

Caio César de Melo e Silva

***Uma abordagem de alta-disponibilidade utilizando-se
de dispositivos móveis em configurações distribuídas***

Florianópolis

Outubro 2010

Caio César de Melo e Silva

***Uma abordagem de alta-disponibilidade utilizando-se
de dispositivos móveis em configurações distribuídas***

Monografia submetida à Universidade Federal
de Santa Catarina como parte dos requisitos
para a obtenção do grau de Bacharel em Ciên-
cias da Computação.

Orientador: Prof. Dr. Mario Antonio Ribeiro Dantas

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Florianópolis

Outubro 2010

Sumário

Resumo

Abstract

1	Introdução	p. 6
1.1	Introdução	p. 6
1.2	Objetivo do trabalho	p. 7
1.3	Justificativa	p. 7
2	Sistemas Distribuídos	p. 9
2.1	Introdução	p. 9
2.2	Agregados computacionais	p. 11
2.3	Grades computacionais	p. 13
2.4	Nuvens computacionais	p. 15
2.5	Alta disponibilidade em configurações distribuídas	p. 16
2.5.1	Pacemaker-Heartbeat	p. 18
2.5.2	DRDB	p. 20
3	Computação móvel	p. 21
3.1	Introdução	p. 21
3.2	Propriedades da computação móvel	p. 22
3.2.1	Comunicação sem fio	p. 22
3.2.2	Mobilidade	p. 23

3.2.3	Portabilidade	p. 24
3.3	Rede sem fio	p. 25
3.4	Desconexão	p. 27
3.5	Computação móvel distribuída	p. 29
4	Proposta	p. 31
5	Estudos de caso	p. 35
6	Conclusão e Trabalhos futuros	p. 39
	Referências Bibliográficas	p. 40

Resumo

Sistemas distribuídos são atualmente configurações computacionais utilizadas para resolver problemas que exigem grande poder computacional. Essas ambientes possibilitam a utilização de alto poder de processamento mesmo em áreas geograficamente distribuídas. Nesse contexto que se insere a computação móvel, com suas características intrínsecas de mobilidade e portabilidade, são excelentes interfaces para configurações distribuídas. Muitos desafios devem ser superados para que seja concretizada a união dessas dois conjuntos computacionais. Nesse trabalho são apresentados tais desafios e possíveis formas para solucionar os mesmos.

Palavras-chave: Computação Móvel, Computação Distribuída de Alto Desempenho.

Abstract

Distributed systems are currently used to solve problems that require large computational power. These environments gives the opportunity to use high processing power even in areas geographically distributed. In this context that the mobile computing appears, with its inherent characteristics of mobility and portability, mobiles are excellent interfaces for distributed configurations. Many challenges must be overcome to union these two computational areas. This work presents such challenges and possible ways to solve them.

Keywords: High Performance Computing, Mobile Computing, Distributed Computing Computing.

1 *Introdução*

1.1 *Introdução*

Desde do surgimento da computação, o ser humano continuamente necessita aumentar o poder computacional das máquinas, com a finalidade de solver problemas cada vez mais complexos de forma mais rápida. A solução inicialmente encontrada foi a criação de computadores poderosos, ou seja, com grande poder de processamento e memória, surgindo assim os *mainframes*, que devido a tecnologia microeletrônica da época, eram máquinas imensas. Apesar de atualmente existirem microcomputadores com poder computacional maior do que os *mainframes*, cada vez mais surgem aplicações que exigem um maior desempenho computacional.

De forma alternativa surgiram os *clusters* e, mais posteriormente, os *grids* ou grades computacionais, acompanhando a crescente demanda por processamento. O que fez desses sistemas uma boa solução foi a possibilidade de obter o mesmo poder de processamento de um supercomputador com o custo mais baixo, uma melhor relação custo benefício. Além disso, os *grids* foram uma solução para utilização conjuntas de recursos distribuídos geograficamente.

Sistemas ditribuídos tiveram sua grande expansão junto com a explosão da Internet, devido à padronização de comunicações entre sistemas heterogêneos. Entretanto, apesar da Internet ser uma ferramenta poderosa para interconexão de máquinas e dispositivos de arquiteturas heterogêneas, praticamente inexiste uma forma eficiente de controlar tais dispositivos. Além disso, graças a uma grande melhora na comunicação sem fio, dispositivos móveis passaram a fazer parte do contexto de sistema distribuído, trazendo mais desafios de controle de dispositivos nesses ambientes, agora, altamente heterogêneos.

Porém, o surgimento de um novo conceito de arquitetura distribuída, parece favorecer a heterogeneidade. As nuvens computacionais, como são chamadas, tem como objetivo “livrar” o usuário do processamento e armazenamento de suas aplicações e dados nos seus ambientes locais, ou seja, a instalação de novos softwares bem como onde os mesmos serão armazenados passa a ser responsabilidade da nuvem, cabendo ao usuário apenas informar o que e o quanto

precisa para suprir suas necessidades.

Inserido nesse contexto, os dispositivos móveis são uma ótima interface para esse ambiente devido suas características intrínsecas, como mobilidade e portabilidade. Conjuntamente, o poder computacional desses dispositivos tem crescido gradativamente favorecendo sua utilização, também, como recurso de ambientes distribuídos. Porém, para que esses dispositivos sejam utilizados de forma eficiente, é necessário um entendimento de suas restrições, como duração reduzida da bateria, o seu desempenho em *hardware*, a instabilidade da rede sem fio, entre outras.

As vertentes *Mobile grid* e mais recentemente *Mobile cloud* tem sido alvo de pesquisas constantes. Pesquisas essas que tem por finalidade minorar as limitações dos dispositivos móveis, objetivando um uso eficiente dos mesmos, que há alguns anos já fazem parte, de forma massiva, do contexto computacional.

1.2 Objetivo do trabalho

- **Objetivo Principal:** Propor e implementar uma abordagem de alta disponibilidade em configurações distribuídas utilizando dispositivos móveis.
- **Objetivos Específicos;**
 - Estudar os ambientes distribuídos e móveis;
 - Entender os desafios encontrados para inserção de dispositivos móveis na gerencia de ambientes distribuídos;
 - Compreender o funcionamento de configurações de alta disponibilidade;
 - Prover uma solução baseada em transparência de mobilidade;
 - Implementar esta solução com facilidades de alta disponibilidade considerando os ambientes móveis.

1.3 Justificativa

A motivação maior desse trabalho consiste em encontrar uma solução para o problema de alta-disponibilidade em ambientes distribuídos, sem dependência de escalonador, utilizando dispositivos móveis. Essa configuração se faz justificável ao passo que sistemas distribuídos atualmente são utilizados de forma quase que absoluta para problemas que demandam alto desempenho. Com a utilização massiva de dispositivos móveis, os mesmos passaram a fazer parte

do contexto distribuído. Porém a integração dessas duas vertentes computacionais ainda é falha sob muitos aspectos, são poucos os trabalhos encontrados referenciando alta-disponibilidade móvel.

O trabalho está dividido em seis capítulos. O primeiro trás uma introdução mostrando os aspectos gerais que envolvem o trabalho, bem como objetivos e justificativa para realização do mesmo. O segundo capítulo faz uma apresentação das arquiteturas distribuídas e mostra como a alta-disponibilidade aparece nessas configurações. O capítulo seguinte, terceiro, mostra os conceitos que envolvem a computação móvel, e quais as formas de integração de dispositivos móveis com sistemas distribuídos, mostrando limitações e vantagens para tal fim. No capítulo quatro é apresentada a proposta propriamente dita, bem como o ambiente experimental configurado. O estudo de caso envolvendo o sistema proposto e apresentado no capítulo cinco. E finalmente no capítulo seis conclusões e trabalhos futuros.

2 *Sistemas Distribuídos*

2.1 Introdução

Sistemas computacionais distribuídos quando surgiram na década de 80, apareceram como uma solução para o compartilhamento de recursos fisicamente separados e de alto custo. Devido aos avanços tecnológicos na computação e a uma exigência de desempenho cada vez maior de aplicações científicas, industriais e comerciais, a demanda por esses serviços tem aumentado gradativamente. De forma que, diversas organizações encontraram, nos sistemas distribuídos, uma solução barata para obtenção de melhor performance em seus programas.

Sob o aspecto de arquitetura de máquinas para execução de aplicativos, sistemas distribuídos devem ser vistos como configurações com grande poder de escala pela agregação dos computadores existentes na rede (DANTAS, 2005), permitindo que aplicações clientes tenham acesso remoto a outras máquinas para compartilhamento de informações e recursos.

Pode ser citada como principal vantagem dos sistemas ditribuídos a possibilidade de compartilhamento de recursos ociosos na rede, fazendo com que as tarefas sejam executadas de forma mais rápida, disponibilizando um maior espaço para armazenamento de informações e sendo ainda possível utilizar qualquer outro recurso conectado. Algumas outras vantagens podem ser notadas quanto ao uso de sistemas distribuídos:

- **Econômica:** agrega máquinas de baixo custo, podendo obter um alto poder de processamento;
- **Velocidade de processamento:** a capacidade de interconctar processadores facilita a obtenção de alta performance;
- **Tolerância a falhas:** quando um nodo falha, o sistema como um todo pode continuar funcionando;
- **Crescimento de poder computacional:** pode-se aumentar o poder computacional com a inserção de novos nodos;

- **Flexibilidade:** permite que usuários compartilhem dados e periféricos;
- **Distribuição inerente:** algumas aplicações possuem uma natureza distribuída. Ex.: Uma aplicação para rede de supermercado, jogos, etc.

Dado as vantagens que os sistemas distribuídos oferecem, surge uma vertente computacional voltada para o alto desempenho, nominada de computação distribuída de alto desempenho. *Clusters*, *Grids* e *Clouds* são atualmente, os representantes, sob o aspecto de arquitetura, dessa categoria computacional. Essas configurações podem ter como característica uma grande heterogeneidade no que diz respeito aos recursos nela disponíveis. Podendo ser citados como exemplos: PC's, notebooks, supercomputadores, PDA's, celulares, etc.

Existe um anseio para que os mais variados dispositivos sejam utilizados da melhor forma possível em ambientes de alto desempenho. Atualmente, devido principalmente à comunicação sem fio, os dispositivos móveis tem sido alvo de várias pesquisas que tem como intuito inserir tais dispositivos em ambientes distribuídos. Pois, além da portabilidade e mobilidade que esses dispositivos oferecem, houve um crescimento significativo na capacidade de processamento dos mesmos, possibilitando a utilização de dispositivos móveis não só como interface do ambiente mas também como recurso.



Figura 2.1: Arquitetura modelo de um sistema distribuído.

2.2 Agregados computacionais

Conceitualmente, um agregado computacional ou *cluster*, consiste em um grupo de máquinas físicas fracamente acopladas capazes de suportar o mesmo conjunto de máquinas virtuais (HUGHES-FENCHEL, 1997). Agregados podem ser também definidos como configurações distribuídas, nas quais seus nodos possuem poder de processamento e memória, vistos como um recurso computacional único.

Um cluster, sob a perspectiva física, é um agrupamento de computadores, mais especificamente, dois ou mais computadores interconectados por um meio. Geralmente esse meio é uma tecnologia de rede já estabelecida, como por exemplo a Ethernet.

