5 usa um único semáforo para proteger às estruturas internas dos dados que podem ser modificadas por múltiplas *threads* em execução. Entretanto, isto proporciona a arquitetura somente uma segurança para as *threads*, chamado *thread-safety*, não suportando acessos concorrentes a arquivos HDF5 (FOLK et al., 2011). Já existem trabalhos sendo desenvolvidos, que visam permitir acesso concorrente para estruturas de dados individuais dentro dos conjuntos de dados armazenados.

Em relação à segurança lógica dos dados, o HDF5 não tem como função primordial proporcionar camadas de controle de acesso aos arquivos. Fica a cargo de cada um dos sistemas operacionais instalados nos servidores de armazenamento de dados, no caso de um cenário distribuído ou máquinas isoladas, a responsabilidade de preservar o acesso, a modificação e a remoção de todo ou de parte dos conjuntos de dados persistidos.

Sobre as propriedades similares as ACID (*Atomicity, Consistency, Isolation, Durability*), comumente encontradas nos bancos de dados relacionais, como já mencionado, o HDF5 não possui um modelo de transação para modificações de dados que garanta atomicidade das operações. O acesso por múltiplas *threads* de um processo não pode modificar os dados com garantia de consistência, inclusive acesso a grupos e conjuntos de dados diferentes, uma vez que há estruturas globais que são alteradas durante o processo de atualização da informação (PRADO, 2012).

Este conceito na computação é conhecido como *thread-safety*, e nestes casos cabe ao utilizador da biblioteca HDF5 criar mecanismos para garantir tal propriedade, se assim necessitar. Ainda assim é possível habilitar uma primitiva de sincronização, que usa a abstração de semáforos na interface de programação, para permitir que múltiplos processos possam fazer leituras enquanto um único processo estiver modificando o arquivo em HDF5, porém tal funcionalidade deve ser selecionada durante o processo de construção da biblioteca HDF5 (PRADO, 2012).

2.5 INDEXAÇÃO

Os índices podem ser definidos como estruturas de acesso adicionais que são utilizadas para aumentar a velocidade da recuperação de registros na resposta de certas condições de busca. Os índices fornecem um acesso eficiente aos registros a partir de campos de indexação que são utilizados para construir o índice. Há uma quantidade significativa de índices, em que cada um deles usa uma estrutura de dados específica para aumentar a velocidade

da busca (ELMASRI; NAVATHE, 2009).

Os tipos de índice mais utilizados são baseados em arquivos ordenados (índices de nível único) e estruturas de dados de árvores (índices multiníveis e árvores-B). Porém, eles podem ser construídos em *hashing* ou em outras estruturas de busca de dados (ELMASRI; NAVATHE, 2009). Como ilustração, alguns exemplos de tipos de indexação podem ser *bitmap index*, *dense index*, *sparse index*, *reverse index* e *inverted index*.

Esses mecanismos, em sua totalidade, também auxiliam o processo de construção de consultas (*queries*) para recuperação de determinada parte de informação. Como já mencionado, o HDF5 não provê mecanismos para elaboração de *queries* de alto nível do tipo SQL, para que os programadores integrem suas aplicações com o *backend*. Neste tese, para construir estes mecanismos de consultas sobre os dados armazenados nos arquivos HDF5, foi definido o uso de índices invertidos (*inverted index*) para indexação dos termos de busca, isto é, os metadados das imagens DICOM. Essa estratégia foi definida por permitir a indexação homogênea, extensível e flexível de todos os tipos de metadados que estão associados a imagens DICOM em qualquer nível da hierarquia estruturada nos arquivos HDF5.

Na literatura pode-se encontrar uma grande quantidade de ferramentas, que proporcionam mecanismos de busca e indexação de código aberto, como por exemplo Apache Lucene, Sphinx, Glimpse e o Xapian, entre outros. Cada um destes tem características próprias e modelos de operação distintos, porém tem como propósito facilitar tarefas intensivas de busca e recuperação de dados. O Apache Lucene, é um software da Apache Foundation escrito na linguagem de programação Java e conta com uma variação chamada, CLucene, que foi portada para linguagem C++. O principal, ou ainda, mais conhecido utilizador deste mecanismo de indexação é a enciclopédia colaborativa Wikipédia.

O CLucene pode ser definido como uma API de busca e de indexação de alto desempenho, escalável, flexível, multiplataforma, de código aberto e repleta de características importantes para desenvolvedores. Ele clama por ser mais rápido que o Apache Lucene, visto que foi desenvolvido em C++. É uma API muito madura, com ampla documentação, com escalabilidade acima da média que resulta em melhores níveis de desempenho (CLUCENE, 2013).

Pelas razões expostas, o CLucene se apresenta como a escolha neste trabalho para a estratégia de utilizar índices invertidos. Ainda, o CLucene se apresenta como uma escolha conveniente para indexação de conjuntos de dados HDF5, devido a suas semelhanças de arquitetura, como podem ser vistas abaixo:

- Linguagem de programação: como mencionado, ele é escrito em C++ o que torna compatível com o HDF5.

- Escalável: com as características de escalabilidade ilimitada do HDF5 não faria sentido escolher uma solução que não atendesse a este requisito.
- Multiplataforma: assim como o HDF5, o CLucene é multiplataforma. Neste requisito, não faria sentido escolher um sistema de buscas que não trabalhasse em diferentes arquiteturas.
- Flexibilidade: ele contém características de flexibilidade interessantes que remetem diretamente a liberdade que o HDF5 proporciona para construir estruturas de persistência de dados.

Por fim, os requisitos acima levantados podem ser aplicados ao Apache Lucene, escrito em Java, pois atendem a maioria dos pontos discutidos. Entretanto, como um dos focos deste trabalho é o desempenho, pode-se encontrar algumas avaliações de desempenho, como por exemplo em (BROADBENT; SAUNDERS; EKSTROM, 2006), em que o CLucene foi mais rápido do que o Apache Lucene e o Glimpse. Uma abstração de onde esta camada de indexação, é localizada na arquitetura, pode ser vista na Figura 21.

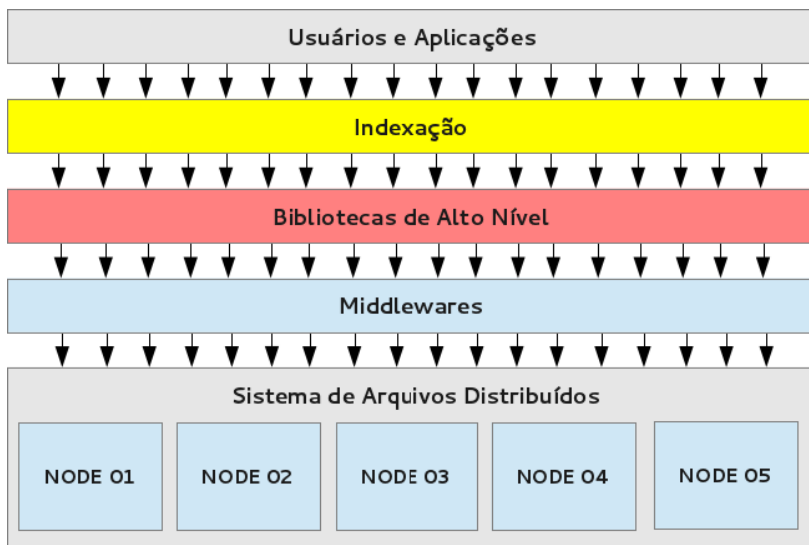


Figura 21: Abstração da Arquitetura com a Camada de Indexação.

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram vistos assuntos relacionados a fundamentação teórica desta tese, onde foram realizados estudos para embasar o modelo que esta sendo proposto. Inicialmente foi dissertado sobre conceitos teóricos sobre Telemedicina e características técnicas do Sistema Integrado de Telemedicina e Telessaúde (STT). Em seguida foi abordado o padrão DICOM, abordando de suas peculiaridades e quesitos técnicos.

Após isso, foi dissertado sobre os Sistemas de Arquivos Distribuídos (SAD), falando de suas características e seus componentes. Na sequência, foi falado sobre esquemas de paralelização de escritas e leituras em SAD, usando MPI-IO. Ainda, foi realizada uma análise dos SAD do mercado e na sequência foram detalhados os sistemas CEPH, PVFS, Lustre, FhGFS e o HDF5, que servirão de *backend* para o modelo.

Na sequência, foi dissertado sobre os Formatos de Dados, definindo o que eles são e pra que são utilizados. Foram abordados formatos de dados da família XML e formatos binários, como o CDF, NetCDF e o HDF5. Após isto, foi dissertado sobre os componentes do formato de dados HDF5 e sua arquitetura. Ainda, foram abordados aspectos de segurança e consistência de dados no HDF5. Por fim, foi falado sobre mecanismos de indexação de dados e a ferramenta que fará este papel no modelo, no caso o CLucene.

3 ESTADO DA ARTE

Para verificar o ineditismo, ou ainda, a originalidade do modelo proposto, neste capítulo será discutido o estado da arte do tema desta tese, buscando averiguar a existência de trabalhos relacionados ou similares. Para realização deste levantamento, foi utilizado o método Revisão Sistemática da Literatura, proposto e disseminado por Kitchenham (KITCHENHAM, 2004). Assim, serão discutidos os tópicos de relacionados a contextualização da pesquisa, questões de pesquisa, definições de bases de dados a serem pesquisadas, termos de pesquisa, procedimentos e critérios de seleção dos estudos, procedimentos de avaliação e qualidade dos estudos, estratégias de extração e sintetização dos dados e identificação de estudos relevantes. Ainda, serão tratados os trabalhos relacionados ao tema e os trabalhos realizados para construção deste modelo. Por fim, serão apresentadas as considerações finais do capítulo.

3.1 REVISÃO SISTEMÁTICA DA LITERATURA

O revisão de literatura é uma tarefa fundamental para a análise de áreas de conhecimento. Ela visa, primariamente, proporcionar métodos para identificação dos principais autores, obras e metodologias já desenvolvidas sobre determinado assunto. Com isto, é possível identificar *gaps* nas pesquisas já realizadas, dando uma visão geral sobre a área pesquisada. Em Kitchenham (2009) (KITCHENHAM et al., 2009) é realizada uma Revisão Sistemática da Literatura (RSL), do inglês *Systematic Literature Review* (SLR), acerca do tema engenharia de software, onde é proposto uma RSL sobre o próprio tema. Esta revisão foi elaborada seguindo a metodologia proposta, pela mesma autora em (KITCHENHAM, 2004). Desta forma, na RSL aqui redigida, será usada este método, que é internacionalmente aceito.

3.1.1 Contextualização da Pesquisa

A falta de padronização para o armazenamento de dados e informações, em sistemas médicos baseados em imagens (PACS, RIS e Telemedicina) motivou o desenvolvimento de vários tipos de abordagens para a solução do armazenamento de longo prazo para estes tipos de sistema. Estes tipos de soluções vão deste a utilização de bancos de dados baseados no paradigma relacional, passando por armazenamento natural em sistemas de arquivos e

indo até o armazenamento baseado no paradigma *NoSQL* em nuvens computacionais. Desta forma, o desenvolvimento deste levantamento bibliográfico é motivado pela tentativa de se desvendar quais estratégias têm sido desenvolvidas, que estejam alinhadas com o tema desta tese.

3.1.2 Questões Específicas da Pesquisa

Esta pesquisa visa responder a seguinte pergunta:

- Quais estratégias têm sido desenvolvidas para a aplicação de uma camada de persistência hierárquica e distribuída para sistemas médicos baseados em imagens?

3.1.3 Definições de Bases de Dados e Termos de Pesquisa

Esta pesquisa por trabalhos relacionados foi realizada a partir da rede da Universidade Federal de Santa Catarina – UFSC, pois nesta rede há a possibilidade de acesso irrestrito as principais bases via Internet. A metodologia seguiu o padrão de acesso as principais bases de dados de documentos científicos da área. A saber, as bases de dados consultadas foram:

- ACM Digital Library;
- IEEEExplore;
- Springer Link;
- PubMed;
- Science Direct.

As pesquisas realizadas foram limitadas aos trabalhos publicados no período de Janeiro de 1993 até Dezembro de 2013. Este definição foi traçada devido a data de lançamento do padrão DICOM. As buscas foram realizadas através da utilização de palavras-chave nos mecanismos de busca das bases de dados consultadas, sempre adequando as *strings* solicitadas aos mecanismos de cada base. As palavras-chave foram divididas em três classes, relacionadas ao objeto, as operações realizadas, as bases tecnológicas, visando assim fornecer uma maior acuracidade ao levantamento.

O objeto relacionado a esta tese é o *DICOM*, que necessita de operações de ‘*indexing*’, ‘*storage*’, ‘*store*’, ‘*index*’, ‘*query*’, ‘*retrieval*’, em bases tec-

nológicas de armazenamento hierárquico “HDF” e “*Hierarchical Data Format*”. Como informação adicional, os contextos (PACS, HIS, RIS e Telemedicina) foram retirados das buscas, pois independente do contexto e da forma com que ele se aplica, o objeto de pesquisa ainda permanece o mesmo. Assim, de acordo com a RSL, a primeira fase da pesquisa foi limitada somente ao título, *abstract* e as palavras-chave dos artigos. Por último, para um primeiro refinamento, foi excluída a palavra chave “web”, pois estava causando poluição aos resultados, resgatando trabalhos de outros escopos.

3.1.4 Procedimentos e Critérios de Seleção dos Estudos

Os artigos selecionados para análise, foram publicados na língua inglesa, durante o período de 1993 até 2013, devido as razões acima levantadas. Somente foram analisados artigos publicados em conferências ou em *journals*, devido a relevância dos mesmo para a ciência, e ainda por este estabelecerem processos de revisão por pares. Dessa forma, foram excluídos artigos técnicos da *Web*, visto que eles não passaram pelo crivo da comunidade científica.

Além disto, foram tratados como relevantes, trabalhos que buscassem responder a questão de pesquisa levantada, acerca da utilização de formatos de dados hierárquicos para armazenamento e recuperação de imagens DICOM, bem como os metadados a elas relacionados. Como critério de seleção dos artigos, foi realizado um filtro inicial, realizado através da leitura do título do trabalho, o resumo e as palavras-chave.

3.1.5 Procedimentos para Avaliação da Qualidade dos Estudos

Em relação a avaliação da qualidades dos estudos, foram levantados critérios para a seleção dos mesmos. Tais critérios foram gerados levando em consideração a natureza da proposta, bem como seus componentes, sua metodologia, sua arquitetura e suas tecnologias relacionadas. Estes foram criados para fins de se obter um melhor alinhamento com a proposta de modelo aqui apresentada.

1. O estudo deve apresentar experimentos realizados e com metodologia concreta e passível de ser reproduzida;
2. O estudo deve estar alinhado com o tema de armazenamento de imagens DICOM;
3. O estudo não pode ser relacionado ao armazenamento de imagens DI-

COM em SGBD's;

4. O estudo deve ser aplicado ao contexto da Telemedicina, RIS, HIS ou PACS;
5. O estudo não deve ser relacionado à itens que estejam fora do escopo do trabalho.

3.1.6 Estratégia para Extração e Sintetização dos Dados

Para estratégia de extração e sintetização dos dados dos artigos recuperados, os mesmos serão expostos citando seus autores, ano de publicação, título do artigo, domínio de aplicação, modelo de armazenamento utilizado, modelo de gerenciamento proposto, experimentos realizados, discussão sobre os resultados e ainda, relacionando o trabalho com o tema desta proposta de pesquisa. Desta forma, facilitando assim a análise e a apreciação das abordagens encontradas.

3.1.7 Identificação dos Estudos Relevantes

Como resultado da identificação dos estudos relevantes nas bases de conhecimento, na Tabela 5 pode ser visualizada a relação dos repositórios consultados, os termos pesquisados e a quantidade de resultados separados por base.

3.1.8 Estudos Relevantes Sintetizados

Após a fase de identificação dos estudos relevantes para esta revisão sistemática, foram identificados um total de 545 trabalhos nos repositórios internacionais de artigos, que tinham algum nível de relação com o modelo apresentado nesta tese. Após uma primeira leitura do título, *abstract* e palavras-chave, foram aplicados os critérios de exclusão expostos, onde os artigos tinham que estar alinhados com o tema desta tese, apresentando experimentos, com metodologia concreta e passível de ser reproduzida, relacionados com armazenamento hierárquico de imagens médicas DICOM, que não estivessem relacionadas com o uso de sistema gerenciadores de bancos de dados relacionais e dentro de um contexto de PACS, RIS, HIS ou Telemedicina. Após esta primeira leitura, foram selecionados um total de 32 artigos.

Depois desta primeira fase, os 32 artigos selecionados foram analisa-

Base de Dados	Termos	Número de Resultados
ACM Digital Library	((Abstract:DICOM and (Abstract:query or Abstract:retrieval or Abstract:store or Abstract:storage)) and (not Abstract:web)) or ((Abstract:hdf) AND (Abstract:query or Abstract:retrieval or Abstract:index or Abstract:indexing))	4
IEEEExplore	(("Abstract":DICOM) AND ("Abstract":query OR "Abstract":retrieval OR "Abstract":store OR "Abstract":storage) AND NOT "Abstract":web) OR (("Abstract":hdf OR "Abstract":hierarchical data format) AND ("Abstract":indexing OR "Abstract":index OR "Abstract":query OR "Abstract":retrieval))	156
Springer Link	'ab:((dicom and (query or retrieval or storage)) and not web) or (hdf and (index or query or retrieval)))' published between '1 Jan 1993' and '21 Sep 2013'	79
PubMed	((DICOM[Title/Abstract] AND (query[Title/Abstract] OR retrieve[Title/Abstract] OR store[Title/Abstract] OR storage[Title/Abstract])) NOT web[Title/Abstract] OR ((hdf[Title/Abstract] or hierarchical data format[Title/Abstract]) AND (retrieval[Title/Abstract] OR query[Title/Abstract] OR index[Title/Abstract] OR indexing[Title/Abstract])))	232
Science Direct	((Title-Abstr-Key(DICOM) AND (Title-Abstr-Key(query) OR Title-Abstr-Key(retrieval) OR Title-Abstr-Key(store) OR Title-Abstr-Key(storage))) AND NOT Title-Abstr-Key(web)) OR ((Title-Abstr-Key(hdf) OR Title-Abstr-Key(hierarchical data format)) AND (Title-Abstr-Key(query) OR Title-Abstr-Key(retrieval) OR Title-Abstr-Key(indexing) OR Title-Abstr-Key(index)))	74
TOTAL		545

Tabela 5: RSL – Artigos Encontrados nas Bases de Conhecimento.

dos na íntegra, buscando encontrar relações sólidas com o modelo apresentado nesta tese. Foram encontrados estudos que vão desde armazenamento em sistemas de arquivos (COSTA et al., 2009), passando por estudos de indexação de dados para o formato HDF5 no âmbito da física (GOSINK et al., 2006), trabalhos na área de armazenamento de objetos DICOM em nuvens computacionais (TENG et al., 2010), entre outros. Entretanto, excetuando-se trabalhos relacionados de autoria de Douglas D. J. de Macedo, não foram identificados trabalhos que estejam 100% alinhados com o tema desta pesquisa. Porém, foram encontrados estudos relacionados, que contém algum tipo de relação com componentes encontrados no modelo proposto. Exemplos destes pode ser vistos na seção 3.2.

3.2 TRABALHOS RELACIONADOS

No artigo *Scientific formats for object-relational database systems: a study of suitability and performance* (COHEN et al., 2006) foi proposta uma extensão de um modelo tradicional de armazenamento, que foi comparado com um modelo objeto relacional e outros modelos usando o HDF (versões 4 e 5) e o NetCDF. Como consequência, este estudo indicou resultados promissores para os formatos de dados científicos analisados, em relação aos bancos de dados objeto relacionais. Dos formatos analisados, o HDF5 apresentou um desempenho na ordem de 200%.

Uma importante contribuição deste estudo foi a indicação de que os dados armazenados sob os formatos de dados científicos são mais fáceis de gerenciar, se comparada a persistência usando bancos de dados objeto relacionais. A relação deste estudo com o modelo aqui apresentado, esta na utilização de formatos de dados científicos para persistência de dados, ao invés da utilização tradicional. Ainda, este estudo indica que entre os formatos de dados científicos analisados, o HDF5 é o que obtém um melhor desempenho, além de ter um nível de gerenciamento mais amigável e simples do que os modelos tradicionais avaliados, usando bancos de dados objeto relacionais.

Em (TENG et al., 2010), o trabalho intitulado *A medical image archive solution in the cloud* relatam sobre o armazenamento de imagens médicas DICOM em cenários distribuídos, no caso, nuvens computacionais. Neste estudo foi usado a plataforma Windows Azure para persistir as imagens, sob um banco de dados relacional. Foi proposto um modelo de armazenamento, busca e recuperação para PACS. Como resultado, este artigo demonstrou que o protótipo desenvolvido teve boa viabilidade de implementação de plataformas de armazenamento de longo prazo. A relação deste trabalho com o mo-

delo aqui proposto encontra-se na preocupação de distribuição de imagens em cenários altamente distribuídos para persistência de longo prazo.

No trabalho intitulado *HDF5-FastQuery: Accelerating Complex Queries on HDF Datasets using Fast Bitmap Indices* (GOSINK et al., 2006), foi desenvolvido um modelo de gerenciamento de dados usando índice de *bitmap* chamado HDF5-FastQuery. O modelo desenvolvido foi comparado com outra solução usando árvore-R* e ainda, comparando com a leitura nativa oferecida pelo HDF5. Como resultado, o HDF5-FastQuery obteve um desempenho superior em até duas vezes, se comparada as outras abordagens. Foi observado ainda que o índice usando árvore-R não suportou o aumento de volume de informações, apresentando falhas de consistência. A relação deste estudo com o modelo aqui apresentado esta na parte de construção de índices auxiliarem para indexação de dados sob o formato HDF5. Como complemento, vale ressaltar que o objeto do estudo foi na área de física e não na área médica.

Em (FOLINO; SHAH; KRANSNOGOR, 2009) o trabalho intitulado *On the storage, management and analysis of (multi) similarity for large scale protein structure datasets in the grid* foi apresentada uma proposta de comparação de similaridade entre proteínas para concepção de uma base de conhecimento. No trabalho foi usado um modelo de persistência utilizando bancos dados relacionais (Oracle) e o formato de dados HDF5, por meio de consultas SQL e a própria interface de interação nativa do HDF5. Como resultado, a utilização do HDF5 foi superior ao Oracle, em termos de desempenho e gerenciamento de dados. A relação deste trabalho com o modelo aqui proposto encontra-se na interseção de dados críticos armazenados em bancos de dados relacionais e em formatos de dados científicos, onde o HDF5 teve desempenho superior.

No artigo *Indexing and retrieving DICOM data in disperse and unstructured archives* (COSTA et al., 2009) foi apresentado um esquema de recuperação de objetos DICOM usando sistemas de arquivos locais, usando índices invertidos. Para isto foi usada a versão original do Lucene, escrito em Java. Como resultados, os autores concluíram que esta pode ser uma nova forma de armazenamento e recuperação de dados ou ainda, que este pode estender o modelo relacional, tradicionalmente utilizado. A relação deste estudo com o modelo aqui apresentado esta na fase de indexação de objetos DICOM para armazenamento e posterior recuperação. Entretanto, eles não utilizaram nenhuma camada de abstração ou hierarquização para armazenamento dos dados, sendo isto feito localmente, com o sistema de entrada e saída do próprio sistema operacional. Ainda, a abordagem adotada é local e não distribuída.

Em (ABDULJWAD; NING; DE, 2010) o artigo intitulado *SMX/R: Ef-*

efficient way of storing and managing XML documents using RDBMSs based on paths aborda também aspectos de armazenamento, de busca e de recuperação de informações usando o formato de dados hierárquico XML. Neste trabalho foi usado o modelo de bancos de dados relacionais, em conjunto com o formato XML buscando um melhor desempenho em relação ao modelo anteriormente desenvolvido pelo mesmo grupo, chamado XRel. Como resultado, o modelo proposto teve um desempenho superior ao XRel. A relação deste trabalho com o modelo aqui apresentado resume-se na concepção de um modelo de armazenamento, busca e recuperação usando formatos de dados hierárquicos XML.

Como adição aos trabalhos relacionados, durante a fase de seleção dos estudos, baseados nos critérios de exclusão, foram identificados dois trabalhos, que tem relação direta com o tema, pois tratam de armazenamento de imagens médicas DICOM de forma hierárquica com o formato de dados HDF5, sendo eles (DOUGHERTY, 2009) e (DOUGHERTY et al., 2009). Apesar de ambos trabalhos estarem alinhados com o tema, eles não tratam componentes como indexação e distribuição de dados. Estes trabalhos foram excluídos da análise final, visto que não continham experimentos, modelos ou arquiteturas validadas com metodologia clara, entretanto, cabem aqui serem relacionados.

Em 2009, o pesquisador Matthew T. Dougherty, do Centro Nacional de Imagens Macromoleculares (*National Center for Macromolecular Imaging*), em Houston, no Texas, escreveu um artigo, intitulado, “*Envisioning a Standard Image Storage Framework*” (DOUGHERTY, 2009), onde foi descrita a possibilidade da criação de um *framework* visando persistência e interoperabilidade para imagens médicas sob o formato HDF5. Porém, este foi um estudo teórico, onde não houveram experimentos, metodologias, arquiteturas e resultados que pudessem comprovar a ideia.

Ainda em outubro de 2009, o mesmo pesquisador, juntamente com especialistas de várias áreas, como armazenamento e imagens médicas, e até o presidente do HDF Group, Michael J. Folk, publicaram um artigo intitulado (DOUGHERTY et al., 2009) intitulado *Unifying Biological Image Formats with HDF5*. Neste trabalho foi proposta uma discussão sobre a adoção do formato HDF5 como plataforma básica para imagens biológicas. Este trabalho teórico foi desenvolvido por vários pesquisadores de centros de excelência, que concluíram que o assunto não é se as ciências biomédicas vão adotar o HDF5 como um *framework* para suporte a imagens, pois isto aos poucos já está acontecendo, mas quanto tempo isso vai demorar e quão difícil será integrar os muitos tipos de dados médicos sobre o formato HDF5.

Entretanto, estes dois artigos foram publicados em uma conferência em 18 a 22 de março de 2009 e em um *journal* em outubro de 2009, respecti-

vamente, e anteriormente a este período, já haviam sido publicados três artigos citando a possibilidade de armazenamento de imagens médicas DICOM sob o formato HDF5, pelo pesquisador Douglas D. J. de Macedo (MACEDO et al., 2009) (MACEDO et al., 2008d) (MACEDO et al., 2008a), que já continham experimentos, arquiteturas e testes validados. Dito isto, destes 32 artigos selecionados, nenhum deles continha um nível de alinhamento suficiente para ser relacionado com o modelo aqui apresentado. Todos os trabalhos encontrados, que tinham relação de armazenamento hierárquico distribuído de imagens médicas DICOM, estão relacionados com o grupo de pesquisa que vem desenvolvendo este modelo desde 2007, com autoria ou co-autoria de Douglas D. J. de Macedo, o que atesta o ineditismo desta proposta de modelo. Na Seção 3.3 estes trabalhos serão expostos, para um melhor entendimento da construção deste modelo.

3.3 TRABALHOS REALIZADOS

Em complemento, durante as pesquisas, foram encontrados trabalhos de autoria ou co-autoria de Douglas D. J. de Macedo (MACEDO, D.D.J.) tratando sobre o tema desta tese. Em sua totalidade, foram experimentos realizados e publicados em conferências ou periódicos internacionais visando averiguar, com a comunidade, sobre a aceitabilidade de partes do modelo proposto. Os trabalhos iniciaram em 2007, com a tentativa de armazenar dados médicos em outros formatos. Os primeiros resultados foram publicados em (MACEDO et al., 2008d) e (MACEDO et al., 2008a), onde foi demonstrada uma arquitetura básica usando o formato de dados HDF5, como base do modelo.

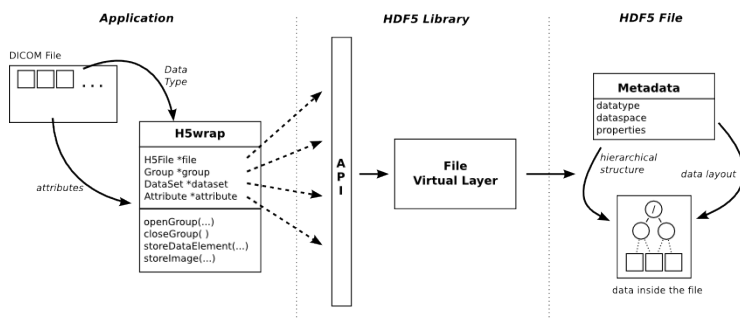


Figura 22: Arquitetura proposta em (MACEDO et al., 2009).

No ano de 2009, foi proposta uma arquitetura completa para imagens

DICOM usando o formato de dados HDF5 (MACEDO et al., 2009) que tinha como *backend* um sistema de arquivo distribuído. Esta arquitetura apesar de funcional, tinha sério problemas com a escrita de grandes volumes de dados e ainda, com a recuperação de dados dos múltiplos nodos do agregado computacional. Na Figura 22 é possível visualizar a primeira arquitetura proposta. Ainda em 2009, foi proposta uma estrutura de replicação de dados (AMARAL et al., 2009), usando um sistema de arquivo distribuído mais simples, no caso o NFS (*Network File System*) e comparada com um sistema de replicação de dados usando um banco de dados relacional, no caso, o PostgreSQL.

Em 2011, motivados pelos resultados obtidos em trabalhos anteriores e somado ao fato motivação de superar os problemas encontrados quando a escrita de grandes volumes de dados e a sua posterior recuperação com desempenho satisfatório, foi proposto um trabalho de otimização de código em (MACEDO et al., 2011), onde foram comparados os resultados desta nova abordagem, com os resultados obtidos anteriormente. Ainda em 2011, o trabalho (SOARES et al., 2011) focou os esforços em escrita paralela de imagens DICOM no formato de dados HDF5 em sistemas de arquivos distribuídos. O foco deste estudo foi somente na fase de escrita dos dados, deixando de lado a parte de recuperação. Na Figura 23 é possível visualizar esta arquitetura.

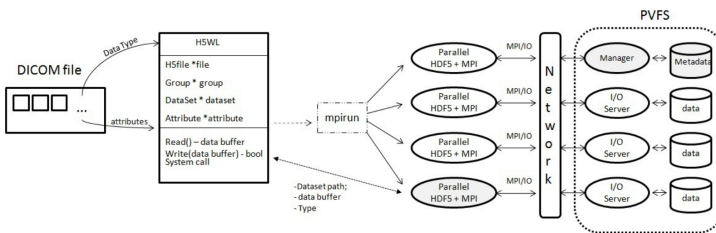


Figura 23: Arquitetura proposta em (SOARES et al., 2011).

Em 2012, motivados com os bons resultados obtidos usando escrita paralela de objetos DICOM, foi proposta uma melhoria no modelo de gravação, usando MPI-IO, em (SOARES et al., 2012) e (SOARES et al., 2012), cada um destes demonstrando fases distintas do processo de escrita em múltiplos nodos em um cenário distribuído. O primeiro deste foi focado na área de comunicação entre os nodos e no custo desta no processo de gravação. O segundo uniu a parte de custos de comunicação com o processo completo de gravação, onde iniciava no servidor de imagens medicas, até a alocação final no nodo do sistema de arquivo distribuído.

Ainda em 2012, foi proposta uma nova arquitetura para persistência

para imagens médicas, porém esta era focada no formato de dados NetCDF (MAGNUS et al., 2012). Neste estudo, foi comparado o processo de armazenamento e de recuperação de dados no formato NetCDF, em comparação com as outras arquiteturas já validadas, usando HDF5. Como resultado, foi constatado que o formato NetCDF tem desempenho inferior para persistência de imagens DICOM, se comparado com o HDF5. O estudo indica que o NetCDF tem desempenho satisfatório para este fim, se estiver integrado ao formato HDF5 para efetuar a parte de hierarquização dos dados.

Em 2013, foi proposta uma extensão do modelo de armazenamento, porém focado em nuvens computacionais privadas (SOARES et al., 2013). O foco do estudo foi na área de gerenciamento de grandes volumes de dados. Foi apresentado um modelo de interoperabilidade para armazenamento e recuperação de dados médicos usando formato de dados HDF5. Ainda em 2013, foi proposta uma melhoria no sistema de replicação de dados (AMARAL et al., 2013), onde foram analisadas as replicações de dados entre múltiplos nodos em um esquema de bancos de dados distribuídos e esta abordagem foi comparada com os resultados obtidos em cenários distribuídos no formato HDF5.

