ESTUDO DE CASO DE PROCESSAMENTO DE ETL EM PLATAFORMA BIG DATA

Maurício Tavares Santana Lins e Leandro Mendes Ferreira
Everis do Brasil – Divisão de Tecnologia – Laboratório de Big Data
Av. das Nações Unidas, 14171, São Paulo - SP, 04794-000, Brasil

RESUMO

O processamento realizado entre sistemas legados e transacionais para os sistemas de tomada de decisões gerenciais, atualmente possui suas tecnologias amplamente consolidadas nas empresas. Adicionalmente, as metodologias e as práticas utilizadas neste tipo de processamento são conceitos bem desenvolvidos no mercado. O aumento de volume de dados gerados pelas mídias sociais, aplicativos móveis, sensores, entre outros, alavancou o surgimento de novas tecnologias que viabilizassem o processamento tradicional com este novo e imenso volume de dados, permitindo também o cruzamento com os dados já conhecidos, gerando vantagens estratégicas para as empresas. Com o aumento no volume de dados, aumentou-se também a exigência para atualização das tecnologias tradicionais. Destaca-se que a velocidade não atinge o que se espera na maioria dos casos, e com isto eleva-se muito o custo com aumento da infraestrutura e licenciamento de *software*. Em contrapartida, as novas tecnologias, intituladas como *Big Data*, possuem um poder computacional que permitem o processamento rápido de grandes volumes de dados, com elevados níveis de complexidade de processamento, e com um baixo custo, por se tratarem de tecnologias *open source*. O objetivo geral deste artigo é apresentar um estudo de caso de ganhos reais e altamente relevantes com as tecnologias de *Big Data*.

PALAVRAS-CHAVE

Extract Transform Load, Apache Hadoop, Apache Spark, Data Warehouse

1. INTRODUÇÃO

De acordo com Grus (2016), vive-se em um mundo com grande volume de dados. Dados estes que se fazem necessários para o enriquecimento das análises de negócio das organizações ou até mesmo para trazer novos *insights* sobre o comportamento de clientes e do próprio negócio. O autor conclui que a partir destes dados estão as respostas para as inúmeras questões, as quais nunca foram levantadas.

Quando se fala de processamento de dados, pode-se elencar uma série de tecnologias capazes de realizar esta tarefa. Estas tecnologias, atualmente, possuem suas arquiteturas bem definidas, principalmente quando focamos nos sistemas de informações gerenciais (Bazzotti e Garcia, 2005) ou o *Business Intelligence (BI)*.

O BI permite que as organizações obtenham vantagens competitivas criando conhecimento de suas próprias informações, analisando descritivamente seus dados, organizando os dados em *Data Warehouse (DWs)*, que por sua vez são providos por Sistemas Gerenciadores de Bancos de Dados (SGDBs). Os SGBDs tradicionais de mercados possuem problemas de escalabilidade, pois para seus fornecedores atenderem as demandas das organizações, disponibilizam *appliances*. Estes são arquiteturas que possuem *hardware* e *software* customizados para obterem eficiência, seja de armazenamento ou processamento, porém estas estruturas possuem alto custo de implantação, manutenção e licenciamento (Ferreira, 2016).

Outra característica dos sistemas de BI são as ferramentas utilizadas para processamento de transformações e regras de negócio chamadas de *Extract Transform Load* (ETL) as quais realizam a integração dos dados entre os sistemas legados e um DW central. Assim como os SGBDs, estas ferramentas possuem problemas de escalabilidade e custo elevado.

Atualmente pode-se contar com soluções desenvolvidas para este ambiente de grande volume de dados que possuem uma arquitetura para processamento e armazenamento distribuído disponíveis de forma gratuita e aberta. Pode-se citar como uma das principais ferramentas para processamento e armazenamento de grande volume de dados o *Apache Hadoop* (Achari, 2015).