Agregados apresentam as seguintes características (PFISTER, 1998) (SHIRAZI; KAVI; HURSON, 1995):

- Usualmente pertencem a um domínio administrativo único, e por isso é usado para resolver problemas de uma determinada empresa ou grupo de pesquisa;
- Externamente, são vistos como um único sistema computacional. Seus nodos, não possuem identidade própria, são apenas engrenagens de um sistema maior, o *cluster*.
- A latência nas comunicações é pequena, da ordem de micro-segundos ou menor;
- A taxa de transmissão disponível é elevada;
- Pode se beneficiar de protocolos de comunicação mais eficientes entre seus nodos, pois a rede de interconexão, pertence ao mesmo domínio administrativo, é um recurso controlado;

Cada cluster será construído dependendo da aplicação que nele irá rodar. Temos basicamente dois tipos de cluster, os de alta disponibilidade (HA – High Availability) e os clusters voltados para o alto desempenho de aplicações (HPC- High Performance Computation).

HPC's são normalmente utilizados quando temos tarefas que demandam muito processamento, que se caso fossem executadas em microcomputadores convencionais demandariam muito tempo. Já os HA's são utilizados quando um sistema computacional não pode apresentar falhas quanto a disponibilidade do sistema (HOCHSTETLER et al., 2004).



Figura 2.2: Imagem de um *cluster* real

- **Vantagens:**

- *Escalabilidade*: Pode crescer muito mais do que um único computador;
- *FailOver*: Pode tolerar defeitos em nodos e continuar a oferecer serviços;
- *Baixo custo*: Pode ser construído a partir de componentes COTS (*Common off-the-shelf*);
- *Implementação*: Por hardware é mais eficiente, porém pouco adaptável, já por software há um menor custo. Cabe ao arquiteto do sistema verificar qual a melhor opção dado o problema a ser resolvido.

- **Desvantagens:**

- Geralmente pertence a um único domínio administrativo;
- Buscando facilitar a programação das tarefas a serem executadas no cluster os hardwares e softwares empregados são homogêneos;
- Não é possível executar programas convencionais (seriais) com alto desempenho sem nenhuma modificação de código;
- O desempenho total do sistema é diretamente proporcional ao desempenho individual de cada estação, a qualidade do meio de comunicação e ao número de nodos que estarão conectados ao sistema;
- Maior desempenho desejado requer maior espaço físico, mais dinheiro a ser empregado, pessoas qualificadas para fazer manutenções constantes;

- O cluster necessita ficar alocado em lugar seguro, seco e refrigerado, portanto é imprescindível a utilização de equipamento de ar-condicionado dimensionado conforme o calor gerado pelos equipamentos do cluster.

2.3 Grades computacionais

Segundo (BUYAYA, 2010), grades computacionais são um tipo de sistema paralelo e distribuído que possibilita compartilhamento, seleção e alocação de recursos autônomos, geograficamente distribuídos, em tempo de execução dependendo da sua disponibilidade, capacidade, performance, custo e qualidade de serviço requerido pelos usuários de um *grid*. Ou seja, um sistema paralelo e distribuído, assim como o cluster, que tem a capacidade de dividir recursos com usuários.

Com isso, empresas conseguem um alto poder de processamento de dados sem nenhuma alteração em seu conjunto de máquinas. Por exemplo, uma empresa que pretende utilizar um *grid*, pode usufruir de períodos noturnos, momento em que seus funcionários, usuários dos nodos da empresa, não estejam utilizando os mesmos, para distribuir tarefas e processar dados em um "supercomputador" recém formado. Diferentemente dos clusters, *grids* não determinam que os recursos devam ser dedicados única e exclusivamente ao compartilhamento. Em ambientes de grade, é possível que um usuário determine quanto de cada um de seus recursos ele pretende compartilhar e em quais momentos isto deve ser feito.

Dessa forma, pode ser notada uma redução no custo, pois agora é possível obter um poder de processamento grande utilizando estações de trabalho, sem a necessidade de dedicação exclusiva. Os recursos são usados pela grade em períodos ociosos, ou seja, momentos onde usuários locais não estão utilizando seus próprios recursos. Nas grades computacionais o usuário local pode interromper a execução na sua própria estação, sem que qualquer restrição da grade. Porém, isto gera uma série de problemas, que devem ser tratados pelos programadores e administradores dos *grids*. Devem haver garantias relativas à integridade do sistema, à privacidade do usuário, bem como uma validação e aceitação dos resultados obtidos, além de determinar o que deve ser feito em caso de perda de controle do recurso. Estes são apenas alguns dos possíveis problemas encontrados ao se configurar ambientes de grade computacional.

Uma outra característica dos *grids* é a possibilidade de se utilizar nodos em redes completamente diferentes, geograficamente separadas. Muitas das grades atuais funcionam dessa forma, utilizam a rede da internet para ser o seu canal de comunicação. Iniciativas como a **SETI@home** (ANDERSON et al., 2002), **distributed.net** (DISTRIBUTED.NET, 2010), **Inte-**

Grade (GOLDCHLEGER, 2004), entre muitos outros. Essas configurações de grid funcionam da seguinte maneira, o usuário é encorajado a instalar um *software* com a finalidade de estabelecer um cliente em sua máquina. Uma vez instalado, o computador fará parte da grade, porém, não necessariamente, o computador estará disponibilizando todos os seus recursos a todo tempo.

Dentro desse contexto aparecem os *grids* oportunistas (GOLDCHLEGER, 2004). Algumas dessas grades disponibilizam ao usuário a opção de configurar o quanto de seus recursos o mesmo pretende compartilhar com a grade. Outros sistemas utilizam os recursos dos usuários somente quando o computador estiver ocioso. Algumas grades disponibilizam informações sobre o processamento da grade para os usuários em um protetor de tela, por exemplo, mostrando estatisticamente ao usuário como a grade utiliza seus recursos.

As aplicações clientes das grades oportunistas costuma ser muito pequenas e transparentes ao usuário, e respeitam o poder maior que, no caso das grades oportunistas, é do usuário local. Esse poder de controle se deve ao fato de que, diferentemente aos clusters, cada nodo da grade possui seu próprio gerenciador de recursos, que prioriza as aplicações locais.

O Boinc (*Berkley Open Infrastructure for Network Computing*) (ANDERSON, 2004) é uma plataforma aberta para projetos de processamento distribuído, inicialmente desenvolvido pela Universidade de Berkley, California. baseado-se nesta plataforma que muitas das grades atuais se sustentam, como, por exemplo, a SETI@Home (ANDERSON et al., 2002), Einsten@home (ABBOTT et al., 2009), entre outras. Na página oficial do projeto (BOINC, 2010) é possível obter informações passo-a-passo de como criar sua própria grade.

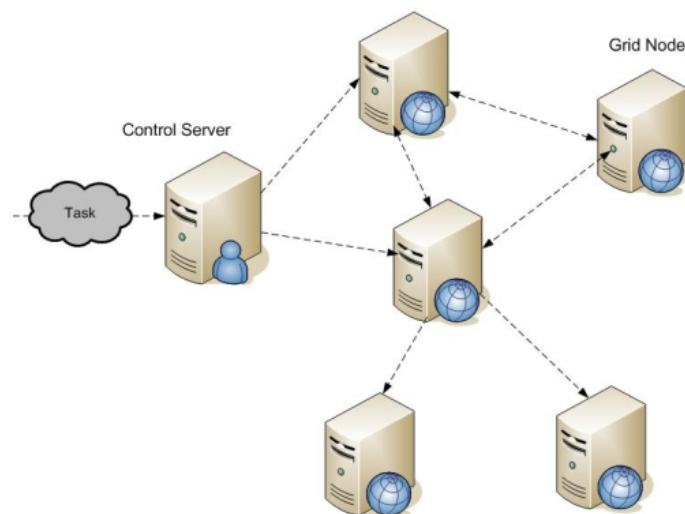


Figura 2.3: Grid computacional

2.4 Nuvens computacionais

Nuvem computacional é uma denominação relativamente recente, baseada nos conceitos de virtualização, computação distribuída, *utility computing*, redes, web e serviços de software. Ainda não existe uma definição estabelecida, muitas discordâncias são encontradas na literatura. Segundo (WANG et al., 2008), computação nas nuvens se define por um conjunto de serviços disponíveis na rede, provendo escalabilidade, garantia de qualidade de serviço, com plataformas computacionais de baixo custo, que podem ser acessadas de forma simples e pervasiva.

Nuvens computacionais podem disponibilizar, como serviço, uma plataforma computacional integrada. Para que isso ocorra, é necessário que serviços de acesso a *hardware*, *software* e dados estejam disponíveis ao usuário. De forma mais clara:

- O sistema *Hardware-as-a-service* (HaaS) é uma modalidade que permite que usuários utilizem recursos de hardware de forma virtual, flexível, escalável e configurável atendendo as necessidades do usuário (HAAS, 2006).
- No *Software-as-a-service* (SaaS) as aplicações são disponibilizadas ao usuário de forma virtual, através da internet. Assim, não há necessidade, por parte do usuário, de instalar e rodar a aplicação no seu próprio computador (WANG et al., 2008).
- Já no *Data-as-a-service* (DaaS) dados de diferentes fontes e em vários formatos, podem ser acessados pelos usuários através da rede, em outras palavras, o usuário faria a manipulação dos dados remotos como se estivessem em seu disco local (WANG et al., 2008).

Através dos mecanismos HaaS, SaaS e DaaS, a nuvem pode oferecer, para os usuários, uma plataforma integrada como serviço, chamada na literatura de *Platform-as-a-service* (PaaS). Assim, cada usuário pode escolher uma plataforma que melhor supre suas necessidades quanto a configuração de *hardware*, *softwares* instalados e acesso a dados. A figura 2.4 apresenta um esquema dessas funcionalidades na nuvem.

Alguns dos conceitos que envolvem a computação nas nuvens a distingue de outros paradigmas computacionais. Os recursos e serviços disponibilizados para os usuários são sob demanda, dessa forma, usuários podem personalizar seu ambiente computacional. As configurações computacionais, providas pela nuvem, podem garantir qualidade de serviço (QoS) para os usuários, como velocidade de CPU, tamanho de memória, ou seja, performance em hardware. Outra característica das nuvens computacionais é sua autonomicidade. *Hardware*, *software* e dados dentro das nuvens podem ser automaticamente reconfigurados, apresentando uma nova plataforma ao usuário.

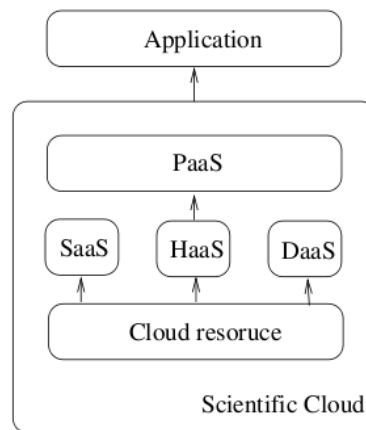


Figura 2.4: Nuvem científica

Outros dois aspectos de suma importância quanto à caracterização de nuvens computacionais são a escalabilidade e a flexibilidade. Os serviços e as plataformas computacionais oferecidas pela nuvem, devem ser disponibilizadas em diferentes regiões geográficas, com diferentes configurações de hardware e software, sendo que essas plataformas devem ser flexíveis para que se adaptem aos mais diversos pedidos de um potencial grande número de usuários.