3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi visto o estado da arte relacionado ao tema desta pesquisa, afim de verificar o ineditismo do modelo proposto. Para isto, foi realizada uma Revisão Sistemática da Literatura (RSL), baseado na abordagem de (KITCHENHAM et al., 2009). Foi dada uma contextualização da pesquisa e, posteriormente, foi levantada uma questão de pesquisa, para guiar a revisão. Durante o processo de revisão sistemática, foram consultadas as principais bases de conhecimento, sendo elas: ACM Digital Library, IEEEExplore, Springer Link, PubMed e Science Direct.

Além disto, foram definidos termos de busca alinhados com a pergunta de pesquisa e no processo de busca de estudos, foram levantados um total de 545 estudos. Após a primeira etapa de aplicação de critérios de exclusão, restaram 32 trabalhos alinhados com o tema. Na segunda fase, com leitura e análise integral dos artigos, foi identificado que nenhum deles estava totalmente alinhado com o modelo proposto. Por fim, foram levantados alguns trabalhos relacionados que continham partes do modelo proposto e após isto, foram explicitados os artigos de autoria ou co-autoria de Douglas D. J. de Macedo, que tratam especificamente do modelo proposto.

4 MODELO PROPOSTO

Dado o contexto apresentado sobre o tema, e baseando-se no levantamento bibliográfico e, posteriormente na revisão sistemática da literatura, pôde-se desenvolver um modelo de armazenamento para imagens médicas DICOM de forma hierárquica e distribuída. Neste capítulo, é discutido o modelo conceitual do estudo de caso (STT/RCTM), bem como sua arquitetura, definição dos componentes contidos nela e as relações entre estes. Além disto, o modelo proposto para persistência é apresentado, juntamente com o modelo de dados e de indexação concebidos para o modelo. Como complemento, é apresentada uma arquitetura baseada no novo modelo proposto, onde são traçadas observações sobre todos os componentes da arquitetura, suas interligações, fundamentos e funções dentro do contexto que é apresentado. Por fim, são dadas as considerações finais do capítulo.

4.1 MODELO CONCEITUAL DO STT/RCTM

Para um melhor entendimento do modelo que esta sendo proposto, é importante conhecer o modelo de armazenamento atual que inspirou a concepção de um novo modelo, como ele funciona, quais seus componentes e sua arquitetura. Na Figura 24 é possível visualizar o modelo conceitual do STT/RCTM utilizado para persistência de dados, que serve como estudo de caso nesta tese. Nele pode-se ver um modelo em camadas, bastante utilizado em sistemas de informação e/ou conhecimento em geral, que são baseados no paradigma cliente-servidor. Neste modelo são encontrados usuários, equipamentos, interfaces de comunicação, uma camada de imagens e aplicação segmentadas, um interpretador de comandos e a camada de armazenamento de fato.

Em relação à camada de usuários, estes podem ser tanto pessoas, softwares, computadores, *tablets*, celulares, etc. Usuários são considerados aqui qualquer entidade que estabelece uma relação com o modelo, inclusive até os equipamentos médico-hospitalares. Quanto aos equipamentos, estes podem ser desde aparelhos de eletrocardiograma, passando por equipamentos de ressonância magnética, tomografia computadorizada, até equipamentos de raio-x digital. Os equipamentos estão nesta camada devido ao fato de que em vários momentos, eles atuam no modelo como agentes solicitados operações, de consulta, leituras, buscas, recuperações, gravações, etc. e desta forma, também atuam como clientes em alguns momentos.

Na interface, é ilustrado uma camada de interação com os usuários ou

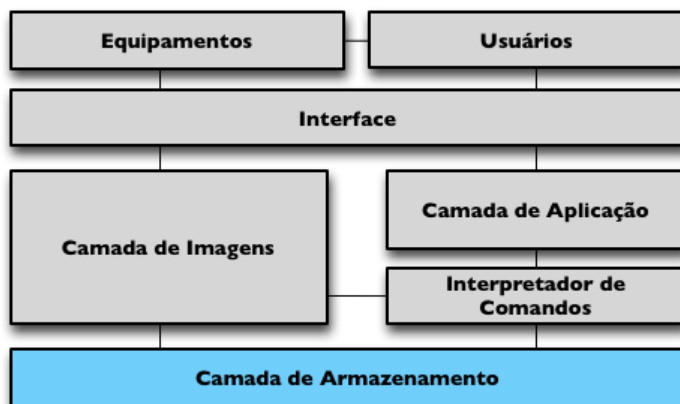


Figura 24: Modelo Conceitual Atual do STT/RCTM.

equipamentos, com o restante do modelo. Nesta camada, encontra-se toda a parte de comunicação entre as entidades, desde a parte de hardware, como por exemplo *links* de comunicação, roteadores, *switches*, *modems*, entre outros, até na parte de lógica, com algoritmos de roteamento, *firmwares*, sistemas de controle de fluxo, *proxies*, etc. Nesta camada acontece toda a parte de comunicação com o mundo exterior ao modelo e é a entrada principal do modelo, do ponto de vista dos usuários e equipamentos.

Na camada abaixo, o modelo é dividido em dois blocos, sendo a camada de imagens e a camada de aplicação e a camada de interpretação dos comandos. Na camada de aplicação se encontra o próprio software, neste caso específico, escrito na linguagem de programação PHP e abaixo dela esta seu interpretador de comandos. Neste ponto da camada, vale a pena ressaltar que o software poderia estar escrito em outra linguagem, isto dependendo do contexto que esta sendo aplicado. Neste caso específico, foi escrito em PHP por se tratar de um sistema Web, mas poderia ter sido escrito em Java, Python, ASP.net, entre outros. Ainda nesta camada, encontra-se o servidor Web, que é o responsável em receber as requisições externas, repassar para as páginas relacionadas, que irão usar o interpretador de comandos para atingir a base de dados, ou ainda, o servidor de imagens médicas.

Na camada de imagens, pode-se encontrar os servidores de imagens, como por exemplo o CyclopsDCMServer, que neste modelo é o responsável em receber as imagens advindas dos equipamentos médico-hospitalares, des-construí-las e enviá-las para o banco de dados que irá persistir o exame. Esta

camada é responsável em interagir através da interface com os usuários e fazer a intermediação de suas solicitações com as camadas inferiores, ou ainda, com o interpretador de comandos, dependendo do contexto da requisição. É neste ponto do modelo que irá ser iniciado o processo de desenvolvimento de um novo modelo, visto que a idéia a princípio é proporcionar um novo método de armazenamento.

Na base do modelo, a camada de armazenamento, que é a responsável por persistir de fato os exames dos pacientes. Nesta camada, no modelo tradicional que esta sendo aplicado, encontra-se um sistema gerenciador de bancos de dados relacionais (PostgreSQL), que contém uma base de dados específica para armazenamento de imagens em formato binário (BLOBs) e outra base de dados com exames já segmentados e convertidos para outros formatos convencionais na Web, como JPEG2000 ou PNG. Este é um ponto de interesse para o novo modelo que esta sendo apresentado, pois será nesta camada a principal inovação, onde este modelo tradicional de armazenamento será substituído por outro modelo de persistência.

Resumindo, quando um paciente se dirige até um hospital para realizar um exame médico baseado em imagens, estas imagens são enviadas pelos próprios equipamentos ou ainda, por seus operadores, através das interfaces de comunicação que interligam os pontos. Desta forma, é disparada uma conexão direta do equipamento para a camada de imagens, onde se encontra o servidor DICOM que irá receber estas imagens e tratá-las. Após receber estas imagens, o servidor de imagens irá fazer o *parsing* da imagem e segmentá-la, separando os metadados da imagens em si.

Este é um ponto chave do modelo, pois aqui o servidor de imagens DICOM irá enviar solicitações de gravação para a camada de armazenamento, gravando os metadados em uma base distinta, juntamente com as imagens DICOM já convertidas para formatos usuais na Web, isto para facilitar a visualização no próprio *browser*. Ainda, em uma segunda etapa, o servidor de imagens irá enviar uma solicitação de armazenamento do objeto binário da imagem DICOM (BLOB) em outra base de dados, pois em alguns casos, os especialistas necessitam de uma melhor resolução ou flexibilidade para visualizar a imagem, como por exemplo usando software de visualização de imagens DICOM, como o DIMP (web), OsiriX (stand-alone), entre outros.

4.2 ARQUITETURA DO STT/RCTM

Baseados no modelo conceitual do STT/RCTM apresentado, foi constituída uma arquitetura computacional que implementa o modelo. Na Figura 25 é possível visualizar esta arquitetura em detalhes. Basicamente a arquitetura

tura é composta por usuários (médicos, pacientes e técnicos), softwares (STT e Servidor DICOM), servidores e dispositivos de armazenamento. Todos estes componentes tem papéis bem definidos dentro da arquitetura e nesta seção será tratada cada um destes papéis, que inferem diretamente em operações na arquitetura.

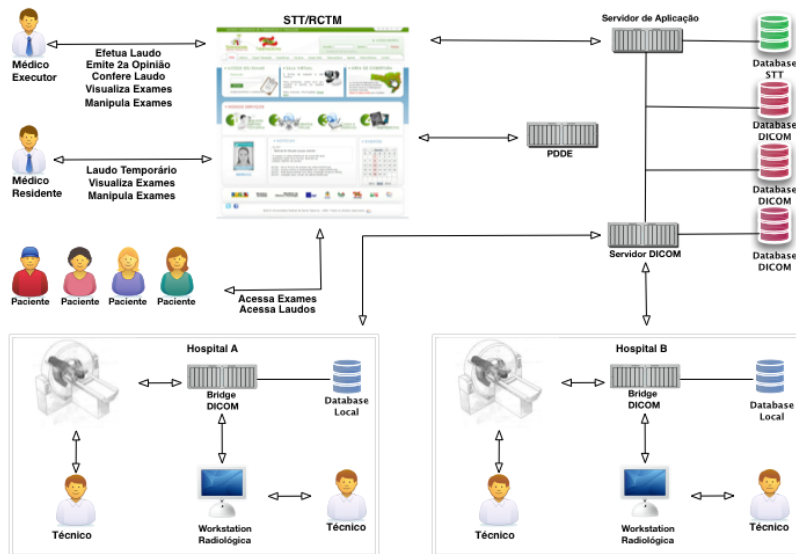


Figura 25: Arquitetura Atual do STT/RCTM

O processo inicia quando um paciente vai até um hospital para realizar um exame médico baseado em imagens. Técnicos (enfermeiros, técnicos de equipamentos, entre outros) realizam o procedimento com o paciente e as imagens são geradas. Neste momento há dois processos possíveis, o primeiro deles o técnico irá verificar se a geração das imagens ocorreu com sucesso através da *workstation* radiológica ou ainda, em cenários mais tradicionais, o próprio médico pode laudar os exames na *workstation*. Nessa mesma estação, anteriormente pode ter sido realizado o cadastro do paciente, com todos os metadados que irão compor as imagens DICOM.

Após este processo, as imagens podem ser transferidas para o servidor de imagens médicas diretamente, em tempo real, através de um processo síncrono, ou ainda, podem ser armazenadas em uma *bridge* local, que será enviada posteriormente, através de um processo assíncrono. A *bridge* é um importante componente nesta arquitetura, visto que ela preserva os dados em

casos de falhas - como comunicação, energia, etc, onde os exames não podem ser enviados em tempo real. Ainda, há casos de grandes hospitais que podem gerar tantos exames, que o *link* de comunicação com o servidor de imagens remoto fica congestionado, e desta forma a *bridge* faz um papel de escalonador no envio das imagens, dando-lhes a prioridade necessária.

Independente do tipo de processo de envio, assíncrono ou síncrono, a comunicação entre os equipamentos médicos ou a *bridges* é realizada através do protocolo de transporte TCP, realizando uma comunicação direta na porta (4007) do servidor de imagens, que esta em *standby* aguardando exames. Quando o servidor de imagens recebe os estudos enviados, ele executa um processo de *parsing* (desconstrução) dos estudos, séries e imagens. Neste processo ele extrai todos os mais de 2000 metadados dos exames, bem como a imagem (*pixeldata*).

Após o processo de envio, recebimento e posterior desconstrução, o servidor DICOM estabelece uma conexão com o banco de dados de produção do sistema Web, onde ele envia metadados selecionados anteriormente na modelagem do banco de dados. Entre estes metadados podem citar o estudo, série, imagens convertidas em formatos para Web (JPEG2000, PNG, JPG, etc), nome do paciente, data de nascimento, nome da mãe, datas de realização dos exames, modalidade do exame, identificadores, entre outros metadados.

Após o armazenamento dos metadados e da imagens já convertidas no banco de dados de produção para o sistema Web, o servidor DICOM armazena as imagens DICOM em formato binário (BLOBs) em outros bancos de dados relacionais, específicos para este fim. Terminado as fases de realização do exame, envio das imagens, recebimento, posterior desconstrução e armazenamento das bases de dados, finaliza-se a fase e inicia-se uma nova fase na arquitetura, onde os médicos e pacientes estão envolvidos.

No caso do médico especialista, ele dentro do sistema irá receber uma mensagem em sua lista de tarefas que há um exame pendente de determinada modalidade para efetivação de um laudo. Em relação aos perfis de médicos possíveis dentro do sistema, há dois casos, sendo um deles o médico executor e o médico residente. O médico executor pode emitir um laudo, efetivar uma segunda opinião em relação a um caso complexo, pode conferir um laudo dado por um médico residente, visualizar exames e ainda, manipulá-los. Em relação ao médico residente, este tem um perfil restrito onde apenas visualiza e manipula exames e pode efetuar um laudo temporário, que será validado por um médico executor.

No momento que um médico efetua um laudo em um exame de um paciente é a fase onde o conhecimento é explicitado. Neste instante o médico, baseado em seu conhecimentos e experiências determina uma opinião sobre uma determinada doença que pode estar relacionada com a imagem que é

mostrada a ele pelo sistema. Desta forma, ele emite sua opinião sobre o caso, que é gravado na base de dados e sincronizada com os servidores de imagem médica, para efeitos de armazenamento. Por último, o paciente após a realização do exame no hospital, recebe seu código de acesso ao exame e com este código, acessa o sistema Web onde ele terá acesso ao seu exame completo, com imagens e laudo relacionado.

4.3 MODELO PROPOSTO

4.3.1 Procedimentos Metodológicos

Os procedimentos metodológicos empregados para o desenvolvimento do modelo que esta sendo proposto iniciaram com a revisão da literatura, buscando embasar às escolhas que foram realizadas com o que há de mais atual nas bases de conhecimento nacionais e internacionais. Após isto, houve uma etapa de especificação e análise sobre as partes do modelo, onde baseados no modelo conceitual atual e sua consequente arquitetura, somada a entrevistas com especialistas pôde-se conceber um modelo conceitual para o armazenamento e recuperação das imagens médicas.

Desta foram, baseando-se no modelo conceitual proposto, foi desenvolvida uma arquitetura computacional que cobriu a totalidade do modelo conceitual. Esta arquitetura foi desenvolvida utilizando-se de componentes que têm funções específicas e bem determinadas na operação da arquitetura. Para a avaliação da arquitetura proposta, foi elaborado um estudo de caso onde o modelo, figurado através da arquitetura, foi implantando em um cenário real. Desta forma houve uma etapa de verificação para determinar a efetividade do modelo proposto para o que ele se propôs. Esta última fase será apresentada no Capítulo 5, onde será apresentada a validação da arquitetura proposta.

4.3.2 Modelo Conceitual

Dado o contexto apresentado, nesta seção é detalhado o modelo conceitual para persistência de exames médicos baseados em imagens DICOM. Na Figura 26 é possível visualizá-lo em detalhes. Uma das importantes características deste novo modelo é a adaptabilidade e flexibilidade para integrações a modelos mais tradicionais, visto que com pouco esforço pode-se adaptar a sistemas de Telemedicina, RIS, HIS ou PACS. Ainda, como complemento, a partir desta integração, toda a parte superior de modelos tradicionais de sis-

temas de informação ou conhecimento permanece exatamente com seu modo de operação, não havendo impacto para os usuários em geral.

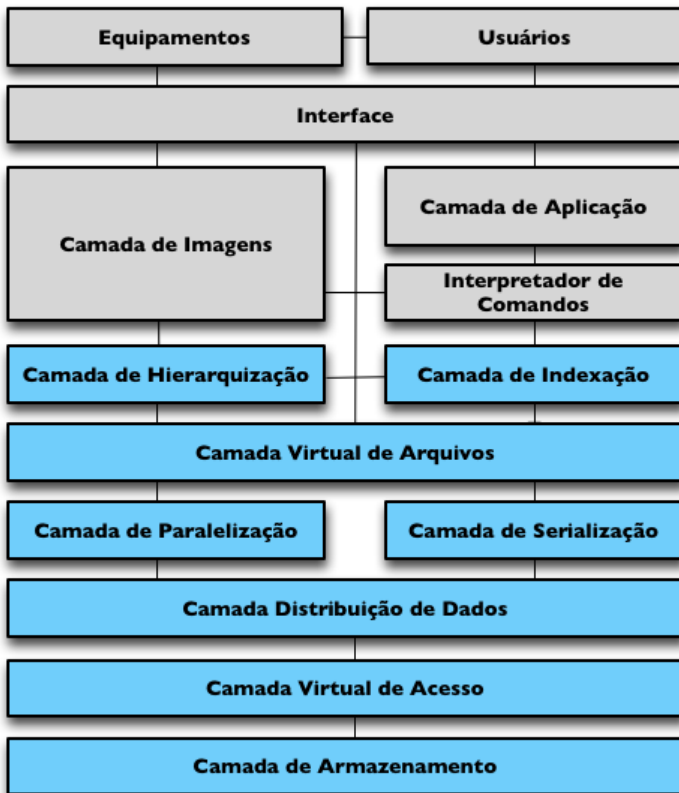


Figura 26: Modelo Conceitual de Armazenamento Proposto.

Dessa forma, as camadas relacionadas a equipamentos médicos, usuários em geral, interfaces, camada de imagens, camada de aplicação e interpretador de comandos já foram dissertadas e na concepção deste novo modelo, elas não serão alteradas, visto que um dos requisitos para uma boa aceitação de um novo modelo de armazenamento é a adaptabilidade a modelos pré-existentes, desta forma, atendendo a este importante critério. Nesta proposta de modelo foram criadas as camadas de indexação, hierarquização, virtual de acesso, paralelização, serialização, distribuição de dados, virtual de acesso e por fim a camada de armazenamento.

A **Camada de Indexação** foi concebida no sentido de indexar os metadados advindos do servidor de imagens, visto que isto é um requisito muito importante para um sistema desta natureza. Nessa camada estão implementadas funções de busca (*C-Find*) e recuperação (*C-Get* e *C-Move*) de de alto nível para que os desenvolvedores da aplicação possam usá-las de forma mais usual. Vale ressaltar que em um cenário clássico, os desenvolvedores integrariam sua aplicação usando conectores e interfaces, da linguagem de programação, aos sistema gerenciadores de bancos de dados relacionais.

A **Camada de Hierarquização** é responsável por receber os dados advindos do servidores de imagens, entre eles os metadados (desconstruídos) e hierarquizá-los dentro do formato de dados especificado. Ela tem função de proporcionar preservação dos dados, acesso randômico, portabilidade e multiplataforma. Num conceito de mais alto nível, nesta camada que esta localizado a modelagem do sistema de armazenamento de dados, onde há descritores que indicam como determinado metadados vai ser hierarquizado e em que ponto da hierarquia, para posteriormente ser armazenado.

A **Camada Virtual de Arquivos** pode ser visualizada na Figura 27. Este é um componente do modelo que tem a função de tratar as operações de entrada e saída (I/O) em baixo nível, de forma que a aplicação possa ler e gravar dados utilizando variados recursos de armazenamento de dados. Esta camada serve no modelo como uma interface entre as camadas inferiores e superiores, intermediando as operações de armazenamento, nos meios que forem declarados. Como exemplo, nesta camada que o MPI-IO pode ser integrado, abstraindo assim a complexidade de desenvolver para este tipo de sistema de armazenamento paralelo.

A **Camada de Paralelização** e **Camada de Serialização** são as camadas que determinam qual tipo de escrita e leitura será realizada para o modelo como um todo. Neste caso, é possível realizar escritas de forma paralela, dividindo os dados em partes e gravando simultaneamente em diferentes partes de um agregado computacional. Por outro lado, na camada de serialização esta usa tipos comuns de escrita e leitura, que pode ser tanto usada para agregados computacionais, como também em sistemas de arquivos convencionais. Para este modelo, esta sendo uma abordagem distribuída, entretanto, devido aos quesitos de flexibilidade já expostos, é possível também usar o mesmo modelo de forma local, em um único nodo computacional ou repositório de dados.

A **Camada de Distribuição de Dados** é a responsável por gerenciar a distribuição dos dados no ambientes distribuídos aplicados nas arquiteturas advindas deste modelo. A função é abstrair dos desenvolvedores as tarefas de alocação de dados nos nodos. Este é uma camada intermediária que serve como interface entre os métodos de gravação e leitura (serial ou paralelo) e

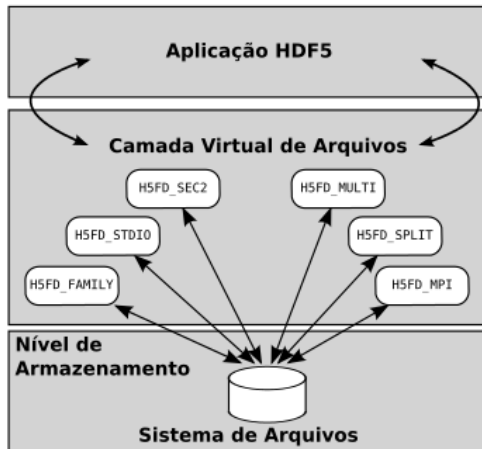


Figura 27: Camada Virtual de Arquivos

os sistemas de arquivos distribuídos que persistem de fato os exames.

A **Camada Virtual de Acesso** tem a função de intermediar as solicitações de distribuição de dados e os nodos computacionais onde os dados realmente estão armazenados. Nesta camada estão alocados *middlewares* dos sistemas de arquivos distribuídos que intermediarão os processos de gravação ou recuperação de dados. Por fim, a **Camada de Armazenamento** tem a função de servir de *backend* para os dados. Nela de fato que os objetos DICOM, já hierarquizados são armazenados. Nesta camada estão localizados os agregados computacionais, servidores de dados, sistemas de armazenamento de dados (*storages*).

A função geral das camadas expostas é expressar, de forma conceitual, como um sistema de armazenamento hierárquico é constituído. Modelos como estes servem de base para concepção e posterior desenvolvimento de arquiteturas que o implementarão na prática para suportar cenários reais. Na Seção 4.4 uma arquitetura baseada neste modelo será proposta, buscando implementar cada uma das camadas expostas, afim de validar o modelo que esta sendo proposto nesta tese.

4.3.3 Modelo de Dados e Indexação

Passada a etapa de construção do modelo conceitual de persistência, da mesma forma que em um modelo tradicional, os dados devem ser modelados

antes de serem persistidos. Desta forma, notou-se a necessidade de definir um modelo de dados específico para as imagens médicas. Para isto, foram realizadas reuniões com especialistas em imagens médicas, técnicos, gestores hospitalares e médicos para definir como seria o modelo que seria utilizado. Assim, seguiu-se estreitamente a documentação do padrão DICOM, com relação a estruturas e tipos de dados. Desta foi concebido o modelo de dados que é possível ser visto na Figura 28.

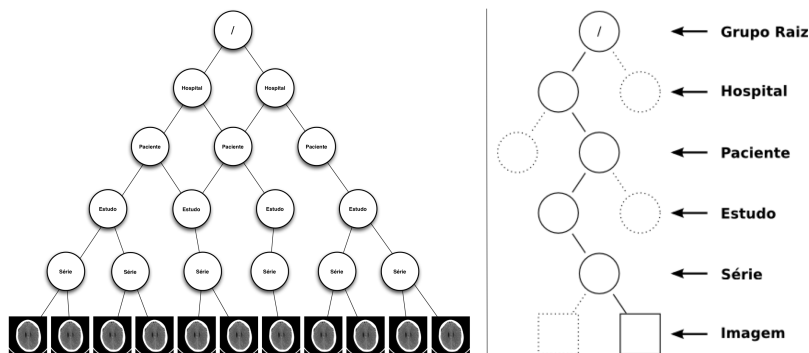


Figura 28: Modelo de Dados Hierárquicos.

Este modelo é constituído em um grupo raiz (/), do próprio mecanismo do HDF5, onde iniciam-se as operações de escritas e a hierarquia do arquivo. Logo abaixo, optou-se por determinar o hospital que o estudo está sendo realizado. Neste ponto, é importante dizer que mesmo o paciente possa fazer múltiplos exames, em instituições distintas, o modelo de dados prevê esta atividade e integrará o novo estudo ao prontuário do paciente. Isto é realizado através do sistema de links de *datasets* do HDF5.

Logo abaixo, está localizado o paciente, com todos os seus metadados de identificação, como nome, número de identificação, cartão SUS, endereço, nome da mãe e do pai, data de nascimento, entre outros. Um paciente pode ter um ou mais estudos, que contém uma ou mais séries, que contém uma ou mais imagens. Cabe ressaltar aqui que este modelo usa metadados de identificação do estudo, da série e da instância em cada um dos pontos de hierarquia e que estes dados servem para localizar e ordenar os exames. Este modelo de dados tem como característica ser altamente flexível, podendo ser modificado no momento da criação das estruturas do arquivo HDF5, não havendo implicações com o tamanho final do arquivo que irá persistir os exames.

Concomitante ao modelo de dados proposto, um modelo de indexação dos dados foi desenvolvido. Este modelo, exposto na Figura 29, foi desen-

volvido utilizando o mecanismo de indexação de documentos do CLucene, já apresentado anteriormente. O CLucene se apresenta como a escolha neste trabalho para a estratégia de utilizando índices invertidos. Ainda, o CLucene se apresenta como uma escolha conveniente para indexação de conjuntos de dados HDF5, devido a suas semelhanças de arquitetura, já apresentadas anteriormente.

Na Figura 29 é possível visualizar ainda como este processo de indexação é realizado. O processo inicia quando um objeto DICOM é desconstruído e os metadados repassados para o extrator de termos do CLucene que inicia a geração dos documentos. Nesta etapa do processo são gerados três documentos distintos, que estão interligados entre si, que são relacionados aos estudo, a série e a instância (imagem). Uma informação importante é que só metadados selecionados são indexados, com objetivo de localizar os arquivos no sistema de persistência.

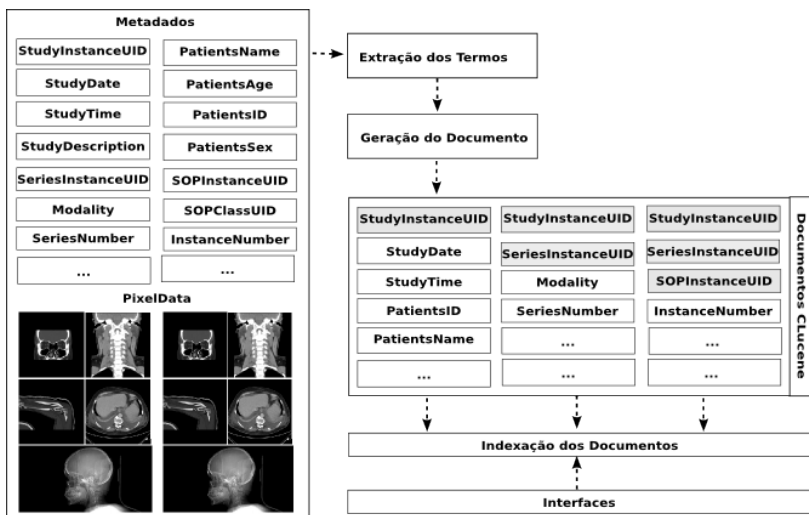


Figura 29: Extração dos Metadados e Geração do Índice.

Os metadados indexados foram definidos de acordo com o padrão DICOM, sendo estes os obrigatórios para sistemas de busca e recuperação de dados, de acordo com documentos da NEMA. Entre estes metadados, no nível do estudos são o StudyInstanceUID (identificação do estudo), StudyDate (data do estudo), StudyTime (hora do estudo), PatientID (identificação do paciente dono do estudo), PatientsName (nome do paciente). No nível da série, os metadados indexados são o StudyInstanceUID (relação com o estudo), SeriesInstanceUID (identificação da série), Modality (tipo de moda-

lidade da série), SeriesNumber (número da série). Por fim, no nível da imagem, ou instância, os metadados indexados são o StudyInstanceUID (relação com o estudo) SeriesInstanceUID (relação com a série), SOPInstanceUID (identificação da instância) e InstanceNumber (número da instância).

No final do processo de construção dos documentos CLucene, relacionados com os estudos, séries e instâncias, os 3 documentos são indexados e o processo tem finaliza. Por fim, os metadados selecionados para comporem o esquema de indexação deste modelo podem ser estendidos ou ainda, modificados, de acordo com a necessidade da aplicação, indexando novos termos, relacionados a metadados, que podem posteriormente serem recuperados pelo CLucene.

4.4 ARQUITETURA PROPOSTA

Baseado no modelo proposto nesta tese, que tem uma função conceitual como já mencionado, nesta seção será exposta uma arquitetura que foi desenvolvida para representar o modelo. Na Figura 30 é possível visualizar a arquitetura proposta para esta tese. Ela foi dividida em três camadas, sendo elas: Entrada de Dados (1), Tratamento de Dados (2) e Armazenamento de Dados (3). Estas camadas e suas funcionalidades, componentes e respectivas características serão expostas nesta seção.

Na camada de Entrada de Dados (1), são retratados os usuários, hospitais e o servidores de imagens médicas DICOM. Esta camada representa o modelo tradicional de operação de arquiteturas convencionais de persistência de dados. É possível visualizar usuários, que podem ser pacientes, técnicos, médicos, residentes, entre outros, se relacionando com o hospital. Este relacionamento esta figurado nas modalidades de exames que podem ser realizadas, como por exemplo ressonância magnética, raio-X digital, ultrassonografia, tomografia computadorizada, entre outros.

Após o processo de realização dos exames médicos baseado em imagem, nos respectivos equipamentos, há uma interface de comunicação que intermedia as operações de manipulação do exame em uma *workstation* radiológica ou o processo de gravação do exame na *bridge* DICOM. A *bridge* DICOM irá efetuar o envio via redes de comunicações de forma síncrona ou assíncrona, dependendo do tipo de ambiente a qual a arquitetura estiver operando. Foi abstraído desta arquitetura tratamentos de qualidade e garantia de *links* de comunicação entre os hospitais e o servidor de imagens, visto que isto esta fora do escopo desta proposta de modelo.