Hadoop é uma composição de *softwares* que atuam como um *framework* para permitir o processamento de grandes volumes de dados, podendo ser utilizado poucos computadores ou até milhares de computadores, os quais trabalham em paralelo e compartilham recursos entre si. Destaca-se que o *framework Hadoop* tem como características principais: a alta disponibilidade e a tolerância a falhas (Hadoop, 2015).

O desenvolvimento deste artigo é baseado na forma de prova de conceito POC (acrônimo em inglês de *Proof of Concept*), onde o objetivo foi desenvolver com tecnologias do *framework Hadoop* o processo de integração de dados de sistemas legados que originalmente utilizava tecnologias tradicionais como SGBDs relacionais e ferramentas de processamento ETL. Este trabalho foi implementado em uma empresa multinacional bancária de grande porte.

A ferramenta de processamento ETL utilizada no processo original é o *Informatica Powercenter*[1] e para comparação na prova de conceito utilizou-se a distribuição de ferramentas do *framework Hadoop* da empresa *Cloudera*[2], na versão *open source*, com o objetivo de comparar e avaliar o tempo de desenvolvimento, a performance do processamento, o custo relacionado a licenciamento e o da composição da arquitetura.

Na seção 2 descreve-se a arquitetura tradicional de BI, na seção 3 descrevem-se algumas das principais tecnologias que compõem o *framework Hadoop*, na seção 4 é apresentada uma comparação de arquitetura tradicional de processamento de ETL com a arquitetura de processamento e integração de dados sobre ferramentas de *Big Data*, que são partes da prova de conceito desenvolvida neste trabalho. Por fim na seção 5 apresentam-se os resultados obtidos e uma análise sobre a performance e o desempenho de cada arquitetura.

2. ARQUITETURA TRADICIONAL DE APLICAÇÕES DE BUSINESS INTELLIGENCE

A proposta tradicional de arquitetura de BI foi desenvolvida originalmente por Bill Inmon no final da década de 1970, e os seus conceitos foram difundidos pela IBM através do artigo *"An architecture for a business information system"* [3] onde as principais definições sobre a construção de um DW foi apresentada na época e continua sendo implementadas com poucas alterações até os dias atuais. Ralph Kimball aperfeiçoou o modelo de Inmon introduzindo também o conceito de pequenos DWs, conhecidos como *Data Marts* (DMs) no início dos anos de 1990 (KEMMP, 2012).

Vale mencionar que Kimball *et al.* (2013) e Inmon (2005) definem as seguintes camadas na arquitetura de um sistema de BI:

***Data Source* (DSs):** São fontes de dados que podem ser arquivos sequenciais, bancos de dados relacionais de qualquer natureza, sistemas transacionais ou quaisquer outras fontes de dados.

Extract, Transform and Load (ETL): Processo que extrai dados dos DSs e realiza transformações, que podem ser uma agregação de dados, limpeza dos dados, realizando uma adaptação nos dados para que os mesmos se adequem com a próxima camada.

***Stanging Area*:** é uma área temporária para armazenar os dados antes das transformações, utilizada para que o processo ETL não consuma diretamente a fonte de dados de origem. Possui os dados brutos que são acessados por desenvolvedores.

Operational Data Store (ODS): é uma área de integração, contém os dados das fontes de dados, porém já transformados e com algumas validações básicas. Por exemplo, se uma empresa possui dois sistemas distintos com informações de produtos, na camada de ODS, os dados destes ambos sistemas são armazenados juntos, onde são transformados e padronizados para que sejam armazenados em um único local.

Visualização de Dados: É uma camada composta por ferramentas que apresentam os dados de forma multidimensional ou OLAP (*On-line Analytical Processing*). Aplicações que possibilitam a construção de *Dashboards* (painéis gerenciais), que apresentam os indicadores por meio de gráficos, tabelas com marcadores de performance.