Tecnologias de virtualização como VMware (VMWARE, 2010), a Web 2.0 (WEB2.0, 2010), serviços Web, arquitetura orientada a serviços (SOA), sistemas de armazenamento distribuído, todos esses conceitos e tecnologias orquestram a computação nas nuvens, de forma que a partir deles é viável possuir flexibilidade e escalabilidade em serviços de *hardware*, bem como uma melhor interconectividade e iteratividade de aplicações Web.

As tecnologias móveis já fazem parte dos sistemas distribuídos, e a sua utilização nesses ambientes está em constante crescimento. São encontrados alguns desafios para a inserção desses dispositivos, como limitação de processamento e armazenamento e fonte de energia finita. Porém, uma vez concretizada, a computação móvel distribuída será uma ótima ferramenta para solução de problemas tanto de âmbito acadêmico como comercial.

2.5 Alta disponibilidade em configurações distribuídas

No contexto de sistemas computacionais em geral, Alta Disponibilidade (HA) refere-se à existência de componentes no sistema capazes de substituir instantaneamente outros componentes que tornem-se inativos, independentemente da causa dessa inatividade. Desse modo garantindo que um sistema nunca, ou quase nunca, ficará indisponível em consequência de uma falha.

Novamente, como ocorre com supercomputadores, existe uma variedade de maneiras para alcançar esta característica “variando de soluções que utilizam hardware customizado e redundante para garantir disponibilidade, a soluções baseadas em software usando componentes de hardware off the shelf”(IEEE, 2010). A primeira maneira possibilita um maior nível de disponibilidade, porém tem um custo significativamente mais elevado que a segunda. Tal fato levou a uma popularização da segunda classe de soluções, que tipicamente tratam de um ponto único de falha no sistema.

Num ambiente de cluster, a alta disponibilidade é alcançada através da redundância do nodo mestre. O papel do nodo mestre secundário (redundante) é exatamente o mesmo que o nodo mestre primário, porém a maior parte do tempo de vida do cluster, este fica inativo, apenas monitorando o nodo mestre primário. Ao detectar a indisponibilidade do nodo mestre primário, o secundário assume seu papel, executando todas as funções do mestre primário de maneira idêntica. Desse modo essa substituição entre mestre primário e secundário é transparente para os usuários ou aplicações que utilizam o cluster.

Para o melhor entendimento do assunto devemos ter claros os conceitos de alguns termos relacionados a alta disponibilidade:

- **Disponibilidade Contínua:** Implica em serviço disponível 100 por cento do tempo, sem ocorrer períodos em que esteja indisponível. Isso representa um estado ideal não alcançado por nenhum sistema existente atualmente. “Alta Disponibilidade não implica em Disponibilidade Contínua”(IEEE, 2010).
- **Tolerância a Falhas:** É um meio de alcançar níveis muito altos de disponibilidade. Um sistema tolerante a falhas tem a capacidade de continuar disponível apesar de uma falha de hardware ou software, e é caracterizado por redundância de hardware, incluindo CPU, memória, e subsistemas de E/S. “Alta Disponibilidade não implica em Tolerância a Falhas”(IEEE, 2010).
- **Ponto Único de Falha:** Um componente de hardware ou software cuja perda resulta na perda do serviço. Tais componentes não têm backup de componentes redundantes (IEEE, 2010).
- **Recuperação de Falhas (Failover):** Quando um componente em um sistema HA falha resultando em uma perda de serviço, o serviço é iniciado pelo sistema HA em outro componente do sistema. Essa transferência de serviço após uma falha no sistema é chamada de Recuperação de Falhas (IEEE, 2010).

2.5.1 Pacemaker-Heartbeat

O Pacemaker é um gerenciador de recursos de cluster. Atinge o máximo de disponibilidade para os serviços de cluster (recursos) através da detecção e recuperação de falhas de nó em nível de recursos, fazendo uso dos recursos de mensagens e de filiação fornecidas pela sua infra-estrutura de cluster preferencial (ou OpenAIS ou batimentos cardíacos). Ele pode fazer isso para clusters de praticamente qualquer tamanho e vem com um modelo de dependência poderosa que permite que o administrador possa precisamente expressar as relações (tanto de encomenda e localização) entre os recursos de cluster. Praticamente tudo o que pode ser script podem ser gerenciados como parte de um cluster de pacemakers. Vale a pena ressaltar que o Pacemaker é não um *fork* do Heartbeat, como parece ser um equívoco comum. O Pacemaker é uma continuação do CRM (v2 gerenciador de recursos), que foi originalmente desenvolvido para o Heartbeat, mas desde então se tornou parte do projeto em si. (CLUSTERLAB, 2010)

Características principais (CLUSTERLAB, 2010):

- Detecção e recuperação de falhas de nós e de nível de serviço;
- Armazenamento agnóstico, não há qualquer requisito de armazenamento compartilhado;
- Recursos agnósticos, qualquer coisa que possa ser *script* podem ser clusterizadas.
- Suporte para cluster de pequeno e grande porte;
- Suporte para praticamente qualquer tipo de redundância, incluindo: Ativa/Ativa, Ativa/Passiva, N+1, N+M, N para 1, N para N (as figuras 2.5, 2.6 e 2.7 ilustram algumas dessas configurações)
- Tipos de recursos avançados como suporta para serviços que precisam ser ativados em vários nós, bem como para diferentes modos(mestre/escravo, primário/secundário);

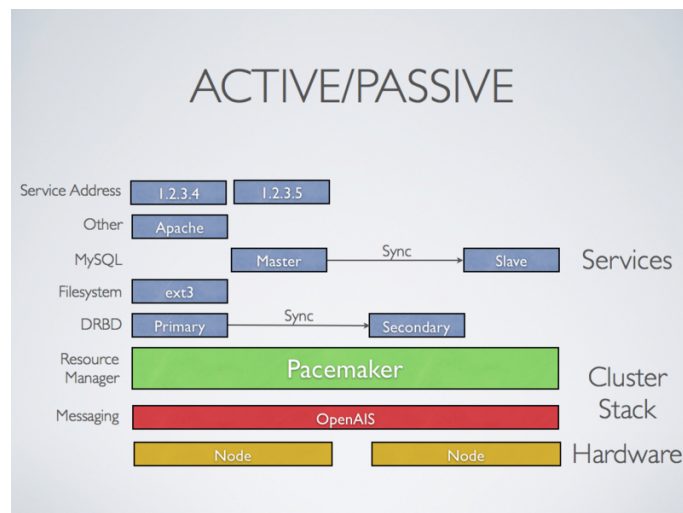


Figura 2.5: Modelo Ativo/Passivo

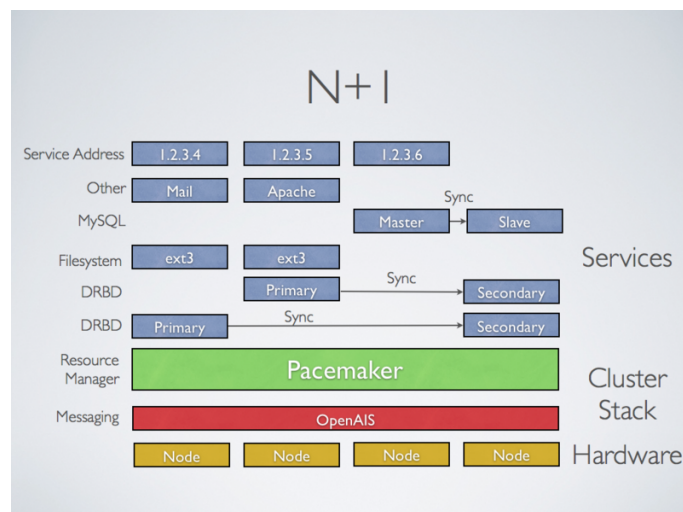


Figura 2.6: Modelo N+1

RSYNC

O rsync é um software para sistemas Unix que sincroniza arquivos e diretórios de um local para outro, minimizando a transferência de dados. Uma característica importante do rsync não encontrado na maioria dos programas e protocolos similares é que o espelhamento ocorre apenas com uma transmissão em cada direção. O rsync pode copiar ou exibir o conteúdo do diretório e copiar arquivos, opcionalmente, usando compressão e recursão. No modo daemon, rsync escuta na porta TCP padrão 873, servindo os arquivos no protocolo rsync nativa ou através de um shell remoto como o RSH ou SSH. (RSYNC, 2010)

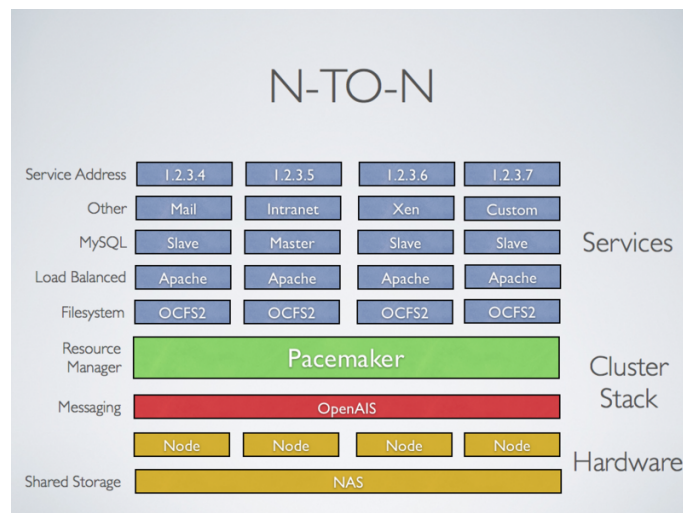


Figura 2.7: Modelo N para N

2.5.2 DRDB

DRBD (Distributed Replicated Block Device) é um sistema de armazenamento distribuído para a plataforma GNU / Linux. É constituído por um módulo de kernel, vários aplicativos de gerenciamento de userspace e alguns scripts shell e é normalmente usado em alta disponibilidade (HA) clusters. DRBD tem semelhanças com o RAID 1, exceto que ele é executado em uma rede. DRBD refere-se tanto o software (módulo do kernel e ferramentas associadas userspace), e também a determinados dispositivos de bloco lógico é gerenciado pelo software. DRBD dispositivo DRBD dispositivo de bloco e também são frequentemente utilizados para o último (DRDB, 2010).

Apesar de ser uma poderosa ferramenta o DRDB nesse trabalho não foi completamente utilizado. Foi utilizado, por sua facilidade de integração com o pacote Pacemaker-Heartbeat, para fazer os testes referentes a alta-disponibilidade.

3 *Computação móvel*

3.1 Introdução

A computação móvel sob uma abordagem de sistemas distribuídos, evidenciou uma significativa expansão no últimos anos, possibilitando que os usuários de dispositivos móveis tenham acesso a um ambiente de rede e compartilhem dados, recursos e serviços. A crescente utilização deste paradigma é originada, sobretudo, dos avanços obtidos recentemente nas tecnologias de rede sem fio, que proporcionam aos usuários acesso a uma infra-estrutura compartilhada, independente de sua localização física (FORMAN; ZAHORJAN, 1994). Outro aspecto que pode ser observado é a disseminação da utilização destes equipamentos devido à redução de valores, facilidades de transporte, tamanho reduzido e leveza, além de mais funcionalidades incorporadas ao aparelho. Como exemplo podem ser citados os *smartphones* que reúnem funcionalidades de dispositivos computacionais e de telefone celular.

