

L. W. Junges and M. Ferrandin

Study on time series databases in the context of IoT

Abstract—¹ The purpose of this research project is to understand the operation of time series databases and to compare the main commercial time series databases with a relational database. An IoT scenario test is also performed with a time series database.

Keywords— Time Series Databases, Internet of Things.

I. INTRODUÇÃO

Estamos vivendo a terceira fase da internet, chamada de Internet das coisas (IoT), onde muitos itens usados no dia-a-dia se conectam à internet a fim de facilitar nossas vidas. Os grandes avanços na área de sistemas embarcados e na microeletrônica vistos nos últimos anos têm possibilitado cada vez mais a expansão dessa nova fase da internet, que a cada dia conta com mais aplicações. Deve-se estudar a tecnologia referida a fundo para seja possível aplicá-la com eficácia nas suas mais variadas formas.

Existe um tipo de banco de dados, que é clássico no meio da informática, e que muitos pensam ser o único, ou no mínimo o melhor tipo de bancos de dados existente para se armazenar qualquer tipo de dado. O nome do tipo de banco de dados referido é o banco de dados relacional. Entretanto, quando se fala em dispositivos conectados a internet que enviam dados periodicamente e que a cada envio se tem no mínimo três tipos de informação (uma de tempo, uma de localização e outra de valor), o banco de dados relacional pode não ser a melhor solução. Nesse contexto, paradigmas como o NoSQL e os bancos de dados de séries temporais (TSDB's, Time Series Databases) podem trazer ganhos quanto a performance, escalabilidade e requisitos de memória.

Em geral, os TSDB's, embora antigos, são pouco conhecidos, até mesmo por profissionais da área de computação e informática. Dessa forma, quando alguém está interessado em aplicar um TSDB surgem muitas dúvidas sobre o seu funcionamento e qual banco de dados melhor satisfaz os requisitos pretendidos.

O objetivo desse projeto de pesquisa é compreender o funcionamento dos bancos de dados de séries temporais e comparar os principais TSDB's comerciais a um banco de dados relacional. Também é realizado um teste em cenário de IoT com um banco de dados de série temporal.

O presente artigo está estruturado da seguinte maneira: A Seção II contém a revisão bibliográfica e trabalhos relacionados. A Seção III contém os materiais e métodos utilizados. A Seção IV contém a implementação e resultados obtidos. Por fim, a Seção V consiste das conclusões obtidas.

II. REVISÃO DA LITERATURA

Os bancos de dados relacionais são apontados como pouco escaláveis e isso acontece porque respeitam rigorosamente as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), o que acarreta num tempo maior de consulta e inserção de dados [1]. Segundo [2], é verdade que a escalabilidade pode ser alcançada em banco de dados com o modelo relacional de banco de dados, entretanto, pode ser uma tarefa de custo elevado e uma tarefa bastante complexa.

Os principais problemas encontrados com a utilização do Modelo Relacional estão principalmente na dificuldade de conciliar o tipo de modelo com a demanda da escalabilidade que está cada vez mais frequente [3]. NoSQL pode ser uma alternativa real para banco de dados que necessitam de uma alta escalabilidade, desempenho e que podem possuir tolerância a falhas [2].

A utilização de bancos de dados não relacionais é apontada como uma das soluções para resolver o problema de escalabilidade.

III. MATERIAIS E MÉTODOS

A fim de comparar a eficiência dos TSDB's, pesquisou-se os principais bancos de dados comerciais desse tipo, e foi encontrado InfluxDB, Cassandra e OpenTSDB dentre as mais recomendadas [4]. Então comparou-se esses com o banco de dados MySQL, que é o banco de dados (relacional) mais usado na atualidade. Foi coletada a base de dados "Gas sensor array temperature modulation Data Set" em [5] e os dados foram inseridos no InfluxDB, Cassandra, OpenTSDB e MySQL. Foi utilizado o software R para ajustar a importação dos dados à formatação padrão de cada sistema de banco de dados. A figura 1 apresenta dados ajustados à formatação padrão do InfluxDB.

1

L.W. Junges, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, Brasil, lucaswilliamjunges@gmail.com

M. Ferrandin, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, Brasil, mauri.ferrandin@ufsc.br

IV. RESULTADOS E DISCUSSÃO

Os tempos de consulta e inserção de dados foram coletados e seguem na tabela 1.