As etapas de construção de um sistema de BI são demonstradas graficamente na Figura 1:

[1] https://www.informatica.com/br/products/data-integration/powercenter.html
[2] http://www.cloudera.com/
[3] Disponível em http://ieeexplore.ieee.org/document/5387658/

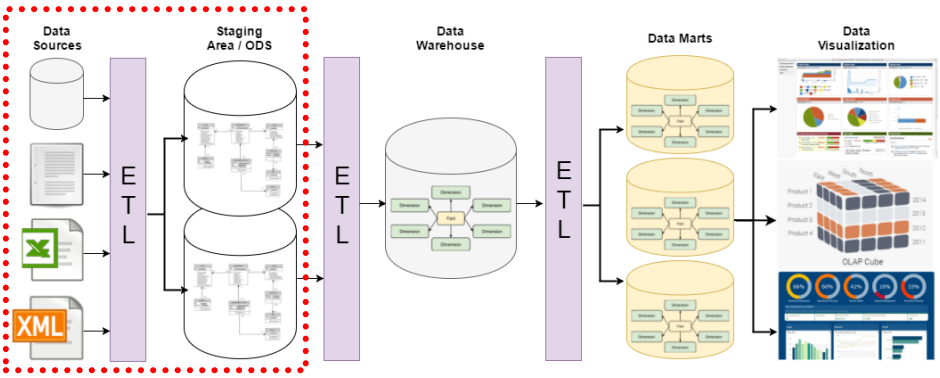

Figura 1. Modelo tradicional de construção de aplicações BI – Fonte: Adaptado de Ferreira (2015)

Observa-se que a Figura 1 apresenta todas as fases do ciclo completo da construção de um sistema de BI. Vale ressaltar que este trabalho tem como foco a primeira camada que está destacada, a qual faz a relação entre DSs e ODS.

3. APLICAÇÕES E TECNOLOGIAS BIG DATA

Beyer (2011) define aplicações de *Big Data* como um conjunto de tecnologias para processamento de dados que são baseados em Volume, Velocidade e Variedade. Entende-se assim que tecnologias de *Big Data* tem auto poder de processamento de grandes volumes de dados, nas mais variadas formas com alta velocidade de performance. Dentre as tecnologias de *Big Data* disponíveis hoje, pode-se destacar a plataforma *Hadoop*.

Segundo White (2012), *Hadoop* é mais conhecido pelo *MapReduce* e por seu sistema de arquivos distribuídos (o HDFS, *Hadoop Distributed FileSystem*). O termo também é usado para uma família de projetos que estão sob o ecossistema de infraestrutura para a computação distribuída, processamento em larga escala e alta disponibilidade. Vale ressaltar dentre os projetos que pertencem a família *Hadoop* foram utilizadas para a prova de conceito as seguintes ferramentas:

- ***MapReduce*:** Modelo de processamento de dados distribuídos, foi desenvolvido para processar de forma paralela em múltiplos nós de *clusters* composto por computadores comuns.
- ***HDFS*:** Sistema de arquivos distribuído que é executado em grandes *clusters* em máquinas comuns.
- ***Hive*:** Ferramenta que trabalha no formato de um armazém de dados distribuído. Gerencia o armazenamento de dados no HDFS e fornece uma linguagem de consulta baseada em SQL para busca de dados.
- ***Impala*:** Ferramenta para consulta de dados em linguagem SQL, sob o HDFS, *HBase* e *Hive*.
- ***Spark*:** É o padrão aberto e flexível para processamento distribuído de dados em memória que permite análises em *batch*, em tempo real e análises avançadas na plataforma sob a plataforma *Hadoop*.
- ***Parquet:*** É uma biblioteca de agrupamento e compressão para maior eficiência na organização dos dados no formato colunar (Parquet, 2016).