O advento da computação móvel propicia o acesso a informações *anywhere-anytime* (PERRY et al., 2001) o que traz benefícios aos usuários principalmente no que diz respeito à possibilidade de mobilidade, porém alguns desafios surgiram juntamente com essa tecnologia. Alguns autores (BORGES; DANTAS, 2006) (BRUNEO et al., 2003) (CLARKE; HUMPHREY, 2002) (LITKE; SKOUTAS; VARVARIGOU, 2004) (PARK; KO; KIM, 2003) (PHAN; HUANG; DULAN, 2002) citam diversos problemas e desafios que são impostos pelo paradigma da computação móvel e que estão em fase de pesquisas ou ainda em aberto. Do mesmo modo, os avanços na computação móvel tornam possível que mais usuários acessem serviços através de uma infra-estrutura de rede independente de localização, todavia, as suas características inerentes revelam grandes desafios. Algumas dessas características que suscitam problemas alvos de pesquisas, a fim de alcançar soluções apropriadas, são:

- duração reduzida de bateria em relação às necessidades dos usuários;
- pequena capacidade de processamento;

- reduzida capacidade de memória;
- heterogeneidade de dispositivos (diferentes tipos de interfaces com maneiras de interação diferentes e limitada ou ainda baixa resolução e pequenas dimensões do monitor);
- instabilidade e qualidade da rede;
- alta mobilidade.

O trabalho de (FORMAN; ZAHORJAN, 1994) aponta que os desafios da computação móvel não são triviais, ou seja, fogem do paradigma conhecido de sistemas distribuídos. Conforme (CHEN; KOTZ, 2000), (IMIELINSKI; VISWANATHAN; BADRINATH, 1994), (SATYA-NARAYANAN, 2001), (KLEIN et al., 2010) e (SAMIMI; MCKINLEY; SADIJADI, 2006) o desenvolvimento de sistemas para estes ambientes, que envolvem dispositivos móveis e redes sem fio (wireless), necessitam de um tratamento diferenciado do projeto de uma aplicação convencional de sistemas distribuídos. Os elementos que devem ser considerados neste processo envolvem, sobretudo, as características inerentes destes ambientes, o padrão de comunicação e a rede.

3.2 Propriedades da computação móvel

Nesta seção são apresentadas as principais propriedades da computação móvel: 3.2.1 comunicação sem fio; 3.2.2 mobilidade; e 3.2.3 portabilidade. Além disso, evidenciam-se os desafios oriundos de tais propriedades da computação móvel que estão intimamente relacionadas. De fato, os elementos descritos devem ser considerados de forma conjunta como parte fundamental dos projetos de sistemas para estes ambientes, diferentemente das abordagens tradicionais.

3.2.1 Comunicação sem fio

A comunicação sem fio, ou comunicação *wireless*, estabelece a troca de mensagens entre dois pontos ou dispositivos por meio de uma rede que utiliza um mecanismo de ondas eletromagnéticas para transmitir os dados. Segundo (PITOURA; BHARGAVA, 1994), em função do acesso sem fio, a comunicação na computação móvel defronta-se com muitos obstáculos devido ao fato que o ambiente interage com o sinal. Além disso, em contraste com computadores estacionários, os quais ficam conectados a uma única rede, computadores móveis encontram mais conexões de rede heterogêneas em diversos caminhos (FORMAN; ZAHORJAN, 1994).

Alguns dos problemas decorrentes da comunicação sem fio são: largura de banda limitada ou altamente variável, alta taxa de erros, aumento na latência de comunicação e retransmissões.

Nesse contexto, nota-se que as redes sem fio (*wireless*) possuem um recurso de largura de banda escasso em relação à rede cabeada, mesmo considerando as melhorias alcançadas nos últimos anos. Ademais, outros fatores que tornam mais complexa a comunicação são o fato da largura de banda ser dividida entre os usuários que estão em uma célula e ainda o movimento dos usuários com seus dispositivos. (MATEUS; LOUREIRO, 1998) afirma que devido às altas taxas de erro na comunicação *wireless*, a eficiência do canal é comprometida. Em decorrência disso, as desconexões na comunicação *wireless*, tornam-se frequentes.

(FORMAN; ZAHORJAN, 1994) observam ainda que, ao considerar que a conexão a um link sem fio é fácil, a segurança da comunicação pode ser comprometida mais facilmente que na comunicação por meio de cabos (*wired*), especialmente se a transmissão estende-se por uma grande área. Isto requer que os projetos de software incluam medidas de segurança. Um mecanismo de comunicação segura que pode ser empregado em canais não seguros é a criptografia.

3.2.2 Mobilidade

Mobilidade é a habilidade de mudar de localização enquanto conectado a uma rede (FORMAN; ZAHORJAN, 1994). No entanto, a partir desta habilidade, característica inerente dos sistemas de computação móvel, surgem outros fatores relevantes, tais como, informações que em ambientes fixos são considerados estáticos, em ambientes móveis tornam-se dinâmicos. Por exemplo, em um computador fixo, configura-se um servidor a ser usado, já no dispositivo móvel necessita de um mecanismo para determinar qual servidor usar dependendo do local. Assim, podemos observar que em um ambiente móvel, a localização do usuário pode ser estimada com um item de dado no qual valores mudam a cada movimento (PITOURA; SAMARAS; SAMARAS, 1998).

A computação móvel se distingue da clássica conexão fixa devido a: 1) a mobilidade de usuários nômades e seus computadores e 2) restrições dos recursos móveis, tais como largura de banda *wireless* escassa e tempo de vida da bateria limitado. A mobilidade de usuários nômades implica que os usuários podem conectar-se de diferentes access points através de links *wireless* e podem querer ficar conectados enquanto se movem, apesar da possibilidade de desconexão.

A mobilidade apresenta benefícios para os usuários móveis na medida em que estes não precisam conhecer a localização do servidor, dos dados ou dos recursos que desejam acessar. Neste sentido, os sistemas devem proporcionar mecanismos que dêem suporte à mobilidade de

maneira transparente para o usuário final. Segundo (IMIELINSKI; VISWANATHAN; BADRI-NATH, 1994), mobilidade é um comportamento com implicações para ambas as redes, fixas e sem fio. Na rede fixa, os usuários móveis podem estabelecer uma conexão de diferentes tipos de portas em locais diferentes. A conexão sem fio permite mobilidade e conectividade irrestrita de qualquer local dentro da área de cobertura. De acordo com (SATYANARAYANAN, 1996) a mobilidade possui restrições intrínsecas:

- elementos móveis são recursos pobres em relação a elementos estáticos: considerando diversos aspectos tais como tamanho, energia, ergonomia, recursos computacionais (processador, memória e capacidade de disco).
- restrição de mobilidade: refere-se ao aspecto de vulnerabilidade quanto a perda e roubo, ou seja, além dos benefícios da mobilidade, herdamos as fragilidades.
- conectividade é altamente variável e impacta no desempenho e confiabilidade: algumas construções podem oferecer confiabilidade e alta largura de banda, porém outras podem somente oferecer baixa largura de banda. Em ambientes abertos este problema pode ser ainda maior, ocasionando lacunas na cobertura da rede.
- elementos móveis dependem de uma fonte de energia finita: enquanto não estiverem disponíveis melhorias quanto às fontes de energia para dispositivos móveis, será necessário considerar sensivelmente este aspecto nos projetos de software e hardware.

3.2.3 Portabilidade

A portabilidade em computação móvel refere-se à capacidade de portar um dispositivo com capacidade de computação, de pequenas dimensões, leve e com certo tempo de autonomia de energia (PITOURA; SAMARAS; SAMARAS, 1998). A portabilidade traz diversos benefícios a usuários que necessitam deslocar-se com frequência e ao mesmo tempo carecem de informações ou acesso de computação para manter sua produtividade. No entanto, juntamente com estas características e benefícios, menores recursos estão disponíveis quando comparados com computadores convencionais. Cabe também destacar que existem várias categorias de dispositivos de computação móvel, desde laptops com maiores recursos, PDA, smartphones, até os aparelhos celulares, com recursos mais limitados. Outro aspecto que merece atenção é o fato destes dispositivos não possuírem muito espaço para baterias, ocasionando um tempo de autonomia de energia limitado. (FORMAN; ZAHORJAN, 1994) aponta nestes termos que as aplicações podem conservar energia pela redução da sua necessidade por computação, comunicação e memória.

3.3 Rede sem fio

O objetivo das redes sem fio é fornecer conectividade em uma infra-estrutura sem fios e cabos. Estas redes proporcionam aos usuários móveis meios de comunicação com outras entidades (usuários, recursos, redes). (COULOURIS; DOLLIMORE; KINDBERG, 2007) apresenta, conforme tabela 3.1, quatro tipos de redes sem fio: WPAN, WLAN, WMAN e WWAN. Na tabela é possível observar o alcance de cada tipo de rede sem fio, bem como a largura de banda e a latência.

Rede	Exemplo	Alcance	Largura de banda (Mbps)	Latência (ms)
WPAN	Bluetooth (IEEE 802.15.1)	10-30m	0,5-2	5-20
WLAN	Wifi (IEEE 802.11)	0,15-1,5 Km	2-54	5-20
WMAN	WiMax (IEEE 802.16)	5-50 Km	1,5-20	5-20
WWAN	Redes telefônicas GSM, 3G	Mundial	0,010-2	100-500

Tabela 3.1: Tipos de rede sem fio (COULOURIS; DOLLIMORE; KINDBERG, 2007)

- WPAN - (*Wireless Personal Area Network*) - rede pessoal sem fio. Fornece conectividade com uma pequena abrangência para dispositivos móveis, ou seja, que estejam fisicamente próximos. É interessante para interligar teclados, impressoras, telefones móveis, agendas eletrônicas, computadores de mão, câmeras fotográficas digitais, mouses e outros. Os padrões mais utilizados para estabelecer a comunicação são ondas de raio infravermelho.
- WLAN - (*Wireless Local Area Networks*) - redes locais sem fio - rede para fornecer conectividade para dispositivos móveis dentro de uma área (casa, prédio) e prover acesso à Internet ou outra rede fixa (LAN, WAN). Possui variantes de padrões, oferecendo largura de banda de 10 e 100 Mbps com abrangência de até 1,5 quilômetros.
- WMAN - (*Wireless Metropolitan Area Network*)- rede metropolitana sem fio - possuem, geralmente, maior abrangência que as WLAN, por exemplo, uma cidade ou região metropolitana.
- WWAN - (*Wireless Wide Area Networks*) - rede de longa distância sem fio - é por meio deste tipo de rede que os usuários de telefonia móvel têm acesso com ampla cobertura, seja por antenas ou sistemas de satélite. É importante salientar que estas redes oferecem taxas de transmissão de dados relativamente baixas quando comparadas com as demais redes.

Conforme (PITOURA; BHARGAVA, 1994), uma estrutura de sistemas distribuídos que suporta mobilidade possui uma arquitetura que consiste em dois tipos distintos de *hosts*: móveis e fixos. Os *hosts* móveis são considerados computadores/dispositivos portáteis interligados em rede por meio de uma conexão sem fio, possibilitando a mobilidade. Já os *hosts* fixos (computadores) estão em uma rede cuja localização e conectividade são constantes. Alguns dos *hosts* fixos, chamados de estação base ou estação de suporte móvel, são acrescidos com uma interface *wireless* para comunicar com *hosts* móveis.