Após o envio por parte da *bridge* DICOM, o servidor DICOM recebe estes exames e efetua a desconstrução das imagens DICOM, extraindo to-

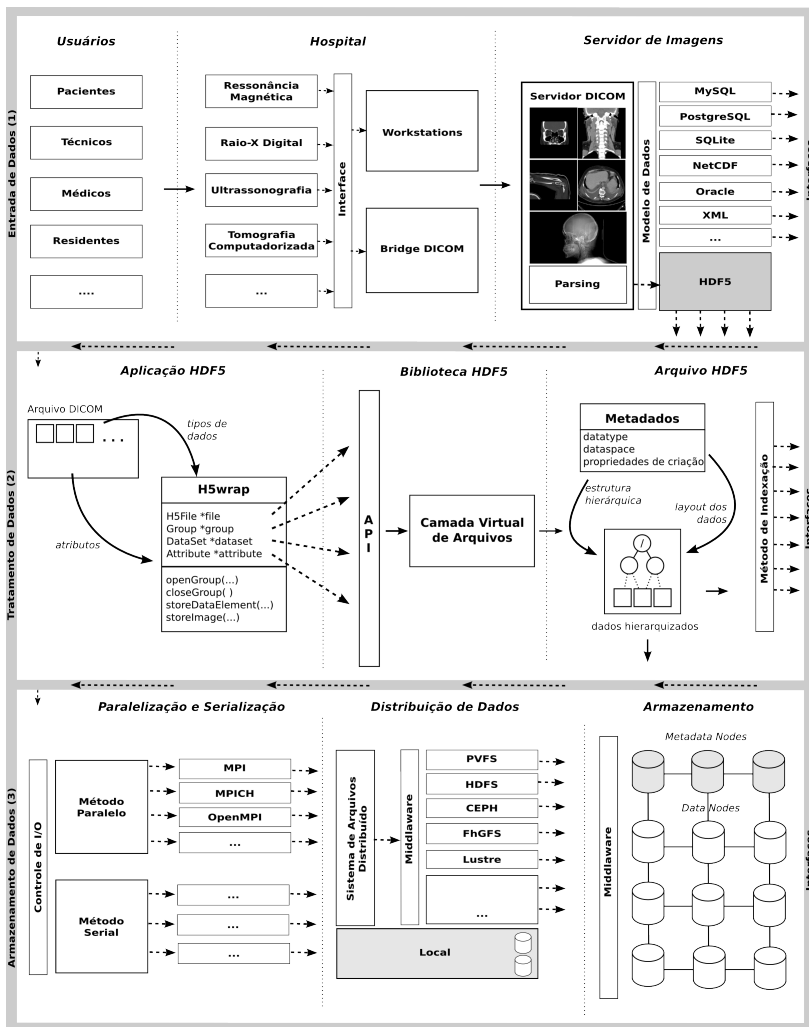


Figura 30: Arquitetura Proposta.

dos os metadados e as respectivas imagens (*pixel data*). Com as imagens do exame, desconstruídas, há-se a necessidade de modelá-las de acordo com o modelo de dados desejado para o sistema de persistência de dados. Os tipos de persistência de dados suportados são os sistemas gerenciadores de bancos de dados relacionais (MySQL, PostgreSQL, SQLite e Oracle), formatos de dados científicos, como o NetCDF ou HDF5 e até estruturas de persistências baseadas no formato XML.

No caso da arquitetura aqui apresentada, o sistema de persistência definido para hierarquização dos dados foi o HDF5, pelas razões já apresentadas. Desta forma, os exames passarão pela fase de modelagem de dados, onde os dados serão definidos e a informação passada para a camada inferior. Na camada de Tratamento dos Dados (2) receberá os dados já desconstruídos e modelados e enviará para o H5Wrap. O H5Wrap é um componente de software que tem a função de invocar os métodos da API do HDF5, preparando os dados para a hierarquização, criando os grupos, definindo os *datasets*, tipos de dados, entre outras características. Os dados já modelados e definidos são passados para a biblioteca do HDF5, onde os métodos serão executados na API e os dados passados para a camada virtual de arquivos do HDF5, onde será definido o mecanismo de persistência que será utilizado.

Por fim o arquivo de fato será gerado, ou ainda, integrado a uma hierarquia já existente, contendo os metadados enviados, juntamente com sua respectiva imagem. Nesta fase ainda, o mecanismo de indexação de dados, implementado usando índice invertidos usando o sistema CLucene, este irá capturar as *tags* propostas para indexação, que são: *StudyInstanceUID*, *PatientsName*, *PatientsID*, *PatientsBirthDate*, *PatientsBirthTime*, *PatientSex*, *StudyDate*, *StudyTime*, *AccessionNumber*, *StudyID*, *SeriesInstanceUID*, *Modality*, *SeriesNumber*, *SOPInstanceUID* e *InstanceNumber*. Não está contido a indexação do *pixeldata*, visto que isso causaria uma sobrecarga no sistema, e ainda, não há a necessidade, pois a função do indexador é proporcionar localização dos dados, para posterior recuperação.

Terminada a fase de envio dos dados e posterior tratamento dos mesmo, com estes já hierarquizados em uma estrutura HDF5 e com os metadados já indexados usando índice invertidos pelos mecanismos do CLucene, os dados de fato começam a ser persistidos. Uma importante observação sobre a arquitetura proposta é que os dados serão criados em tempo real já no *backend* selecionado, no caso um sistema de arquivos distribuídos. Com isto, ganha-se desempenho, pois não há uma etapa preliminar onde os dados já contidos em um arquivo HDF5 tenham obrigatoriamente que ser migrados para os servidores de dados remotos.

A camada de Armazenamento de Dados (3) tem a função de receber as chamadas da camada superior onde já vem com definições de tipo de

método de armazenamento que será utilizado para persistir os dados. Este métodos podem ser de forma paralela, onde processos concorrentes são criados para aumentar a vazão do processo e assim, ganhar desempenho e ainda, pode ser de forma serial, onde somente um processo é acionado e este grava sequencialmente os dados no *backend* a ser utilizado. No caso desta arquitetura foi definida a utilização do MPI-IO para o método paralelo, pois este é um método amplamente suportado nos múltiplos sistemas de arquivos distribuídos. Entretanto, pode-se utilizar outro métodos de paralelização, como o MPICH ou ainda o OpenMPI, que estão fora do escopo deste trabalho.

Passada a fase de seleção do método de gravação dos dados, o fluxo é direcionado para a fase de distribuição de dados. Nesta etapa, múltiplos sistemas de arquivos distribuídos podem ser integrados a arquitetura, para habilitar opções de persistência distintas. Para esta arquitetura, foram definidas e implementadas a arquitetura os seguintes sistemas: PVFS, HDFS, CEPH, FhGFS e o Lustre. Os métodos de gravação, tanto de forma paralela, quanto serial foram integradas aos *middlewares* de operação destes sistemas de arquivos para fornecer um maior nível de transparência para o processo.

Selecionados o método de gravação dos dados, e posteriormente o sistema de arquivos distribuídos que irá persistir os exames, os dados são passados para o sistema de escalonamento e armazenamento dos dados, onde estes serão divididos e alocados nos servidores de dados e nos servidores de metadados. Cada um sistemas de arquivos têm comportamentos distintos quanto a divisão, distribuição, armazenamento e recuperação dos dados. Em suma, todos eles dividem os arquivos em partes e armazenam eles e para efeitos de localização e redundância, usam os servidores de metadados. Uma importante informação aqui é que estes metadados não tem relação com os metadados do das imagens DICOM, pois estes já foram hierarquizados e estão contidos em um arquivo binário que será armazenado.

Ainda, nesta camada, pode haver cenários onde a distribuição dos dados não é um requisito do sistema, em que o hospital necessita ou opta por armazenar os arquivos localmente, em servidores locais. Como opção, esta arquitetura prevê este comportamento e desta forma, o processo de armazenamento dos exames encerra-se com os dados nos discos locais do servidor. Como adendo, é importante salientar que este não é o comportamento padrão da arquitetura, mas sim mais uma opção de armazenamento que busca proporcionar maiores níveis de adaptabilidade a arquiteturas tradicionais já existentes.

Para todas as três camadas da arquitetura, é possível visualizar interfaces de comunicação e interoperabilidade com o mundo externo à arquitetura. No caso da interface da camada de entrada de dados, esta faz relação a conexão com componentes externos a arquitetura, tanto de entrada, como de

saída, como por exemplo conexão com bancos de dados relacionais, descritores XML, etc. Na camada de tratamento de dados, a interface se faz presente para receber solicitações externas de componentes que necessitam de um exame e desta forma, vão conectar ao indexador e solicitar a informação. Por fim, na camada de armazenamento de dados, a interface esta presente para administração dos dados, por parte de administradores do agregado computacional que estará suportando os dados.

4.4.1 Workflow da Arquitetura Proposta

De forma prática, a arquitetura proposta pode ser vista, na forma de fluxo de funcionamento (*workflow*) na Figura 31. Nesta figura é possível visualizar que o processo de forma completa. O processo inicia com a realização de um exame médico em um equipamento médico-hospitalar o qual gera imagens no formato DICOM. Estes dados são enviados via redes de comunicação baseadas na pilha de protocolos TCP/IP, para o servidor de imagens DICOM.

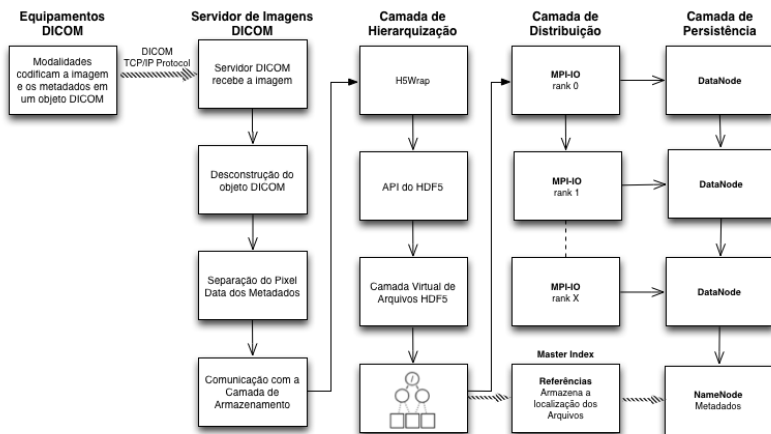


Figura 31: WorkFlow da Arquitetura Proposta para o Modelo.

O servidor de imagens irá receber a imagem e desconstruí-la, separando os metadados que descrevem a imagem ou o conjunto delas, da figura (*pixel data*). Os metadados segmentados e modelados serão enviados para a camada de hierarquização que executa o **H5Wrap**, invocando os métodos de criação do arquivo hierarquizado, na API do HDF5. O arquivo será criado e indexado, já solicitando um método de escrita (ou leitura), no caso do *work-*

flow apresentado aqui, de forma paralela. Desta forma, o MPI-IO irá disparar os múltiplos processos de escrita (ou leitura) para o agregado computacional.

4.4.2 Design do Arquivo no Formato HDF5

Em relação ao *design* do arquivo criado, ele esta de fato hierarquizado, no formato HDF5. Todas os exames, estudos, séries e respectivas imagens estão armazenadas em formato binário, em um único arquivo. Esta foi uma decisão de arquitetura tomada visando proporcionar um maior desempenho, visto que dividir o arquivo, em arquivos menores, causa um problema de gerenciamento de dados e ainda, um *overhead* desnecessário ao sistema. A tarefa de divisão do arquivo foi empregada aos *middlewares* dos sistemas de arquivos distribuídos, que irão dividir o arquivo criado, em múltiplos servidores de dados, com a redundância necessária, para caso de falhas de hardware ou software. A interação com este *container* de dados poderá ser realizada naturalmente pela arquitetura desenvolvida, ou ainda, usando as ferramentas de administração do próprio HDF5.

Para ilustrar, na prática, como uma imagem DICOM é representada dentro do *container* de dados HDF5, na Figura 32 é possível visualizar parte de uma imagem DICOM no formato HDF5, onde somente um pequeno trecho do cabeçalho foi exposto, e é possível ver alguns dos metadados, como por exemplo o *PatientID* e o *PatientsName*. A imagem DICOM no formato proposto não foi exposta por completo, pois uma imagem desta natureza, com todos seus metadados e o *pixeldata* é representada no formato HDF5 dentro da média de 300.000 (trezentas mil) linhas de texto.

À primeira vista, isto pode parecer ter um alto custo computacional, se comparada a modelo tradicionais, pois processar uma quantidade significativa de texto, em tempo real, pode causar um *overhead* nos sistemas de persistência. Entretanto vale ressaltar novamente que o arquivo é construído em formato binário, buscando assim um maior desempenho. A Figura 32 a qual foi recuperada usando ferramentas de extração do próprio conjunto de ferramentas do HDF5, onde ela efetuou um *dump* do arquivo binário, extraindo assim os dados solicitados em formato texto.

4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi apresentado o modelo proposto para armazenamento hierárquico de imagens médicas no formato HDF5. Foram apresentados o modelo conceitual do STT/RCTM, usado atualmente e sua arquitetura rela-

```

HDF5 "sample_dicom.h5" {
  GROUP "/" {
    GROUP "HU" {
      GROUP "MACEDO, DOUGLAS D. J." {
        GROUP "1.2.840.113619.2.22.287.1.000.20060315.203920" {
          GROUP "1.2.840.113619.2.22.287.1.000.6.20060315.204732" {
            GROUP "1.2.840.113619.2.22.287.1.000.6.20.20060315.204917" {
              DATASET "16_16" {
                DATATYPE H5T_STRING {
                  STRSIZE H5T_VARIABLE;
                  STRPAD H5T_STR_NULLTERM;
                  CSET H5T_CSET_ASCII;
                  CTYPE H5T_C_S1;
                }
                DATASPACE SIMPLE { ( 1 ) / ( 1 ) }
                DATA {
                  (0): "MACEDO, DOUGLAS D. J."
                }
                ATTRIBUTE "dataElementName" {
                  DATATYPE H5T_STRING {
                    STRSIZE H5T_VARIABLE;
                    STRPAD H5T_STR_NULLTERM;
                    CSET H5T_CSET_ASCII;
                    CTYPE H5T_C_S1;
                  }
                  DATASPACE SIMPLE { ( 1 ) / ( 1 ) }
                  DATA {
                    (0): "patientsName"
                  }
                }
              }
            }
          }
        }
        DATASET "16_32" {
          DATATYPE H5T_STRING {
            STRSIZE H5T_VARIABLE;
            STRPAD H5T_STR_NULLTERM;
            CSET H5T_CSET_ASCII;
            CTYPE H5T_C_S1;
          }
          DATASPACE SIMPLE { ( 1 ) / ( 1 ) }
          DATA {
            (0): "500907"
          }
          ATTRIBUTE "dataElementName" {
            DATATYPE H5T_STRING {
              STRSIZE H5T_VARIABLE;
              STRPAD H5T_STR_NULLTERM;
              CSET H5T_CSET_ASCII;
              CTYPE H5T_C_S1;
            }
            DATASPACE SIMPLE { ( 1 ) / ( 1 ) }
            DATA {
              (0): "patientID"
            }
          }
        }
      }
    }
  }
}

```

Figura 32: Exemplo de uma Imagem DICOM Hierarquizada

cionada, que implementa o modelo. Além disto, foram descritos todos os pontos de relacionamento entre os componentes desta arquitetura, buscando proporcionar um melhor entendimento do modo de operação atual do objeto de estudo de caso.

Em seguida, foi apresentado o modelo conceitual da proposta, que é composto pelas camadas de hierarquização, indexação, virtual de arquivos, paralelização, serialização, distribuição de dados, virtual de acesso e por fim, a camada de armazenamento de dados. Este modelo conceitual foi desenvolvido buscando abstrair tecnologias, e focando em componentes que um modelo desta natureza deve ter. Ainda, foi apresentado o modelo de dados hierarquizados no formato HDF5 e o modelo de indexação usando índices invertidos, por meio do CLucene.

Adiante, foi apresentada a arquitetura proposta para implementar o modelo conceitual proposto nesta tese. Esta arquitetura foi dividida em três camadas, sendo elas a camada de entrada de dados - que trata do recebimento dos dados no sistema, a camada de tratamento dos dados - que tem a função de tratar os dados e convertê-los para o formato proposto, e por último a camada de armazenamento de dados, onde de fato os processos de armazenamento são disparados, definindo o método de armazenamento e seu respectivo *backend*.

5 AMBIENTE E RESULTADOS EXPERIMENTAIS

Neste capítulo são apresentados os experimentos baseados no modelo proposto e em sua arquitetura. É explicada a metodologia procedural de testes para avaliação do modelo, bem como a arquitetura computacional utilizada nos testes. Em outra palavras, são mostradas as configurações dos computadores e os pacotes de software utilizados no protótipo. Finaliza-se o capítulo com considerações gerais relacionadas a esta fase da pesquisa.

5.1 ARQUITETURA COMPUTACIONAL

Com o intuito de prover maiores níveis de repetibilidade e/ou reprodução dos experimentos apresentados, na Figura 33 é possível visualizar a arquitetura computacional onde os experimentos foram realizados. Trata-se de uma arquitetura de sistemas de arquivos distribuídos convencional, composta por oito nodos que foram utilizados para armazenamento e recuperação das imagens médicas. No topo da arquitetura é possível ver dois componentes, separados em dois computadores distintos, sendo que um atuou como cliente DICOM e outro como servidor DICOM.

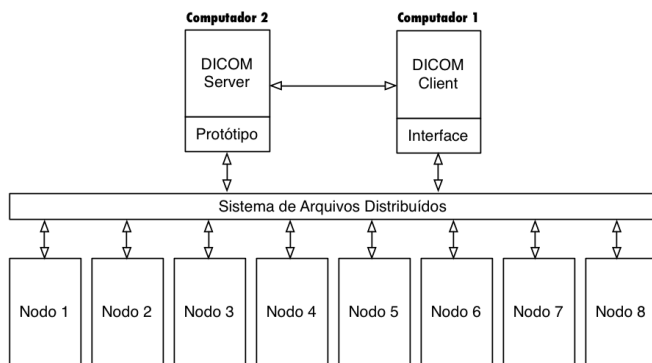


Figura 33: Arquitetura Computacional do Cluster utilizado nos Testes

Na Tabela 6 é possível visualizar a configuração básica dos computadores que compuseram a arquitetura computacional, onde são ilustrados os nomes dos *hosts* para identificação, quantidade de memória RAM, tipo de processador e quantidade de espaço em disco. Uma importante informação sobre os discos rígidos é que em relação aos nodos do *cluster*, a quantidade

Host	Memória	Processador	Disco
DICOM Server	16 Gb	Intel Core i5 2.8 GHz	500 Gb
DICOM Client	16 Gb	Intel Core i5 2.8 GHz	160 Gb
Nodo 1	8 Gb	Intel Core i5 2.8 GHz	20 Gb
Nodo 2	8 Gb	Intel Core i5 2.8 GHz	20 Gb
Nodo 3	8 Gb	Intel Core i5 2.8 GHz	20 Gb
Nodo 4	8 Gb	AMD Athlon x2 2.0 GHz	20 Gb
Nodo 5	8 Gb	Intel Core i5 2.8 GHz	20 Gb
Nodo 6	8 Gb	Intel Core i5 2.8 GHz	20 Gb
Nodo 7	4 Gb	AMD Athlon x2 2.0 GHz	20 Gb
Nodo 8	8 Gb	Intel Core i5 2.8 GHz	20 Gb

Tabela 6: Configurações dos Computadores utilizados nos Testes

disponibilizada para integração com o SAD foi de 20Gb por nodo, desta forma resultando em um espaço de armazenamento único de 160 Gb, visto que como característica dos SADs escolhidos, esses integram a quantidade de armazenamento entregue aos mesmos.

Ainda sobre a composição da arquitetura de testes, esta foi composta por máquinas sem dedicação exclusiva para os testes, que são utilizadas no dia a dia para desenvolvimento de software. Desta forma, derivaram em um *cluster* não-dedicado, sendo que todos os testes foram realizados em horários alternativos quando estas máquinas estavam ociosas. Por fim, sobre a parte de comunicação, todos os testes foram realizados em um barramento de 1Gb/s Ethernet.

Sobre os pacotes de software que compuseram os testes e suas respectivas versões, o sistema operacional utilizado no *cluster* foi o CentOS 6.2. O motivo para definição deste sistema operacional foi sua compatibilidade com os pacotes de sistemas de arquivos distribuídos utilizados nos experimentos. Já nas duas máquinas que continham o cliente DICOM e o servidor DICOM, o sistema operacional utilizado foi o Debian 6.0.8, posto a sua compatibilidade com o servidor de imagens médias DICOM (CyclopsDCMServer). Sobre os pacotes de software utilizados, abaixo segue a lista com suas respectivas versões.

- HDF5 – Versão 1.8.9
- MPI – MPICH2 – versão 1.4.1-p1
- PVFS – OrangeFS – versão 2.8.5.
- Lustre – Versão 2.1.55

- FHGFS – Versão 2011.04-r19
- CEPH – Versão 0.47.2-4
- CLucene – Versão 2.4.0
- PostgreSQL – Versão 8.4

5.2 METODOLOGIA DOS EXPERIMENTOS

Para a avaliação do modelo proposto nesta tese, foi idealizada uma metodologia de experimentos e nesta seção esta é discutida. A validação deste modelo esta sendo aplicada como estudo de caso, o primeiro experimento foi analisar o armazenamento e recuperação de imagens médicas no modelo atual, utilizando o sistema gerenciador de bancos de dados relacionais PostgreSQL. Desta forma, com os resultados do modelo atual é possível traçar comparações estatísticas com modelo que esta sendo proposto. Assim, todos os outros resultados obtidos serão comparados diretamente, com os resultados adquiridos nos testes do modelo atual.

Em um segundo momento a validação do modelo é efetuada a comparação da aplicação do modelo em diferentes softwares para construção de SADs. Assim, identificando qual destes pacotes de software tem o comportamento mais eficiente, se integrado ao modelo proposto. Sobre o sistemas de arquivos distribuídos que serão utilizados nos experimentos, estes são o Lustre, PVFS, CEPH e o FhGFS, por razões já expostas.

Como informação, o sistema de arquivos distribuídos HDFS, do Hadoop, foi excluído das rotinas de experimentos devido a problemas de estabilidade que o mesmo apresentou quando integrado com o modelo proposto, ficando este como um dos trabalhos futuros a serem desenvolvidos. Problemas destas natureza, utilizando o HDFS já foram relatados na literatura, como por exemplo em (TANTISIRIROJ et al., 2011).

A metodologia de experimentos para o SGBD foi a mesma dos testes com os SADs, ou seja, foram realizadas operações com 1000, 2500, 5000 e 10000 imagens, onde foram coletados os tempos de inserção por objeto e ainda os tempos totais da operação. Baseados nestes dados, foram geradas as médias, medianas, desvio padrão, maior número e menor número. O foco dos experimentos foi averiguar o desempenho e a escalabilidade do modelo hierárquico proposto. Nestes experimentos foram tratadas as operações de armazenamento e recuperação de imagens médicas DICOM, tando de forma serial, como de forma paralela.

Por exemplo: foram armazenadas 1000 imagens DICOM de forma serial no sistema de arquivos distribuídos CEPH, onde foram coletados o tempo

de cada uma das inserções no *container* HDF5 e o tempo total. Por fim, buscando garantir o mínimo de significância estatística aos resultados dos testes, e ainda, buscando proporcionar maiores níveis de confiabilidade nos dados apresentados, cada um dos testes foi realizados 25 vezes, onde extraiu-se a média, mediana, desvio padrão, menor valor e maior valor.

Sobre o tipo de exames que compõem a base de dados, estes foram retirados aleatoriamente de uma base de dados de testes, com exames devidamente anonimizados, para preservar a identidade dos pacientes. A proporção da quantidade de arquivos, por modalidades, foi determinada de acordo com a realidade de um sistema PACS de produção. Na Tabela 7 é possível visualizar os tipos de modalidades que foram utilizadas, a quantidade de imagens por modalidade e ainda, o tamanho destes conjuntos de dados. A heterogeneidade dos tipos de exames, de variadas modalidades diferentes é importante para simular um cenário real de um PACS em produção.

Modalidade	Quantidade	Tamanho
Raio-X Digital (CR)	100	1.1 Gb
Ressonancia (RM)	300	0.135 Gb
Tomografia (CT)	18530	9.2 Gb
Angiografia (XA)	1977	3.4 Gb
Ultra-som (US)	2370	2.1 Gb
TOTAL	23277	15.935 Gb

Tabela 7: Composição da Base de Dados

Acerca do modo de operação dos testes, o fluxo de informação foi iniciado no cliente DICOM, que envia para o servidor DICOM uma quantidade X de exames a serem armazenados, ou ainda, uma quantidade Y que deseja que sejam recuperados. O servidor DICOM, no caso do armazenamento, recebe a solicitação de armazenamento, e em tempo real inicia o recebimento das imagens e a posterior desconstrução das imagens e entrega para o modelo proposto (protótipo), que iniciará o procedimento de hierarquização, indexa os metadados do exame, faz a seleção do *backend* a ser utilizado, define a forma de armazenamento e de fato, finaliza com o armazenamento do dado. No caso do processo de recuperação de dados, o cliente DICOM solicita ao servidor DICOM a localização de uma determinada imagem. O servidor entrega a solicitação ao indexador do modelo proposto (protótipo). O indexador fará uma consulta interna em sua base de dados e informará ao servidor DICOM qual a localização do exame solicitado dentro do *container* de dados.

Com relação ao modo de operação dos testes, no caso dos processos de avaliação do modelo proposto, foram utilizados o cliente DICOM, o servidor DICOM, o Nodo 1 atuou como servidor de metadados e dados, enquanto

os outros sete servidores atuaram como servidores de dados. No caso de avaliação do modelo atual, foram utilizados o cliente DICOM, o servidor DICOM e o Nodo 1 atual como servidor de bancos de dados, onde de fato estava instalado o sistema gerenciador de bancos de dados PostgreSQL.

Finalizando o cenário relativo a metodologia, na Tabela 8 pode ser visualizado o tamanho dos conjuntos de dados e qual sua relação de tamanho para cada um dos *backends*, a saber, HDF5, CLucene, PostgreSQL e no sistema de arquivos. No caso do HDF5, esta informação faz referência ao tamanho do *container* HDF5 quando armazenado em um disco local. No caso do CLucene, esta informação faz referência ao tamanho do índice que será pesquisado. No caso do PostgreSQL, esta informação faz referência ao tamanho dos conjuntos de dados quando persistidos neste SGBD. No caso específico do *filesystem*, esta informação faz referência ao tamanho dos conjuntos de dados quando persistidos nos sistemas de arquivos distribuídos.

Imagens	1000	2.500	5.000	10.000
HDF5	695 MB	1.7 GB	3.4 GB	6.6 GB
CLucene	1004 KB	1.2 MB	2 MB	3 MB
PostgreSQL	488 MB	1.2 GB	2.4 GB	4.7 GB
FileSystem	562 MB	1.3 GB	2.7 GB	5.3 GB

Tabela 8: Tamanho da Base de Dados

5.3 EXPERIMENTOS REALIZADOS

Nesta seção é feito um detalhamento sobre a aplicação da metodologia e do ambiente nos experimentos realizados. Na Tabela 9 é possível visualizar o resumo dos testes aplicados na validação desta tese. Para cada um dos tipos de *backend*, foram realizadas as operações de armazenamento e recuperação, de conjuntos de dados compostos de imagens médicas, nos tamanhos de 1000, 2500, 5000 e 10000 imagens. No caso específico dos sistemas de arquivos distribuídos, ainda, foram realizadas operações de armazenamento e recuperação, de forma serial e paralela. Sobre a quantidade de vezes, como citado na metodologia, os testes foram realizados 25 vezes, para garantir o mínimo de significância estatística na validação dos resultados.

No total, foram realizados 18 diferentes testes de validação, sendo 16 avaliando o modelo proposto, utilizando os SADs Lustre, PVFS, CEPH e FhGFS e dois avaliando o modelo atual, utilizando o SGBD PostgreSQL. Como os testes foram realizados executando diferentes tipos de operação, para diferentes conjuntos de dados, isto resultou em um total de 72 diferentes

No	Backend	Tipo de Operação	Imagens	Qtde
1	Lustre	Armazenamento Serial	1000, 2500, 5000, 10000	25 vezes
2	Lustre	Armazenamento Paralelo	1000, 2500, 5000, 10000	25 vezes
3	Lustre	Recuperação Serial	1000, 2500, 5000, 10000	25 vezes
4	Lustre	Recuperação Paralelo	1000, 2500, 5000, 10000	25 vezes
5	PVFS	Armazenamento Serial	1000, 2500, 5000, 10000	25 vezes
6	PVFS	Armazenamento Paralelo	1000, 2500, 5000, 10000	25 vezes
7	PVFS	Recuperação Serial	1000, 2500, 5000, 10000	25 vezes
8	PVFS	Recuperação Paralelo	1000, 2500, 5000, 10000	25 vezes
9	CEPH	Armazenamento Serial	1000, 2500, 5000, 10000	25 vezes
10	CEPH	Armazenamento Paralelo	1000, 2500, 5000, 10000	25 vezes
11	CEPH	Recuperação Serial	1000, 2500, 5000, 10000	25 vezes
12	CEPH	Recuperação Paralelo	1000, 2500, 5000, 10000	25 vezes
13	FhGFS	Armazenamento Serial	1000, 2500, 5000, 10000	25 vezes
14	FhGFS	Armazenamento Paralelo	1000, 2500, 5000, 10000	25 vezes
15	FhGFS	Recuperação Serial	1000, 2500, 5000, 10000	25 vezes
16	FhGFS	Recuperação Paralelo	1000, 2500, 5000, 10000	25 vezes
17	PostgreSQL	Armazenamento	1000, 2500, 5000, 10000	25 vezes
18	PostgreSQL	Recuperação	1000, 2500, 5000, 10000	25 vezes

Tabela 9: Resumo dos Experimentos.

tipos de testes. Estes 72 (setenta e dois) diferentes tipos de testes, foram executados 25 (vinte e cinco) vezes, cada um deles, resultando num total de 1.800 (mil e oitocentos) testes. Esta quantidade de experimentos resultou em um total de 18.000 para os testes com 1000 objetos, 45.000 para os testes de 2500 objetos, 90.000 para os testes com 5000 objetos e 180.000 para os testes com 10000 objetos, totalizando 8.325.000 amostras que foram analisadas.

Então, esta quantidade significativa de amostras foram sintetizadas por médias e quantidades totais, gerando uma soma do tempo total e uma média para cada um dos testes. Os gráficos foram gerados para expressar as amostras de uma forma mais lógica. As amostras foram agrupadas por tipos de operação e posteriormente separadas em quatro experimentos distintos. O Experimento 01 tratou do armazenamento de forma serial, o Experimento 02, tratou do armazenamento de forma paralela, o Experimento 03 tratou a recuperação de forma serial e por fim o Experimento 04 tratou a recuperação de forma paralela.

Como observação final, em relação a execução dos experimentos, para todos os gráficos dos testes, os dados de armazenamento e recuperação do modelo atual, utilizando bancos de dados relacionas, foram inseridos nos gráficos para efeitos comparativos. Assim, foi possível traçar observações sobre o desempenho e a escalabilidade do modelo atual *versus* o proposto. As fontes semi-sintetizadas dos gráficos desta seção, estão expressas nos Apêndices C, D, E e F. Nas próximas sub-seções os experimentos são apresentados.

5.3.1 Experimento 01: Armazenamento Serial

Nesta sub-seção é visto o Experimento 01, que trata do armazenamento serial de dados nos sistemas de arquivos distribuídos (SADs). A metodologia de experimentos foi aplicada e na Figura 34 pode-se visualizar o tempo médio individual para cada umas das 25 tentativas e na Figura 35 é possível visualizar o tempo serial total para cada umas das 25 tentativas (operações). Para os dados de médias, os dados estão explicitados em milionésimos (7 casas decimais) de segundos e para os totais apresentados, estão expressos em centésimos (duas casas) de segundos.

O objetivo da primeira parte do experimento é identificar qual o tempo médio de armazenamento de uma imagem DICOM (instância) através do modelo proposto, assim as imagens são hierarquizadas e armazenadas de forma serial nos sistemas de arquivos distribuídos selecionados. Ainda, consta nos gráficos os tempos das mesmas tentativas, usando o modelo atual utilizando uma arquitetura de bancos de dados relacionais PostgreSQL. No mesmo gráfico, ainda é possível verificar a diferença de desempenho e estabilidade entre os próprios SADs selecionados, projetando assim qual deles tem o melhor desempenho no experimento.

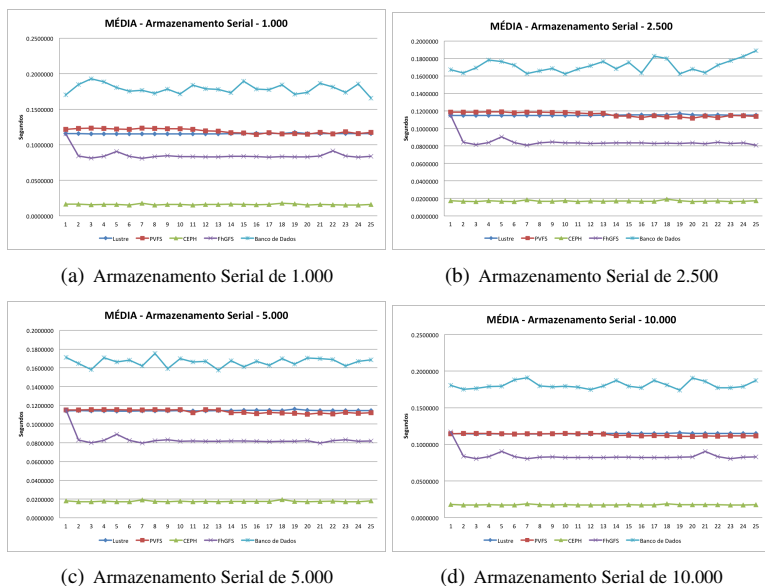


Figura 34: Gráficos do Tempo Médio de Armazenamento Serial.

