Análise da Influência da Modelagem e Formato de Dados no Desempenho de *Data Warehouse* Baseado em Hadoop-Hive

Beatriz Fragnan P. de Oliveira[1], Aline S. Oliveira Valente[1], Marcio Victorino[2], Edward Ribeiro, Maristela Holanda[1]

[1]Dep. de Ciência da Computação – Universidade de Brasilia (UNB)
Caixa Postal 4.466 – 70.910-900 – Brasilia – DF – Brazil

[2]Fac. de Ciência da Informação – UNB – Brasilia – DF – Brazil

{beatrizfra, asoliveirav}@gmail.com,

{mcvictorino, edwardribeiro, mholanda}@unb.br

Abstract. *The advancement of data warehousing in cloud environments has grown. In this context, there is no defined model or pattern on how to handle data. Therefore, this work aims to present a comparative analysis of the performance in the use of the Hive platform with the snowflake model and the totally denormalized. The used data for this analysis are those of the Brazilian Army Open Data in the Google Cloud environment. The analysis is performed for different quantities of lines in Hive, for a cluster configuration scene and for two types of storage of tables. Lastly, using the Parquet format on the tables, a performance four times superior was achieved to that of the CSV format.*

Resumo. *O desenvolvimento de* data warehouse *em ambientes em nuvem tem crescido. A modelagem de dados neste ambiente ainda não tem um padrão definido. Assim, esse artigo tem como objetivo apresentar uma análise comparativa de desempenho do uso da plataforma Hive no modelo floco de neve e totalmente desnormalizado. Os dados utilizados para análise são os dados abertos do Exército Brasileiro no ambiente Google Cloud. As análises são realizadas para diferentes quantidades de linhas no Hive, para um cenário de configuração do* cluster *e para dois tipos de armazenamento das tabelas. Por fim, utilizando o formato Parquet nas tabelas, obteve-se um desempenho mais de quatro vezes superior ao do formato CSV.*

1. Introdução

Em *Data Warehouses* (DW) é comum ter uma arquitetura centralizada para facilitar a análise dos dados. Contudo, atualmente os volumes de dados de análise estão atingindo tamanhos críticos [Jacobs 2009], trazendo a necessidade de diferentes soluções para o processamento e gerenciamento de *Big Data*. Desta forma, é possível ter DW distribuídos e com diferentes estruturas [Mohanty et al. 2013]. Neste contexto, as empresas têm desenvolvido um novo ecossistema de plataformas que utilize não só os dados dos DW tradicionais implementados, mas também arquiteturas de DW híbridas arquitetadas, um *Big Data Warehouse* [Mohanty et al. 2013]. Os *Big Data Warehouses* diferem substancialmente dos DW tradicionais uma vez que o seu esquema deve ser baseado em modelos lógicos novos e mais flexíveis que os modelos relacionais [Di Tria et al. 2014].

Apache Hadoop é uma estrutura de código aberto, escrito em Java, que permite o processamento distribuído de um grande conjunto de dados em *cluster* de computadores. Hive é uma infraestrutura de DW *open source*, construída sobre o Hadoop, que facilita as consultas e a gestão de grandes volumes de dados armazenados em ambiente distribuído [Cassavia et al. 2014][Sandoval 2015]. O processamento no Hive pode depender de fatores como: parâmetros de configuração do software, consulta realizada, volume de dados em questão e modelagem dos dados, dentre outras características. No Hive, não existe um formato único no qual os dados devem ser armazenados, sendo que este suporta arquivos de texto com separação de atributos através de vírgulas, *comma-separated values* (CSV) ou tabulação, *tab-separated values* (TSV), ou outros formatos como o Parquet e *Optimized Row Columnar* (ORC) . Parquet é um formato de código aberto disponível para projetos no ecossistema Hadoop, que foi projetado para um formato de armazenamento colunar [Weintraub et al. 2021], [Rodrigues et al. 2019], diferente de arquivos baseado em linha, como o CSV. O Hive tem sido utilizado para implementação de arquiteturas de *Big Data Warehouse*.

A escolha do modelo dos dados assim como do formato dos arquivos a serem lidos podem influenciar no desempenho de determinadas consultas. Dessa forma, o presente trabalho tem como objetivo analisar o desempenho do Hive, diante de dois modelos de dados distintos. No primeiro, os dados são colocados em uma única tabela desnormalizada e no segundo, os mesmos são inseridos em um modelo dimensional de floco de neve. Além disso, também analisou-se a influência do formato dos dados das tabelas no tempo de execução das consultas, fazendo-se uma comparação entre os formatos CSV e parquet. Dessa forma, foram feitas simulações com diferentes cargas de trabalho e então os resultados foram analisados.