A figura 3.1 apresenta a arquitetura de rede que suporta conexão sem fio e mostra a interação entre os diferentes elementos que constituem este cenário. A área geográfica abrangível por uma estação base é chamada de célula. Cada *host* móvel pode diretamente comunicar com uma estação base, enquanto se movimentava na área geográfica de abrangência. O processo durante o qual um *host* móvel entra em uma nova célula chama-se hand-off. Geralmente, algumas áreas de abrangência de uma célula sobrepõem a área das outras.

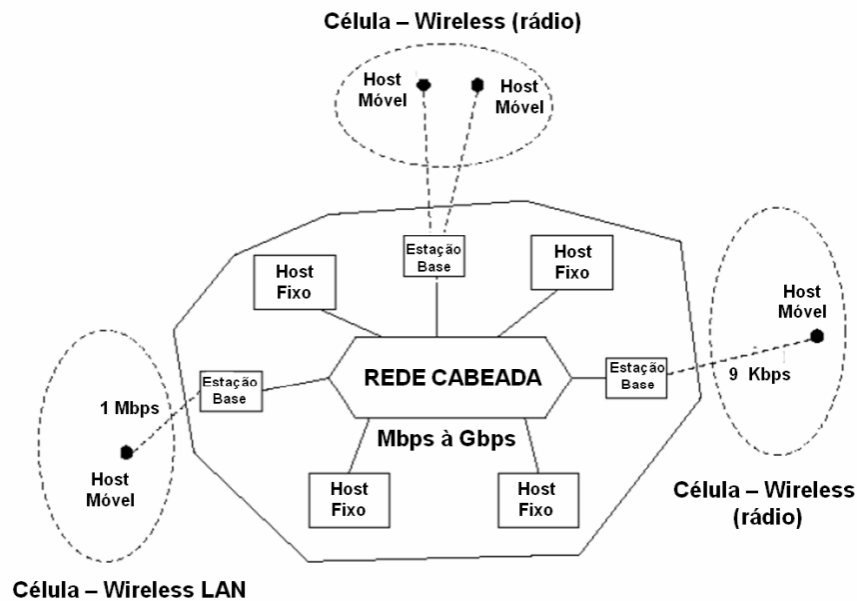


Figura 3.1: Arquitetura de rede com suporte a conexão *wireless* (PITOURA; BHARGAVA, 1994)

Enquanto em sistemas distribuídos não móveis, um *host* opera de dois modos diferentes, conectado ou desconectado. No ambiente móvel, o *host* pode operar de diversas formas, sendo um dos fatores determinantes a disponibilidade de largura de banda. O autor cita que as desconexões frequentes, sejam elas totais ou parciais, não devem ser tratadas como falhas, ou seja, um *host* móvel deve ser capaz de operar em casos de impossibilidade de conexão.

Os autores (PITOURA; BHARGAVA, 1994) ainda comentam que outro fator que pode determinar o modo de operação do dispositivo é o tempo de vida de bateria. Neste caso, o dispositivo poderia entrar em um estado de economia de energia. Outro aspecto importante diz respeito à área de abrangência da célula que pode variar amplamente, desde uma pequena sala (infravermelho), por alguns quilômetros (rádio) ou ainda muitos quilômetros (satélite).

A mobilidade também pode causar a perda ou degradação da conexão *wireless*, visto que os usuários podem mover-se entre as células. Neste contexto, o número de dispositivos em uma célula pode variar dinamicamente, por exemplo, um evento que reúna muitos usuários móveis, pode sobrecarregar a capacidade da rede *wireless*.

3.4 Desconexão

No cenário do ambiente móvel pode-se entender que um *host* móvel está conectado quando está em completo modo de operação. Por outro lado, considera-se desconectado o *host* móvel que se encontra inacessível ao restante da rede. As desconexões em uma ambiente com características de mobilidade podem ocorrer por diversas razões, por exemplo (CONCEIÇÃO; KON, 2006):

- alta variabilidade da qualidade da conexão;
- necessidade de economia de recursos (bateria) ou término da energia;
- hand-offs (pode ocorrer um salto na força do sinal quando se desconecta do ponto de acesso de sinal mais fraco e conecta-se a outro de sinal mais forte);
- interferências (interferência eletromagnética que pode ocorrer com ondas de microondas, sinais de motores ou outros equipamentos elétricos);
- sombreamento (variação de sinal causada pela obstrução de objetos tais como montanhas, prédios, outdoors, móveis ou paredes; outra forma ocorre quando entre duas estações existe uma terceira que impede a comunicação entre as duas primeiras).

Corroborando com as propriedades e problemas vistos anteriormente, é possível afirmar que o ambiente móvel é suscetível a desconexões, o que se pode considerar como uma característica inerente da computação móvel. Para (MATEUS; LOUREIRO, 1998) e (PITOURA; SAMARAS; SAMARAS, 1998) as desconexões de rede podem ser classificadas como: a) voluntárias ou forçadas; b) previsíveis ou repentinas; c) curtas ou longas.

Voluntárias: quando a desconexão é intencional, por exemplo, o usuário pode evitar o acesso à rede para diminuir custo da tarefa de comunicação, o consumo de energia ou o uso de largura de banda.

Forçadas: quando a região onde o usuário está não tem cobertura para acesso à rede;

Previsíveis: as desconexões previsíveis podem ser de natureza voluntária, em função da variação na taxa sinal ruído (SNR - Signal-to-Noise Ratio), ou ainda quanto a energia disponível na bateria (em situações quando atinge certo limiar pode fazer com que todos os recursos do ambiente passem a trabalhar com outra qualidade de serviço);

Súbitas: desconexões que ocorrem repentinamente, de forma abrupta;

Curtas ou Longas: as desconexões podem durar um pequeno período de tempo, por exemplo, quando há algum obstáculo no sinal, ou ainda ter duração longa em função, por exemplo, de falta de energia da bateria.

Conforme citado em (FORMAN; ZAHORJAN, 1994), as falhas da rede são uma grande preocupação em computação móvel. Nos projetos de aplicações para computação móvel, deve-se decidir como serão feitos os investimentos buscando amenizar as limitações do ambiente: disponibilizar mais recursos na rede, tentar prevenir desconexões, ou possibilitar aos sistemas lidar com as desconexões de maneira refinada quando possível. Porém, nem tudo pode ser mascarado para o usuário; nestes casos, boas interfaces de usuário podem ajudar fornecendo feedback sobre quais operações estão indisponíveis em função da desconexão.

Em um sistema distribuído convencional, um *host* opera em um ou dois modos: conectado ou desconectado. Já em um ambiente móvel, há mais modos de operação de um *host* móvel (PITOURA; BHARGAVA, 1994). Pode-se observar no diagrama de estados da figura 2 que os estados possíveis são: totalmente conectado, parcialmente conectado, desconectado e modo “cochilo” (doze).

Segundo (PITOURA; BHARGAVA, 1994) o grau de conexão é relativo à disponibilidade da largura de banda. Além disso, como desconexões parciais ou totais são muito frequentes, elas não deveriam ser tratadas como falhas. Pelo contrário, um *host* móvel deveria ser capaz de operar sob baixa ou sem conexão com a rede estática. Um *host* móvel opera em um modo de desconexão parcial quando a comunicação pela rede está fraca. Em adição, um *host* móvel pode alternar para modo cochilo com o objetivo de economizar energia. Em decorrência disso, a velocidade de *clock* da UCP é reduzida e nenhuma operação do usuário é executada até receber alguma mensagem externa para retornar a operação normal.

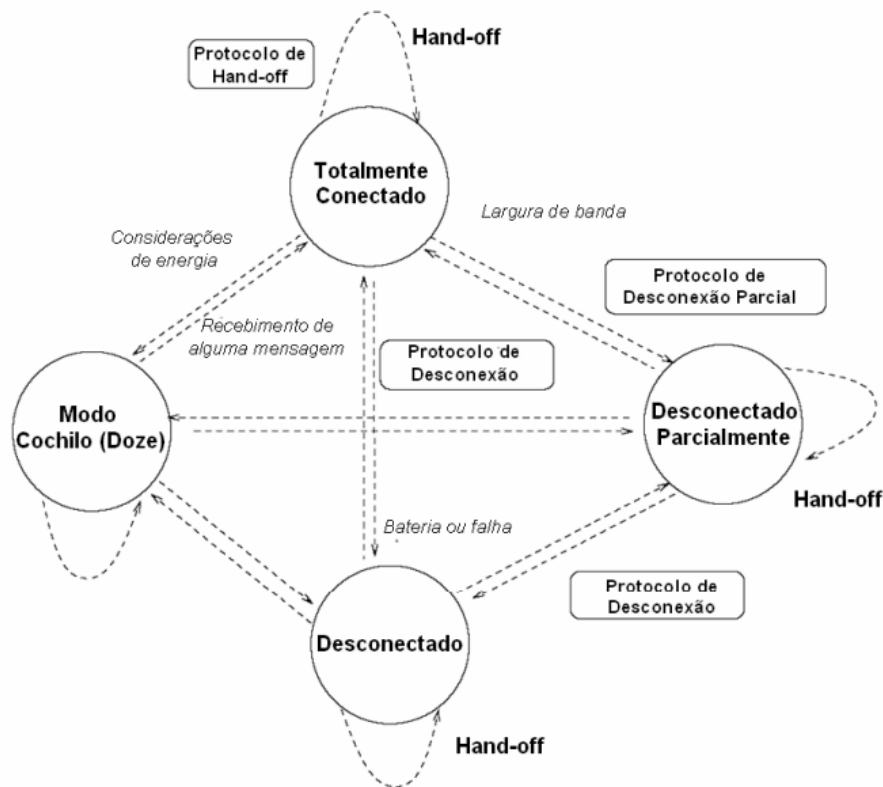


Figura 3.2: Estados de operação de um elemento móvel (PITOURA; BHARGAVA, 1994)

3.5 Computação móvel distribuída

(SATYANARAYANAN, 1996) menciona que a mobilidade intensifica a tensão entre autonomia e interdependência, que é característica de qualquer sistema distribuído, em função da relativa escassez de recursos de elementos móveis bem como sua baixa confiança. Assim, segundo o autor, a adaptação é a chave da mobilidade. Neste sentido é imprescindível que sistemas projetados para computação móvel observem eventos/recursos do sistema e a partir destes elementos façam adaptações em tempo de execução, primando pela entrega correta do serviço e alcançando assim um estado de confiabilidade. No trabalho de (KLEIN et al., 2010) é proposto um *framework* que tem como objetivo gerenciar o acesso de dispositivos móveis em cenários heterogêneos, ou seja, ambientes uma gama de diferentes tecnologias para comunicação sem fio.

O conceito de *Mobile Cloud Computing* (MCC) tem como foco tirar vantagem das nuvens computacionais para usuários de dispositivos móveis, mas também prover funcionalidades adicionais à nuvem. MCC irá ajudar a superar limitações dos dispositivos móveis principalmente no que diz respeito a poder de processamento e armazenamento de dados. E talvez ajude no

aspecto de eficiência energética já que a execução será realizada pela nuvem. MCC também parece uma solução para um mercado fragmentado de sistemas operacionais para *mobiles*. A inserção de dispositivos móveis em ambientes de nuvem proporcionaria uma maior de segurança para esses dispositivos, devido a um controle centralizado bem como uma manutenção dos softwares utilizados (KLEIN et al., 2010).