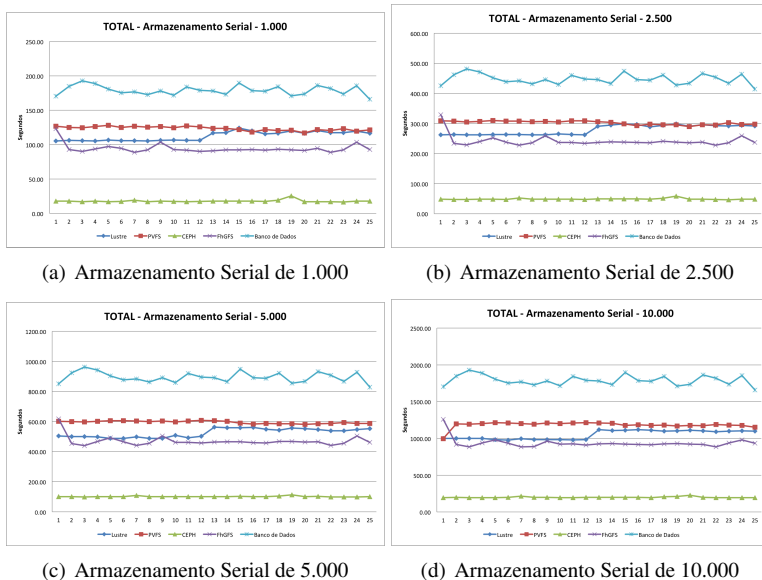


Figura 35: Gráficos do Tempo Total de Armazenamento Serial.

Na segunda parte do Experimento 01, a Figura 35 ilustra os tempos totais de armazenamento, esta segue a mesma metodologia da primeira parte do Experimento, com a diferença que esta, tem como objetivo identificar os tempos totais para persistência dos conjuntos de dados de forma serial nos sistemas de arquivos distribuídos e utilizando bancos de dados, no modelo atual. Com este experimento é possível prever quanto tempo em média, é o custo computacional para armazenar 1000, 2500, 5000 e 10000 imagens, tanto para o modelo proposto nesta tese, quanto no modelo relacional.

Como resultado preliminar deste primeiro experimento, é possível constatar que o modelo proposto, utilizando o HDF5 e sistemas de arquivos distribuídos como *backend*, para qualquer um dos casos de armazenamento serial apresentados, tem desempenho e estabilidade superiores a abordagem usando bancos de dados.

Esta inferência é válida tanto para os resultados de médias, da primeira parte do experimento, quanto para os resultados totais, na segunda parte do experimento. No capítulo 6 uma análise mais detalhada será realizada, notando as diferenças entre às abordagens.

5.3.2 Experimento 02: Armazenamento Paralelo

Nesta sub-seção é visto o Experimento 02, que trata do armazenamento paralelo de dados nos sistemas de arquivos distribuídos (SADs). A metodologia de teste dos experimentos foi aplicada, e na Figura 36 pode-se visualizar o tempo médio individual para cada uma das 25 tentativas (operações) e na Figura 37 é possível visualizar o tempo total de armazenamento paralelo para cada uma das 25 tentativas (operações) de persistência. Para os dados de médias, os dados estão explicitados em milionésimos de segundos e para totais apresentados, os dados estão expressos em centésimos de segundos.

O objetivo da primeira parte do experimento, expresso na Figura 36 é identificar qual o tempo médio de armazenamento, no modo paralelo, de uma imagem DICOM (instância) através do modelo proposto. Ainda na mesma Figura, são ilustrados os tempos médios de gravação, de forma convencional, do modelo atual, usando bancos de dados. No gráfico é possível ainda, visualizar quais dos sistemas de arquivos distribuídos tem melhor desempenho executando o experimento.

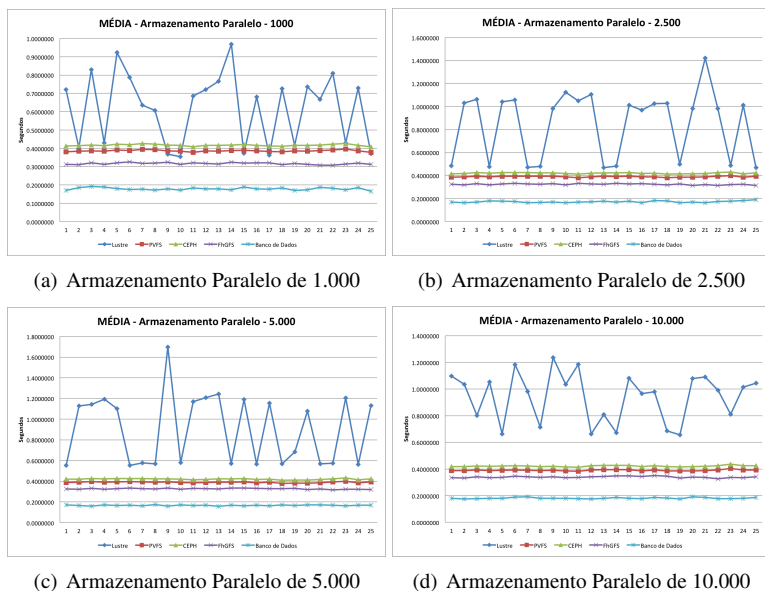


Figura 36: Gráficos do Tempo Médio de Armazenamento Paralelo.

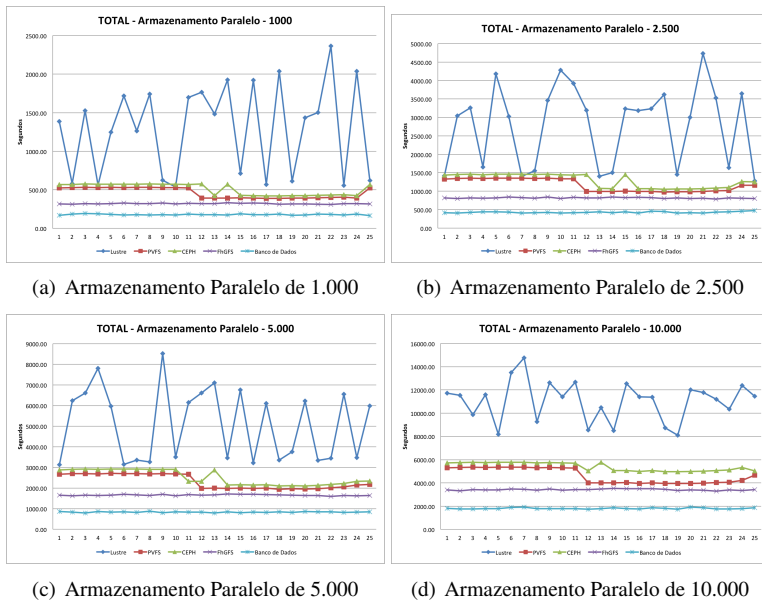


Figura 37: Gráficos do Tempo Total de Armazenamento Paralelo.

Na segunda parte do experimento, que pode ser visto na Figura 37, este trata os tempos totais de armazenamento das imagens de forma paralela, usando o modelo proposto. Esta etapa segue a mesma metodologia da primeira parte, entretanto esta etapa difere pois tem como objetivo averiguar os tempos totais para persistência dos conjuntos de dados e compará-los entre si e contra o modelo atual. Com a execução deste experimento é possível prever qual o tempo total em média para armazenar os conjuntos de dados.

Como resultados preliminares deste primeiro experimento, é possível constatar que o modelo proposto nesta tese, usando armazenamento de imagens médicas, em sistemas de arquivos distribuídos, de forma paralela, não teve desempenho superior ao modelo atual, usando bancos de dados. Ainda, foi possível constatar que o modelo, integrado ao sistema de arquivos distribuídos Lustre em específico, apresentou uma instabilidade insatisfatória, com alta variação dos tempos na operação dos testes. No capítulo 6 uma análise mais detalhada será traçada, observando as diferenças entre as abordagens.

5.3.3 Experimento 03: Recuperação Serial

Nesta sub-seção é visto o Experimento 03, que tem o intuito de abordar a parte de recuperação das imagens DICOM, de forma serial, usando o modelo proposto nesta tese, e ainda, usando o modelo atual utilizando bancos de dados. A metodologia de testes dos experimentos foi aplicada e na Figura 38 pode-se visualizar o tempo médio individual da recuperação de uma imagem, para cada umas das 25 tentativas (operações). Na Figura 39 é possível visualizar o tempo total da recuperação dos conjuntos de dados de forma serial, usando o modelo proposto. Para os dados de médias, estes estão explicitados em milionésimos de segundos e para os dados totais, estes estão expressos em centésimos de segundos.

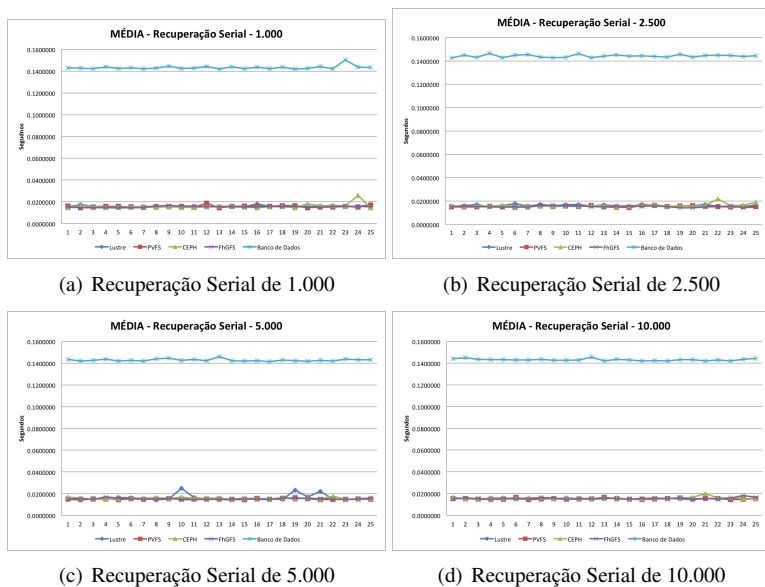


Figura 38: Gráficos do Tempo Médio de Recuperação Serial.

O objetivo da primeira parte do experimento, expresso na Figura 38 é identificar qual o tempo médio de recuperação, de forma serial, de uma imagem DICOM através do modelo proposto. Ainda na mesma figura, são apresentados os tempos médios de recuperação utilizando bancos de dados. A diferença deste experimento, para o experimento de armazenamento é que, enquanto o primeiro trata o armazenamento de 1000, 2500, 5000 e 10000 imagens, o experimento de recuperação busca, através do indexador do mo-

delo, uma (1) imagem dentro de um conjunto de dados de 1000, 2500, 5000 e 10000 imagens, desta forma, os tempos de recuperação não variam tanto, pois trata-se de apenas um objeto.

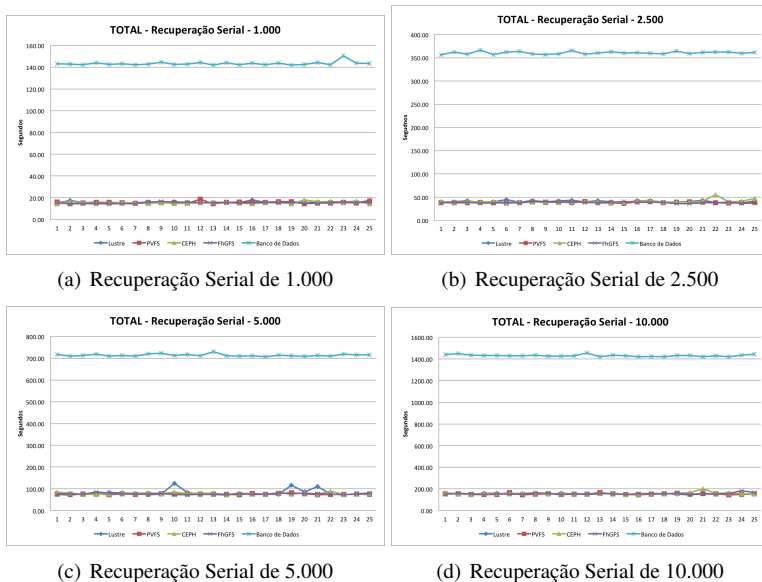


Figura 39: Gráficos do Tempo Total de Recuperação Serial.

Já na segunda parte do experimento, que pode ser visualizado na Figura 39, este trata da recuperação de forma serial, de 1000, 2500, 5000 e 10000 imagens aleatórias, dentro dos *containers* de dados já hierarquizados, utilizando o modelo proposto. Esta etapa segue a mesma metodologia dos testes anteriores, entretanto ela tem o intuito de averiguar qual o tempo total de recuperação de forma serial dos conjuntos de dados, e ainda, compará-los com o mesmo processo de recuperação para os mesmos conjuntos de dados, usando o modelo atual. Com a execução deste experimento é possível prever o tempo total em média de recuperação.

Sobre a primeira parte do Experimento 03, visto na Figura 38 é importante salientar que independente do tamanho do conjunto de dados, os tempos de recuperação não variam significativamente, visto que a recuperação proposta foi apenas para um objeto. Esta parte do experimento foi apresentada de forma separada, por tamanho dos conjuntos de dados, por fatores de normalização com o restante dos experimentos.

Como resultados preliminares do Experimento 03, que tratou da re-

cuperação de forma serial de imagens médicas DICOM, foi possível constatar que o modelo proposto nesta tese, teve desempenho superior ao modelo atual, utilizando bancos de dados. Ainda foi possível constatar uma estabilidade, com uma mínima variação, utilizando ambas abordagens. No Capítulo 6 uma análise mais detalhada será traçada, observando as diferenças entre as abordagens.

5.3.4 Experimento 04: Recuperação Paralela

Nesta sub-seção é visto o Experimento 04, que trata o problema de recuperação de imagens médicas DICOM, utilizando o modelo proposto, de forma paralela. Neste experimento, são apresentados os resultados da recuperação de dados utilizando o modelo atual, usando bancos de dados. A metodologia de testes foi aplicada e na Figura 40 é possível visualizar o tempo médio de recuperação de uma (1) imagem de forma paralela. Na Figura 41 é possível visualizar o tempo total de recuperação de forma paralela dos conjuntos de dados completos. Para os dados das médias apresentadas, estes estão explicitados em milionésimos de segundos e para os dados totais, estes estão expressos em centésimos de segundos.

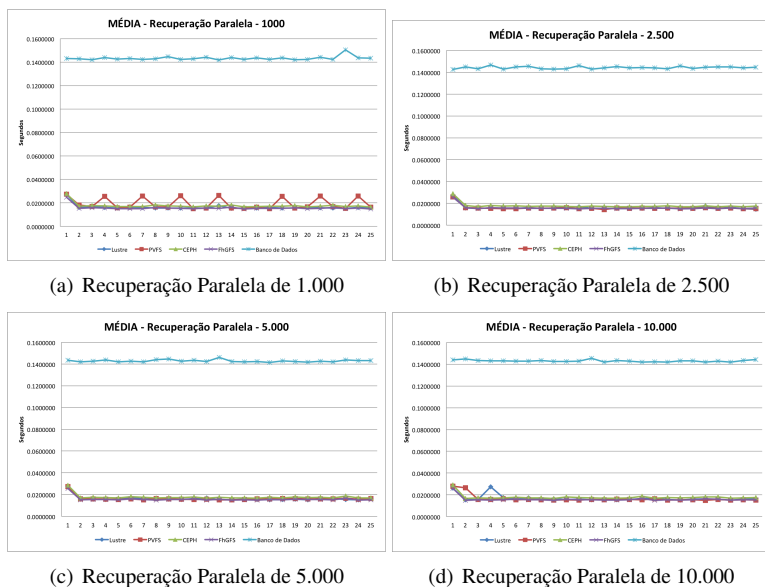


Figura 40: Gráficos do Tempo Médio de Recuperação Paralela.

O objetivo da primeira parte do experimento, visualizado na Figura 40 é identificar qual o tempo médio de recuperação de um objeto, de forma paralela, através do modelo proposto. Na Figura, se apresentou os tempos médios de recuperação usando bancos de dados, por fatores comparativos. Para esta primeira etapa, o mesmo modo de operação do experimento de recuperação serial foi utilizado, então, independente do tamanho do conjunto de dados apresentado, os tempos de recuperação variam pouco, pois trata-se de apenas um objeto.

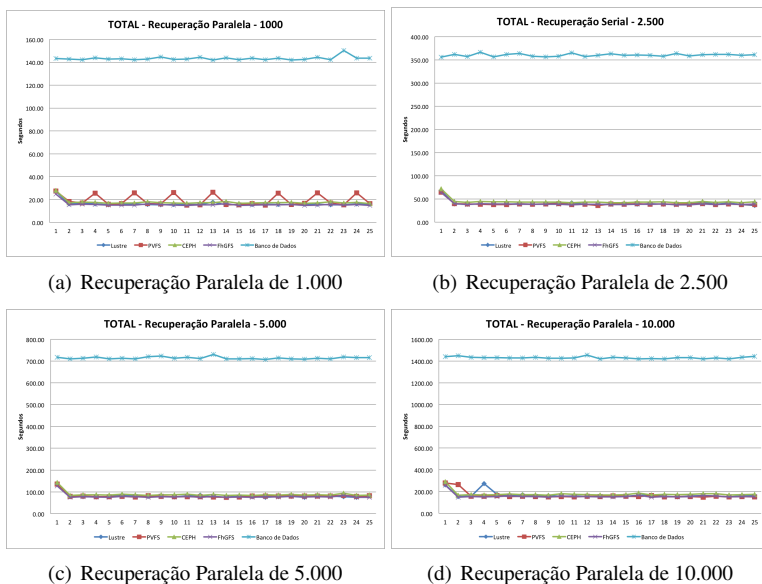


Figura 41: Gráficos do Tempo Total de Recuperação Paralela.

Ainda sobre o primeiro experimento, é importante salientar que os tempos de recuperação usando bancos de dados não foram feitos de forma paralela, mas sim, com seu modo de operação convencional, usando operações internas do SGBD, como por exemplo o comando SELECT. À primeira vista, pode-se parecer que o que esta sendo comparado, não é comparável, entretanto o intuito não é uma comparação direta, mas sim uma estimativa de desempenho para cada um dos modelos que estão sendo avaliados.

Na segunda parte do experimento, que pode ser visualizado na Figura 41, este trata da recuperação de forma paralela da totalidade dos conjuntos de dados. Esta etapa segue a mesma metodologia dos testes anteriores, entretanto tem o objetivo de averiguar qual o tempo total de recuperação para

cada conjunto, de forma a averiguar qual tem o melhor desempenho. Com este experimento é possível prever quanto tempo, ou seja, qual o desempenho no processo de recuperação dos *datasets*.

Como resultados preliminares do Experimento 04, que trata da recuperação de forma paralela de imagens DICOM, foi possível constatar que o modelo proposto, independente do tipo de *backend* a ser utilizado, tem um desempenho superior para recuperação de imagens DICOM de forma paralela. Salienta-se que de forma similar a recuperação de forma serial, a estabilidade no processo de recuperação das imagens foi o destaque do experimento, utilizando ambos os modelos.

5.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram demonstrados os experimentos realizados baseados no modelo proposto e em sua arquitetura. Foi apresentada a metodologia de testes para avaliação do modelo, bem como a arquitetura computacional utilizada nos testes, onde foram mostradas as configurações dos computadores e os pacotes de software utilizados no protótipo. Os experimentos propostos foram explicados e divididos em 72 diferentes tipos de testes, que foram executados 25 vezes, buscando proporcionar uma maior significância estatística para as avaliações. Desta forma, o processo de avaliação do modelo culminou em um total de 1800 testes, que derivaram 8.325.000 amostras a serem analisadas.

As amostras foram divididas em quatro experimentos distintos, que visaram analisar os tempos de armazenamento serial, armazenamento paralelo, recuperação serial e recuperação paralela. Observações preliminares foram traçadas onde pôde-se constatar que o modelo proposto é superior em três dos quatro experimentos. O Experimento 03, que trata do armazenamento de forma paralela não obteve resultados superiores a abordagem do modelo atual.

6 ANÁLISES E DISCUSSÕES DOS EXPERIMENTOS

Neste capítulo é efetuada a análise dos experimentos realizados no Capítulo 5. Busca-se traçar observações sobre os resultados obtidos e ainda, elaborar comparações entre o modelo proposto, utilizando formatos de dados hierárquicos de forma distribuída e paralela, para armazenamento e recuperação de imagens médicas DICOM *versus* o modelo atual, utilizando o sistema gerenciador de bancos de dados PostgreSQL. O capítulo está dividido em duas partes, sendo a primeira delas a análise dos resultados, que foi dividida na análise do processo de armazenamento e posteriormente, a análise do processo de recuperação de dados. Na segunda parte, é proposta uma discussão sobre os resultados e as análises obtidas.

6.1 ANÁLISE

Em relação a maneira que as análises são traçadas, estas estão divididas em duas partes. A primeira relativa ao processo de armazenamento de imagens médicas usando o modelo proposto. A segunda parte refere-se a análise do processo de recuperação das imagens. Como já mencionado na metodologia dos experimentos, as comparações serão realizadas em dois momentos distintos. No primeiro momento, os resultados dos experimentos com o modelo proposto serão comparados com os mesmos processos usando o modelo atual. Em um segundo momento, há uma análise onde dentro do modelo proposto, são analisados os resultados dos diferentes tipos de sistemas de arquivos distribuídos.

A análise apresentada caracteriza-se pelas avaliações de armazenamento de dados, onde há uma avaliação sobre o desempenho global, das médias totais dos experimentos, e na sequência há uma análise das médias dos conjuntos de dados, para que assim a análise fique mais sólida. Foi realizada a análise das avaliações de recuperação de dados, onde também há uma avaliação global, levando em consideração as médias totais dos experimentos e depois há uma análise das médias dos conjuntos de dados, desta forma, conseguindo uma análise mais precisa.

Uma importante observação sobre as análises de comparação de resultados, diz respeito aos dados do modelo atual, usando bancos de dados. À primeira vista, pode-se parecer que o que está sendo comparado, não é comparável, visto que os dados do modelo foram processados em outros tipos de *backend* e de forma distribuída. Entretanto o intuito não é uma comparação direta e definitiva, mas sim, adquirir uma estimativa de desem-

penho, para cada um dos modelos que estão sendo avaliados. Desta forma, este estudo serve também de base para tomadas de decisões sobre métodos de persistência de imagens DICOM.

6.2 ANÁLISE DAS AVALIAÇÕES DE ARMAZENAMENTO DE DADOS

6.2.1 Análise do Armazenamento

Na Figura 42 pode-se visualizar o tempo médio total, em segundos, do processo de armazenamento para os conjuntos de dados de 1000, 2500, 5000 e 10000 imagens médicas DICOM. Nos gráficos considera-se os tempos médios totais, para o processo de armazenamento, tanto serial, quanto paralelo. Por fatores comparativos, foram incluídos os tempos médios totais usando o modelo atual, por meio do SGBD PostgreSQL.

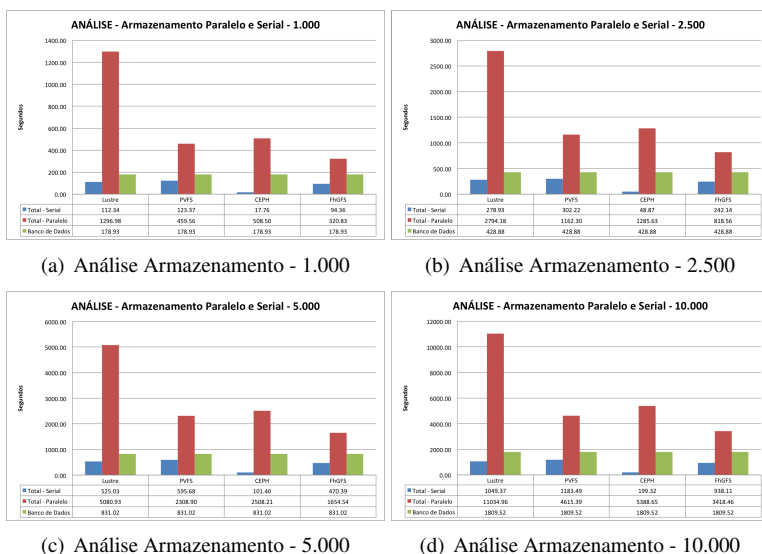


Figura 42: Gráficos da Análise das Avaliações de Armazenamento de Dados.

Na Figura 42a é possível ver os tempos analisados para o *dataset* com 1000 objetos. Em relação a comparação entre o modelo atual versus o modelo proposto, o modelo atual teve desempenho superior ao modelo proposto, para todos os tipos de sistema de arquivos distribuídos, se usados de forma paralela. Enquanto o tempo máximo usando escrita paralela no modelo proposto,

usando o Lustre como *backend*, foi de 1296,98 segundos, o tempo máximo de armazenamento de 1000 objetos usando bancos de dados foi de 178.93 segundos, resultando em uma diferença de 624.85%.

Em relação ao PVFS trabalhando em modo paralelo, a diferença de desempenho foi de 156.84%. Em relação ao CEPH também trabalhando de modo paralelo, a diferença de desempenho foi de 184.19%. E por fim, em relação ao FhGFS, a diferença foi de 79.30%. Nas Figuras 42b, 42c e 42d, que tratam dos conjuntos de dados de 2500, 5000 e 10000 objetos, esta tendência se confirmou e em todas as análises realizadas de forma paralela, o modelo atual usando bancos de dados teve desempenho superior ao modelo proposto usando sistemas de arquivos distribuídos. Esta é uma tendência natural, visto que às médias individuais tendem a variar, entretanto as médias totais se mantêm.

Finalizada a análise do modelo atual *versus* o modelo proposto de forma paralela, para armazenamento de imagens, como proposto, agora os sistemas de arquivos distribuídos são comparados entre si, a fim de verificar quais deles tem melhor desempenho. Em relação aos SADS, na Figura 42a, que refere-se ao armazenamento de 1000 objetos, o mais eficiente foi o FhGFS, realizando a operação em um tempo máximo de 320.83 segundos, em segundo lugar o PVFS realizou a operação em 459.56 segundos, em terceiro lugar veio o CEPH, que realizou a operação em 508.50 segundos e por último, em quarto lugar, o Lustre realizou a operação em 1296.98 segundos. Nas Figuras 42b, 42c e 42d, que tratam dos conjuntos de dados de 2500, 5000 e 10000 objetos, esta tendência se confirmou, pelo motivos já expostos e o FhGFS continuou sendo o mais eficiente fazendo o armazenamento dos objetos de forma paralela.

Ainda na Figura 42a, que trata do armazenamento de 1000 objetos, porém agora analisando o modelo proposto trabalhando de forma serial, para todos os quatro tipos de sistemas de arquivos distribuídos, estes tiveram desempenho superior ao modelo atual usando banco de dados. Enquanto o tempo mínimo para armazenamento de forma serial usando o modelo proposto foi do CEPH com 17.76 segundos, o tempo mínimo do modelo atual usando banco de dados foi de 178.93 segundos, resultando em uma diferença de desempenho de 907.49% em favor do CEPH.

Em relação ao Lustre, a diferença de desempenho foi de 74.84% mais eficiente. Em relação ao PVFS, a diferença de desempenho foi de 45.04% mais eficiente. E em relação ao FhGFS, a diferença de desempenho foi de 89.62% mais eficiente. Nas Figuras 42b, 42c e 42d, que tratam dos conjuntos de dados de 2500, 5000 e 10000 objetos, esta tendência se confirmou, e o CEPH continuou sendo o que obteve o mais eficiente desempenho realizando a operação dos objetos de forma paralela.

Finalizada a análise do modelo atual *versus* o modelo proposto de forma serial, para armazenamento de imagens, como proposto, agora os sistemas de arquivos distribuídos são comparados entre si, a fim de verificar quais deles tem melhor desempenho. Em relação aos SADs trabalhando de forma serial, na Figura 42a, que trata do armazenamento de 1000 objetos, o mais eficiente desempenho foi o CEPH, realizando a operação em 17.76 segundos, seguindo pelo FhGFS realizando a operação em 94.36 segundos, em terceiro lugar veio o Lustre, realizando a operação em 112.34 segundos e por último, em quarto lugar, veio o PVFS, realizando a operação em 123.37 segundos. Nas Figuras 42b, 42c e 42d, que tratam dos conjuntos de dados de 2500, 5000 e 10000 objetos, esta tendência se confirmou e o CEPH continuou sendo o que obteve o mais eficiente desempenho fazendo o armazenamento dos objetos de forma serial.

Por fim, na Figura 43 o gráfico evolutivo do armazenamento, tanto em modo serial, quanto paralelo, pode ser visualizado. O intuito do gráfico é mostrar a tendência de crescimento dos tempos, entre os sistemas de arquivos distribuídos usando o modelo proposto. No gráfico é possível visualizar, as mesmas conclusões já observadas durante a seção, desta forma, vê-se que o CEPH foi o mais eficiente SAD para armazenamento em modo serial e o FhGFS obteve mais eficiente desempenho trabalhando de forma paralela. Ainda, é possível observar que o pior desempenho do processo de armazenamento, em modo serial ficou com o PVFS e o pior, em modo paralelo foi o Lustre.

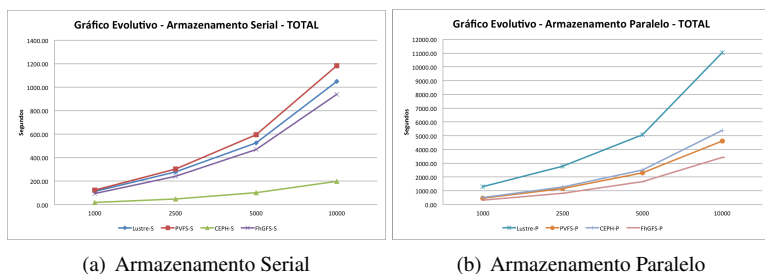


Figura 43: Gráficos Evolutivos das Avaliações de Armazenamento de Dados.

6.2.2 Tabela da Análise do Armazenamento

As análises dos experimentos definidas na seção anterior trataram dos tempos totais dos experimentos relacionados ao armazenamento, em modo

paralelo e serial. Nesta sub-seção uma análise similar será realizada, entretanto esta será realizada com os números das médias dos experimentos, e não com os números totais. Desta forma, espera-se uma análise ainda mais minuciosa e detalhada, pois os dados que serão analisados são apresentados na casa dos milionésimos de segundos. A tabela expressa na Figura 44 trata do armazenamento de imagens médicas DICOM, em modo serial e paralelo, usando o modelo atual com bancos de dados e o modelo proposto usando sistemas de arquivos distribuídos.

ARMAZENAMENTO						
	Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
Serial	1000	0.1156758	0.1193615	0.0160673	0.0852056	0.1789303
	2500	0.1152376	0.1160568	0.0170224	0.0847161	0.1715523
	5000	0.1143639	0.1134307	0.0175967	0.0834891	0.1662040
	10000	0.1146419	0.1131705	0.0175359	0.0843476	0.1809517
	Media	0.1149798	0.1155049	0.0170556	0.0844396	0.1744096
	Menor	0.1152188	0.1145939	0.0151985	0.0809648	0.1658424
	Maior	0.1175296	0.1236557	0.0179551	0.1153306	0.1927189
	Variação	0.0023109	0.0090618	0.0027566	0.0343658	0.0268765
	Desvio P.	0.0004653	0.0032223	0.0006728	0.0066792	0.0067928
	Resultado	51.69%	51.00%	922.60%	106.55%	110.14%
Paralelo	Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
	1000	0.6200134	0.3862493	0.4171493	0.3167072	0.1789303
	2500	0.8473340	0.3891680	0.4203014	0.3233841	0.1715523
	5000	0.9023495	0.3891348	0.4202656	0.3265248	0.1662040
	10000	0.9407164	0.3902256	0.4214437	0.3394586	0.1809517
	Media	0.8276033	0.3886944	0.4197900	0.3265187	0.1744096
	Menor	0.3550379	0.3782672	0.4085285	0.3080535	0.1658424
	Maior	0.9689758	0.3953390	0.4269661	0.3264247	0.1927189
	Variação	0.6139379	0.0170718	0.0184376	0.0183712	0.0268765
	Desvio P.	0.1931047	0.0043644	0.0047136	0.0051296	0.0067928
Resultado	374.52%	122.86%	140.69%	87.21%	181.32%	

Figura 44: Análise dos Experimentos de Armazenamento de Dados.