Este trabalho está estruturado da seguinte forma: Na Seção 2 foram apresentados os trabalhos relacionados; na Seção 3, o estudo de caso; na Seção 4 são apresentados os resultados; na Seção 5, a discussão dos resultados e na Seção 6, a conclusão e trabalhos futuros.

2. Trabalhos Relacionados

Na literatura existem diferentes trabalhos com o tema de DW e Hive para *Big Data*, como [Costa et al. 2017], [Weintraub et al. 2021], [Santos and Costa 2016] e [Rodrigues et al. 2019].

[Costa et al. 2017] explora o impacto da modelagem de dados nas consultas, comparando o modelo estrela com uma tabela única desnormalizada para o *Star Schema Benchmark* (SSB). [Weintraub et al. 2021] apresentou um estudo de otimizações para consultas em ambiente em nuvem, com aplicações em um *Data Lake*. [Santos and Costa 2016] mostrou que o modelo de dados pode mudar de acordo com os requisitos do DW e ressaltou a importância da escolha do modelo mais adequado para cada necessidade. Já [Rodrigues et al. 2019] avalia diferentes ferramentas de processamento de *Big Data* como Drill, HAWQ, Hive, Impala, Presto e Spark usando o *Transaction Processing Council* (TPC-H) *benchmark*.

Diferentemente dos trabalhos apresentados, este artigo, além de comparar o desempenho de dois modelos de dados no Hive, analisa também a influência do formato dos dados das tabelas no tempo de execução das consultas, fazendo também uma comparação

entre os formatos CSV e Parquet.

3. Estudo de Caso

Nesta seção, são apresentados a infraestrutura empregada para a realização das consultas, o conjunto de dados, algumas características das consultas realizadas, os cenários de teste e o método aplicado para obtenção dos resultados.

3.1. Conjunto de Dados e Consultas

Esse trabalho usou um conjunto de dados abertos do serviço militar brasileiro, com registros referentes ao alistamento militar. O objetivo é avaliar o desempenho de diferentes modelagens de dados, comparando o modelo dimensional floco de neve com um modelo totalmente desnormalizado, chamado neste artigo de tabela única. Além disso, é feita uma comparação de desempenho das consultas, variando a forma como os dados são armazenados, Parquet e CSV. O DW em questão é formado por 1 tabela FATO e 23 dimensões, porém as consultas realizadas neste trabalho abordaram apenas a FATO e mais três dimensões. As características das tabelas envolvidas nas consultas deste trabalho são apresentadas na Tabela 1.

Tabela 1. Características das tabelas utilizadas

Nome da Tabela	Quantidade de Colunas
FATO_EVENTO_CIDADAO	11
DIM_GEOGRAFIA	86
DIM_CIDADAO	48
DIM_EVENTO_SERMIL	4

Foram realizadas duas consultas diferentes, denominadas Q1 e Q2. A consulta Q1, consiste em uma função de agregação, "*SELECT COUNT(*) FROM FATO_EVENTO_CIDADAO;*". A consulta Q2, de maneira abstrata é: "*SELECT – 8 colunas COUNT – (distinct 1 atributo) FROM – tabela FATO e inner join WHERE – 3 operações join – (FATO, DIM_GEOGRAFIA, DIM_CIDADAO e DIM_EVENTO_SERMIL) GROUP BY – 8 atributos ORDER BY – 3 atributos*".

Os parâmetros de Q2 acima, se referem à quantidade de colunas, além disso, as 8 colunas do *SELECT* são oriundas das 4 tabelas que sofreram junção.

3.2. Metodologia

Para avaliar o desempenho das duas modelagens propostas, foi construído um *cluster* no Google Cloud com cinco nós. A métrica selecionada para mensurar o desempenho foi o tempo de execução das consultas. As consultas foram realizadas cinco vezes e então calculou-se a média das medições.

Os dados utilizados são referentes aos anos de 2011 a 2018. Os mesmos foram inseridos de maneira incremental ano a ano, aumentando assim a quantidade de linhas a ser consultada. O modelo dimensional de floco de neve é composto por 1 tabela FATO, com 11 colunas e 23 tabelas de dimensões, com quantidade de colunas que variam de 4 a 86. Os dados das tabelas são relacionados ao cidadão alistado e contêm informações como número de registro, idade, dados biométricos, cidade, se deseja servir, nível de escolaridade, se possui ocupação, entre outros. Para facilitar a construção dos gráficos, cada

ano/grupo de anos tem um nome específico de referência. Na Tabela 2, pode-se conferir os anos analisados, seus nomes de referências e as respectivas quantidades de linhas, como é possível observar, para cada período foram adicionadas diferentes quantidades de linhas, chegando a um total de 35.073.178 de linhas.