Porém, alguns desafios devem ser superados para que dispositivos móveis possam se inserir de vez no contexto de nuvens computacionais (KLEIN et al., 2010):

- MCC exige uma conectividade contínua;
- MCC exige conectividade *wireless* sob-demanda com largura de banda escalável;
- MCC requer a escolha da rede, dispendendo energia.

4 *Proposta*

O ambiente de cluster com alta disponibilidade foi montado no LaPeSD (Laboratório de Pesquisa em Sistemas Distribuídos), no departamento de Informática e Estatística localizado dentro da Universidade Federal de Santa Catarina. Este ambiente foi montado para a execução de uma aplicação real que tinha como principal objetivo realizar meta-escalonamento nesse mesmo cluster, um esquema do funcionamento do meta-escalonador pode ser visto na figura 4.1. O cluster era composto por cinco máquinas com sistema operacional Ubuntu Server 10.04, uma máquina mestre com Ubuntu 10.04 LTS interconectadas por um *switch* FastEthernet. O dispositivo móvel escolhido para prover alta disponibilidade foi um laptop com sistema operacional Ubuntu 10.04 LTS. As configurações das máquinas são as seguintes:

- **Ubuntu Server (5x):**
 - *Processador*: Intel Pentium III 733 MHz
 - *Disco Rígido*: 15 GB
 - *Memória RAM*: 256 MB

- **Ubuntu 10.04 (Nodo mestre):**
 - *Processador*: AMD Athlon XP 2400+
 - *Disco Rígido*: 80 GB
 - *Memória RAM*: 512 MB

- **Ubuntu 10.04 (Laptop):**
 - *Processador*: Core2 Duo
 - *Disco Rígido*: 320 GB
 - *Memória RAM*: 3 GB

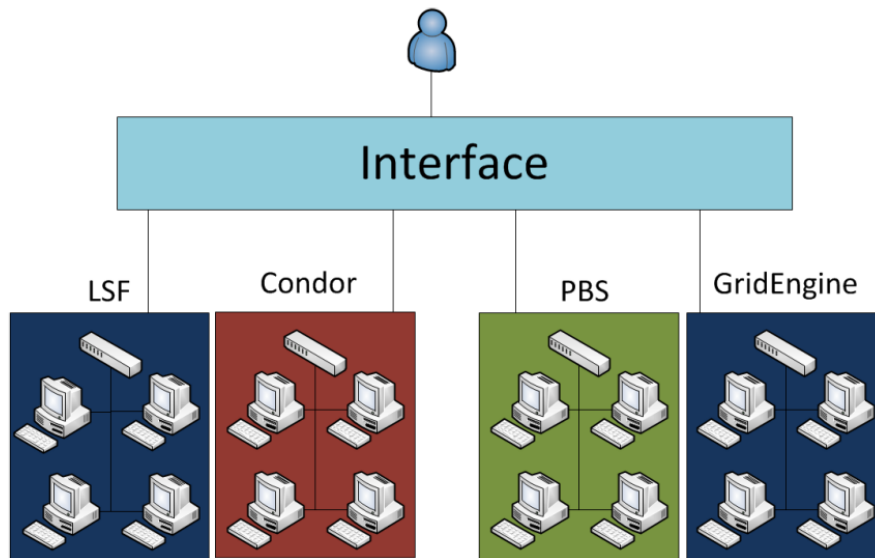


Figura 4.1: Modelo de funcionamento meta-escalonador

Para a formação do ambiente paralelo, foi escolhido o MPI (*Message Passing Interface*) com a biblioteca OpenMPI. O MPI é a especificação formal de uma API padrão para desenvolvimento de aplicações paralelas através do modelo de passagem de mensagens (MPI, 2010). Os gerenciadores de recursos escolhidos foram o *Oracle Grid Engine* e o *Condor*. O *Oracle Grid Engine* é um gerenciador de recursos distribuídos desenvolvido para distribuir as workloads dos usuários entre os recursos computacionais disponíveis (OGE, 2010). Já o *Condor* é uma ferramenta que visa desenvolver, implementar, distribuir, e avaliar mecanismos e políticas que suportam a computação de alto desempenho em grandes coleções de recursos computacionais de propriedade compartilhada.

No trabalho de (MOZER, 2004) foi proposto um ambiente de alta-disponibilidade, um cluster utilizando o escalonador OSCAR juntamente com o software HA-OSCAR para a garantia da alta-disponibilidade. Porém, essa solução faz com que o criador do ambiente obrigatoriamente utilize o escalonador OSCAR, que hoje já não possui mais suporte, se tornou obsoleto. Além do que, ambientes distribuídos hoje em dia possuem, em alguns casos, uma heterogeneidade no que diz respeito aos escalonadores utilizados. O ambiente proposto e implementado fornece a alta-disponibilidade a nível de sistema operacional, dando liberdade para utilização do(s) escalonador(es). Outra característica do ambiente foi a utilização de dispositivo(s) móvel(is) para prover a alta-disponibilidade, nos testes realizados foi utilizado um notebook. Dispositivos móveis são utilizados também como gerenciadores de processo, e o ambiente prove a portabilidade desses gerentes, ou seja, aonde estiver, o gerente tem uma posição do que está acontecendo na configuração. Essas funcionalidades unidas fornecem uma continuidade dos

serviços, diminuindo o tempo de inoperância do sistema. O software utilizado como infraestrutura de gerenciamento da alta-disponibilidade do Multi-cluster foi o Pacemaker-Heartbeat. A figura 4.2 apresenta a proposta do ambiente de alta disponibilidade móvel.

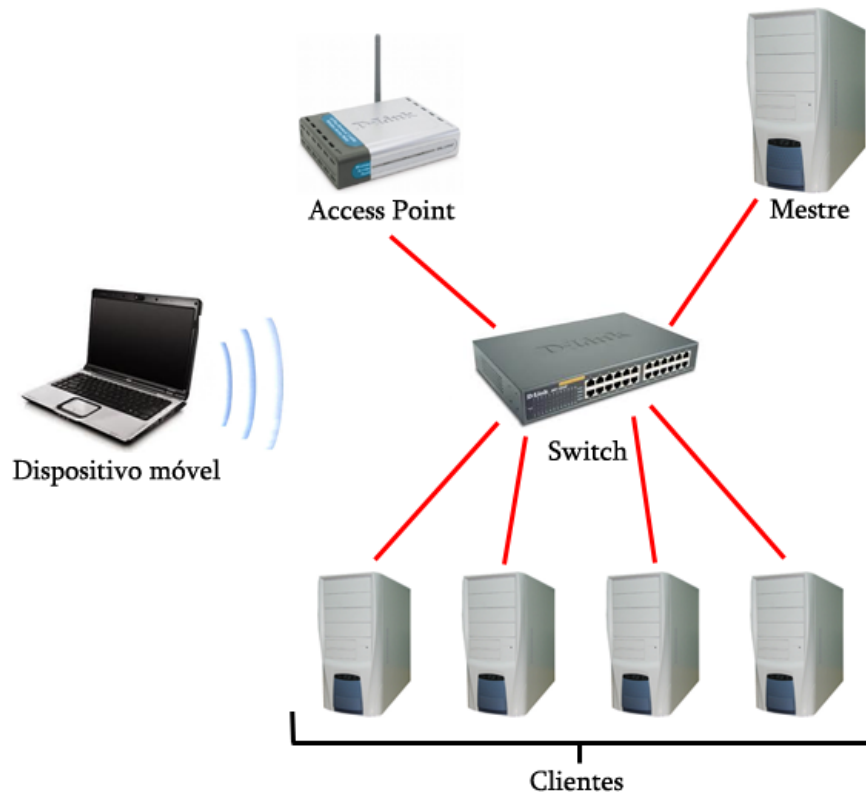
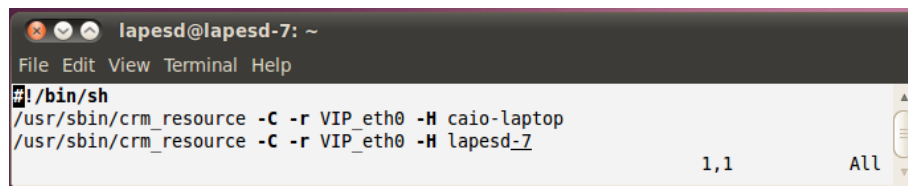


Figura 4.2: Ambiente de alta disponibilidade proposto

A dinâmica do que ocorre no ambiente após uma falha no servidor é a seguinte:

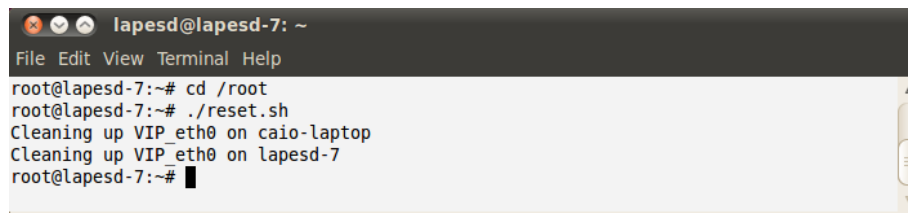
- Primeiro Ato: É reconhecida a falha pelo gerenciador de alta-disponibilidade (DGDB);
- Segundo Ato: Um dos candidatos móveis, que se comunicam com o sistema pelo *access point* e monitoram o servidor através do Heartbeat, toma o seu lugar após a falha;
- Terceiro Ato: Para efetivação do candidato como mestre do sistema, o IP virtual que é referência a para os clientes, passa a apontar para o dispositivo móvel que possui todas as configurações para se tornar mestre;
- Quarto Ato: O sistema volta a funcionar normalmente, agora com o dispositivo móvel funcionando como mestre do ambiente;
- Quinto Ato: Quando restabelecido o nó que falhou, o sistema volta a sua configuração normal.

Para que o sistema volte a funcionar de onde havia parado anteriormente, é necessário fazer um compartilhamento das pastas que contém os arquivos de estado do sistema /oracle no Oracle Grid Engine e /condor no Condor. Essa sincronia é realizada através do pacote RSYNC. Quando o servidor original é restabelecido o Pacemaker-Heartbeat executa um comando para cada nodo (candidatos e servidor), esse comando tem a função de efetuar um “clean“ na comunicação e consequentemente a sincronização dos componentes de alta-disponibilidade é realizada. Para testar o funcionamento dos comandos fiz um script “reset.sh“ com os comandos que Pacemaker-Heartbeat executa. A descrição do script e a respectiva resposta da sincronização podem ser vistas nas figuras 4.3 e 4.4.



```
lapesd@lapesd-7: ~
File Edit View Terminal Help
# /bin/sh
/usr/sbin/crm_resource -C -r VIP_eth0 -H caio-laptop
/usr/sbin/crm_resource -C -r VIP_eth0 -H lapesd-7
1,1 ALL
```

Figura 4.3: Comandos a serem executados pelo Pacemaker-heartbeat para a sincronização dos candidatos com o servidor



```
lapesd@lapesd-7: ~
File Edit View Terminal Help
root@lapesd-7:~# cd /root
root@lapesd-7:~# ./reset.sh
Cleaning up VIP_eth0 on caio-laptop
Cleaning up VIP_eth0 on lapesd-7
root@lapesd-7:~# █
```

Figura 4.4: Resposta positiva da sincronização

O teste foi realizado para o simples entendimento do funcionamento do mecanismo de sincronização realizado pelo pacote Pacemaker. Obviamente para uma sincronização efetiva com todos os candidatos devem ser executados 'N' comandos, um para cada candidato, com o ultimo parametro alterado com o respectivo nome do candidato.