Na tabela é possível visualizar os quatro conjuntos de dados (1000, 2500, 5000 e 10000), ainda é possível visualizar os cinco tipos de *backends*, à saber Lustre, PVFS, CEPH, FhGFS e Bancos de Dados (PostgreSQL). Na tabela, são apresentados os dados das médias, menor amostra, maior amostra, variação, desvio padrão e por fim, o resultado. Quanto ao resultado, este foi expresso em porcentagem (%) para uma melhor visualização dos resultados obtidos. Uma importante observação é que a porcentagem expressa representa o percentual de desempenho em relação ao modelo atual, utilizando bancos de dados.

Para os testes com armazenamento serial, seguindo os resultados já expostos na seção anterior, o modelo atual teve desempenho inferior. Esta inferioridade de desempenho ficou exposta em 110.14% pior do que as médias dos sistemas de arquivos distribuídos que são usando como *backend* de dados

do modelo proposto nesta tese. O destaque deste experimento foi o CEPH, que obteve o menor tempo, que representa 922.60% mais eficiente que do os experimentos usando bancos de dados. Na sequência, o segundo mais eficiente foi o FhGFS, que foi 106.55% mais eficiente, seguindo pelo Lustre com 51.69% e por último, o PVFS com 51.00%.

Para os testes com armazenamento paralelo, os mesmos também seguiram os resultados já expostos na seção anterior, onde o modelo atual foi mais eficiente do que todas as análises realizadas com o modelo proposto, usando SADS. No caso, a abordagem usando bancos de dados foi 181.32% mais eficiente do que a média dos resultados do modelo proposto. O mais eficiente sistema de arquivos distribuídos foi o FhGFS, que obteve um desempenho 87.21% pior que o modelo atual, sendo seguido pelo PVFS que foi 122.86% pior, em terceiro lugar ficou o CEPH com 140.69% e em último lugar, o Lustre obteve um desempenho 374.52% pior.

Em relação a variação, nos testes de armazenamento serial, o que obteve a menor variação foi o Lustre, sendo seguido pelo CEPH, em terceiro lugar o PVFS, em quarto lugar veio a abordagem usando banco de dados e por último, com maior variação ficou o FhGFS. Em relação aos testes com armazenamento paralelo, o que obteve a menor variação foi o PVFS, em segundo lugar foi o FhGFS, em terceiro lugar foi o CEPH, em quarto lugar foi a abordagem usando bancos de dados e em último lugar foi o Lustre. Desta forma, pode-se inferir que no geral, a maior estabilidade é apresentada pelo Lustre, usado de forma serial, e ainda, a pior estabilidade também é do Lustre, só que em modo paralelo.

6.3 ANÁLISE DAS AVALIAÇÕES DE RECUPERAÇÃO DE DADOS

6.3.1 Análise da Recuperação

Na Figura 45 pode-se visualizar o tempo médio total, em segundos, do processo de recuperação para os conjuntos de dados de 1000, 2500, 5000 e 10000 imagens médicas DICOM. Nos gráficos foram inseridos os tempos médios totais, para o processo de recuperação, tanto em modo serial, quanto em paralelo, e ainda, por fatores comparativos e seguindo a metodologia proposta, foram inseridos os tempos médios totais usando o modelo atual, através do SGBD PostgreSQL.

Na Figura 45a é possível ver os tempos analisados para o *dataset* com 1000 objetos. Em relação a comparação entre o modelo atual *versus* o modelo proposto, o modelo proposto teve desempenho superior na totalidade dos testes, tanto trabalhando em modo serial, quanto em modo paralelo, utilizando

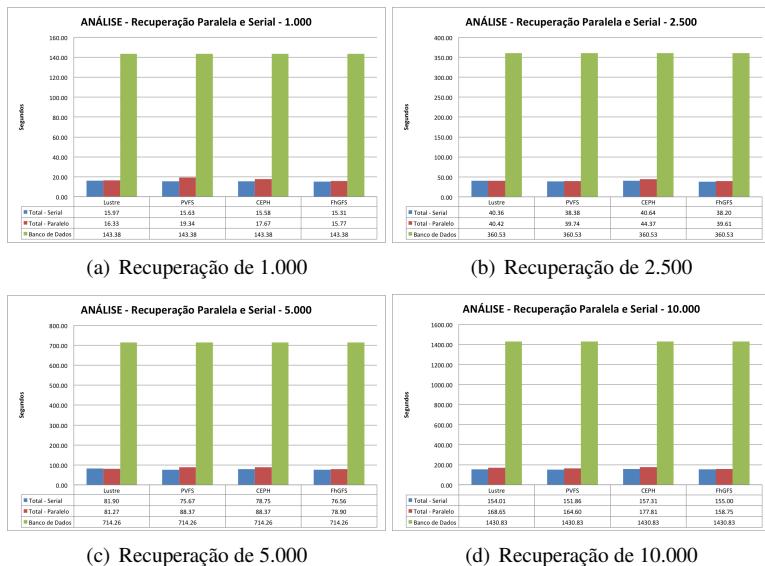


Figura 45: Gráficos da Análise das Avaliações de Recuperação de Dados.

qualquer um sistemas de arquivos distribuídos analisados. Enquanto o tempo máximo do modelo atual atingiu 143.38 segundos, o SAD que teve o pior desempenho, ou seja, com tempo máximo foi o PVFS, em modo paralelo, com 19.34 segundos, resultando em uma diferença de 641.37%.

Como comparação, nos testes de recuperação de 1000 objetos em modo paralelo, o FhGFS foi o que obteve a maior eficiência, obtendo 15.77 segundos, resultando em uma diferença de 809.19% em relação ao modelo atual. Em segundo lugar o Lustre obteve 16.33 segundos, resultando em uma diferença de 778.02%. Em terceiro lugar o CEPH obteve 17.67 segundos, resultando em uma diferença de 711.43%. E por fim, como já citado, o pior desempenho foi do PVFS, com um resultado de 19.34 segundos, resultando na menor diferença entre o modelo proposto, executado em modo paralelo, com 641.37%.

Como complemento, no restante dos experimentos de recuperação em modo paralelo, o modelo proposto foi mais eficiente, se comparado ao modelo atual. Nos testes de 2500 objetos, que pode ser visto na Figura 45b a ordem de desempenho entre os SADs foi alterada, sendo que o FhGFS manteve o melhor desempenho (809.85%), seguido pelo PVFS que obteve (806.97%), em terceiro lugar o Lustre (791.71%) e em último lugar o CEPH (712.33%). Nos testes de 5000 objetos, que pode ser visto na Figura 45c, o FhGFS man-

teve melhor desempenho (805.27%), em segundo lugar o Lustre (778.87%) e em terceiro e quarto lugares, empatados o PVFS e CEPH (708.26%). Por fim, nos testes de 10000 objetos, que pode ser visto na Figura 45d, o FhGFS manteve o melhor desempenho (800.74%), em segundo lugar o PVFS (769.28%), em terceiro lugar o Lustre (748.40%) e em último lugar o CEPH (704.40%).

Ainda, sobre a recuperação, na Figura 45a, que trata da recuperação de 1000 objetos, porém agora analisando os dados de recuperação serial, para os quatro tipos de SAD analisados como *backend* no modelo proposto, estes tiveram desempenho superior se comparados ao modelo atual, usando bancos de dados. Enquanto o tempo máximo utilizando bancos de dados atingiu 143.48 segundos, o tempo máximo de recuperação de 1000 objetos, usando o modelo proposto, usando o Lustre como *backend* atingiu 15.97 segundos, resultando em uma diferença de desempenho de 797.81%.

Como comparação, nos testes de recuperação de 1000 imagens em modo serial, o FhGFS foi o que obteve o melhor resultado, que obteve 15.31 segundos, culminando em uma diferença de desempenho de 836.51% em relação ao modelo atual. Em segundo lugar, o CEPH obteve 15.58 segundos, resultando em uma diferença de 820.28%. Em terceiro lugar o PVFS obteve 15.63 segundos, resultando em uma diferença de 817.34%. E por fim, como já citado, o pior desempenho foi do Lustre, com 15.97 segundos, resultando na menor diferença entre o modelo proposto e o modelo atual, executando processo de recuperação 797.81% mais rápido do que usando bancos de dados, porém sendo o pior SAD avaliado para este conjunto de dados.

Como complemento, no restante dos experimentos de recuperação em modo serial, o modelo proposto obteve melhor desempenho, se comparado ao modelo atual. Nos testes de 2500 objetos, que pode ser visto na Figura 45b a ordem de desempenho foi alterada, sendo o FhGFS manteve o melhor desempenho (843.80%), seguindo pelo PVFS (839.37%), em terceiro lugar o Lustre (793.29%) e em último lugar o CEPH (787.13%). Nos testes de 5000 objetos, que pode ser visto na Figura 45c, o PVFS foi o que obteve o melhor resultado (843.91%), sendo seguido pelo FhGFS (832.94%), em terceiro lugar o CEPH (807.00%) e em último o Lustre (772.11%). Por fim, nos testes de recuperação serial de 10000 objetos, que pode ser visto na Figura 45d, o PVFS obteve o melhor desempenho (842.20%), sendo seguido pelo Lustre (829.05%), em terceiro lugar o FhGFS (823.12%) e em último lugar o CEPH (809.56%).

Por fim, na Figura 46 é possível visualizar o gráfico evolutivo do processo de recuperação, tanto serial, quanto paralela. O objetivo do gráfico é proporcionar uma visualização geral sobre a tendência de crescimento dos tempos, entre os diferentes tipos de *backends* que integrados ao modelo proposto. No gráfico, é possível visualizar ainda, as mesmas inferências já ob-

servadas durante a seção, desta forma, vê-se uma evolução constante entre os diferentes tipo, mostrando baixa variação nos resultados.

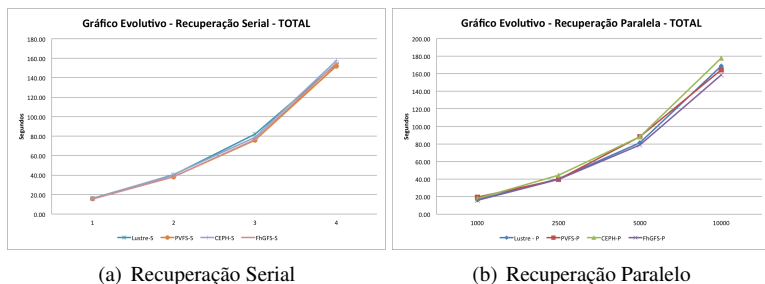


Figura 46: Gráficos Evolutivos das Avaliações de Recuperação de Dados.

6.3.2 Tabela da Análise da Recuperação

As análises dos experimentos definidas na seção anterior trataram dos tempos totais dos experimentos relacionados ao processo de recuperação, em modo serial e paralelo. Nesta sub-seção uma análise similar é realizada, entretanto esta será realizada levando em consideração os números das médias dos experimentos, e não os números totais. Desta forma, espera-se uma análise ainda mais minuciosa e detalhada, pois os dados que serão analisados são apresentados na casa dos milionésimos de segundos. A tabela expressa na Figura 47 trata do processo de recuperação de imagens médicas DICOM, usando o modelo proposto, tanto de forma serial, quanto paralelo e ainda, apresenta-se os dados do modelo atual, utilizando bancos de dados.

Na tabela é possível visualizar os quatro conjuntos de dados (1000, 2500, 5000 e 10000), ainda é possível visualizar os cinco tipos de backends, à saber, Lustre, PVFS, CEPH, FhGFS e Bancos de Dados (PostgreSQL). Na tabela, são apresentados dados das médias, menor amostra, maior amostra, variação, desvio padrão e por fim, o resultado. Quanto ao resultado, este foi expresso em porcentagem (%) para uma melhor visualização dos resultados obtidos. A porcentagem expressa representa o percentual de desempenho em relação ao modelo atual, utilizando bancos de dados.

Para os testes de recuperação serial, seguindo os resultados já expostos na seção anterior, porém agora detalhados, o modelo atual usando bancos de dados teve desempenho inferior, em comparação ao modelo proposto. Esta inferioridade de desempenho ficou exposta em 815.46% pior do que as médias dos sistemas de arquivos distribuídos que foram utilizados como *bac-*

RECUPERAÇÃO						
	Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
Serial	1000	0.0159706	0.0156286	0.0159387	0.0153139	0.1433799
	2500	0.0161429	0.0153509	0.0162566	0.0152789	0.1442123
	5000	0.0163808	0.0151335	0.0157492	0.0153118	0.1428511
	10000	0.0153962	0.0152728	0.0158879	0.0155811	0.1430825
	Media	0.0159726	0.0153465	0.0159581	0.0153714	0.1433815
	Menor	0.0149510	0.0144482	0.0145199	0.0144060	0.1419181
	Maior	0.0179551	0.0185201	0.0258980	0.0165241	0.1504240
	Varição	0.0030041	0.0040720	0.0113780	0.0021181	0.0085060
	Desvio P.	0.0007341	0.0008422	0.0022352	0.0005827	0.0016701
	Resultado	797.67%	834.30%	798.49%	832.78%	815.46%
	Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
Paralelo	1000	0.0163343	0.0193440	0.0176675	0.0157746	0.1433799
	2500	0.0161663	0.0158947	0.0177468	0.0158453	0.1442123
	5000	0.0162549	0.0163266	0.0176738	0.0157801	0.1428511
	10000	0.0168654	0.0164605	0.0177805	0.0158755	0.1430825
	Media	0.0164052	0.0170064	0.0177171	0.0158189	0.1433815
	Menor	0.0153019	0.0150459	0.0164797	0.0147140	0.1419181
	Maior	0.0270691	0.0274680	0.0278725	0.0248861	0.1504240
	Varição	0.0117671	0.0124221	0.0113928	0.0101721	0.0085060
	Desvio P.	0.0022964	0.0048093	0.0021749	0.0019419	0.0016701
	Resultado	774.00%	743.10%	709.28%	806.39%	756.68%

Figura 47: Análise dos Experimentos de Recuperação de Dados.

kend para o modelo proposto nesta tese. O destaque deste experimento foi o PVFS, que obteve o menor tempo em média, com 0.0153465 milionésimos de segundo, representando uma superioridade 834.30% mais eficiente que o modelo atual. Na sequência, o segundo mais eficiente, muito próximo foi o FhGFS com 0.0153714, representando uma superioridade 832.78% mais eficiente, em terceiro lugar o CEPH com 0.0159581, representando 798.49% mais eficiente e por último, em quarto lugar o Lustre com 0.0159726, que representa 797.67% mais eficiente que a abordagem usando bancos de dados.

Para os testes de recuperação paralela, seguindo os resultados já expostos, o modelo proposto obteve melhor desempenho em relação ao modelo atual, levando em consideração qualquer um dos *backends* utilizados para persistência dos dados. No caso da recuperação paralela, o modelo atual teve um desempenho 756.68% pior que a média dos sistemas de arquivos distribuídos. O mais eficiente SAD foi o FhGFS, representando uma superioridade de 806.39% mais eficiente se comparado com o modelo atual. Em segundo lugar o Lustre, sendo este 774.00% mais eficiente que o modelo atual. Em terceiro lugar o PVFS, sendo o mesmo 743.10% mais eficiente que o modelo atual. Em último lugar, o CEPH foi 709.28% mais eficiente.

Em relação à variação, nos testes de recuperação serial, o que obteve a menor variação foi o FhGFS, sendo seguido pelo Lustre, em terceiro lugar o PVFS, em quarto lugar a abordagem usando bancos de dados e em último

lugar o CEPH. Em relação à variação aos testes de recuperação paralela, o que obteve menor variação foi a abordagem usando bancos de dados, em segundo lugar o FhGFS, em terceiro lugar o CEPH, em quarto lugar o Lustre e em último lugar o PVFS. Desta forma, é possível inferir que a abordagem que tem a melhor estabilidade em relação a recuperação de forma serial é a que aplica o modelo proposto, usando o FhGFS, e ainda, a que representa a pior estabilidade é o PVFS, recuperando dados de forma paralela.

6.4 DISCUSSÕES FINAIS DOS EXPERIMENTOS

Na Figura 48 pode ser visualizada uma tabela com o estrato final dos experimentos, levando em consideração as médias dos testes realizados. O objetivo desta tabela é mostrar, de forma visual e direta, a classificação final (*ranking*) dos testes realizados, levando em consideração o modelo proposto, com os quatro (4) tipos de *backend*, à saber Lustre, PVFS, CEPH e FhGFS, em conjunto com o modelo atual, utilizando bancos de dados (PostgreSQL).

Armazenamento			
Armazenamento Serial	1º	CEPH	922.60%
	2º	FhGFS	106.55%
	3º	Lustre	51.69%
	4º	PVFS	51.00%
	5º	Banco de Dados	110.14%
Armazenamento Paralelo	1º	Banco de Dados	181.32%
	2º	FhGFS	87.21%
	3º	PVFS	122.86%
	4º	CEPH	140.69%
	5º	Lustre	374.52%
Recuperação			
Recuperação Serial	1º	PVFS	834.30%
	2º	FhGFS	832.78%
	3º	CEPH	798.49%
	4º	Lustre	797.67%
	5º	Banco de Dados	815.46%
Recuperação Paralela	1º	FhGFS	806.39%
	2º	Lustre	774.00%
	3º	PVFS	743.10%
	4º	CEPH	709.28%
	5º	Banco de Dados	756.68

Figura 48: Análise Consolidada dos Experimentos.

Baseado nos dados da tabela, para o processo de armazenamento de

dados, em modo serial, o primeiro lugar ficou com o CEPH, em segundo lugar o FhGFS, em terceiro lugar o Lustre, em quarto lugar o PVFS e em quinto e último lugar, a abordagem usando bancos de dados. Entretanto, nos testes de armazenamento de dados, em modo paralelo, a abordagem usando bancos de dados ficou em primeiro lugar, em segundo lugar o FhGFS, em terceiro lugar o PVFS, em quarto lugar o CEPH e em quinto e último lugar o Lustre.

Desta forma, após a finalização dos experimentos e suas respectivas análises, é possível inferir que, para as operações de armazenamento de imagens médicas DICOM, em modo serial, o modelo proposto nesta tese foi superior, atingindo melhor desempenho, se comparado ao modelo atual, utilizando bancos de dados. A abordagem proposta nesta tese mostrou-se superior na totalidade dos experimentos desta categoria. Ainda, é possível inferir, que baseados nos experimentos de armazenamento serial, que o CEPH foi o sistema de arquivos distribuídos que obteve um resultado mais eficiente, atingindo uma superioridade de 922.60% se comparado com a abordagem utilizando bancos de dados. Além disto, dentro das análises de *backends* para o modelo proposto, o CEPH foi o que obteve o melhor desempenho entre os sistemas de arquivos distribuídos analisados.

Já para operações de armazenamento, em modo paralelo, o modelo atual, utilizando bancos de dados, obteve melhor desempenho do que qualquer um dos tipos de sistemas de arquivos distribuídos utilizados como *backend* para o modelo proposto. Baseado nos experimentos realizados, é possível inferir que o custo de comunicação (*socket*), para o armazenamento de forma paralela dos conjuntos de dados, causou perdas significativas de desempenho, perdas estas que afetaram os resultados finais. Foi constatado, em um conjunto de 1000 amostras coletadas durante os testes, que o custo de comunicação atingiu a média de 56% do tempo total de armazenamento de uma imagem DICOM. Desta forma, isto escalado para a casa dos milhares de exames, resultou nos dados apresentados e na consequente perda de desempenho. Por fim, de acordo com experimentos, ainda é possível inferir que, baseado nas análises de armazenamento paralelo, para o modelo proposto, que o FhGFS obteve melhor desempenho entre o SADs analisados nos testes.

Ainda, baseado nos dados da tabela, para o processo de recuperação de dados, em modo serial, o primeiro lugar ficou com o PVFS, em segundo lugar o FhGFS, em terceiro lugar o CEPH, em quarto lugar o Lustre e em quinto e último lugar a abordagem usando bancos de dados. Por outro lado, nos experimentos de recuperação de dados, em modo paralelo, o primeiro lugar ficou com o FhGFS, em segundo lugar o Lustre, em terceiro lugar o PVFS, em quarto lugar o CEPH e em quinto e último lugar a abordagem usando bancos de dados.

Desta forma, é possível inferir que o modelo proposto, usando forma-

tos de dados hierárquicos de forma distribuída, para persistência de imagens médicas, obteve melhor desempenho nos experimentos de recuperação serial e também nos experimentos de recuperação paralela, se comparado com os dados de recuperação utilizando o modelo atual, com bancos de dados. Entretanto, para os experimentos de recuperação serial, houve uma variação entre os SADs que obtiveram melhor desempenho, dependendo do tamanho do conjunto de dados.

Para os conjuntos de 1000 e 2500 objetos, o FhGFS foi o SAD mais eficiente e para os conjuntos de 5000 e 10000 objetos, o PVFS foi o que obteve melhor desempenho. No geral, na comparação entre os *backends* do modelo proposto, o PVFS foi o sistema que obteve o melhor desempenho, sendo 843.30% mais rápido que o modelo atual. Sobre a recuperação em modo paralelo, o FhGFS foi o SAD que obteve o melhor desempenho entre os sistemas analisados, sendo este 806.39% melhor que o modelo atual utilizando bancos de dados.

Sobre a estabilidade das abordagens averiguadas nos experimentos, esta foi traduzida através da captura das variações de tempo entre as médias dos testes. Em relação ao armazenamento serial, a melhor estabilidade apresentada foi do Lustre e a pior foi do FhGFS. Com isto, é possível inferir que, mesmo o Lustre tendo ficado em terceiro lugar geral no ranking de desempenho, sua estabilidade é um ponto forte. Ainda, sobre o armazenamento serial, mesmo o FhGFS tendo ficado em segundo lugar geral, dentro destes experimentos, ele apresenta pior estabilidade, com uma variação de tempos acima da média.

Em relação ao armazenamento paralelo, a melhor estabilidade apresentada foi a abordagem usando bancos de dados, sendo esta 181.32% mais eficiente que a média do modelo proposto. Para os SADs analisados no modelo proposto, o PVFS e a pior estabilidade foi do Lustre. Com isto, é possível inferir que, mesmo o PVFS tendo ficado como terceiro lugar entre os sistemas analisados para o modelo proposto, este obteve melhor estabilidade. Ainda, sobre o armazenamento paralelo e sua consequente estabilidade, o Lustre foi o que obteve pior estabilidade e ainda, pior desempenho no geral.

Ainda sobre a estabilidade das abordagens averiguadas nos experimentos, em relação a recuperação serial o FhGFS foi o que obteve menor variação nos testes e o CEPH foi o que obteve maior variação e por consequência, pior estabilidade. Com isto, é possível inferir que mesmo o FhGFS ficando em segundo lugar geral, sendo este 832.78% mais eficiente que a abordagem atual, ele foi o mais estável. Ainda, sobre recuperação serial, mesmo o CEPH sendo o que obteve maior variação, ele foi 709.28% mais eficiente que a abordagem usando bancos de dados.

Em relação a recuperação paralela, a menor variação foi da abordagem

usando bancos de dados e a pior foi do PVFS. Com isto é possível inferir que mesmo a abordagem usando bancos de dados sendo a mais estável, no geral, contando-se a média geral dos experimentos do modelo proposto, esta é 756.68% pior do que a média geral para este experimento. Ainda, sobre recuperação paralela, mesmo o PVFS sendo o mais instável, este ainda ficou em terceiro lugar geral, para esta modalidade de testes.

6.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo abordou-se um conjunto de análises dos experimentos realizados. O objetivo foi traçar observações sobre os resultados obtidos e ainda, elaborando comparações entre o modelo proposto, utilizando formatos de dados hierárquicos de forma distribuída e paralela, para armazenamento e recuperação de imagens médicas DICOM *versus* o modelo atual, utilizando o sistema gerenciador de bancos de dados PostgreSQL. O capítulo foi dividido em duas partes, sendo a primeira delas a análise dos resultados, que foi dividida na análise do processo de armazenamento e posteriormente, a análise do processo de recuperação de dados. Na segunda parte, foi proposta uma discussão sobre os resultados e análises obtidas nos experimentos.

Como conclusão da etapa de análises dos experimentos realizados, o mais eficiente desempenho de armazenamento, contando o modo serial e paralelo, ficou com o CEPH, trabalhando em serial e o menos eficiente no processo de armazenamento no geral ficou com o Lustre, trabalhando em modo paralelo. Sobre a recuperação, o mais eficiente desempenho no geral ficou com o PVFS, trabalhando em modo serial e a menos eficiente foi a abordagem usando bancos de dados. Desta forma, com os resultados já expostos e analisados, é possível inferir o sucesso do modelo proposto e sua aplicação nos processos de armazenamento e posterior recuperação de imagens médicas DICOM.

7 CONCLUSÕES E TRABALHOS FUTUROS

Nesta tese foi apresentado um novo modelo de armazenamento para sistemas médicos baseados em imagens. Este modelo foi concebido a partir de um paradigma hierárquico, seguindo o padrão para imagens médicas internacionalmente reconhecido (DICOM). Assim, representando os dados de uma forma mais natural, simplificando o processo de busca e de recuperação, tanto do *pixeldata*, como dos metadados relacionados. Ao final dos experimentos e das etapas de análises dos resultados, o modelo proposto mostrou ter um desempenho superior, se comparado ao modelo clássico de SGBDs. Desta forma, o modelo, juntamente com sua posterior validação figura como principal contribuição desta tese.

Em relação aos objetivos propostos, foi possível (1) identificar os requisitos para elaboração de uma proposta de modelo hierárquico de armazenamento de dados para imagens médicas no formato DICOM. Esta identificação foi baseada através revisão bibliográfica onde foram elencados as características para este tipo de sistema e quais são as necessidades para sistemas desta natureza. Também, foi possível (2) identificar as plataformas computacionais de sistemas de arquivos distribuídos que melhor se adequam ao modelo criado, visando sempre manutenção de longo prazo. Esta identificação, também, foi realizada através da revisão da literatura, pois nesta, foram levantados os preceitos técnicos para este tipo de sistema, onde foram pesquisados o Lustre, o PVFS, o CEPH, o FhGFS e o HDFS como plataforma de persistência do modelo. Em complemento, o HDFS foi retirado das rotinas dos experimentos, devido a problemas de estabilidade apresentados.

Após as fases de levantamento bibliográfico e revisão sistemática da literatura, foi proposto um modelo hierárquico de armazenamento e recuperação utilizando o formato de dados hierárquicos HDF5 (3), que pôde ser discutido no Capítulo 4. Para implementar o modelo conceitual proposto, foi desenvolvida uma arquitetura computacional dividida em camadas, já integradas a arquitetura do STT/RCTM, que serviu como estudo de caso. Feita esta integração foi possível demonstrar (4) a viabilidade do modelo proposto por meio da implementação de um protótipo integrado com o estudo de caso. Foram realizados experimentos (5), buscando realizar uma análise comparativa utilizando o modelo proposto *versus* o modelo clássico de SGBDs.

Como conclusão, baseando-se na análise dos resultados experimentais, o mais eficiente desempenho de armazenamento, levando em consideração os modos serial e paralelo, foi o CEPH, trabalhando em serial e o menos eficiente no processo de armazenamento no geral foi o Lustre, trabalhando em modo paralelo, pelas razões relativas a parte de comunicação e sincronização.

Sobre a recuperação, o mais eficiente desempenho no geral ficou com o PVFS, trabalhando em modo serial e o menos eficiente desempenho foi a abordagem usando bancos de dados. Desta forma, com os resultados já expostos e analisados, é possível inferir o sucesso do modelo proposto e sua aplicação nos processos de armazenamento e posterior recuperação de imagens médicas DICOM.

Finalizando, é importante salientar que o intuito dos experimentos e sua posterior análise foi comprovar que o modelo proposto nesta tese é viável para persistência de imagens médicas DICOM de forma hierárquica e distribuída. Ainda, como já mencionado, devido a grande disseminação e uso de sistemas gerenciadores de bancos de dados relacionais, e sua aplicação em massa em sistemas de medicina em geral, esta tese não teve a pretensão de substituir imediatamente estes sistemas de persistência. O que se espera, é que quando engenheiros e desenvolvedores, deste tipo de sistema, se deparem com os problemas de pesquisa relacionados, busquem este modelo como uma das alternativas possíveis para melhoria dos tempos de resposta dos sistemas.

Por fim, este trabalho, partiu de uma contextualização, que identificou um problema de pesquisa e que este derivou em uma pergunta de pesquisa. Dito isto, após todas as etapas de revisão da literatura, que buscou embasar os preceitos técnicos do trabalho, após a etapa de revisão sistemática da literatura, buscando validar o ineditismo da tese, após a construção do modelo teórico e sua consequente arquitetura, e finalmente, após a etapa experimental e a análise dos dados, é possível responder a pergunta, inferindo que sim, que a aplicação de uma camada de persistência hierárquica e distribuída se mostrou viável e eficaz para persistência de sistemas médicos baseados em imagens.

Desta forma, verificou-se que o modelo proposto na tese, teve significativo reconhecimento pela comunidade científica, tanto da área médica, quanto da área da ciência da computação e da área interdisciplinar. Este reconhecimento pode ser aferido através da aceitação de vários artigos científicos publicados em *journals* e eventos internacionais.

7.1 TRABALHOS FUTUROS

Como sugestão de trabalhos futuros, sugere-se que o modelo proposto neste trabalho seja estendido, com integrações com outros tipos de *backend*, distribuídos e paralelos. Ainda neste sentido, a integração deste modelo com outros tipos de sistemas de arquivos distribuídos para Internet, sob paradigmas NoSQL, como por exemplo HDFS, do Hadoop, MongoDB, Casandra, Hypertable, entre outros.

A análise dos experimentos realizados nesta pesquisa foram direcionados ao estudo de caso, onde o modelo atual utilizou como *backend* o sistema gerenciador de bancos de dados PostgreSQL. Dito isto, outra pesquisa seria, a análise de experimentos para este modelo, levando em consideração outros tipos de SGBDs, como por exemplo o MySQL, Microsoft SQL Server, Oracle, entre outros.

Sobre a área de comunicação de dados, este trabalho não teve o intuito de realizar uma análise comparativa de métodos de comunicação, realizando tratamento dos métodos de comunicação, protocolos, sincronização, entre outros aspectos. Desta forma, indica-se como trabalhos futuros a realização de análises de comunicação, principalmente de forma paralela, pois estas foram as que obtiveram o pior desempenho, quando usadas para armazenamento de dados.

O modelo concebido foi criado com o objetivo de servir e suportar imagens médicas. Entretanto, trata-se de um modelo de armazenamento de imagens de propósito geral, desta forma, indica-se como outra pesquisa a aplicação do modelo para outros tipos de imagens, como por exemplo: imagens geoespaciais, imagens digitais para prospecção de solos, entre outros tipos de imagens, que além de conter *pixeldatas*, contenham significado através de metadados.

Por fim, outra sugestão refere-se ao cenários dos experimentos realizados para validação do modelo. Neste foi utilizado um *cluster* computacional formado por 8 máquinas, sendo uma deles servidor de metadados e as 8 servidores de dados. Desta forma, medições de escalabilidade e desempenho, quando o *backend* tiver 2, 4, 8, 16, 32, ou mais nodos não foram realizadas. Desta forma, acredita-se que um trabalho futuro de avaliação de desempenho do crescimento do *cluster* e ainda, nos sistemas de replicação de dados embutidos poderá ser diferencial.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDULJWAD, F.; NING, W.; DE, X. Sxm/r: Efficient way of storing and managing xml documents using rdbmss based on paths. In: IEEE. *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*. [S.l.], 2010. v. 1, p. V1–143.
- ALASDAIR, R.; BRUCE, A.; MILLS, J. *Smith AG CHIMP-MPI user guide*. [S.l.], 1994.
- ALTED, F.; VILATA, I. *OPSI: The indexing system of PyTables 2 Professional Edition*. [S.l.]: Exhaustive description and benchmarks about the indexing engine that comes with PyTables Pro. <http://www.pytables.org/docs/OPSI-indexes.pdf>, 2007.
- AMARAL, E. et al. An analysis of medical images data replication and recovery considering a database management system and a distributed file system. In: *The 18th IEEE Symposium in Computer and Communication (ISCC)*. [S.l.: s.n.], 2013.
- AMARAL, E. et al. Replicação distribuída de imagens médicas sob o formato de dados hdf5. In: *I2TS 2009* (. [s.n.], 2009. <<http://XXXXXX/62761.pdf>>.
- ANDRADE, R.; WAGNER, H.; WANGENHEIM, A. von. Telemedicina em santa catarina: um projeto sustentável. *XIII Congresso Brasileiro de Informática na Saúde*, 2012.
- ASSOCIATION, N. E. M. *Digital Imaging and Communications in Medicine (DICOM)*. [S.l.]: The Association, 1993.
- AZAGURY, A. et al. Towards an object store. In: IEEE. *Mass Storage Systems and Technologies, 2003.(MSST 2003). Proceedings. 20th IEEE/11th NASA Goddard Conference on*. [S.l.], 2003. p. 165–176.
- BAKER, M.; OUSTERHOUT, J. Availability in the sprite distributed file system. *ACM SIGOPS Operating Systems Review*, ACM, v. 25, n. 2, p. 95–98, 1991.
- BAKKER, A. R. His, ris, and pacs. *Computerized Medical Imaging and Graphics*, v. 15, n. 3, p. 157 – 160, 1991. ISSN 0895-6111. <http://www.sciencedirect.com/science/article/pii/089561119190004F>>.