4. COMPARAÇÃO ENTRE ARQUITETURA TRADICIONAL DE PROCESSAMENTO DE ETL E ARQUITETURA DE BIG DATA PROPOSTA PARA PROVA DE CONCEITO

A prova de conceito foi desenvolvida a partir de um problema real de uma instituição bancária multinacional que utilizava uma arquitetura tradicional de processamento de ETL para construção de sua camada de ODS que é consumida por diversos sistemas de inteligência de negócios da instituição. Devido ao aumento de volume de dados e o custo de licenciamento de software para atualizar a arquitetura, a instituição decidiu buscar alternativas de processamento mais eficientes e com custos reduzidos.

4.1 Arquitetura Utilizada pela Instituição

Para a construção da camada de ODS eram realizadas as seguintes etapas de processamento: Tratamento dos dados originais dos sistemas *Mainframe*, validação dos seus tipos de dados, verificação de duplicidade, validações de domínios, homogeneização de dados, verificação de integridade referencial e disponibilização das bases de dados integradas na camada de ODS para os sistemas consumidores subsequentes.

Este processamento era efetuado em ambiente produtivo utilizando tecnologias tradicionais para processamento de ETL, neste caso utilizando a plataforma *Informatica Powercenter*, cujo fluxo do processo é apresentado na Figura 2.

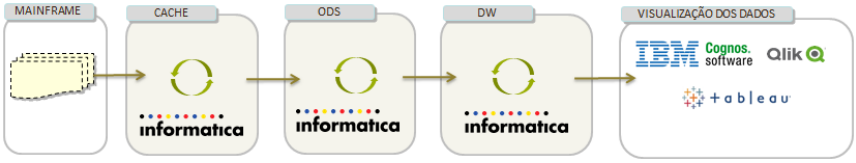

Figura 2. Fluxo do processamento realizado no modelo tradicional

No fluxo apresentado na FIGURA 2 destacam-se as etapas para o processamento dos dados que são: as camadas de *ODS* e *DW* e posteriormente a camada de Visualização de Dados, onde se encontram as ferramentas utilizadas para consumir os dados disponibilizados.

Adicionalmente destaca-se o processamento da camada chamada de *cache*, que é um processo onde os dados mais utilizados de clientes são armazenados em memória para que os processamentos posteriores possam ser executados de forma paralela com maior performance.

4.2 Proposta de Arquitetura Big Data para Integração de Dados

Como a prova de conceito visava gerar um processamento de grande volume de dados no menor intervalo de tempo possível optou-se em utilizar a ferramenta *Spark*, que possui seu processamento distribuído em memória. Adicionalmente a ferramenta *Spark* e é resiliente no caso de uma possível falha no processamento. Utilizou-se o HDFS para a camada de persistência de dados, a ferramenta *Hive* para estruturação dos dados e consulta primária, além da ferramenta *Impala* para consultas *ad-hoc* de dados. Na Figura 3 é apresentado o fluxo do processamento sob a arquitetura proposta.

Figura 3. Fluxo do processamento com o uso das tecnologias de *Big Data*

Conforme apresentado na Figura 3 desenvolveu-se um processamento com uma abordagem diferente da arquitetura que era utilizada pela instituição. O processamento da arquitetura proposta ocorre em memória, que tecnologicamente é mais eficiente, e para a persistência do dado processado usou-se o *HDFS*.

Vale mencionar que se realizou outra mudança no processamento efetuado, onde na nova arquitetura foi retirada a camada de preparação de *cache*, pois todo o processamento em *Spark* é realizado em paralelo e em memória, sendo assim, não se necessita, criar uma camada para possibilitar o paralelismo.