Tabela 2. Linhas analisadas por período

Anos	Referência	Quantidade de Linhas da Fato	Linhas Retornadas	
			Q1	Q2
2011	A1	3.645.403	1	28
2011 a 2012	A2	9.458.748	1	80
2011 a 2013	A3	14.621.794	1	138
2011 a 2014	A4	18.206.124	1	185
2011 a 2015	A5	22.191.384	1	226
2011 a 2016	A6	26.059.430	1	277
2011 a 2017	A7	30.515.025	1	2660
2011 a 2018	A8	35.073.178	1	7666

3.3. Cenários de Teste

Neste trabalho, foram realizadas simulações em um ambiente no Google Cloud, que consiste em um Hadoop *cluster* com 5 nós, configurados como 1 mestre (YARN *Resource Manager*) e 4 trabalhadores (YARN *Node Managers*). A configuração do nó mestre foi denominada pelo nome: Otimizado para computação C2 *Standard* 4, com 4 CPU com 16 GB de memória e um disco principal de 500 GB do tipo HDD. Por fim, cada nó trabalhador neste *cluster* possui 2 CPU com 7,5 GB de memória e um disco principal com 300 GB de disco tipo HDD.

Foram comparados dois cenários. Ambos utilizaram o *cluster* de 5 nós. No cenário um (C1), comparou-se o modelo dimensional e a tabela única desnormalizada e os dados foram armazenados como CSV. Já o cenário dois (C2) visou construir um cenário semelhante ao C1, porém utilizando tabelas armazenadas como Parquet.

4. Resultados

O primeiro cenário avaliado foi o C1 utilizando os anos de A1 a A8. Os resultados são mostrados de acordo com cada consulta realizada. No gráfico da consulta Q1, mostrado na Figura 1, o desempenho da tabela única foi tão superior ao do modelo dimensional, que não apareceu visivelmente no gráfico. Também pôde-se observar que o tempo de consulta não variou muito à medida que o número de linhas foi crescendo, ou seja, ao longo dos anos. Conforme pode-se verificar na Figura 2, o desempenho da tabela única desnormalizada foi ligeiramente superior em quase todas as simulações, sendo ultrapassado somente em A7.

Para o cenário C2 foram analisados os anos A1 a A8. Os resultados são mostrados de acordo com cada consulta realizada. De acordo com a Figura 3, em C2, os tempos de execução da consulta Q1 para o modelo dimensional e a tabela única ficaram bem similares para a maioria dos anos, com exceção apenas de A1. Esse comportamento foi bem diferente do resultado em C1, onde a tabela única apresentou um resultado muito superior ao modelo dimensional. Assim, é possível dizer que o formato Parquet das tabelas favoreceu o esquema do modelo dimensional para esse conjunto de dados.