BASHSHUR, R. On the definition and evaluation of telemedicine. *Telemedicine Journal*, v. 1, n. 1, p. 19–30, 1995.

BIDGOOD, W. D. et al. Understanding and using dicom, the data interchange standard for biomedical imaging. *Journal of the American Medical Informatics Association*, BMJ Publishing Group Ltd, v. 4, n. 3, p. 199–212, 1997.

BIRRELL, A. D. et al. The echo distributed file system. Citeseer, 1993.

BLAZONA, B.; KONCAR, M. HI7 and dicom based integration of radiology departments with healthcare enterprise information systems. *International Journal of Medical Informatics*, Elsevier, v. 76, p. S425–S432, 2007.

BMJ. *Regulamento sobre a Prevenção de Danos por Raios X*. [S.l.], 1987. 28.

BORTHAKUR, D. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, v. 11, p. 21, 2007.

BRAAM, P. et al. *The Lustre storage architecture*. 2004.

BROADBENT, R. E.; SAUNDERS, G. S.; EKSTROM, J. J. An infrastructure for the evaluation and comparison of information retrieval systems. In: *Proceedings of the 7th conference on Information technology education*. New York, NY, USA: ACM, 2006. (SIGITE '06), p. 123–127. ISBN 1-59593-521-5. <<http://doi.acm.org/10.1145/1168812.1168842>>.

CABRAL J.E., J.; KIM, Y. Multimedia systems for telemedicine and their communications requirements. *Communications Magazine, IEEE*, v. 34, n. 7, p. 20–27, jul 1996. ISSN 0163-6804.

CARVALHO, A.; LEON, F. Ponce de et al. Grandes desafios da pesquisa em computação no brasil 2006 2016. *SEMINÁRIO GRANDES DESAFIOS DE PESQUISA EM COMPUTAÇÃO NO BRASIL*, v. 2016, 2006.

CFM: Conselho Federal de Medicina. *CFM. RESOLUÇÃO CFM No 1.821/07*. [S.l.], 2007.

CLUCENE. <http://clucene.sourceforge.net/>. Setembro 2013. On Line. <<http://clucene.sourceforge.net/>>.

COHEN, S. et al. Scientific formats for object-relational database systems: a study of suitability and performance. *ACM SIGMOD Record*, ACM, v. 35, n. 2, p. 10–15, 2006.

COLLEN, M. His concepts, goals and objectives. *Towards new hospital information systems*. North Holland: Elsevier Science Publishers, 1988.

CORBETT, P. et al. Overview of the mpi-io parallel i/o interface. In: *Input/Output in Parallel and Distributed Computer Systems*. [S.l.]: Springer, 1996. p. 127–146.

COSTA, C. et al. Indexing and retrieving dicom data in disperse and unstructured archives. *International journal of computer assisted radiology and surgery*, Springer, v. 4, n. 1, p. 71–77, 2009.

COULOURIS, G. F. *Distributed Systems: Concepts and Design, 4/e*. [S.l.]: Pearson Education India, 2009.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, ACM, v. 51, n. 1, p. 107–113, 2008.

DOI, K. Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology. *Physics in Medicine and Biology*, IOP Publishing, v. 51, n. 13, p. R5, 2006.

DOLIN, R. et al. The hl7 clinical document architecture. *Journal of the American Medical Informatics Association*, BMJ Publishing Group Ltd, v. 8, n. 6, p. 552–569, 2001.

DOLIN, R. et al. HL7 clinical document architecture, release 2. *Journal of the American Medical Informatics Association*, BMJ Publishing Group Ltd, v. 13, n. 1, p. 30–39, 2006.

DOUGHERTY, M. Envisioning a standard image storage framework. In: *Proceedings of the 5th High-End Visualization Workshop*. [S.l.: s.n.], 2009.

DOUGHERTY, M. T. et al. Unifying biological image formats with HDF5. *Communications of the ACM*, v. 52, n. 10, p. 42–47, out. 2009. ISSN 0001-0782 (print), 1557-7317 (electronic).

DUENNEBEIL, S. et al. Strategies for development and adoption of ehr in german ambulatory care. In: IEEE. *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*. [S.l.], 2010. p. 1–8.

EGC. *Departamento de Engenharia e Gestão do Conhecimento*. Fevereiro 2013. <<http://www.egc.ufsc.br>>.

ELMASRI, R.; NAVATHE, S. *Fundamentals of database systems*. [S.l.]: Pearson, 2009.

FOLINO, G.; SHAH, A. A.; KRANSNOGOR, N. On the storage, management and analysis of (multi) similarity for large scale protein structure datasets in the grid. In: IEEE. *Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium on*. [S.l.], 2009. p. 1–8.

FOLK, M. et al. An overview of the hdf5 technology suite and its applications. In: ACM. *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*. [S.l.], 2011. p. 36–47.

FOLK, M.; POURMAL, E. Balancing performance and preservation lessons learned with hdf5. In: ACM. *Proceedings of the 2010 Roadmap for Digital Preservation Interoperability Framework Workshop*. [S.l.], 2010. p. 11.

FRAUNHOFERFS. *FhGFS - The Fraunhofer Parallel Cluster File System*. February 2013. On Line. <<http://www.fhgfs.com>>.

GABRIEL, E. et al. Open mpi: Goals, concept, and design of a next generation mpi implementation. In: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. [S.l.]: Springer, 2004. p. 97–104.

GEORGIEVA, J.; GANCHEVA, V.; GORANOVA, M. Scientific data formats. In: *AIC*. [S.l.: s.n.], 2009. v. 9, p. 19–24.

GOSINK, L. et al. Hdf5-fastquery: Accelerating complex queries on hdf datasets using fast bitmap indices. In: IEEE. *Scientific and Statistical Database Management, 2006. 18th International Conference on*. [S.l.], 2006. p. 149–158.

GRAY, J. et al. Scientific data management in the coming decade. *ACM SIGMOD Record*, ACM, v. 34, n. 4, p. 34–41, 2005.

GROPP, W. et al. A high-performance, portable implementation of the {MPI} message passing interface standard. *Parallel Computing*, v. 22, n. 6, p. 789 – 828, 1996. ISSN 0167-8191. <<http://www.sciencedirect.com/science/article/pii/0167819196000245>>.

GROUP, C. *Cyclops Group*. February 2013. <<http://www.cyclops.ufsc.br>>.

HDFGROUP. *HDF5 Introduction*. Setember 2013. On Line. <HDF Group>.

HEALTH, T. G. C. in G. *The Grand Challenges in Global Health*. Jan 2013. On Line. <<http://http://www.grandchallenges.org/>>.

HOWARD, J. H. et al. Scale and performance in a distributed file system. *ACM Trans. Comput. Syst.*, ACM, New York, NY, USA, v. 6, n. 1, p. 51–81, fev. 1988. ISSN 0734-2071. <<http://doi.acm.org/10.1145/35037.35059>>.

HSIEH, R. et al. Telemedicine in china. *International journal of medical informatics*, Elsevier, v. 61, n. 2, p. 139–146, 2001.

HU, J.; LEE, K. P. *DICOM XML DTD/schema generator*. [S.l.]: Google Patents, abr. 20 2004. US Patent 6,725,231.

HUANG, H. *PACS and imaging informatics: basic principles and applications*. [S.l.]: Wiley-Blackwell, 2011.

HUSSEIN, R. et al. Dicom structured reporting part 1. overview and characteristics1. *Radiographics*, Radiological Society of North America, v. 24, n. 3, p. 891–896, 2004.

JACKSON, M.; SIMMONS, J.; GRAEF, M. D. Mxa: a customizable hdf5-based data format for multi-dimensional data sets. *Modelling and Simulation in Materials Science and Engineering*, IOP Publishing, v. 18, n. 6, p. 065008, 2010.

JEPSEN, T. It in healthcare: progress report. *IT professional*, IEEE, v. 5, n. 1, p. 8–14, 2003.

JUTRAS, A. Teleröntgen diagnosis by means of video-tape recording. *The American journal of roentgenology, radium therapy, and nuclear medicine*, v. 82, p. 1099, 1959.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, p. 2004, 2004.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, Elsevier, v. 51, n. 1, p. 7–15, 2009.

KUNKEL, J.; LUDWIG, T. Performance evaluation of the pvfs2 architecture. In: IEEE. *Parallel, Distributed and Network-Based Processing, 2007. PDP'07. 15th EUROMICRO International Conference on*. [S.l.], 2007. p. 509–516.

LEE, K. P.; HU, J. Xml schema representation of dicom structured reporting. *Journal of the American Medical Informatics Association*, BMJ Publishing Group Ltd, v. 10, n. 2, p. 213–223, 2003.

LEE, K.-P.; SPENCE, P. L. view_hdf: visualization and analysis tool for hierarchical data format files. In: IEEE. *OCEANS'02 MTS/IEEE*. [S.l.], 2002. v. 2, p. 744–750.

LEIF, R. C.; LEIF, S. H. A cytometryml table of contents that describes relationships between elements based upon dicom and flow cytometry standard. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *SPIE BiOS*. [S.l.], 2011. p. 790217–790217.

LEVY, E.; SILBERSCHATZ, A. Distributed file systems: concepts and examples. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 22, n. 4, p. 321–374, dez. 1990. ISSN 0360-0300. <<http://doi.acm.org/10.1145/98163.98169>>.

LIGON W.B., I.; ROSS, R. Implementation and performance of a parallel file system for high performance distributed applications. In: *High Performance Distributed Computing, 1996., Proceedings of 5th IEEE International Symposium on*. [S.l.: s.n.], 1996. p. 471–480. ISSN 1082-8907.

LODWICK, G. Pictorial information systems and radiology improving the quality of communications. *NATO ASI series. Series F: computer and system sciences*, Plenum, v. 19, p. 1–27, 1986.

MACEDO, D. D. J. de et al. An improvement of a different approach for medical image storage. In: REDDY, S.; TATA, S. (Ed.). *WETICE*. IEEE Computer Society, 2011. p. 140–142. <<http://dx.doi.org/10.1109/WETICE.2011.26>>.

MACEDO, D. D. J. de et al. Armazenamento distribuído de imagens médicas dicom no formato de dados hdf5. In: *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web*. New York, NY, USA: ACM, 2008. (WebMedia '08), p. 20–27. ISBN 978-1-60558-170-5.

MACEDO, D. D. J. de et al. Asynchronous data replication: A national integration strategy for databases on telemedicine network. In: *CBMS*. IEEE Computer Society, 2008. p. 638–643. ISBN 978-0-7695-3165-6. <<http://doi.ieeecomputersociety.org/10.1109/CBMS.2008.76>>.

MACEDO, D. D. J. de et al. An interoperability approach based on asynchronous replication among distributed internet databases. In: *ISCC*. IEEE, 2008. p. 658–663. <<http://dx.doi.org/10.1109/ISCC.2008.4625712>>.

MACEDO, D. D. J. de et al. Armazenamento de imagens médicas dicom em sistemas de arquivos distribuídos. In: *IX Simpósio Brasileiro de Sistemas Computacionais*. Campo Grande (MS): [s.n.], 2008. (WSCAD08).

MACEDO, D. D. J. de et al. An architecture for DICOM medical images storage and retrieval adopting distributed file systems. *IJHPSA*, v. 2, n. 2, p. 99–106, 2009. <<http://dx.doi.org/10.1504/IJHPSA.2009.032027>>.

MAGNUS, M. et al. A study of netcdf as an approach for high performance medical image storage. v. 341, n. 1, p. 012016, 2012.

MAIA, R.; WANGENHEIM, A. von; NOBRE, L. A statewide telemedicine network for public health in brazil. In: *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*. [S.l.: s.n.], 2006. p. 495–500.

MALIK, A. Telemedicine country report-pakistan. In: *IEEE. e-Health Networking, Application and Services, 2007 9th International Conference on*. [S.l.], 2007. p. 90–94.

MAY, C. et al. Telepsychiatry evaluation in the north-west of england: preliminary results of a qualitative study. *Journal of Telemedicine and Telecare*, Royal Society of Medicine Press Ltd, v. 6, n. suppl 1, p. 20–22, 2000.

MEDEIROS, C. Grand research challenges in computer science in brazil. *Computer*, v. 41, n. 6, p. 59–65, june 2008. ISSN 0018-9162.

MILDENBERGER, P.; EICHELBERG, M.; MARTIN, E. Introduction to the dicom standard. *European radiology*, Berlin: Springer International, c1991-, v. 12, n. 4, p. 920–927, 2002.

MOORE, M. The evolution of telemedicine. *Future Generation Computer Systems*, Elsevier, v. 15, n. 2, p. 245–254, 1999.

NAM, B.; SUSSMAN, A. Improving access to multi-dimensional self-describing scientific datasets. In: *IEEE. Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on*. [S.l.], 2003. p. 172–179.

NEMA. *Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definition*. [S.l.], 2003.

NEMA. *Digital Imaging and Communications in Medicine (DICOM) Part 4: Service Class Specifications*. [S.l.], 2004.

NEMA. *Digital Imaging and Communications in Medicine (DICOM) Part 6: Data Dictionary*. [S.l.], 2004.

NEMA. *Digital Imaging and Communications in Medicine (DICOM). Part 10: Media Storage and File Format for Media Interchange*. Rosslyn, Virginia, 2011.

PAL, A. et al. Telemedicine diffusion in a developing country: the case of india (march 2004). *Information Technology in Biomedicine, IEEE Transactions on*, IEEE, v. 9, n. 1, p. 59–65, 2005.

PRADO, T. C. *Otimização da Persistência de Dados em PACS Empregando Modelos de Dados Hierárquicos Indexados*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2012.

PRADO, T. C. et al. Optimization of pacs data persistency using indexed hierarchical data. *Journal of Digital Imaging*, Springer US, p. 1–12, 2014.

RADIOLOGISTS, T. R. C. of. *Retention and Storage of Images and Radiological Patient Data*. [S.l.], February 2008.

REPKO, A. *Interdisciplinary research: Process and theory*. [S.l.]: Sage Publications, Incorporated, 2008. 416 p.

RON, E. Cancer risks from medical radiation. *Health Physics*, LWW, v. 85, n. 1, p. 47–59, 2003.

SATYANARAYANAN, M. et al. Coda: A highly available file system for a distributed workstation environment. *Computers, IEEE Transactions on*, IEEE, v. 39, n. 4, p. 447–459, 1990.

SHASHARINA, S. G. et al. Distributed technologies for remote access of hdf data. In: IEEE. *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. WETICE 2007. 16th IEEE International Workshops on*. [S.l.], 2007. p. 255–260.

SHVACHKO, K. et al. The hadoop distributed file system. In: IEEE. *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*. [S.l.], 2010. p. 1–10.

SICKLES, E.; WOLVERTON, D.; DEE, K. Performance parameters for screening and diagnostic mammography: Specialist and general radiologists1. *Radiology*, Radiological Society of North America, v. 224, n. 3, p. 861–869, 2002.

SNIR, M. et al. *MPI: The Complete Reference*. Cambridge, MA, USA: MIT Press, 1995. ISBN 0262691841.

- SOARES, T. et al. An approach using parallel architecture to storage dicom images in distributed file system. v. 341, n. 1, p. 012021, 2012.
- SOARES, T. S. et al. A data management in a private cloud storage environment utilizing high performance distributed file systems. In: *IEEE. Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on.* [S.l.], 2013. p. 158–163.
- SOARES, T. S. et al. A parallel architecture using hdf for storing dicom medical images on distributed file systems. In: *WorldComp.* [S.l.: s.n.], 2011.
- SOARES, T. S. et al. Ph5wrap: A parallel approach to storage server of dicom images. *WorldComp*, 2012.
- TANENBAUM, A. S.; STEEN, M. V. *Distributed systems.* [S.l.]: Prentice Hall, 2002.
- TANTISIRIROJ, W. et al. On the duality of data-intensive file system design: reconciling hdfs and pvfs. In: *ACM. Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis.* [S.l.], 2011. p. 67.
- TENG, C.-C. et al. A medical image archive solution in the cloud. In: *IEEE. Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on.* [S.l.], 2010. p. 431–434.
- THAKUR, R.; GROPP, W.; LUSK, E. An abstract-device interface for implementing portable parallel-i/o interfaces. In: *IEEE. Frontiers of Massively Parallel Computing, 1996. Proceedings Frontiers' 96., Sixth Symposium on the.* [S.l.], 1996. p. 180–187.
- THAKUR, R.; GROPP, W.; LUSK, E. On implementing mpi-io portably and with high performance. In: *Proceedings of the sixth workshop on I/O in parallel and distributed systems.* New York, NY, USA: ACM, 1999. (IOPADS '99), p. 23–32. ISBN 1-58113-123-2. <<http://doi.acm.org/10.1145/301816.301826>>.
- THIELSCHER, C.; DOARN, C. Long-term future of telemedicine in germany: the patients, physicians, and payers perspective. *Telemedicine and e-Health*, Mary Ann Liebert, Inc. 140 Huguenot Street 3rd Floor New Rochelle, NY 10801 USA, v. 14, n. 7, p. 701–706, 2008.
- TIAN, F. et al. The design and performance evaluation of alternative xml storage strategies. *ACM Sigmod Record*, ACM, v. 31, n. 1, p. 5–10, 2002.

- WALLAUER, J. et al. Building a national telemedicine network. *IT Professional*, v. 10, n. 2, p. 12–17, march-april 2008. ISSN 1520-9202.
- WANG, Z.; GU, H. A review of telemedicine in china. *Journal of Telemedicine and Telecare*, Royal Society of Medicine Press Ltd, v. 15, n. 1, p. 23–27, 2009.
- WANGENHEIM, A. von et al. Ways to implement large scale telemedicine: The santa catarina experience. *Latin-American Journal of Telehealth*, v. 3, p. 364–376, 2009.
- WEIL, S. et al. Ceph: A scalable, high-performance distributed file system. In: *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*. [S.l.: s.n.], 2006. p. 307–320.
- WEIL, S. et al. Crush: Controlled, scalable, decentralized placement of replicated data. In: *ACM. Proceedings of the 2006 ACM/IEEE conference on Supercomputing*. [S.l.], 2006. p. 122.
- WETERING, R. van de; BATENBURG, R. A pacs maturity model: a systematic meta-analytic review on maturation and evolvability of pacs in the hospital enterprise. *International journal of medical informatics*, Elsevier, v. 78, n. 2, p. 127–140, 2009.
- WHANCLOUD. *The Lustre Manual v2*. February 2013. On Line. <<http://www.whancloud.com>>.
- WILSON, E. Asynchronous health care communication. *Communications of the ACM*, ACM, v. 46, n. 6, p. 79–84, 2003.
- WITTSON, C.; AFFLECK, D.; JOHNSON, V. Two-way television in group therapy. *Mental Hospitals; Mental Hospitals*, 1961.
- XUE, Y.; LIANG, H. Analysis of telemedicine diffusion: the case of china. *Information Technology in Biomedicine, IEEE Transactions on*, IEEE, v. 11, n. 2, p. 231–233, 2007.
- YOU, X.; YAO, Z. Application of xml in dicom. In: *Proc. of SPIE Vol.* [S.l.: s.n.], 2005. v. 5748, p. 447.
- YU, H. et al. High performance file i/o for the blue gene/l supercomputer. In: *IEEE. High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*. [S.l.], 2006. p. 187–196.

APÊNDICE A – Motivação Pessoal

A.1 MOTIVAÇÃO PESSOAL

Após mais de 15 anos trabalhando com infraestrutura e engenharia de sistemas críticos, na parte de desenvolvimento e operação, tive a oportunidade de passar por grandes parques computacionais, como o do Correios, da Universidade Federal de Rondônia, da Receita Federal, da Ambev, da IBM e da Universidade Federal de Santa Catarina. Foi nesta última, em 2006, que tive a oportunidade de me engajar no projeto da Rede Catarinense de Telemedicina (RCTM) (WALLAUER et al., 2008) (MAIA; WANGENHEIM; NOBRE, 2006).

A RCTM (WALLAUER et al., 2008) (MAIA; WANGENHEIM; NOBRE, 2006) foi desenvolvida pelo Laboratório de Telemedicina, da Universidade Federal de Santa Catarina, em parceria com a Secretaria de Estado da Saúde (SES) do Estado de Santa Catarina. A motivação para a criação desta rede foi a necessidade do Estado em diminuir custos de transporte de pacientes para os centros especializados mais próximos, ainda atingindo melhores níveis de qualidade e de tempo de resposta.

Como plataforma tecnológica, a RCTM funciona muito bem. Ela basicamente é composta por uma arquitetura cliente–servidor *web*, desenvolvida na linguagem de programação PHP (PHP Hypertext Preprocessor) e os dados são armazenados em um sistema gerenciador de bancos de dados relacionais PostgreSQL. Desta forma, trata-se de uma arquitetura computacional bem tradicional, entretanto de grande valia para a população em geral.

Entretanto, no decorrer da operação, nos deparamos com um grave problema de escalabilidade desta rede, pois a medida que ela crescia, o desempenho da mesma ia caindo para níveis alarmantes. Após vários experimentos, testes e protótipos, em todos os componentes da arquitetura da RCTM, foi constatado que o principal gargalo estava no banco de dados. Foram gastos extensos recursos a fim de melhorar as configurações de memória e processador das máquinas que hospedavam o banco de dados e ainda, muito tempo revendo o *design* e o *tunning* do banco de dados como um todo, para constatarmos que o problema era bem mais profundo, e que isto necessitaria de uma pesquisa mais detalhada.

Foi então, que a partir de 2006, iniciei pesquisas de aplicação de estratégias de Sistemas Distribuídos aplicados em Sistemas de Telemedicina, visando persistência de dados a longo prazo, com requisitos de transparência e segurança, para o armazenamento de dados, informações e conhecimento, que estão embutidos nestes tipos de cenários. Isto culminou em uma série de estudos, como aplicação de replicação de dados distribuídos (MACEDO et al., 2008b), (MACEDO et al., 2008c). Ainda, após vários experimentos, vimos que para o armazenamento somente de textos, a estratégia de replicação

estava indo muito bem, entretanto, quando colocávamos volume no sistema, com a inserção das imagens dos exames, o desempenho era prejudicado. Desta forma, surgiu a pergunta:

Porque usar o paradigma relacional?

Desta forma, no início de 2007, culminando com o final do mestrado, nós iniciamos pesquisas utilizando outros formatos de dados científicos para persistência dos dados (MACEDO et al., 2008a), (MACEDO et al., 2008d). Nossa ideia era criar cenários onde abandonássemos completamente o paradigma relacional, ou ainda, criássemos infraestruturas híbridas para persistência dos dados. Graças a muita pesquisa e trabalho da equipe, tivemos bastante sucesso nos primeiros protótipos e então quando entrei no doutorado, foquei todos os esforços em criar um modelo de armazenamento de conhecimento baseado em outros paradigmas, que não o relacional.

APÊNDICE B – Publicações Relacionadas

B.1 INTRODUÇÃO

Neste capítulo estão as publicações relacionadas ao tema desta tese, que foram desenvolvidas durante a fase do doutorado, (2009 – 2013) até o presente momento. Ainda, foram publicados mais 5 trabalhos acadêmicos, sendo 1 capítulo de livro (ERRC'09), 1 livro (CNI'12), 1 Workshop (WGRS/SBRC'12), 1 artigo regional (ERRC'11) e 1 artigo em revista (e-Tech'13). Entretanto, estes últimos 5 não estão totalmente alinhados com o tema desta tese e não serão citados abaixo.

A título de observação, os pesquisadores MSc. Tiago Steimetz Soares (SOARES, T.S.), Marcone Magnus (MAGNUS, M.), Hilton Perantunes (PERANTUNES, H. W. G.), MSc. Thiago Coelho Prado (PRADO, T.C.), Marcus Vinícius Florêncio e Souza e MSc. Elias Amaral (AMARAL, E.) foram alunos e pesquisadores que participaram da elaboração e no desenvolvimento de partes desta pesquisa, sob supervisão de Douglas Dyllon Jeronimo de Macedo (MACEDO, D. D. J.), utilizando esta pesquisa como base para os seus trabalhos de conclusão de curso, seja ele graduação ou mestrado.

B.2 JOURNALS INTERNACIONAIS

- 1.SOARES, T. S. ; MACEDO, D. D. J. ; DANTAS, M.A.R. ; BAUER, M. . A Data Management in a Private Cloud Storage. The Computer Journal. 2014 (ACEITO)
- 2.PRADO, T. C; MACEDO, D.D.J. ; WANGENHEIM, A. V. ; DANTAS, M.A.R. Optimization of PACS Data Persistency using Indexed Hierarchical Data. Journal of Digital Imaging. 2014.
- 3.MAGNUS, M. ; PRADO, T. C. ; WANGENHEIM, A. V. ; MACEDO, D. D. J. ; DANTAS, M.A.R. . A Study of NetCDF as an Approach for High Performance Medical Image Storage. Journal of Physics. Conference Series (Online), 2012.
- 4.SOARES, T. S. ; DANTAS, M.A.R. ; MACEDO, D. D. J. ; BAUER, M. . An Approach Using Parallel Architecture to Storage DICOM Images in Distributed File System. Journal of Physics. Conference Series (Online), 2012.
- 5.MACEDO, D. D. J. ; PERANTUNES, H. W. G. ; WANGENHEIM, A. V. ; DANTAS, M.A.R. . An architecture for DICOM medical images storage and retrieval adopting distributed file systems. International

Journal of High Performance Systems Architecture (Print), v. 2, p. 99-106, 2009.

B.3 CONFERÊNCIAS INTERNACIONAIS

1. SOARES, T. S. ; MACEDO, D. D. J. ; DANTAS, M.A.R. ; BAUER, M. . A Data Management in a Private Cloud Storage Environment Utilizing High Performance Distributed File Systems. In: The 22nd IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013, New Jersey. Proceedings of the 22nd IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013.
2. AMARAL, E. ; COMUNELLO, E. ; MACEDO, D. D. J. ; CAPRETZ, M. A. M. ; PRADO, T. C. ; DANTAS, M.A.R. . An Analysis of Medical Images Data Replication and Recovery Considering a Database Management System and a Distributed File System. In: The 18th IEEE Symposium in Computer and Communication (ISCC), 2013, Split - Croatia. Proceedings of the 18th IEEE Symposium in Computer and Communication (ISCC), 2013.
3. SOARES, T. S. ; PRADO, T. C.; MACEDO, D. D. J. ; DANTAS, M.A.R. ; BAUER, M. . PH5WRAP: A Parallel Approach To Storage Server of Dicom Images. In: The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'12), 2011, Las Vegas. The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLD-COMP'12), 2012.
4. SOARES, T. S. ; MACEDO, D. D. J. ; DANTAS, M.A.R. ; BAUER, M. . An Parallel Architecture Using HDF for Storage DICOM Medical Images on Distributed File Systems. In: The 2011 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'11), 2011, Las Vegas. The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLD-COMP'11), 2011.
5. MACEDO, D. D. J. ; CAPRETZ, M. A. M. ; COELHO, T. P. ; WANGENHEIM, A. V. ; DANTAS, M.A.R. . An Improvement of a Different Approach for Medical Image Storage. In: 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2011, Paris. 20th IEEE International

Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). Paris, 2011. p. 140-142.

6. MACEDO, D. D. J. ; COELHO, T. P. ; COMUNELLO, E. ; DANTAS, M.A.R. . A Study Comparison of Store and Retrieve of DICOM Images using High Performance Data Format. In: High Performance Computing Symposium (HPCS'09), 2009, Kingston. Lecture Notes in Computer Science, 2009.

7. AMARAL, E. ; PRADO, T. C. ; COMUNELLO, E. ; MACEDO, D. D. J. ; DANTAS, M.A.R. . Replicação Distribuída de Imagens Médicas sob o Formato de Dados HDF5. In: 8th International Information and Telecommunication Technologies Symposium, 2009, Florianópolis. Proceeding of the 8th International Information and Telecommunication Technologies Symposium (I2TS'09), 2009.