Nome	Tempo de consulta (s)	Tempo de inserção (s)	Tipo
MySQL	2.98	0.023	Relacional
InfluxDB	2.27	0.015	Série temporal
Cassandra	2.61	0.013	Série temporal
OpenTSDB	2.54	0.023	Série temporal

Tabela 1 - Comparativo entre os bancos de dados utilizados

Após analisar o tempo de consulta dos TSDB's utilizados, foi selecionado aquele com o menor tempo de consulta, InfluxDB, e aplicado este num projeto que o autor havia feito no passado com um banco de dados MySQL. O projeto referido consiste em coletar os dados referentes à alimentação de estudantes no almoço de uma escola a fim de monitorar a alimentação dos mesmos.

A figura 1 mostra o esquema do banco de dados relacional que estava sendo empregado.

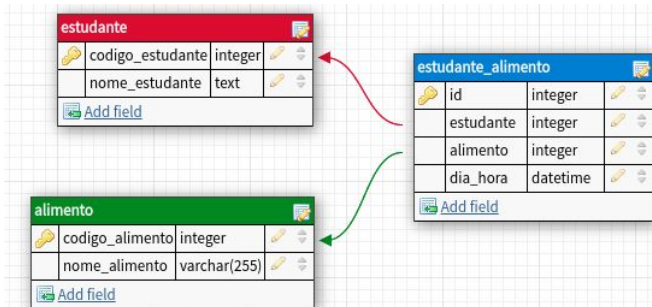


Figura 1 - esquema do banco de dados

Com a implementação do banco de dados de séries temporais, InfluxDB, os comandos de inserção ficaram como na figura 3.

```

1 insert codigo_estudante=1, nome_estudante=Lucas, dia=20/06/2017, massa=20, codigo_alimento=2 nome_alimento="arroz" 1497959272;
2 insert codigo_estudante=1, nome_estudante=Lucas, dia=20/06/2017, massa=30, codigo_alimento=3 nome_alimento="feijao" 1497959272;
3 insert codigo_estudante=1, nome_estudante=Lucas, dia=20/06/2017, massa=40, codigo_alimento=4 nome_alimento="cenoura" 1497959272;
4 insert codigo_estudante=1, nome_estudante=Lucas, dia=20/06/2017, massa=13, codigo_alimento=5 nome_alimento="alface" 1497959272;
5 insert codigo_estudante=1, nome_estudante=Lucas, dia=20/06/2017, massa=15, codigo_alimento=6 nome_alimento="brocolini" 1497959272;
6 insert codigo_estudante=1, nome_estudante=Igor, dia=20/06/2017, massa=13, codigo_alimento=7 nome_alimento="frango" 1497959272;
7 insert codigo_estudante=2, nome_estudante=Igor, dia=20/06/2017, massa=56, codigo_alimento=8 nome_alimento="couve" 1497959272;
8 insert codigo_estudante=2, nome_estudante=Igor, dia=20/06/2017, massa=13, codigo_alimento=9 nome_alimento="beterraba" 1497959272;
9 insert codigo_estudante=2, nome_estudante=Igor, dia=20/06/2017, massa=8, codigo_alimento=2 nome_alimento="arroz" 1497959272;
10 insert codigo_estudante=2, nome_estudante=Igor, dia=20/06/2017, massa=150, codigo_alimento=2 nome_alimento="arroz" 1497959272;
11 insert codigo_estudante=2, nome_estudante=Igor, dia=20/06/2017, massa=48, codigo_alimento=2 nome_alimento="arroz" 1497959272;
12 insert codigo_estudante=1, nome_estudante=Lucas, dia=21/06/2017, massa=37, codigo_alimento=5 nome_alimento="alface" 1497959272;
13 insert codigo_estudante=1, nome_estudante=Lucas, dia=21/06/2017, massa=29, codigo_alimento=7 nome_alimento="frango" 1497959272;
14 insert codigo_estudante=1, nome_estudante=Lucas, dia=21/06/2017, massa=39, codigo_alimento=2 nome_alimento="arroz" 1497959272;
15 insert codigo_estudante=1, nome_estudante=Igor, dia=21/06/2017, massa=16, codigo_alimento=8 nome_alimento="couve" 1497959272;
16 insert codigo_estudante=1, nome_estudante=Lucas, dia=21/06/2017, massa=68, codigo_alimento=2 nome_alimento="arroz" 1497959272;
17 insert codigo_estudante=1, nome_estudante=Lucas, dia=21/06/2017, massa=87, codigo_alimento=2 nome_alimento="arroz" 1497959272;
18 insert codigo_estudante=1, nome_estudante=Igor, dia=21/06/2017, massa=16, codigo_alimento=8 nome_alimento="couve" 1497959272;
19 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=8, codigo_alimento=2 nome_alimento="arroz" 1497959272;
20 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=46, codigo_alimento=4 nome_alimento="cenoura" 1497959272;
21 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=12, codigo_alimento=2 nome_alimento="arroz" 1497959272;
22 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=16, codigo_alimento=2 nome_alimento="macarrao" 1497959272;
23 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=44, codigo_alimento=4 nome_alimento="cenoura" 1497959272;
24 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=87, codigo_alimento=8 nome_alimento="couve" 1497959272;
25 insert codigo_estudante=2, nome_estudante=Igor, dia=21/06/2017, massa=21, codigo_alimento=6 nome_alimento="brocolini" 1497959272;
26 insert codigo_estudante=1, nome_estudante=Lucas, dia=22/06/2017, massa=8, codigo_alimento=2 nome_alimento="arroz" 1497959272;
27 insert codigo_estudante=1, nome_estudante=Lucas, dia=22/06/2017, massa=43, codigo_alimento=9 nome_alimento="beterraba" 1497959272;