No desenvolvimento deste trabalho foi utilizada a linguagem de programação *Python 2.7 sob* o *Spark*. O processo de desenvolvimento da aplicação é representado na Figura 4:

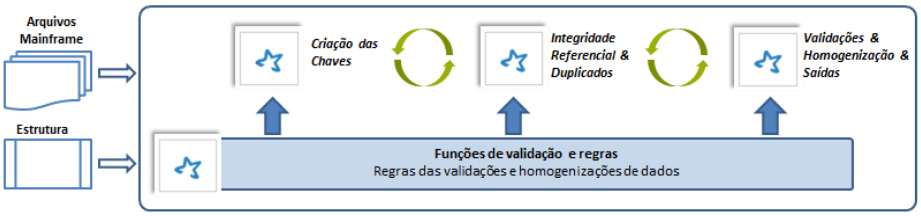

Figura 4. Fluxo do processamento desenvolvido com Spark

Foram desenvolvidas funções para realizar todos os tipos de tratamentos de dados que permitiram de forma simples que o sistema se adequasse a qualquer tipo de dados, tendo como entrada os dados e um arquivo de *metadados* representando sua estrutura, conforme FIGURA 4.

Para a camada de visualização de dados foi realizado um trabalho de preparação e adequação para que os mesmos retornassem no tempo desejável quando consultados por ferramentas de relatórios ou pelas análises de descoberta de dados ou para consultas *ad-hoc*.

Este trabalho consistiu na preparação de duas camadas: os dados brutos resultantes do processamento, sem nenhum tipo de tratamento, dados particionados mensalmente e compactados no formato *Parquet* com compressão do tipo *Snappy*.

Com os dados compactados e particionados permitiu-se que as pesquisas realizadas por ferramentas de visualização de dados retornassem com um ganho significativo de performance. Os dados brutos foram mantidos para comparação de performance.

5. RESULTADOS

Para a prova de conceito foi realizado um processamento de aproximadamente 20 dias de dados gerados pela instituição bancária, o que resultou aproximadamente em um volume de 900 Gigabytes de dados. Os dados foram processados tanto na arquitetura proposta como na arquitetura anteriormente utilizada.

Para este trabalho a infraestrutura utilizada em ambas as arquiteturas eram semelhantes, vide abaixo:

- ***Cluster Arquitetura Atual (Powercenter):*** 6 Máquinas, 63Tb HD, 1.2 Tb RAM, 144 Cores.
- ***Cluster Proposto Cloudera Hadoop****:* 6 Máquinas, 10Tb HD, 1.2 Tb RAM, 192 Cores.

Como se pode ver, o *cluster* da arquitetura proposta possui 25% de acréscimo na capacidade de processamento, entretanto a quantidade de memória é idêntica e o armazenamento é cerca de 85% menor que a arquitetura atual.

Neste trabalho entendeu-se que os resultados foram satisfatórios no que diz respeito ao tempo de desenvolvimento, já que todos os programas e scripts foram desenvolvidos ao longo de 27 dias corridos, por 2 programadores. Tratando-se da performance de processamento, entendeu-se que os resultados também foram satisfatórios, pois em comparação a arquitetura proposta com a arquitetura tradicional utilizada atualmente obteve-se ganhos significativos nos tempos de processamento dos dados, conforme a Tabela 1.

Tabela 1. Tabela comparativa com o tempo de processamento de cada arquitetura

Tipo	Passo	Tempo
Processamento na Arquitetura Tradicional	**Cache**	4 horas
	Processamento Final	30 minutos
Processamento na Arquitetura proposta de *Big Data*	**Cache**	-
	Processamento Final	**6 minutos**

Conforme se pode ver na Tabela 1, o processamento na arquitetura *Big Data* foi realizado 98% mais rápido do que a arquitetura tradicional utilizada pela instituição. O processamento na arquitetura tradicional utilizava 4 horas para o processamento de *cache* e mais 30 minutos para processamento final, somando-se no total 270 minutos de processamento. Esse ganho de performance, deve-se ao fato que não é necessária a alocação de dados em memória que ocorre na camada de *cache*, além das plataforma ser otimizada para processamento paralelo o que não ocorre na arquitetura tradicional. Destaca-se que se obteve uma

ineficiência na camada de visualização e consulta de dados e este problema foi resolvido com o particionamento e compactação dos dados. As consultas sob os dados compactados e particionados retornaram em cerca de 40 segundos. Sem a compactação e o particionamento as consultas não retornavam.