B.4 RESUMO DAS PUBLICAÇÕES

Nome	Ano	Local	Qualis
Journal of Digital Imaging	2014	-	A2
The Computer Journal	2014	-	B1
Journal of Physics. Conference Series (Online)	2012	-	B2
Journal of Physics. Conference Series (Online)	2012	-	B2
Int. Journal of High Perfor. Sys. and Arch (IJHPSA) (Print)	2009	-	B2
WETICE'13	2013	Hammamet, Tunisia	B1
ISCC'13	2013	Split, Croatia	A2
PDPTA'12	2012	Las Vegas, EUA	B2
PDPTA'11	2011	Las Vegas, EUA	B2
WETICE'11	2011	Paris, France	B1
HPCS'09	2009	Kingston, Canada	B3
I2TS'09	2009	Florianópolis, Brasil	B4

APÊNDICE C – Tabelas - Armazenamento Paralelo

Armazenamento Paralelo HDF - 1000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.7210977	0.3819444	0.4125000	0.3127136	0.1702966
2	0.4023855	0.3836923	0.4143876	0.3104588	0.1846559
3	0.8296440	0.3865901	0.4175173	0.3207299	0.1927189
4	0.4294642	0.3839663	0.4146836	0.3118601	0.1886186
5	0.9238806	0.3908379	0.4221049	0.3209300	0.1806373
6	0.7876930	0.3884361	0.4195110	0.3264247	0.1753096
7	0.6346807	0.3944913	0.4260506	0.3179014	0.1768480
8	0.6064060	0.3920127	0.4233737	0.3187168	0.1725469
9	0.3675151	0.3864944	0.4174139	0.3234172	0.1782507
10	0.3550379	0.3850482	0.4158520	0.3132448	0.1716480
11	0.6855588	0.3782672	0.4085285	0.3216681	0.1840612
12	0.7213194	0.3855553	0.4163997	0.3180208	0.1789713
13	0.7659211	0.3851378	0.4159488	0.3147664	0.1781507
14	0.9689758	0.3871150	0.4180842	0.3246543	0.1731295
15	0.3732340	0.3899823	0.4211809	0.3185675	0.1895569
16	0.6798691	0.3853080	0.4161327	0.3216586	0.1784963
17	0.3631585	0.3829933	0.4136328	0.3201038	0.1774913
18	0.7264775	0.3805552	0.4109996	0.3109698	0.1843436
19	0.4170280	0.3853483	0.4161762	0.3169526	0.1709795
20	0.7352071	0.3848631	0.4156522	0.3119801	0.1736184
21	0.6674986	0.3877312	0.4187497	0.3080535	0.1863405
22	0.8089139	0.3909490	0.4222249	0.3083197	0.1815829
23	0.4287943	0.3953390	0.4269661	0.3147141	0.1736265
24	0.7294223	0.3853080	0.4161327	0.3187059	0.1855348
25	0.3711517	0.3782672	0.4085285	0.3121467	0.1658424
MÉDIA	0.6200134	0.3862493	0.4171493	0.3167072	0.1789303
MENOR	0.3550379	0.3782672	0.4085285	0.3080535	0.1658424
MAIOR	0.9689758	0.3953390	0.4269661	0.3264247	0.1927189
DESVIO	0.1931047	0.0043644	0.0047136	0.0051296	0.0067928
MEDIANA	0.6798691	0.3853483	0.4161762	0.3179014	0.1782507

Armazenamento Paralelo HDF - 2500 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.4845394	0.3848983	0.4156902	0.3226113	0.1674264
2	1.0289216	0.3865125	0.4174335	0.3188585	0.1635327
3	1.0618811	0.3925120	0.4239129	0.3282180	0.1692225
4	0.4771845	0.3880390	0.4190821	0.3193661	0.1782712
5	1.0414090	0.3933659	0.4248351	0.3268439	0.1767195
6	1.0572853	0.3940152	0.4255364	0.3320913	0.1723219
7	0.4699264	0.3935127	0.4249937	0.3258363	0.1628120
8	0.4790612	0.3923831	0.4237737	0.3247284	0.1658856
9	0.9830997	0.3923227	0.4237085	0.3299564	0.1685090
10	1.1241562	0.3867116	0.4176485	0.3193014	0.1622743
11	1.0486856	0.3803031	0.4107274	0.3301690	0.1679482
12	1.1045482	0.3883927	0.4194641	0.3268322	0.1717804
13	0.4667519	0.3922220	0.4235998	0.3236099	0.1767310
14	0.4801567	0.3906633	0.4219163	0.3317479	0.1681659
15	1.0116253	0.3934256	0.4248996	0.3272425	0.1753953
16	0.9672768	0.3866305	0.4175610	0.3275117	0.1635395
17	1.0251989	0.3886383	0.4197293	0.3234801	0.1828766
18	1.0264843	0.3805583	0.4110029	0.3189318	0.1799275
19	0.4982435	0.3837973	0.4145011	0.3253434	0.1624819
20	0.9815681	0.3842219	0.4149597	0.3129386	0.1680922
21	1.4199045	0.3872602	0.4182410	0.3199310	0.1637880
22	0.9818797	0.3935670	0.4250523	0.3135393	0.1723384
23	0.4863911	0.3988818	0.4307923	0.3203160	0.1774935
24	1.0104145	0.3844678	0.4152253	0.3221922	0.1823444
25	0.4667578	0.3918964	0.4232481	0.3130057	0.1889303
MÉDIA	0.8473340	0.3891680	0.4203014	0.3233841	0.1715523
MENOR	0.4667519	0.3803031	0.4107274	0.3129386	0.1622743
MAIOR	1.4199045	0.3988818	0.4307923	0.3320913	0.1889303
DESVIO	0.2944990	0.0046277	0.0049979	0.0055381	0.0073616
MEDIANA	0.9830997	0.3886383	0.4197293	0.3236099	0.1692225

Armazenamento Paralelo HDF - 5000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.5519125	0.3876296	0.4186400	0.3248571	0.1711119
2	1.1268152	0.3892899	0.4204330	0.3215320	0.1645035
3	1.1434634	0.3948615	0.4264504	0.3295505	0.1581020
4	1.1920910	0.3905427	0.4217862	0.3233761	0.1708966
5	1.1011475	0.3945634	0.4261285	0.3275392	0.1661590
6	0.5517049	0.3951652	0.4267784	0.3331356	0.1682191
7	0.5764807	0.3940380	0.4255610	0.3289632	0.1621494
8	0.5687531	0.3907380	0.4219971	0.3252177	0.1752817
9	1.6965655	0.3924041	0.4237964	0.3329974	0.1591626
10	0.5806112	0.3874178	0.4184112	0.3216285	0.1699194
11	1.1680427	0.3830757	0.4137218	0.3321215	0.1664203
12	1.2070076	0.3869125	0.4178655	0.3280320	0.1667939
13	1.2428811	0.3906459	0.4218975	0.3261540	0.1576142
14	0.5714130	0.3901194	0.4213290	0.3340361	0.1674973
15	1.1902843	0.3929183	0.4243518	0.3338705	0.1611496
16	0.5652008	0.3852729	0.4160947	0.3308940	0.1668824
17	1.1538006	0.3890835	0.4202102	0.3290988	0.1627997
18	0.5692270	0.3789798	0.4092982	0.3278420	0.1698388
19	0.6848950	0.3814069	0.4119195	0.3301225	0.1638554
20	1.0762525	0.3801751	0.4105891	0.3188733	0.1705162
21	0.5690261	0.3850925	0.4158999	0.3249724	0.1698081
22	0.5735493	0.3920829	0.4234496	0.3148571	0.1689066
23	1.2043071	0.3994098	0.4313626	0.3216422	0.1621735
24	0.5628901	0.3840139	0.4147350	0.3229016	0.1668618
25	1.1304164	0.3925309	0.4239334	0.3189042	0.1684764
MÉDIA	0.9023495	0.3891348	0.4202656	0.3265248	0.1662040
MENOR	0.5517049	0.3789798	0.4092982	0.3148571	0.1576142
MAIOR	1.6965655	0.3994098	0.4313626	0.3340361	0.1752817
DESVIO	0.3381539	0.0051185	0.0055280	0.0051707	0.0044433
MEDIANA	1.0762525	0.3901194	0.4213290	0.3275392	0.1668618

Armazenamento Paralelo HDF - 1000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	1.0974845	0.3869131	0.4178661	0.3348954	0.1804359
2	1.0337037	0.3872578	0.4182384	0.3312211	0.1751092
3	0.8017069	0.3910930	0.4223804	0.3405502	0.1765920
4	1.0527933	0.3885791	0.4196654	0.3353982	0.1787404
5	0.6626098	0.3907460	0.4220057	0.3373418	0.1792912
6	1.1821996	0.3922015	0.4235776	0.3454409	0.1881963
7	0.9808240	0.3905733	0.4218192	0.3413544	0.1908806
8	0.7142990	0.3869059	0.4178584	0.3374693	0.1797979
9	1.2351501	0.3893776	0.4205278	0.3413560	0.1785016
10	1.0349941	0.3851109	0.4159198	0.3350139	0.1794217
11	1.1860913	0.3819971	0.4125569	0.3378458	0.1780597
12	0.6616197	0.3929889	0.4244280	0.3419979	0.1747693
13	0.8090124	0.3942393	0.4257784	0.3431969	0.1798123
14	0.6713848	0.3947940	0.4263775	0.3487687	0.1871414
15	1.0808163	0.3956380	0.4272890	0.3478770	0.1795312
16	0.9659851	0.3861799	0.4170743	0.3446290	0.1772397
17	0.9783403	0.3929591	0.4243958	0.3510487	0.1873130
18	0.6862514	0.3860220	0.4169038	0.3458190	0.1810181
19	0.6553802	0.3848245	0.4156105	0.3326778	0.1740306
20	1.0786262	0.3860648	0.4169500	0.3382018	0.1906786
21	1.0894575	0.3878832	0.4189139	0.3355818	0.1860995
22	0.9897515	0.3925917	0.4239990	0.3261347	0.1773581
23	0.8102467	0.4044189	0.4367724	0.3363152	0.1773011
24	1.0141008	0.3933247	0.4247907	0.3343203	0.1791003
25	1.0450808	0.3929559	0.4243924	0.3420103	0.1873723
MÉDIA	0.9407164	0.3902256	0.4214437	0.3394586	0.1809517
MENOR	0.6553802	0.3819971	0.4125569	0.3261347	0.1740306
MAIOR	1.2351501	0.4044189	0.4367724	0.3510487	0.1908806
DESVIO	0.1850569	0.0046755	0.0050495	0.0059250	0.0050173
MEDIANA	0.9897515	0.3905733	0.4218192	0.3382018	0.1794217

Armazenamento Paralelo HDF - 1000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	1387.99	525.08	567.08	317.52	170.30
2	579.18	528.33	570.59	312.43	184.66
3	1525.59	534.31	577.05	321.87	192.72
4	561.34	530.24	572.66	318.73	188.62
5	1246.01	531.62	574.15	322.35	180.64
6	1718.43	530.57	573.01	330.33	175.31
7	1263.75	532.03	574.60	323.18	176.85
8	1740.71	534.12	576.85	319.27	172.55
9	622.98	529.22	571.56	330.17	178.25
10	535.44	529.28	571.62	315.76	171.65
11	1696.65	526.10	568.19	326.18	184.06
12	1765.71	392.06	577.05	319.54	178.97
13	1482.16	391.79	425.06	320.93	178.15
14	1926.86	393.81	574.15	332.21	173.13
15	711.09	398.73	430.62	323.46	189.56
16	1921.61	393.57	425.06	329.59	178.50
17	570.38	390.84	422.11	324.99	177.49
18	2038.34	388.45	419.52	314.58	184.34
19	612.52	392.39	423.78	318.20	170.98
20	1432.68	394.58	426.15	317.02	173.62
21	1504.19	398.25	430.10	313.39	186.34
22	2364.15	400.00	432.00	309.90	181.58
23	556.23	404.03	436.35	320.79	173.63
24	2038.01	393.57	425.06	320.98	185.53
25	622.40	526.10	568.19	317.34	165.84
MÉDIA	1296.98	459.56	508.50	320.83	178.93
MENOR	535.44	388.45	419.52	309.90	165.84
MAIOR	2364.15	534.31	577.05	332.21	192.72
DESVIO	590.64	68.92	73.94	5.80	6.79
MEDIANA	1432.68	404.03	568.19	320.79	178.25

Armazenamento Paralelo HDF - 2500 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	1423.57	1329.16	1435.50	819.58	418.57
2	3046.01	1348.16	1456.01	803.17	408.83
3	3257.55	1350.98	1459.06	822.50	423.06
4	1656.52	1342.35	1449.74	813.04	445.68
5	4182.69	1354.40	1462.75	821.44	441.80
6	3024.78	1351.46	1459.58	840.88	430.80
7	1400.44	1351.05	1459.13	825.85	407.03
8	1557.49	1349.38	1457.33	812.97	414.71
9	3457.66	1350.68	1458.73	841.83	421.27
10	4278.12	1339.09	1446.22	804.47	405.69
11	3921.14	1333.03	1439.68	834.15	419.87
12	3193.76	991.60	1456.01	818.30	429.45
13	1400.58	997.88	1077.71	821.09	441.83
14	1502.05	993.73	1073.23	847.01	420.41
15	3232.30	1002.38	1456.01	830.81	438.49
16	3183.10	990.90	1070.17	838.88	408.85
17	3235.93	992.29	1071.67	824.18	457.19
18	3621.09	973.93	1051.84	803.95	449.82
19	1456.52	985.60	1064.45	816.91	406.20
20	2998.39	982.94	1061.58	799.40	420.23
21	4733.17	992.04	1071.40	809.60	409.47
22	3528.30	1008.77	1089.47	788.09	430.85
23	1641.76	1023.13	1104.98	814.64	443.73
24	3646.52	1164.79	1257.97	811.49	455.86
25	1274.94	1157.82	1250.44	799.71	472.33
MÉDIA	2794.18	1162.30	1285.63	818.56	428.88
MENOR	1274.94	973.93	1051.84	788.09	405.69
MAIOR	4733.17	1354.40	1462.75	847.01	472.33
DESVIO	1085.94	171.86	184.91	14.83	18.40
MEDIANA	3183.10	1157.82	1435.50	818.30	423.06

Armazenamento Paralelo HDF - 5000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	3135.75	2668.51	2881.99	1650.08	855.56
2	6234.24	2692.94	2908.37	1617.62	822.52
3	6603.49	2700.83	2916.89	1652.34	790.51
4	7798.41	2685.56	2900.40	1641.47	854.48
5	5970.96	2708.36	2925.03	1648.47	830.80
6	3137.40	2701.68	2917.82	1688.65	841.10
7	3354.61	2700.56	2916.60	1668.61	810.75
8	3271.46	2684.91	2899.70	1630.24	876.41
9	8526.53	2694.32	2909.87	1700.14	795.81
10	3506.10	2685.37	2900.20	1617.40	849.60
11	6140.03	2664.29	2322.86	1680.04	832.10
12	6603.79	1977.40	2322.86	1644.50	833.97
13	7098.65	1990.57	2881.99	1662.33	788.07
14	3454.72	1986.20	2145.10	1707.35	837.49
15	6755.73	2002.01	2162.17	1689.74	805.75
16	3219.48	1985.17	2143.99	1695.33	834.41
17	6097.06	1992.76	2152.18	1680.61	814.00
18	3360.35	1938.86	2093.97	1658.78	849.19
19	3757.51	1958.56	2115.24	1656.66	819.28
20	6215.85	1948.15	2104.01	1643.36	852.58
21	3335.66	1970.49	2128.13	1640.34	849.04
22	3436.93	2013.13	2174.18	1585.03	844.53
23	6544.81	2051.87	2216.02	1633.78	810.87
24	3476.93	2150.80	2322.86	1626.90	834.31
25	5986.76	2169.16	2342.70	1643.67	842.38
MÉDIA	5080.93	2308.90	2508.21	1654.54	831.02
MENOR	3135.75	1938.86	2093.97	1585.03	788.07
MAIOR	8526.53	2708.36	2925.03	1707.35	876.41
DESVIO	1764.66	348.46	365.67	29.36	22.22
MEDIANA	5970.96	2150.80	2322.86	1650.08	834.31

Armazenamento Paralelo HDF - 10000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	11720.49	5299.22	5723.15	3401.62	1804.36
2	11535.08	5331.05	5757.53	3331.95	1751.09
3	9866.86	5348.05	5775.90	3414.34	1765.92
4	11584.69	5315.64	5740.89	3395.28	1787.40
5	8181.26	5350.61	5778.65	3393.64	1792.91
6	13490.68	5354.64	5783.01	3485.86	1881.96
7	14751.17	5343.53	5771.01	3456.09	1908.81
8	9267.31	5305.13	5729.54	3380.92	1797.98
9	12626.92	5322.92	5748.76	3465.30	1785.02
10	11390.55	5307.68	5732.29	3363.34	1794.22
11	12679.53	5277.77	5699.99	3419.64	1780.60
12	8550.22	4008.11	5049.06	3418.53	1747.69
13	10488.76	4009.77	5778.65	3471.91	1798.12
14	8495.07	4014.87	5057.58	3527.78	1871.41
15	12526.38	4021.73	5066.23	3497.12	1795.31
16	11392.07	3962.73	4991.90	3492.26	1772.40
17	11384.32	4013.18	5055.45	3508.15	1873.13
18	8725.39	3943.74	4967.98	3456.35	1810.18
19	8095.36	3943.46	4967.62	3333.13	1740.31
20	12002.63	3953.22	4979.91	3400.40	1906.79
21	11774.45	3970.85	5002.13	3376.40	1861.00
22	11184.27	4024.39	5069.57	3290.76	1773.58
23	10338.24	4063.16	5118.41	3390.33	1773.01
24	12374.03	4227.71	5325.70	3357.45	1791.00
25	11448.36	4671.68	5045.42	3433.01	1873.72
MÉDIA	11034.96	4615.39	5388.65	3418.46	1809.52
MENOR	8095.36	3943.46	4967.62	3290.76	1740.31
MAIOR	14751.17	5354.64	5783.01	3527.78	1908.81
DESVIO	1734.32	655.74	362.48	60.51	50.17
MEDIANA	11392.07	4227.71	5325.70	3414.34	1794.22

APÊNDICE D – Tabelas - Armazenamento Serial

Armazenamento Serial HDF - 1000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.1157161	0.1217808	0.0164214	0.1153306	0.1702966
2	0.1158922	0.1231070	0.0164200	0.0841458	0.1846559
3	0.1153327	0.1233534	0.0158145	0.0814705	0.1927189
4	0.1154038	0.1232347	0.0161074	0.0839576	0.1886186
5	0.1153523	0.1219763	0.0160419	0.0904685	0.1806373
6	0.1152899	0.1219007	0.0154397	0.0838020	0.1753096
7	0.1153005	0.1236557	0.0176105	0.0809648	0.1768480
8	0.1152326	0.1231965	0.0152551	0.0834579	0.1725469
9	0.1154424	0.1226200	0.0160862	0.0845280	0.1782507
10	0.1154162	0.1227999	0.0160503	0.0834050	0.1716480
11	0.1152188	0.1216557	0.0152836	0.0833843	0.1840612
12	0.1155310	0.1194462	0.0159689	0.0829515	0.1789713
13	0.1154581	0.1190438	0.0161716	0.0830510	0.1781507
14	0.1156856	0.1169373	0.0163866	0.0836414	0.1731295
15	0.1158583	0.1165996	0.0161526	0.0836686	0.1895569
16	0.1161900	0.1145939	0.0158654	0.0834000	0.1784963
17	0.1160245	0.1172435	0.0159764	0.0827239	0.1774913
18	0.1159023	0.1154963	0.0179551	0.0832442	0.1843436
19	0.1175296	0.1157116	0.0169449	0.0829046	0.1709795
20	0.1158569	0.1148527	0.0152608	0.0831570	0.1736184
21	0.1155720	0.1176634	0.0158935	0.0844588	0.1863405
22	0.1157587	0.1155641	0.0157907	0.0915806	0.1815829
23	0.1156291	0.1183654	0.0151985	0.0841499	0.1736265
24	0.1156071	0.1157554	0.0153908	0.0825476	0.1855348
25	0.1156949	0.1174849	0.0161963	0.0837457	0.1658424
MÉDIA	0.1156758	0.1193615	0.0160673	0.0852056	0.1789303
MENOR	0.1152188	0.1145939	0.0151985	0.0809648	0.1658424
MAIOR	0.1175296	0.1236557	0.0179551	0.1153306	0.1927189
DESVIO	0.0004653	0.0032223	0.0006728	0.0066792	0.0067928
MEDIANA	0.1156071	0.1190438	0.0160419	0.0834579	0.1782507

Armazenamento Serial HDF - 2500 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.1149410	0.1186320	0.0175376	0.1153306	0.1674264
2	0.1150538	0.1187532	0.0168362	0.0841458	0.1635327
3	0.1147374	0.1188450	0.0164932	0.0814679	0.1692225
4	0.1149909	0.1189988	0.0172720	0.0839448	0.1782712
5	0.1149025	0.1188900	0.0166876	0.0904677	0.1767195
6	0.1150125	0.1179247	0.0165281	0.0838006	0.1723219
7	0.1148380	0.1187332	0.0185112	0.0809609	0.1628120
8	0.1148643	0.1187625	0.0166265	0.0834570	0.1658856
9	0.1149942	0.1182127	0.0167141	0.0845280	0.1685090
10	0.1150684	0.1182648	0.0172307	0.0834049	0.1622743
11	0.1148440	0.1174863	0.0164953	0.0833840	0.1679482
12	0.1149235	0.1170931	0.0169612	0.0829403	0.1717804
13	0.1152056	0.1174486	0.0168353	0.0830339	0.1767310
14	0.1153966	0.1141735	0.0169576	0.0836275	0.1681659
15	0.1156923	0.1142712	0.0169352	0.0836611	0.1753953
16	0.1154720	0.1125183	0.0168520	0.0833981	0.1635395
17	0.1154292	0.1145642	0.0168706	0.0827187	0.1828766
18	0.1155082	0.1132315	0.0190827	0.0832281	0.1799275
19	0.1169305	0.1131034	0.0172322	0.0828841	0.1624819
20	0.1154467	0.1116937	0.0164075	0.0835054	0.1680922
21	0.1151922	0.1141326	0.0167609	0.0824786	0.1637880
22	0.1155055	0.1123312	0.0170798	0.0841524	0.1723384
23	0.1152918	0.1149769	0.0164859	0.0829872	0.1774935
24	0.1153605	0.1144502	0.0165981	0.0834822	0.1823444
25	0.1153390	0.1139278	0.0175679	0.0809125	0.1889303
MÉDIA	0.1152376	0.1160568	0.0170224	0.0847161	0.1715523
MENOR	0.1147374	0.1116937	0.0164075	0.0809125	0.1622743
MAIOR	0.1169305	0.1189988	0.0190827	0.1153306	0.1889303
DESVIO	0.0004400	0.0025331	0.0006257	0.0066056	0.0073616
MEDIANA	0.1151922	0.1170931	0.0168520	0.0834049	0.1692225

Armazenamento Serial HDF - 5000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados	
	1	0.1141725	0.1151800	0.0182251	0.1151720	0.1711119
	2	0.1142663	0.1151485	0.0173135	0.0828874	0.1645035
	3	0.1140256	0.1154454	0.0170577	0.0799551	0.1581020
	4	0.1142341	0.1153980	0.0177079	0.0826898	0.1708966
	5	0.1137803	0.1153016	0.0172026	0.0891650	0.1661590
	6	0.1139094	0.1149113	0.0172571	0.0826581	0.1682191
	7	0.1139919	0.1152055	0.0191178	0.0796611	0.1621494
	8	0.1139780	0.1152865	0.0174514	0.0823798	0.1752817
	9	0.1140764	0.1152117	0.0172999	0.0832844	0.1591626
	10	0.1144038	0.1152613	0.0178044	0.0818145	0.1699194
	11	0.1140053	0.1121720	0.0171703	0.0818740	0.1664203
	12	0.1141504	0.1153797	0.0174009	0.0817070	0.1667939
	13	0.1145765	0.1151279	0.0172664	0.0817300	0.1576142
	14	0.1145025	0.1121720	0.0174481	0.0821513	0.1674973
	15	0.1146532	0.1126089	0.0175501	0.0820976	0.1611496
	16	0.1146456	0.1111255	0.0173836	0.0818121	0.1668824
	17	0.1146411	0.1124220	0.0174364	0.0814649	0.1627997
	18	0.1142975	0.1116790	0.0194353	0.0817372	0.1698388
	19	0.1161168	0.1114756	0.0175896	0.0817776	0.1638554
	20	0.1147475	0.1105510	0.0170895	0.0821944	0.1705162
	21	0.1143556	0.1119132	0.0175020	0.0796611	0.1698081
	22	0.1143553	0.1107193	0.0178300	0.0823798	0.1689066
	23	0.1142590	0.1125259	0.0171674	0.0832844	0.1621735
	24	0.1144136	0.1114981	0.0171828	0.0818145	0.1668618
	25	0.1145390	0.1120485	0.0180267	0.0818740	0.1684764
MÉDIA	0.1143639	0.1134307	0.0175967	0.0834891	0.1662040	
MENOR	0.1137803	0.1105510	0.0170577	0.0796611	0.1576142	
MAIOR	0.1161168	0.1154454	0.0194353	0.1151720	0.1752817	
DESVIO	0.0004492	0.0018361	0.0005838	0.0068236	0.0044433	
MEDIANA	0.1142975	0.1126089	0.0174364	0.0818740	0.1668618	

Armazenamento Serial HDF - 10.000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.1143515	0.1147017	0.0178880	0.1170859	0.1804359
2	0.1144280	0.1148148	0.0172750	0.0836580	0.1751092
3	0.1142809	0.1148081	0.0171520	0.0804513	0.1765920
4	0.1145189	0.1148535	0.0175160	0.0833869	0.1787404
5	0.1142263	0.1145988	0.0172940	0.0904980	0.1792912
6	0.1142755	0.1141620	0.0172210	0.0833775	0.1881963
7	0.1142223	0.1145122	0.0191010	0.0803082	0.1908806
8	0.1142437	0.1145672	0.0174630	0.0823992	0.1797979
9	0.1143329	0.1145699	0.0173210	0.0827352	0.1785016
10	0.1145701	0.1149039	0.0174960	0.0821722	0.1794217
11	0.1142192	0.1145122	0.0172040	0.0821917	0.1780597
12	0.1141520	0.1148729	0.0174260	0.0821579	0.1747693
13	0.1148109	0.1141057	0.0172510	0.0821660	0.1798123
14	0.1147549	0.1121340	0.0173280	0.0825098	0.1871414
15	0.1149821	0.1122630	0.0174600	0.0825694	0.1795312
16	0.1149228	0.1116710	0.0173540	0.0822442	0.1772397
17	0.1150479	0.1118536	0.0173840	0.0819142	0.1873130
18	0.1148908	0.1118867	0.0188720	0.0822358	0.1810181
19	0.1156688	0.1109040	0.0175320	0.0823697	0.1740306
20	0.1150901	0.1109359	0.0175550	0.0829415	0.1906786
21	0.1147530	0.1117120	0.0174740	0.0904980	0.1860995
22	0.1147649	0.1110399	0.0177160	0.0833775	0.1773581
23	0.1148276	0.1117752	0.0172360	0.0803082	0.1773011
24	0.1147811	0.1114407	0.0171500	0.0823992	0.1791003
25	0.1149319	0.1116648	0.0177280	0.0827352	0.1873723
MÉDIA	0.1146419	0.1131705	0.0175359	0.0843476	0.1809517
MENOR	0.1141520	0.1109040	0.0171500	0.0803082	0.1740306
MAIOR	0.1156688	0.1149039	0.0191010	0.1170859	0.1908806
DESVIO	0.0003719	0.0015733	0.0004746	0.0072385	0.0050173
MEDIANA	0.1147530	0.1141057	0.0174260	0.0823992	0.1794217

Armazenamento Serial HDF - 1000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	105.28	126.72	17.90	123.12	170.30
2	106.22	125.10	17.51	92.56	184.66
3	105.72	124.31	16.94	90.24	192.72
4	105.43	126.24	17.55	93.88	188.62
5	106.61	127.86	16.85	97.26	180.64
6	105.53	124.87	17.32	94.37	175.31
7	105.89	126.50	19.05	88.59	176.85
8	105.10	125.42	16.83	92.27	172.55
9	106.34	126.31	17.58	103.56	178.25
10	106.60	124.49	17.46	92.56	171.65
11	105.95	127.09	16.97	91.77	184.06
12	105.98	125.66	17.09	89.89	178.97
13	116.74	123.73	17.63	91.19	178.15
14	117.36	123.70	17.69	92.19	173.13
15	124.00	121.68	17.63	92.33	189.56
16	120.14	118.01	17.74	92.92	178.50
17	115.44	122.01	17.45	91.79	177.49
18	116.46	120.40	19.27	93.36	184.34
19	119.93	120.79	25.18	92.20	170.98
20	116.49	117.01	16.87	91.51	173.62
21	120.47	121.80	17.03	94.37	186.34
22	117.25	120.48	16.99	88.59	181.58
23	117.18	123.14	16.43	92.27	173.63
24	119.62	119.45	17.50	103.56	185.53
25	116.64	121.54	17.50	92.56	165.84
MÉDIA	112.34	123.37	17.76	94.36	178.93
MENOR	105.10	117.01	16.43	88.59	165.84
MAIOR	124.00	127.86	25.18	123.12	192.72
DESVIO	6.55	2.94	1.67	7.00	6.79
MEDIANA	115.44	123.73	17.50	92.33	178.25

Armazenamento Serial HDF - 2500 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	261.90	309.06	47.91	329.10	425.74
2	262.99	307.65	47.38	234.25	461.64
3	261.87	304.75	47.45	229.30	481.80
4	262.35	306.61	48.49	239.54	471.55
5	263.47	309.63	48.25	252.28	451.59
6	262.90	307.99	47.77	238.34	438.27
7	263.51	307.98	52.47	227.81	442.12
8	261.78	305.39	47.95	235.74	431.37
9	262.54	307.31	48.75	259.42	445.63
10	265.52	304.78	48.48	237.39	429.12
11	263.28	308.57	48.04	236.69	460.15
12	262.60	308.64	47.69	234.14	447.43
13	290.48	306.28	48.94	236.80	445.38
14	294.84	303.56	48.96	238.69	432.82
15	298.82	298.95	49.59	238.26	473.89
16	296.79	293.08	49.05	236.74	446.24
17	288.60	297.84	47.94	235.81	443.73
18	293.47	295.39	51.55	240.52	460.86
19	298.53	295.25	58.47	238.06	427.45
20	291.04	289.43	48.60	235.83	434.05
21	296.17	296.08	48.29	238.34	465.85
22	292.20	294.60	47.04	227.81	453.96
23	291.57	302.85	45.95	235.74	434.07
24	294.22	296.31	48.17	259.42	463.84
25	291.83	297.39	48.64	237.39	414.61
MÉDIA	278.93	302.22	48.87	242.14	447.33
MENOR	261.78	289.43	45.95	227.81	414.61
MAIOR	298.82	309.63	58.47	329.10	481.80
DESVIO	15.90	6.10	2.39	19.73	16.98
MEDIANA	288.60	304.75	48.29	237.39	445.63

Armazenamento Serial HDF - 5000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados	
	1	503.58	602.62	99.87	617.41	851.48
	2	500.90	599.09	101.09	452.60	923.28
	3	500.42	597.12	99.14	440.10	963.59
	4	497.53	601.35	100.12	467.66	943.09
	5	487.64	605.82	99.74	491.78	903.19
	6	487.77	606.60	100.93	465.17	876.55
	7	498.87	603.32	108.40	441.46	884.24
	8	488.10	598.83	100.56	455.95	862.73
	9	489.01	603.66	101.07	505.15	891.25
	10	508.26	598.45	99.77	462.40	858.24
	11	492.90	603.27	100.70	462.10	920.31
	12	501.40	607.22	101.12	456.88	894.86
	13	563.71	606.63	100.60	463.72	890.75
	14	559.79	602.02	101.10	464.88	865.65
	15	559.65	589.79	102.73	465.53	947.78
	16	560.66	584.30	100.53	459.78	892.48
	17	549.39	587.60	100.48	458.01	887.46
	18	541.83	585.31	105.32	467.25	921.72
	19	556.89	586.30	112.23	467.33	854.90
	20	553.09	580.90	100.72	464.47	868.09
	21	546.74	586.03	102.48	465.17	931.70
	22	538.35	586.72	97.62	441.46	907.91
	23	538.48	593.73	98.42	455.95	868.13
	24	547.77	587.06	99.17	505.15	927.67
	25	553.12	588.24	101.05	462.40	829.21
MÉDIA		525.03	595.68	101.40	470.39	894.65
MENOR		487.64	580.90	97.62	440.10	829.21
MAIOR		563.71	607.22	112.23	617.41	963.59
DESVIO		29.15	8.56	3.11	34.57	33.96
MEDIANA		538.35	598.45	100.70	463.72	891.25

Armazenamento Serial HDF - 10.000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	997.80	997.23	196.22	1257.34	1702.97
2	999.65	1195.10	198.40	918.16	1846.56
3	997.91	1192.12	194.53	884.99	1927.19
4	1001.20	1201.64	196.32	939.26	1886.19
5	984.35	1215.60	195.56	979.41	1806.37
6	977.11	1207.79	197.15	933.91	1753.10
7	997.39	1203.22	213.67	885.37	1768.48
8	981.75	1193.70	196.75	888.12	1725.47
9	982.03	1209.18	197.49	964.18	1782.51
10	981.90	1200.23	195.51	920.40	1716.48
11	980.41	1208.45	196.14	927.81	1840.61
12	983.82	1212.51	198.78	911.42	1789.71
13	1117.33	1211.05	196.90	927.66	1781.51
14	1106.79	1205.05	197.46	928.08	1731.29
15	1110.24	1178.16	199.45	923.15	1895.57
16	1118.61	1182.90	197.14	917.53	1784.96
17	1109.99	1175.53	192.52	913.94	1774.91
18	1099.29	1181.46	205.64	924.65	1843.44
19	1102.66	1169.32	209.76	930.47	1709.80
20	1111.73	1175.99	228.36	921.23	1736.18
21	1101.35	1174.51	198.86	918.16	1863.40
22	1091.48	1187.94	194.96	884.99	1815.83
23	1097.78	1179.39	193.96	939.26	1736.26
24	1102.89	1178.44	195.29	979.41	1855.35
25	1098.90	1150.80	196.11	933.91	1658.42
MÉDIA	1049.37	1183.49	199.32	938.11	1789.30
MENOR	977.11	997.23	192.52	884.99	1658.42
MAIOR	1118.61	1215.60	228.36	1257.34	1927.19
DESVIO	60.00	42.10	7.72	70.96	67.93
MEDIANA	1091.48	1192.12	196.90	924.65	1782.51