```

Figura 3 - Comandos de inserção

Os dados específicos do projeto, como alimento e massa do alimento foram obtidos através de balanças conectadas a um Raspberry Pi 3, que insere os dados no InfluxDB através de um cliente sendo executado num programa em Python.

Com o uso do banco de dados de séries temporais nesse projeto, houve um ganho de escalabilidade, o que resulta em inserções e consultas mais rápidas.

V. CONCLUSÃO

Este trabalho apresentou um comparativo de performance entre TSDB's e um banco de dados relacional e fez uma implementação real de um banco de dados de séries temporais.

Para comparar a performance, foi selecionado um dataset e inseridos os dados do mesmo em quatro diferentes bancos de dados (três de séries temporais em um relacional). Os sistemas gerenciadores de bancos de dados foram escolhidos com base na recomendação em [6].

Os resultados mostram que os TSDB's apresentam uma performance superior do que os bancos de dados relacionais ao tipo de aplicação utilizado.

REFERÊNCIAS

[1] MOHAMED, Mohamed A. Relational vs. NoSQL Databases: A Survey. International Journal of Computer and Information Technology, Khartoum, 3 maio 2014. Disponível em: https://www.researchgate.net/profile/Mohamed_Mohamed69/publication/263272704_Relational_Vs_NoSQL_databases_A_survey/links/0b7d53a495312ad22000000/Relational-Vs-NoSQL-databases-A-survey.pdf.

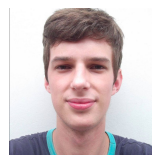
[2] Toth, Renato Molina. Abordagem NoSQL – uma real alternativa. Sorocaba: 2012.

[3] Kuhn, Igor Antonio. ANÁLISE DE SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS PARA ARMAZENAMENTO DE DADOS CLIMÁTICOS. Cuiabá: Universidade Federal do Mato Grosso, 2016.

[4] BADER, Andreas. Comparison of Time Series Databases. Universitätsstraße 38, Stuttgart, p. 37, 13 jan. 2016. Disponível em: https://elib.uni-stuttgart.de/bitstream/11682/9793/1/Diplomarbeit_Comparison_of_Time_Series_Databases.pdf.

[5] Machine Learning Repository, ["https://archive.ics.uci.edu/ml/index.php"](https://archive.ics.uci.edu/ml/index.php).

[6] Namiot, D. (2015). Time Series Databases. Moscow: Lomonosov Moscow State University, p.135.



Lucas William Junges Graduando em Engenharia de Controle e Automação pela Universidade Federal de Santa Catarina (UFSC). <http://lattes.cnpq.br/2866720264812264>.



Mauri Ferrandin received his PhD from Pontifical University of Paraná (PUCPR) and MSc degree from the Federal University of Santa Catarina (UFSC). His main research focuses on classification algorithms to deal with hierarchical classification scenarios, data mining applied to IoT and Cloud environments.

<http://lattes.cnpq.br/4132238007682356>.