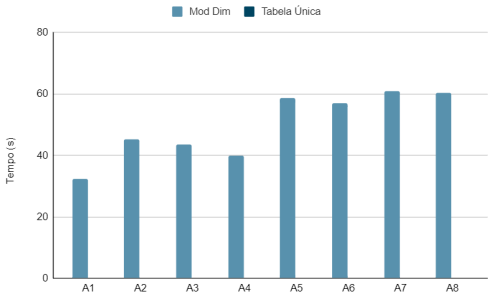

Figura 1. Resultado da consulta Q1 em C1

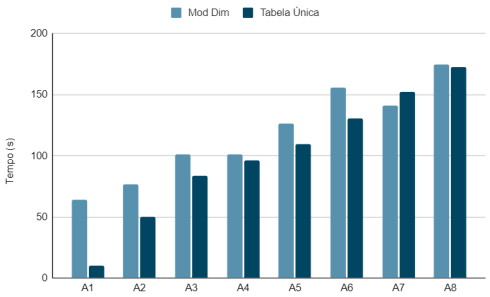

Figura 2. Resultado da consulta Q2 em C1

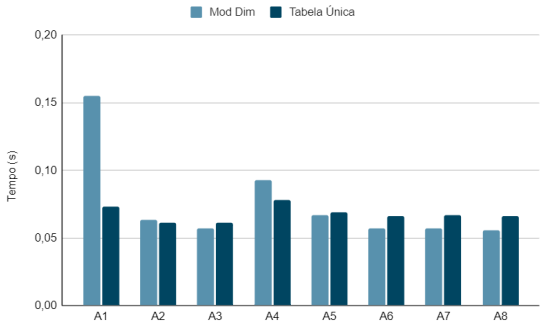

Figura 3. Resultado da consulta Q1 em C2

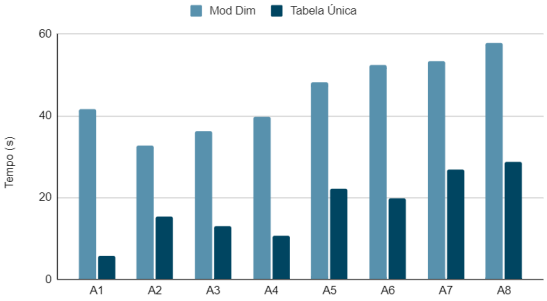

Figura 4. Resultado da consulta Q2 em C2

Conforme pode-se verificar na Figura 4, o desempenho da tabela única desnormalizada foi melhor em todas as simulações, diferente do observado no cenário C1. Comparado a C1, o perfil do comportamento é semelhante, porém as diferenças de tempo são bem maiores. Para esta consulta, o armazenamento das tabelas em formato Parquet desfavoreceu o desempenho do modelo dimensional.

De uma maneira geral, analisando-se o tempo de execução de todas as consultas em ambos os cenários, verifica-se que o desempenho de C2 foi mais de quatro vezes superior ao de C1. Ao analisar o desempenho da tabela única isoladamente, o resultado de C2 é melhor ainda, atingindo um desempenho $5,6$ vezes melhor que o de C1, em termos de tempo de execução da consulta.

5. Discussão dos Resultados

A partir das simulações realizadas é possível inferir que, para esse conjunto de dados, o formato CSV favorece o modelo da tabela única desnormalizada e o formato Parquet equipara os resultados dos dois modelos. Em complemento, para a consulta Q2, o formato Parquet favorece o modelo da tabela única desnormalizada. Assim, de uma maneira geral, o formato Parquet melhora o desempenho das consultas em $4,2$ vezes, o que é algo bem expressivo. Diante dos diferentes resultados obtidos para as duas consultas nos dois cenários, conclui-se que não existe um melhor cenário de uma forma geral e sim para um grupo de consultas específicas. Dessa forma, é necessário agrupar as consultas semelhantes e adequar o contexto a elas. Por exemplo, viu-se que consultas simples com funções de agregação tem melhor desempenho na tabela única, quando este é armazenado em CSV,

porém quando é armazenado em Parquet, o esquema do modelo dimensional é favorecido. Também foi visto que para uma consulta com diferentes operações como *joins, group by* e *order by*, como a Q2, o formato Parquet contribui para a melhora do desempenho da tabela única, diante do modelo dimensional.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma análise de desempenho entre consultas em DW e com isso foi possível observar como o formato Parquet tornou todas as consultas mais rápidas, melhorando o desempenho das mesmas expressivamente. Como trabalhos futuros, há a possibilidade de fazer uma comparação, considerando duas abordagens diferentes, uma com armazenamento em disco do Hadoop e outra com armazenamento no Spark. Além disso, também pode-se realizar uma análise do impacto do cache nas consultas executadas sucessivas vezes. Por fim, pode ser realizado em complemento ao trabalho realizado, algum tipo de partição nas tabelas, analisando sua influência no desempenho das consultas para os dois modelos em questão, o dimensional floco de neve e o desnormalizado.

Referências

Cassavia, N., Dicosta, P., Masciari, E., and Saccà, D. (2014). Data preparation for tourist data big data warehousing. In *International Conference on Data Management Technologies and Applications*, pages 419–426. INSTICC, SciTePress.

Costa, E., Costa, C., and Santos, M. Y. (2017). Efficient big data modelling and organization for hadoop hive-based data warehouses. In Themistocleous, M. and Morabito, V., editors, *European, Mediterranean and Middle Eastern Conference on Information Systems*, pages 3–16. Springer International Publishing.

Di Tria, F., Lefons, E., and Tangorra, F. (2014). Design process for big data warehouses. In *International Conference on Data Science and Advanced Analytics (DSAA)*, pages 512–518.

Jacobs, A. (2009). The pathologies of big data. *Comm. of the ACM*, 52(8):36–44.

Mohanty, S., Jagadeesh, M., and Srivatsa, H. (2013). *Big data Imperatives: Enterprise Big Data Warehouse, BI Implementations and Analytics.* Apress, 1st edition.

Rodrigues, M., Santos, M. Y., and Bernardino, J. (2019). Big data processing tools: An experimental performance evaluation. *WIREs Data Mining and Knowledge Discovery*, 9(2):e1297.

Sandoval, L. J. (2015). Design of business intelligence applications using big data technology. In *2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)*, pages 1–6.

Santos, M. Y. and Costa, C. (2016). Data warehousing in big data: From multidimensional to tabular data models. In *Ninth International C* Conference on Computer Science Software Engineering*, pages 51–60. ACM.

Weintraub, G., Gudes, E., and Dolev, S. (2021). Needle in a haystack queries in cloud data lakes. In *EDBT/ICDT Workshops*. CEUR-WS.org.