APÊNDICE E – Tabelas - Recuperação Paralela

Recuperação Paralela HDF - 1000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0270691	0.0274680	0.0278725	0.0248861	0.1432490
2	0.0161190	0.0182719	0.0171684	0.0153289	0.1429451
3	0.0169301	0.0168941	0.0177700	0.0158660	0.1421980
4	0.0160649	0.0255690	0.0175056	0.0156300	0.1440130
5	0.0154610	0.0159631	0.0167764	0.0149789	0.1427230
6	0.0161610	0.0164800	0.0168906	0.0150809	0.1431911
7	0.0162261	0.0259528	0.0170060	0.0151839	0.1423079
8	0.0155380	0.0165470	0.0181833	0.0162351	0.1427751
9	0.0154991	0.0163012	0.0173846	0.0155220	0.1446501
10	0.0155330	0.0261741	0.0170621	0.0152340	0.1424789
11	0.0157001	0.0151320	0.0167585	0.0149629	0.1428601
12	0.0155981	0.0154669	0.0173390	0.0154812	0.1443121
13	0.0176959	0.0263579	0.0172861	0.0154340	0.1419181
14	0.0153019	0.0156891	0.0182998	0.0163391	0.1440209
15	0.0156660	0.0153401	0.0166645	0.0148790	0.1423050
16	0.0155501	0.0164270	0.0170340	0.0152090	0.1437661
17	0.0160232	0.0150459	0.0171830	0.0153420	0.1423679
18	0.0155370	0.0256600	0.0172089	0.0153651	0.1437021
19	0.0156939	0.0157080	0.0176276	0.0157390	0.1420769
20	0.0159869	0.0166531	0.0167350	0.0149419	0.1424411
21	0.0159149	0.0259099	0.0170599	0.0152321	0.1443461
22	0.0154891	0.0169811	0.0179537	0.0160301	0.1422870
23	0.0154140	0.0152450	0.0170151	0.0151920	0.1504240
24	0.0162239	0.0259371	0.0174239	0.0155571	0.1436378
25	0.0159609	0.0164270	0.0164797	0.0147140	0.1435022
MÉDIA	0.0163343	0.0193440	0.0176675	0.0157746	0.1433799
MENOR	0.0153019	0.0150459	0.0164797	0.0147140	0.1419181
MAIOR	0.0270691	0.0274680	0.0278725	0.0248861	0.1504240
DESVIO	0.0022964	0.0048093	0.0021749	0.0019419	0.0016701
MEDIANA	0.0157001	0.0165470	0.0171830	0.0153420	0.1429451

Recuperação Paralela HDF - 2500 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0272501	0.0257690	0.0287323	0.0256538	0.1424521
2	0.0157659	0.0159180	0.0179334	0.0160120	0.1449229
3	0.0153971	0.0156150	0.0169197	0.0151069	0.1430349
4	0.0158122	0.0154049	0.0179435	0.0160210	0.1465811
5	0.0155079	0.0152550	0.0174575	0.0155871	0.1427682
6	0.0154960	0.0151272	0.0177085	0.0158112	0.1448130
7	0.0159199	0.0156760	0.0172658	0.0154159	0.1455571
8	0.0155659	0.0155439	0.0174327	0.0155649	0.1432819
9	0.0157728	0.0158029	0.0174194	0.0155530	0.1427659
10	0.0164690	0.0159252	0.0172661	0.0154161	0.1432251
11	0.0167670	0.0151401	0.0167531	0.0149581	0.1461750
12	0.0154471	0.0153842	0.0173590	0.0154991	0.1429320
13	0.0150781	0.0144081	0.0172672	0.0154171	0.1441230
14	0.0155239	0.0156910	0.0169096	0.0150979	0.1452542
15	0.0156550	0.0155380	0.0169029	0.0150919	0.1440410
16	0.0155039	0.0158591	0.0172009	0.0153580	0.1443280
17	0.0159280	0.0154200	0.0172178	0.0153730	0.1439490
18	0.0153770	0.0156372	0.0176925	0.0157969	0.1432571
19	0.0153809	0.0154750	0.0167494	0.0149548	0.1456630
20	0.0157359	0.0153549	0.0168538	0.0150480	0.1434820
21	0.0174251	0.0160761	0.0177230	0.0158241	0.1446181
22	0.0154290	0.0152969	0.0169221	0.0151091	0.1448231
23	0.0163889	0.0156701	0.0175985	0.0157130	0.1447721
24	0.0151160	0.0152340	0.0168987	0.0150881	0.1439690
25	0.0144460	0.0151460	0.0175414	0.0156620	0.1445180
MÉDIA	0.0161663	0.0158947	0.0177468	0.0158453	0.1442123
MENOR	0.0144460	0.0144081	0.0167494	0.0149548	0.1424521
MAIOR	0.0272501	0.0257690	0.0287323	0.0256538	0.1465811
DESVIO	0.0023820	0.0020857	0.0023167	0.0020685	0.0011188
MEDIANA	0.0155659	0.0155380	0.0172672	0.0154171	0.1441230

Recuperação Paralela HDF - 5000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0271580	0.0272498	0.0285443	0.0254860	0.143501954
2	0.0156260	0.0159199	0.0168391	0.0150349	0.142104006
3	0.0156870	0.0161619	0.0173790	0.0155170	0.142682154
4	0.0153811	0.0156460	0.0171606	0.0153220	0.143687954
5	0.0156162	0.0154591	0.0168762	0.0150681	0.142066006
6	0.0171928	0.0158851	0.0179088	0.0159900	0.142704154
7	0.0159640	0.0152168	0.0173790	0.0155170	0.142111929
8	0.0159400	0.0164149	0.0167953	0.0149958	0.143982992
9	0.0158460	0.0158820	0.0172917	0.0154390	0.144712133
10	0.0153961	0.0156231	0.0171876	0.0153461	0.142515884
11	0.0163939	0.0155160	0.0178079	0.0158999	0.143531056
12	0.0169969	0.0156269	0.0166722	0.0148859	0.142241822
13	0.0152581	0.0152159	0.0176097	0.0157230	0.146041141
14	0.0151560	0.0150130	0.0168012	0.0150011	0.142163129
15	0.0158291	0.0153911	0.0169603	0.0151432	0.142052907
16	0.0155330	0.0160611	0.0167294	0.0149369	0.142303177
17	0.0150530	0.0160909	0.0174049	0.0155401	0.141376932
18	0.0155010	0.0162601	0.0169579	0.0151410	0.142990086
19	0.0157800	0.0161970	0.0176276	0.0157390	0.142161931
20	0.0150890	0.0160859	0.0171326	0.0152969	0.141663047
21	0.0167360	0.0160952	0.0174327	0.0155649	0.142494056
22	0.0162599	0.0161622	0.0169801	0.0151608	0.142093909
23	0.0154638	0.0167291	0.0187310	0.0167241	0.143677929
24	0.0150099	0.0158579	0.0167507	0.0149560	0.143295228
25	0.0165050	0.0164030	0.0168840	0.0150750	0.143121857
MÉDIA	0.0162549	0.0163266	0.0176738	0.0157801	0.1428511
MENOR	0.0150099	0.0150130	0.0166722	0.0148859	0.1413769
MAIOR	0.0271580	0.0272498	0.0285443	0.0254860	0.1460411
DESVIO	0.0023477	0.0023140	0.0023125	0.0020648	0.0010390
MEDIANA	0.0156870	0.0159199	0.0171606	0.0153220	0.1425159

Recuperação Paralela HDF - 10.000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0262651	0.0278130	0.0290639	0.0259500	0.144083908
2	0.0155280	0.0265651	0.0166890	0.0149009	0.144860056
3	0.0156269	0.0157509	0.0172044	0.0153611	0.143558055
4	0.0272038	0.0157161	0.0169614	0.0151441	0.143283974
5	0.0174758	0.0162101	0.0172469	0.0153990	0.143297913
6	0.0159910	0.0155730	0.0178506	0.0159380	0.142922935
7	0.0171151	0.0156369	0.0171975	0.0153549	0.142960892
8	0.0158012	0.0155160	0.0171585	0.0153201	0.143354916
9	0.0164349	0.0150528	0.0166476	0.0148640	0.142542087
10	0.0155849	0.0155101	0.0179457	0.0160229	0.142666152
11	0.0161581	0.0151291	0.0173780	0.0155160	0.142930994
12	0.0153630	0.0157959	0.0172850	0.0154331	0.145478875
13	0.0156548	0.0155802	0.0169221	0.0151091	0.141961875
14	0.0155041	0.0160060	0.0169635	0.0151460	0.143404063
15	0.0156760	0.0156360	0.0171371	0.0153010	0.142912936
16	0.0167191	0.0154591	0.0186402	0.0166430	0.141921052
17	0.0169170	0.0163090	0.0167585	0.0149629	0.142342848
18	0.0158031	0.0150580	0.0173587	0.0154989	0.141935848
19	0.0150330	0.0154250	0.0171171	0.0152831	0.143043007
20	0.0160091	0.0152948	0.0174429	0.0155740	0.143128888
21	0.0162601	0.0149941	0.0181486	0.0162041	0.142003812
22	0.0158358	0.0156641	0.0179390	0.0160170	0.142742067
23	0.0152540	0.0152349	0.0168727	0.0150650	0.14205803
24	0.0160298	0.0155652	0.0172300	0.0153840	0.143310964
25	0.0163920	0.0150170	0.0173542	0.0154948	0.144357094
MÉDIA	0.0168654	0.0164605	0.0177805	0.0158755	0.1430825
MENOR	0.0150330	0.0149941	0.0166476	0.0148640	0.1419211
MAIOR	0.0272038	0.0278130	0.0290639	0.0259500	0.1454789
DESVIO	0.0030304	0.0032520	0.0023983	0.0021414	0.0008998
MEDIANA	0.0159910	0.0155730	0.0172300	0.0153840	0.1429609

Recuperação Paralela HDF - 1000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados	
	1	27.07	27.47	27.87	24.89	143.25
	2	16.12	18.27	17.17	15.33	142.95
	3	16.93	16.89	17.77	15.87	142.20
	4	16.06	25.57	17.51	15.63	144.01
	5	15.46	15.96	16.78	14.98	142.72
	6	16.16	16.48	16.89	15.08	143.19
	7	16.23	25.95	17.01	15.18	142.31
	8	15.54	16.55	18.18	16.24	142.78
	9	15.50	16.30	17.38	15.52	144.65
	10	15.53	26.17	17.06	15.23	142.48
	11	15.70	15.13	16.76	14.96	142.86
	12	15.60	15.47	17.34	15.48	144.31
	13	17.70	26.36	17.29	15.43	141.92
	14	15.30	15.69	18.30	16.34	144.02
	15	15.67	15.34	16.66	14.88	142.31
	16	15.55	16.43	17.03	15.21	143.77
	17	16.02	15.05	17.18	15.34	142.37
	18	15.54	25.66	17.21	15.37	143.70
	19	15.69	15.71	17.63	15.74	142.08
	20	15.99	16.65	16.73	14.94	142.44
	21	15.91	25.91	17.06	15.23	144.35
	22	15.49	16.98	17.95	16.03	142.29
	23	15.41	15.24	17.02	15.19	150.42
	24	16.22	25.94	17.42	15.56	143.64
	25	15.96	16.43	16.48	14.71	143.50
MÉDIA		16.33	19.34	17.67	15.77	143.38
MENOR		15.30	15.05	16.48	14.71	141.92
MAIOR		27.07	27.47	27.87	24.89	150.42
DESVIO		2.30	4.81	2.17	1.94	1.67
MEDIANA		15.70	16.55	17.18	15.34	142.95

Recuperação Paralela HDF - 2500 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	68.13	64.42	71.83	64.13	356.13
2	39.41	39.80	44.83	40.03	362.31
3	38.49	39.04	42.30	37.77	357.59
4	39.53	38.51	44.86	40.05	366.45
5	38.77	38.14	43.64	38.97	356.92
6	38.74	37.82	44.27	39.53	362.03
7	39.80	39.19	43.16	38.54	363.89
8	38.91	38.86	43.58	38.91	358.20
9	39.43	39.51	43.55	38.88	356.91
10	41.17	39.81	43.17	38.54	358.06
11	41.92	37.85	41.88	37.40	365.44
12	38.62	38.46	43.40	38.75	357.33
13	37.70	36.02	43.17	38.54	360.31
14	38.81	39.23	42.27	37.74	363.14
15	39.14	38.84	42.26	37.73	360.10
16	38.76	39.65	43.00	38.39	360.82
17	39.82	38.55	43.04	38.43	359.87
18	38.44	39.09	44.23	39.49	358.14
19	38.45	38.69	41.87	37.39	364.16
20	39.34	38.39	42.13	37.62	358.70
21	43.56	40.19	44.31	39.56	361.55
22	38.57	38.24	42.31	37.77	362.06
23	40.97	39.18	44.00	39.28	361.93
24	37.79	38.08	42.25	37.72	359.92
25	36.12	37.87	43.85	39.15	361.30
MÉDIA	40.42	39.74	44.37	39.61	360.53
MENOR	36.12	36.02	41.87	37.39	356.13
MAIOR	68.13	64.42	71.83	64.13	366.45
DESVIO	5.96	5.21	5.79	5.17	2.80
MEDIANA	38.91	38.84	43.17	38.54	360.31

Recuperação Paralela HDF - 5000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	135.79	136.25	142.72	127.43	717.51
2	78.13	79.60	84.20	75.17	710.52
3	78.43	80.81	86.90	77.58	713.41
4	76.91	78.23	85.80	76.61	718.44
5	78.08	77.30	84.38	75.34	710.33
6	85.96	79.43	89.54	79.95	713.52
7	79.82	76.08	86.90	77.58	710.56
8	79.70	82.07	83.98	74.98	719.91
9	79.23	79.41	86.46	77.20	723.56
10	76.98	78.12	85.94	76.73	712.58
11	81.97	77.58	89.04	79.50	717.66
12	84.98	78.13	83.36	74.43	711.21
13	76.29	76.08	88.05	78.61	730.21
14	75.78	75.06	84.01	75.01	710.82
15	79.15	76.96	84.80	75.72	710.26
16	77.66	80.31	83.65	74.68	711.52
17	75.27	80.45	87.02	77.70	706.88
18	77.51	81.30	84.79	75.71	714.95
19	78.90	80.98	88.14	78.69	710.81
20	75.45	80.43	85.66	76.48	708.32
21	83.68	80.48	87.16	77.82	712.47
22	81.30	80.81	84.90	75.80	710.47
23	77.32	83.65	93.66	83.62	718.39
24	75.05	79.29	83.75	74.78	716.48
25	82.53	82.01	84.42	75.37	715.61
MÉDIA	81.27	81.63	88.37	78.90	714.26
MENOR	75.05	75.06	83.36	74.43	706.88
MAIOR	135.79	136.25	142.72	127.43	730.21
DESVIO	11.74	11.57	11.56	10.32	5.19
MEDIANA	78.43	79.60	85.80	76.61	712.58

Recuperação Paralela HDF - 10.000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	262.65	278.13	290.64	259.50	1440.84
2	155.28	265.65	166.89	149.01	1448.60
3	156.27	157.51	172.04	153.61	1435.58
4	272.04	157.16	169.61	151.44	1432.84
5	174.76	162.10	172.47	153.99	1432.98
6	159.91	155.73	178.51	159.38	1429.23
7	171.15	156.37	171.97	153.55	1429.61
8	158.01	155.16	171.58	153.20	1433.55
9	164.35	150.53	166.48	148.64	1425.42
10	155.85	155.10	179.46	160.23	1426.66
11	161.58	151.29	173.78	155.16	1429.31
12	153.63	157.96	172.85	154.33	1454.79
13	156.55	155.80	169.22	151.09	1419.62
14	155.04	160.06	169.64	151.46	1434.04
15	156.76	156.36	171.37	153.01	1429.13
16	167.19	154.59	186.40	166.43	1419.21
17	169.17	163.09	167.58	149.63	1423.43
18	158.03	150.58	173.59	154.99	1419.36
19	150.33	154.25	171.17	152.83	1430.43
20	160.09	152.95	174.43	155.74	1431.29
21	162.60	149.94	181.49	162.04	1420.04
22	158.36	156.64	179.39	160.17	1427.42
23	152.54	152.35	168.73	150.65	1420.58
24	160.30	155.65	172.30	153.84	1433.11
25	163.92	150.17	173.54	154.95	1443.57
MÉDIA	168.65	164.60	177.81	158.75	1430.83
MENOR	150.33	149.94	166.48	148.64	1419.21
MAIOR	272.04	278.13	290.64	259.50	1454.79
DESVIO	30.30	32.52	23.98	21.41	9.00
MEDIANA	159.91	155.73	172.30	153.84	1429.61

APÊNDICE F – Tabelas - Recuperação Serial

Recuperação Serial HDF - 1000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0152600	0.0159669	0.0146730	0.0151589	0.143249026
2	0.0174530	0.0147102	0.0165820	0.0145359	0.142945069
3	0.0157621	0.0151951	0.0156929	0.0145180	0.142198008
4	0.0157030	0.0156698	0.0148160	0.0145979	0.144012999
5	0.0151451	0.0156910	0.0149980	0.0144060	0.142723007
6	0.0149510	0.0153499	0.0158329	0.0145490	0.143191055
7	0.0155671	0.0148940	0.0154431	0.0144780	0.142307911
8	0.0160530	0.0155289	0.0149460	0.0159731	0.142775079
9	0.0154462	0.0155771	0.0154319	0.0165241	0.14465014
10	0.0162871	0.0155380	0.0147550	0.0159361	0.142478886
11	0.0156820	0.0148301	0.0148301	0.0161278	0.142860135
12	0.0153348	0.0185201	0.0154071	0.0154521	0.144312149
13	0.0157950	0.0144482	0.0160859	0.0152991	0.141918072
14	0.0159221	0.0156119	0.0156720	0.0153341	0.144020926
15	0.0155420	0.0160260	0.0152810	0.0151310	0.142305005
16	0.0179551	0.0160971	0.0145199	0.0153100	0.143766105
17	0.0156641	0.0156970	0.0154250	0.0157168	0.142367865
18	0.0166602	0.0163720	0.0156610	0.0154071	0.143702113
19	0.0163100	0.0163312	0.0146530	0.0157850	0.142076925
20	0.0155370	0.0145760	0.0181959	0.0149901	0.142441138
21	0.0160539	0.0151720	0.0164359	0.0149791	0.144346101
22	0.0162361	0.0151072	0.0166709	0.0157590	0.142286995
23	0.0159869	0.0156012	0.0159161	0.0159209	0.150424023
24	0.0155470	0.0152729	0.0258980	0.0156059	0.143637825
25	0.0174119	0.0169320	0.0146451	0.0153520	0.143502192
MÉDIA	0.0159706	0.0156286	0.0159387	0.0153139	0.1433799
MENOR	0.0149510	0.0144482	0.0145199	0.0144060	0.1419181
MAIOR	0.0179551	0.0185201	0.0258980	0.0165241	0.1504240
DESVIO	0.0007341	0.0008422	0.0022352	0.0005827	0.0016701
MEDIANA	0.0157621	0.0155771	0.0154319	0.0153341	0.1429451

Recuperação Serial HDF - 2500 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0155411	0.0150919	0.0160520	0.0149970	0.1424521
2	0.0159180	0.0150762	0.0155849	0.0156848	0.1449229
3	0.0170271	0.0150762	0.0160599	0.0152571	0.1430349
4	0.0148730	0.0154750	0.0158160	0.0149610	0.1465811
5	0.0159919	0.0150130	0.0162508	0.0148101	0.1427682
6	0.0178249	0.0148289	0.0153980	0.0153320	0.1448130
7	0.0156631	0.0150890	0.0155780	0.0147891	0.1455571
8	0.0170112	0.0159039	0.0155890	0.0162189	0.1432819
9	0.0160060	0.0152988	0.0158300	0.0162179	0.1427659
10	0.0167470	0.0156810	0.0154381	0.0156569	0.1432251
11	0.0171981	0.0152311	0.0156691	0.0160480	0.1461750
12	0.0158930	0.0161788	0.0160289	0.0152781	0.1429320
13	0.0168669	0.0150728	0.0159280	0.0150690	0.1441230
14	0.0158341	0.0147960	0.0151379	0.0155342	0.1452542
15	0.0151429	0.0144911	0.0156631	0.0160761	0.1440410
16	0.0170949	0.0160081	0.0164139	0.0153968	0.1443280
17	0.0166628	0.0162930	0.0171812	0.0156391	0.1439490
18	0.0152550	0.0151441	0.0156212	0.0153592	0.1432571
19	0.0158701	0.0156801	0.0158269	0.0142050	0.1456630
20	0.0160711	0.0159700	0.0155611	0.0143070	0.1434820
21	0.0171790	0.0156250	0.0164111	0.0150552	0.1446181
22	0.0153971	0.0152249	0.0219541	0.0151031	0.1448231
23	0.0148361	0.0148480	0.0160139	0.0157840	0.1447721
24	0.0148690	0.0153589	0.0165801	0.0144629	0.1439690
25	0.0167999	0.0153172	0.0188282	0.0147300	0.1445180
MÉDIA	0.0161429	0.0153509	0.0162566	0.0152789	0.1442123
MENOR	0.0148361	0.0144911	0.0151379	0.0142050	0.1424521
MAIOR	0.0178249	0.0162930	0.0219541	0.0162189	0.1465811
DESVIO	0.0008480	0.0004623	0.0013926	0.0005629	0.0011188
MEDIANA	0.0159919	0.0152311	0.0158300	0.0152781	0.1441230

Recuperação Serial HDF - 5000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0149400	0.0152481	0.0169251	0.0152199	0.143501954
2	0.0142632	0.0148931	0.0159569	0.0153010	0.142104006
3	0.0155461	0.0151880	0.0147691	0.0148709	0.142682154
4	0.0148571	0.0153391	0.0149350	0.0170670	0.143687954
5	0.0166521	0.0144730	0.0155120	0.0160739	0.142066006
6	0.0163519	0.0152998	0.0159581	0.0153069	0.142704154
7	0.0156791	0.0149450	0.0159829	0.0151989	0.142111929
8	0.0146430	0.0150809	0.0161710	0.0152590	0.143982992
9	0.0151660	0.0155580	0.0155871	0.0158510	0.144712133
10	0.0248940	0.0152609	0.0168941	0.0146101	0.142515884
11	0.0167329	0.0151331	0.0157681	0.0146110	0.143531056
12	0.0151670	0.0152609	0.0159559	0.0147340	0.142241822
13	0.0148740	0.0152659	0.0160711	0.0150001	0.146041141
14	0.0143158	0.0147271	0.0146999	0.0148451	0.142163129
15	0.0158429	0.0145891	0.0161092	0.0153520	0.142052907
16	0.0157120	0.0156460	0.0150521	0.0149009	0.142303177
17	0.0151401	0.0149751	0.0151858	0.0149400	0.141376932
18	0.0149209	0.0156760	0.0163791	0.0160811	0.142990086
19	0.0232871	0.0163410	0.0150752	0.0157990	0.142161931
20	0.0171480	0.0153670	0.0163431	0.0157650	0.141663047
21	0.0221219	0.0144579	0.0150261	0.0153990	0.142494056
22	0.0152521	0.0147378	0.0178168	0.0151072	0.142093909
23	0.0145390	0.0149701	0.0148170	0.0147061	0.143677929
24	0.0153680	0.0151229	0.0152519	0.0151980	0.143295228
25	0.0161068	0.0147820	0.0154879	0.0155969	0.143121857
MÉDIA	0.0163808	0.0151335	0.0157492	0.0153118	0.1428511
MENOR	0.0142632	0.0144579	0.0146999	0.0146101	0.1413769
MAIOR	0.0248940	0.0163410	0.0178168	0.0170670	0.1460411
DESVIO	0.0027889	0.0004167	0.0007661	0.0005648	0.0010390
MEDIANA	0.0153680	0.0151331	0.0157681	0.0152199	0.1425159

Recuperação Serial HDF - 10.000 - MÉDIAS

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	0.0159969	0.0155680	0.0164769	0.0152402	0.144083908
2	0.0153489	0.0155089	0.0150139	0.0158861	0.144860056
3	0.0149331	0.0148480	0.0151210	0.0155001	0.143558055
4	0.0159180	0.0149920	0.0157528	0.0146701	0.143283974
5	0.0159090	0.0148489	0.0157311	0.0152969	0.143297913
6	0.0150750	0.0167100	0.0156159	0.0150859	0.142922935
7	0.0151680	0.0146990	0.0160980	0.0150352	0.142960892
8	0.0156651	0.0151861	0.0162151	0.0162280	0.143354916
9	0.0152822	0.0153642	0.0153592	0.0160570	0.142542087
10	0.0159309	0.0149319	0.0153821	0.0146730	0.142666152
11	0.0154011	0.0150959	0.0156569	0.0155020	0.142930994
12	0.0149570	0.0152681	0.0155790	0.0151560	0.145478875
13	0.0154841	0.0166330	0.0158010	0.0157061	0.141961875
14	0.0159440	0.0153182	0.0159130	0.0154409	0.143404063
15	0.0147929	0.0147290	0.0147200	0.0148180	0.142912936
16	0.0154190	0.0146050	0.0152760	0.0157881	0.141921052
17	0.0158620	0.0150890	0.0159938	0.0157032	0.142342848
18	0.0156660	0.0155110	0.0154390	0.0154569	0.141935848
19	0.0152471	0.0158892	0.0159609	0.0159039	0.143043007
20	0.0144851	0.0150750	0.0166769	0.0148740	0.143128888
21	0.0154681	0.0157750	0.0201340	0.0155230	0.142003812
22	0.0150411	0.0153971	0.0160460	0.0151632	0.142742067
23	0.0154331	0.0145550	0.0162289	0.0158038	0.14205803
24	0.0152199	0.0148780	0.0155649	0.0184081	0.143310964
25	0.0152569	0.0153439	0.0154412	0.0166070	0.144357094
MÉDIA	0.0153962	0.0152728	0.0158879	0.0155811	0.1430825
MENOR	0.0144851	0.0145550	0.0147200	0.0146701	0.1419211
MAIOR	0.0159969	0.0167100	0.0201340	0.0184081	0.1454789
DESVIO	0.0003995	0.0005472	0.0009940	0.0007608	0.0008998
MEDIANA	0.0154011	0.0151861	0.0157311	0.0155001	0.1429609

Recuperação Serial HDF - 1000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados	
	1	15.26	15.97	14.67	15.16	143.25
	2	17.45	14.71	16.58	14.54	142.95
	3	15.76	15.20	15.69	14.52	142.20
	4	15.70	15.67	14.82	14.60	144.01
	5	15.15	15.69	15.00	14.41	142.72
	6	14.95	15.35	15.83	14.55	143.19
	7	15.57	14.89	15.44	14.48	142.31
	8	16.05	15.53	14.95	15.97	142.78
	9	15.45	15.58	15.43	16.52	144.65
	10	16.29	15.54	14.76	15.94	142.48
	11	15.68	14.83	14.83	16.13	142.86
	12	15.33	18.52	15.41	15.45	144.31
	13	15.79	14.45	16.09	15.30	141.92
	14	15.92	15.61	15.67	15.33	144.02
	15	15.54	16.03	15.28	15.13	142.31
	16	17.96	16.10	14.52	15.31	143.77
	17	15.66	15.70	15.42	15.72	142.37
	18	16.66	16.37	15.66	15.41	143.70
	19	16.31	16.33	14.65	15.78	142.08
	20	15.54	14.58	18.20	14.99	142.44
	21	16.05	15.17	16.44	14.98	144.35
	22	16.24	15.11	16.67	15.76	142.29
	23	15.99	15.60	15.92	15.92	150.42
	24	15.55	15.27	16.90	15.61	143.64
	25	17.41	16.93	14.65	15.35	143.50
MÉDIA		15.97	15.63	15.58	15.31	143.38
MENOR		14.95	14.45	14.52	14.41	141.92
MAIOR		17.96	18.52	18.20	16.52	150.42
DESVIO		0.73	0.84	0.88	0.58	1.67
MEDIANA		15.76	15.58	15.43	15.33	142.95

Recuperação Serial HDF - 2500 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados	
	1	38.85	37.73	40.13	37.49	356.13
	2	39.80	37.69	38.96	39.21	362.31
	3	42.57	37.69	40.15	38.14	357.59
	4	37.18	38.69	39.54	37.40	366.45
	5	39.98	37.53	40.63	37.03	356.92
	6	44.56	37.07	38.50	38.33	362.03
	7	39.16	37.72	38.95	36.97	363.89
	8	42.53	39.76	38.97	40.55	358.20
	9	40.01	38.25	39.58	40.54	356.91
	10	41.87	39.20	38.60	39.14	358.06
	11	43.00	38.08	39.17	40.12	365.44
	12	39.73	40.45	40.07	38.20	357.33
	13	42.17	37.68	39.82	37.67	360.31
	14	39.59	36.99	37.84	38.84	363.14
	15	37.86	36.23	39.16	40.19	360.10
	16	42.74	40.02	41.03	38.49	360.82
	17	41.66	40.73	42.95	39.10	359.87
	18	38.14	37.86	39.05	38.40	358.14
	19	39.68	39.20	39.57	35.51	364.16
	20	40.18	39.92	38.90	35.77	358.70
	21	42.95	39.06	41.03	37.64	361.55
	22	38.49	38.06	54.89	37.76	362.06
	23	37.09	37.12	40.03	39.46	361.93
	24	37.17	38.40	41.45	36.16	359.92
	25	42.00	38.29	47.07	36.82	361.30
MÉDIA		40.36	38.38	40.64	38.20	360.53
MENOR		37.09	36.23	37.84	35.51	356.13
MAIOR		44.56	40.73	54.89	40.55	366.45
DESVIO		2.12	1.16	3.48	1.41	2.80
MEDIANA		39.98	38.08	39.58	38.20	360.31

Recuperação Serial HDF - 5000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	74.70	76.24	84.63	76.10	717.51
2	71.32	74.47	79.78	76.50	710.52
3	77.73	75.94	73.85	74.35	713.41
4	74.29	76.70	74.68	85.33	718.44
5	83.26	72.36	77.56	80.37	710.33
6	81.76	76.50	79.79	76.53	713.52
7	78.40	74.73	79.91	75.99	710.56
8	73.21	75.40	80.85	76.30	719.91
9	75.83	77.79	77.94	79.26	723.56
10	124.47	76.30	84.47	73.05	712.58
11	83.66	75.67	78.84	73.06	717.66
12	75.83	76.30	79.78	73.67	711.21
13	74.37	76.33	80.36	75.00	730.21
14	71.58	73.64	73.50	74.23	710.82
15	79.21	72.95	80.55	76.76	710.26
16	78.56	78.23	75.26	74.50	711.52
17	75.70	74.88	75.93	74.70	706.88
18	74.60	78.38	81.90	80.41	714.95
19	116.44	81.70	75.38	79.00	710.81
20	85.74	76.84	81.72	78.82	708.32
21	110.61	72.29	75.13	76.99	712.47
22	76.26	73.69	89.08	75.54	710.47
23	72.70	74.85	74.08	73.53	718.39
24	76.84	75.61	76.26	75.99	716.48
25	80.53	73.91	77.44	77.98	715.61
MÉDIA	81.90	75.67	78.75	76.56	714.26
MENOR	71.32	72.29	73.50	73.05	706.88
MAIOR	124.47	81.70	89.08	85.33	730.21
DESVIO	13.94	2.08	3.83	2.82	5.19
MEDIANA	76.84	75.67	78.84	76.10	712.58

Recuperação Serial HDF - 10.000 - TOTAL

Tentativas	Lustre	PVFS	CEPH	FhGFS	Banco de Dados
1	159.97	155.68	164.77	152.40	1440.84
2	153.49	155.09	150.14	158.86	1448.60
3	149.33	148.48	151.21	155.00	1435.58
4	159.18	149.92	157.53	146.70	1432.84
5	159.09	148.49	157.31	152.97	1432.98
6	150.75	167.10	156.16	150.86	1429.23
7	151.68	146.99	160.98	150.35	1429.61
8	156.65	151.86	162.15	162.28	1433.55
9	152.82	153.64	153.59	160.57	1425.42
10	159.31	149.32	153.82	146.73	1426.66
11	154.01	150.96	156.57	155.02	1429.31
12	149.57	152.68	155.79	151.56	1454.79
13	154.84	166.33	158.01	157.06	1419.62
14	159.44	153.18	159.13	154.41	1434.04
15	147.93	147.29	147.20	148.18	1429.13
16	154.19	146.05	152.76	157.88	1419.21
17	158.62	150.89	159.94	157.03	1423.43
18	156.66	155.11	154.39	154.57	1419.36
19	152.47	158.89	159.61	159.04	1430.43
20	144.85	150.75	166.77	148.74	1431.29
21	154.68	157.75	201.34	155.23	1420.04
22	150.41	153.97	160.46	151.63	1427.42
23	154.33	145.55	162.29	158.04	1420.58
24	152.20	148.78	155.65	184.08	1433.11
25	152.57	153.44	154.41	166.07	1443.57
MÉDIA	153.96	152.73	158.88	155.81	1430.83
MENOR	144.85	145.55	147.20	146.70	1419.21
MAIOR	159.97	167.10	201.34	184.08	1454.79
DESVIO	3.99	5.47	9.94	7.61	9.00
MEDIANA	154.01	151.86	157.31	155.00	1429.61