5 *Estudos de caso*

Para testar o ambiente, foi simulado um defeito desligando um dos servidores. Em um primeiro momento, desligou-se o servidor primário, fazendo com que o secundário assumisse seu papel após aproximadamente 5 segundos. Este tempo se deve ao fato do heartbeat ficar executando ping a cada 5 segundos entre ambos os servidores. Neste momento, todas as interfaces de rede do servidor secundário ficaram ativas. A partir de então, o servidor secundário passou a agir como primário, resolvendo agora as requisições dos clientes, sem que os mesmos soubessem da mudança de mestre. O inverso também foi testado: no momento em que se desligou o servidor secundário, o primário assumiu seu papel novamente, demonstrando assim, a eficiência do sistema. A sequência de figuras abaixo (5.1 a 5.6) mostra o ambiente testado, sendo visualizado pelo DGDB.

Na figura 5.1 é visto o sistema rodando como deveria, ou seja, o servidor primário atendendo as requisições dos clientes. A interface gráfica do DGDB mostra 3 “caixas”, as caixas indicadas pelos nomes *laped-7* e *caio-laptop* são respectivamente o servidor primário e o candidato a assumir a posição de servidor caso esse falhe. Já a caixa indicada por Group (group1) indica quem é o servidor no momento e sob qual IP o servidor atual responde, nesse caso o servidor *laped-7* com o IP 150.162.64.134 (IP virtual). O terminal ao lado mostra o ping no IP virtual, pela latência dos pacotes é fácil verificar que o servidor está em uma rede cabiada.

Já a figura 5.2 mostra o sistema inoperante, ao tentar pingar no IP virtual o servidor não é encontrado.

A figura 5.3 mostra o momento que o laptop assume a posição de servidor do sistema, isso pode ser visto na caixa *Group* agora o servidor é o laptop com o mesmo IP do servidor anterior (150.162.64.134), ou seja, para os clientes o servidor só ficou inoperante por alguns segundos e voltou normalmente ao funcionamento. Verificando o ping no IP, observando mais especificamente o parametro time, é possível verificar que o servidor está na rede sem fio. Nesse momento, a caixa indicada por *laped-7* mostra o servidor tentando se reconectar enquanto o sistema permanece operante.

Em 5.4 o servidor cabiado está em estado de reconexão a rede. Logo em seguida, em 5.5 o servidor retorna a rede porém não assume sua posição original, a resincronização é necessária. E finalmente em 5.6 após a resincronização feita, pelo Pacemaker-Heartbeat, o sistema retoma sua configuração inicial.