6. CONCLUSÃO

O processamento utilizando tecnologias de *Big Data* tem-se tornado uma tendência que traz algumas quebras de paradigmas de processamento de grandes volumes de dados.

Vale ressaltar que as ferramentas utilizadas não foram visuais e todo o processo foi desenvolvido por meio de código, em comparação com a ferramenta *Informatica Power Center*, que faz parte da arquitetura atual, que utiliza uma abordagem de ferramentas de programação visual e esta é uma dificuldade para adequação da empresa devido a diferença de abordagem de desenvolvimento.

Ao avaliar a capacidade de processamento da arquitetura proposta, a robustez, o custo da infraestrutura e de licenciamento, já que todas as ferramentas da arquitetura são *open source* e podem ser executadas em computadores comuns, somando ao tempo de desenvolvimento, pode-se concluir que existe uma grande vantagem na plataforma proposta.

Vale mencionar que a plataforma apresentada possui diversas outras funcionalidades e capacidades, como processamento em *streaming*/*real time*, capacidades analíticas, de construção de algoritmos para *machine learning* e modelagem estatística, processamento e armazenamento em memória e consultas em SQL, que não foram aplicadas para este trabalho, mas que também apresentam um grande ganho na plataforma proposta e sem custo adicional.

AGRADECIMENTOS

Agradecemos em primeiro lugar a Deus. Agradecemos a Evandro Luis Armelin, Pablo Sáez e a Everis do Brasil que permitiram que este trabalho fosse desenvolvido. Agradecemos também a Melina Cintra Lins pelo apoio e a Profª MSc. Luciana Maria da Silva pelas contribuições no trabalho.

REFERÊNCIAS

Achari, S,. *Hadoop Essentials: Delve into the key concepts of Hadoop and get a thorough understanding of the Hadoop ecosystem.* Packt Publishing, Birmingham, UK, pp 34. 2015.

Bazzotti, C.; Garcia, E. A importância do sistema de informação gerencial para tomada de decisões, 2005. Disponível em: http://www.waltenomartins.com.br/sig_texto02.pdf.

Beyer, M. Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. STAMFORD, Conn., June 27, 2011. Disponível em: http://www.gartner.com/newsroom/id/1731916

Ferreira, M. L., 2015 Modelo de Processamento para Criação de BI em Bancos de Dados NoSQL Orientado a Colunas. *3ª Conferência Ibero Americana de Computação Aplicada*. Florianópolis, Santa Catarina, Brasil, pp. 1-2

Gilbert, S.,Lynch, N., Brewer's Conjecture and the Feasibility of Consistent, Avaliable, Partition-Tolerant Web-Services. ACM SIGACT News, New York, NY,USA, v.33, n. 2, p.51,59, Junho, 2002.

Grus, J. 2016. *Data Science do Zero.* Altabooks Editora, Rio de Janeiro, BR.

Hadoop, A. What Is Apache Hadoop? – Disponível em: http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F. Acessado em 02/12/2016, 2016

Inmon, W. H. Building the Data Warehouse, 4º Edition. Wiley Publishing, Inc., 2005.

Kempe. S. A Short History of Data Warehousing. Disponível em: http://www.dataversity.net/a-short-history-of-data-warehousing/ . Acessado em: 02/11/2016. 2012

Kimball, R., R, M., *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition.* John Wiley & Sons, Inc., 2013.

Parquet, A. Apache Parquet Documentation. Disponível em: https://parquet.apache.org/documentation/latest/. Acessado em 02/12/2016, 2016

White, T. 2012. *Hadoop Definitive Guide, Third Edition.* O'Reilly Media, Inc., Sebastopol, USA.