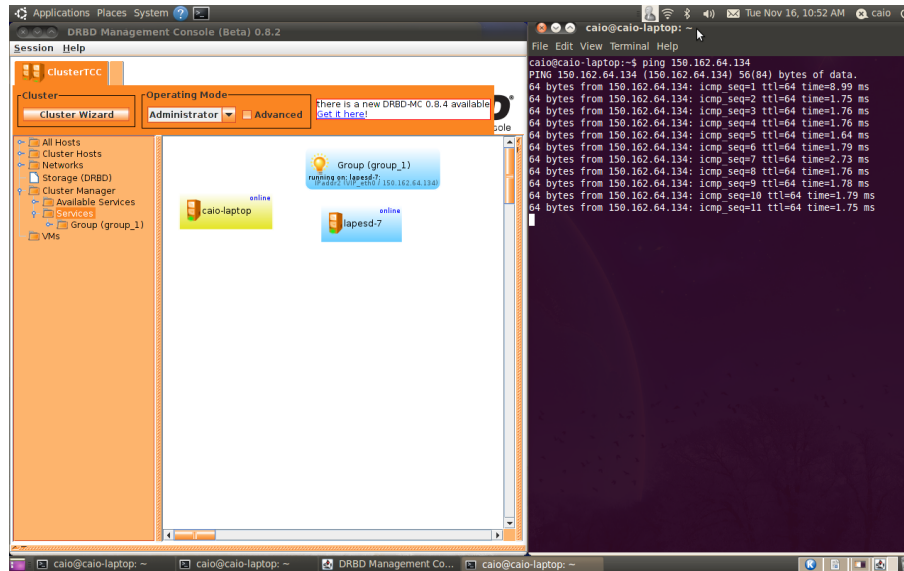


Figura 5.1: Ambiente funcionando normalmente

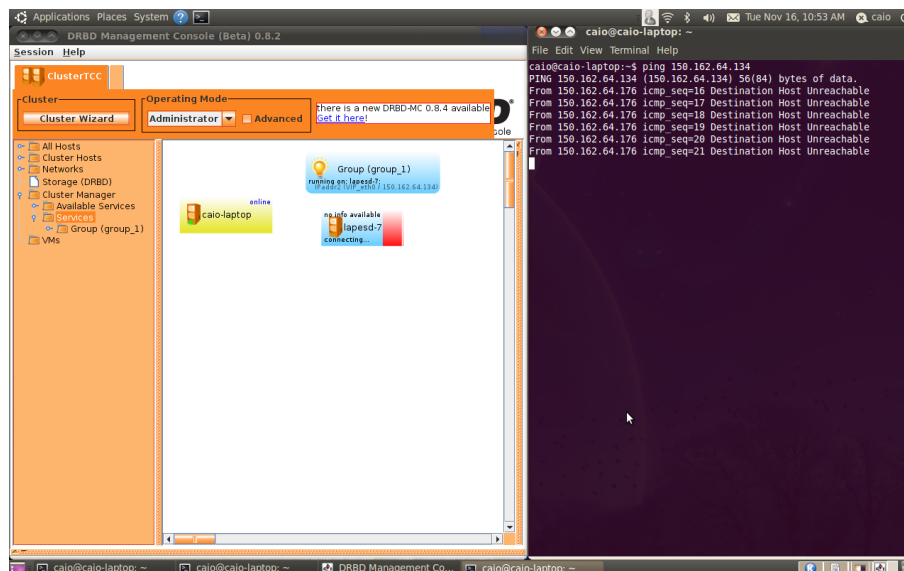


Figura 5.2: Desconexão do servidor primário, sistema inoperante

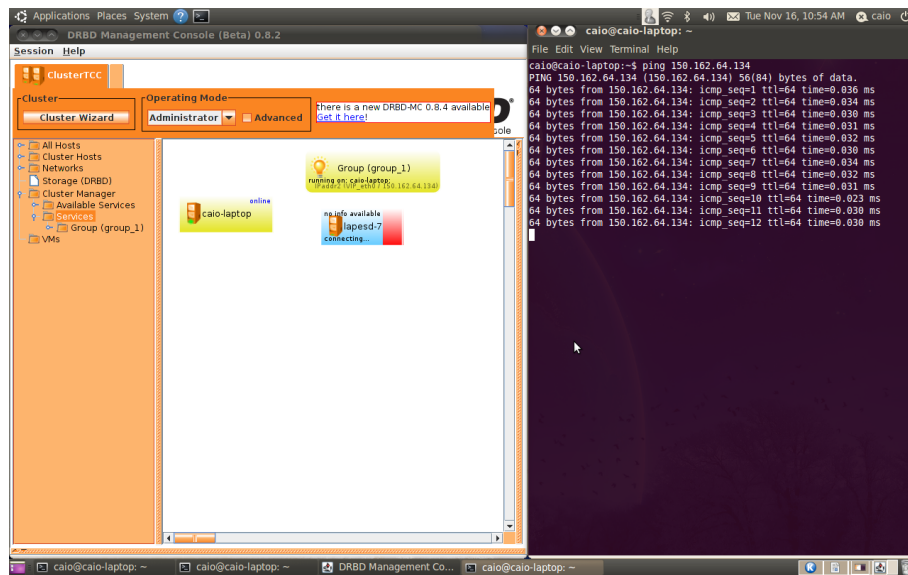


Figura 5.3: Dispositivo móvel assume o papel de mestre

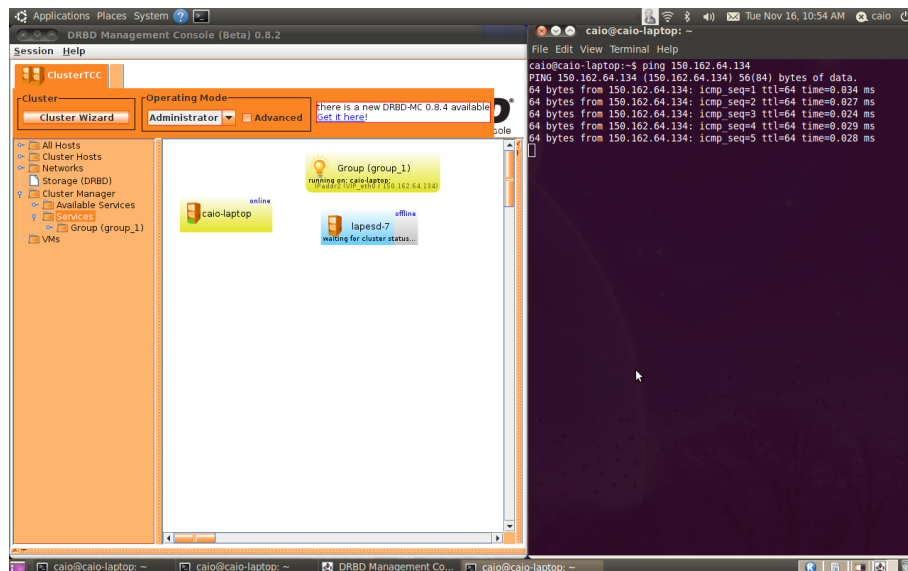


Figura 5.4: Reconexão do servidor primário

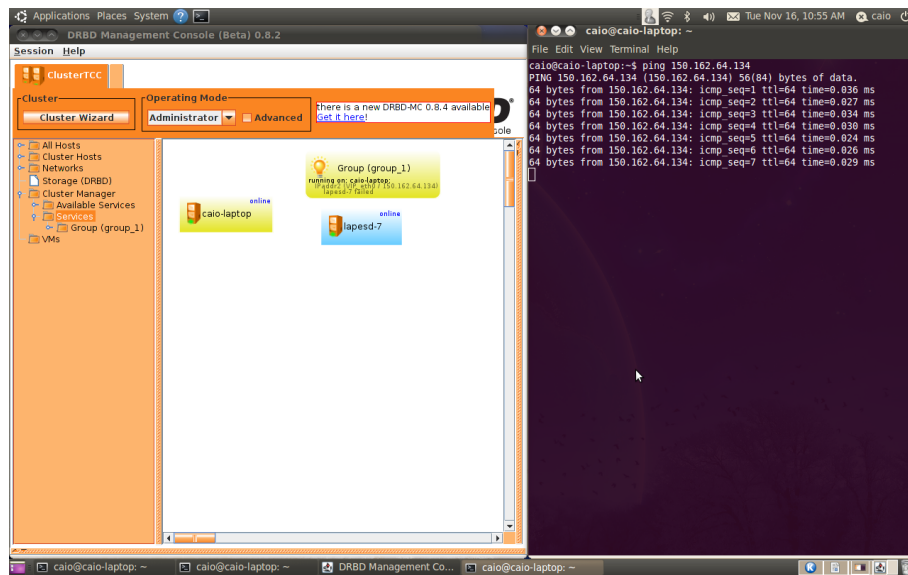


Figura 5.5: Servidor primário reconectado

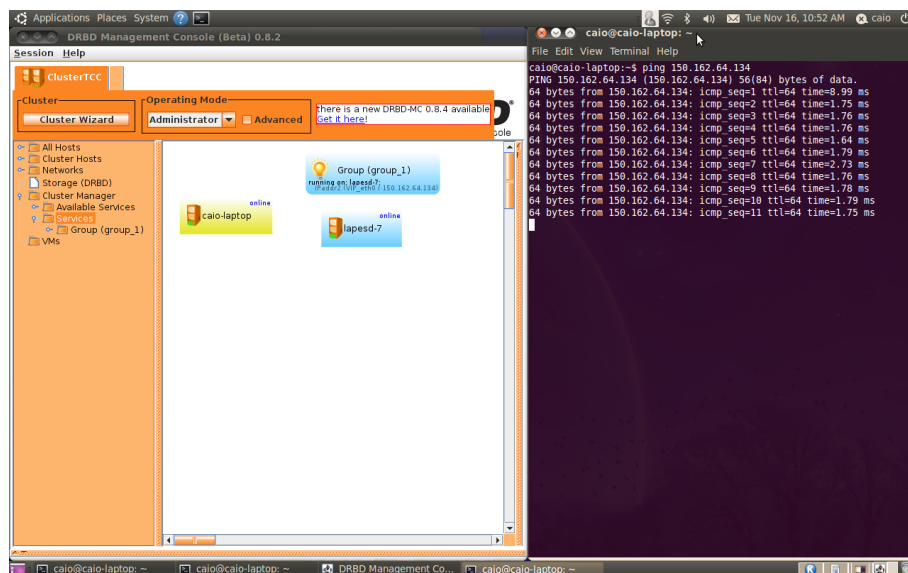


Figura 5.6: Sistema volta ao funcionamento normal

6 *Conclusão e Trabalhos futuros*

À medida que novas tecnologias surgem na área de sistemas distribuídos, quebrando formatos tradicionais de comunicação e interação, mais desafios são impostos para implementação de sistemas computacionais que utilizem eficientemente os benefícios advindos de tais tecnologias. De fato, ao observarmos as características inerentes da computação móvel e dos sistemas distribuídos, verifica-se que a integração das tecnologias é potencial. Entretanto, é conhecido que ambas as tecnologias possuem limitações que não podem ser ignoradas nos projetos e desenvolvimento de sistemas.

Com o intuito de abordar os aspectos relacionados às falhas inerentes destes ambientes, muitas vezes, não considerados nos projetos de sistemas computacionais, a abordagem proposta neste trabalho caracterizou as possíveis falhas na integração de sistemas distribuídos com a computação móvel. Os estudos de caso mostraram que é possível realizar a integração dessas duas vertentes computacionais “driblando” os possíveis problemas que possam aparecer.

Neste trabalho de pesquisa foi proposto um ambiente de alta-disponibilidade através de dispositivos móveis. O ambiente proposto conseguiu atingir os quesitos de alta disponibilidades, além de fornecer uma independência quanto ao escalonador a ser utilizado. Sob o ponto de vista do dispositivo móvel, a inserção deste no ambiente foi satisfatória, ao passo que ele cumpriu com as tarefas desejadas usufruindo de suas características de mobilidade e portabilidade.

Como trabalhos futuros, objetiva-se otimizar a utilização dos dispositivos móveis em ambientes distribuídos, com o foco em nuvens computacionais, sob os aspectos de performance em hardware, eficiência energética e gerenciamento de recursos.

Referências Bibliográficas

ABBOTT, B. et al. Einstein@ Home search for periodic gravitational waves in LIGO S4 data. *Physical Review D*, APS, v. 79, n. 2, p. 22001, 2009.

ANDERSON, D. BOINC: A system for public-resource computing and storage. In: IEEE COMPUTER SOCIETY. *proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. [S.l.], 2004. p. 4–10.

ANDERSON, D. et al. SETI@home: an experiment in public-resource computing. *Communications of the ACM*, ACM, v. 45, n. 11, p. 56–61, 2002.

BOINC. *Documentation*. 2010. Acessado em: agosto de 2010. Disponível em: <<http://boinc-ssl.berkeley.edu/>>.

BORGES, V.; DANTAS, M. Uma abordagem de submissão e monitoração de múltiplas tarefas para ambientes de grade computacional utilizando dispositivos móveis. *XXXIII SEMISH*, p. 403–418, 2006.

BRUNEO, D. et al. Communication paradigms for mobile grid users. In: IEEE. *Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on*. [S.l.], 2003. p. 669–676.

BUYYA, R. *Grid Computing Info Center*. 2010. Acessado em: julho de 2010. Disponível em: <<http://www.gridcomputing.com>>.

CHEN, G.; KOTZ, D. *A survey of context-aware mobile computing research*. [S.l.], 2000.

CLARKE, B.; HUMPHREY, M. Beyond the 'Device as Portal': Meeting the Requirements of Wireless and Mobile Devices in the Legion Grid Computing System. In: CITESEER. *2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (associated with IPDPS 2002), Ft. Lauderdale*. [S.l.], 2002.

CLUSTERLAB. *Pacemaker*. 2010. Acessado em: setembro de 2010. Disponível em: <<http://www.clusterlabs.org/wiki/Pacemaker>>.

CONCEIÇÃO, A. da; KON, F. O uso de pares de pacotes para monitoração da taxa de transmissão e da capacidade de vazão em redes IEEE 802.11. In: *24th Brazilian Symposium on Computer Networks (SBRC), Curitiba-PR, Brazil*. [S.l.: s.n.], 2006.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Sistemas distribuídos-conceitos e projeto*. [S.l.: s.n.], 2007.

DANTAS, M. *Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais*. [S.l.: s.n.], 2005.

- DISTRIBUTED.NET. *Documentation*. 2010. Acessado em: julho de 2010. Disponível em: <http://www.distributed.net>.
- DRDB. *DRDB*. 2010. Acessado em: outubro de 2010. Disponível em: <http://www.drbd.org/>.
- FORMAN, G.; ZAHORJAN, J. The challenges of mobile computing. *COMPUTER*, Published by the IEEE Computer Society, p. 38–47, 1994.
- GOLDCHLEGER, A. *Integrade: Um sistema de middleware para computação em grade oportunista*. Tese (Doutorado) — Citeseer, 2004.
- HAAS, H. comes. 2006. Acessado em: agosto de 2010. Disponível em: http://www.rougtype.com/archives/2006/03/here_comes_haas.php.
- HOCHSTETLER, S. et al. *Linux Clustering with CSM and GPFS*. [S.l.]: IBM, International Technical Support Organization, 2004.
- HUGHES-FENCHEL, G. A flexible clustered approach to high availability. In: PUBLISHED BY THE IEEE COMPUTER SOCIETY. *ftcs*. [S.l.], 1997. p. 314.
- IEEE. *Task force on cluster computing - High Availability*. 2010. Acessado em: agosto de 2010. Disponível em: <http://www.ieeetfcc.org/high-availability.html>.
- IMIELINSKI, T.; VISWANATHAN, S.; BADRINATH, B. Energy efficient indexing on air. In: ACM. *Proceedings of the 1994 ACM SIGMOD international conference on Management of data*. [S.l.], 1994. p. 25–36.
- KLEIN, A. et al. Access Schemes for Mobile Cloud Computing. In: IEEE. *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. [S.l.], 2010. p. 387–392.
- LITKE, A.; SKOUTAS, D.; VARVARIGOU, T. Mobile grid computing: Changes and challenges of resource management in a mobile grid environment. In: *Proceedings of International Conference on Practical Aspects of Knowledge Management*. [S.l.: s.n.], 2004.
- MATEUS, G.; LOUREIRO, A. *Introdução à computação móvel*. [S.l.]: DCC/IM, COPPE/UFRJ, 1998.
- MOZER, S. *ALTA DISPONIBILIDADE: UM ESTUDO DE CASO EM UM AMBIENTE DE IMAGEM ÚNICA DE PRODUÇÃO*. Tese (Doutorado), 2004.
- MPI. *Message Passing Interface*. 2010. Acessado em: setembro de 2010. Disponível em: <http://www.mcs.anl.gov/research/projects/mpi/>.
- OGE. *Oracle Grid Engine*. 2010. Acessado em: setembro de 2010. Disponível em: <http://www.oracle.com/us/products/tools%20-/oracle-grid-engine-075549.html>.
- PARK, S.; KO, Y.; KIM, J. Disconnected operation service in mobile grid computing. *Service-Oriented Computing-ICSOC 2003*, Springer, p. 499–513, 2003.
- PERRY, M. et al. Dealing with mobility: understanding access anytime, anywhere. *ACM Transactions on Computer-Human Interaction (TOCHI)*, ACM, v. 8, n. 4, p. 323–347, 2001.
- PFISTER, G. *In search of clusters*. [S.l.]: Prentice-Hall Englewood Cliffs, NJ, 1998.

- PHAN, T.; HUANG, L.; DULAN, C. Challenge: integrating mobile wireless devices into the computational grid. In: ACM. *Proceedings of the 8th annual international conference on Mobile computing and networking*. [S.l.], 2002. p. 278.
- PITOURA, E.; BHARGAVA, B. Building information systems for mobile environments. In: ACM. *Proceedings of the third international conference on Information and knowledge management*. [S.l.], 1994. p. 371–378.
- PITOURA, E.; SAMARAS, G.; SAMARAS, G. *Data management for mobile computing*. [S.l.]: Citeseer, 1998.
- RSYNC. *RSYNC*. 2010. Acessado em: setembro de 2010. Disponível em: <<http://en.wikipedia.org/wiki/Rsync>>.
- SAMIMI, F.; MCKINLEY, P.; SADIJADI, S. Mobile service clouds: A self-managing infrastructure for autonomic mobile computing services. *Self-Managed Networks, Systems, and Services*, Springer, p. 130–141, 2006.
- SATYANARAYANAN, M. Fundamental challenges in mobile computing. In: ACM. *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*. [S.l.], 1996. p. 1–7.
- SATYANARAYANAN, M. Pervasive computing: Vision and challenges. *IEEE Personal communications*, v. 8, n. 4, p. 10–17, 2001.
- SHIRAZI, B.; KAVI, K.; HURSON, A. *Scheduling and load balancing in parallel and distributed systems*. [S.l.]: IEEE Computer Society Press Los Alamitos, CA, USA, 1995.
- VMWARE. 2010. Acessado em: agosto de 2010. Disponível em: <<http://www.vmware.com/>>.
- WANG, L. et al. Scientific cloud computing: Early definition and experience. In: IEEE. *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*. [S.l.], 2008. p. 825–830.
- WEB2.0. 2010. Acessado em: agosto de 2010. Disponível em: <http://en.wikipedia.org/wiki/Web_2.0>.