Análise de Dados com Base em Processamento Massivo em Paralelo

Arquitetura de Data Warehousing

Profa. Dra. Cristina Dutra de Aguiar

Resumo:

Data warehousing engloba arquiteturas, algoritmos e ferramentas que possibilitam que dados selecionados de fontes de dados autônomas, heterogêneas e distribuídas sejam integrados em um único banco de dados, conhecido como *data warehouse*. Por meio da arquitetura de *data warehousing*, é possível identificar os componentes que participam do ambiente, o relacionamento que existe entre esses componentes e as funcionalidades de cada um. Neste texto são descritos conceitos relacionados à arquitetura de *data warehousing* e às diferenças entre os locais de armazenamento de dados presentes nessa arquitetura. Também são descritos conceitos relacionados a *big data*, incluindo sua definição e os desafios relacionados. Por fim, são ilustrados exemplos de arquiteturas instanciadas por meio de tecnologias, chamadas no mercado de trabalho de *pipelines*.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

Conteúdo

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

1 ARQUITETURA DE DATA WAREHOUSING

A Figura 1 ilustra uma visão geral da arquitetura de *data warehousing*, cujo detalhamento é descrito a seguir.

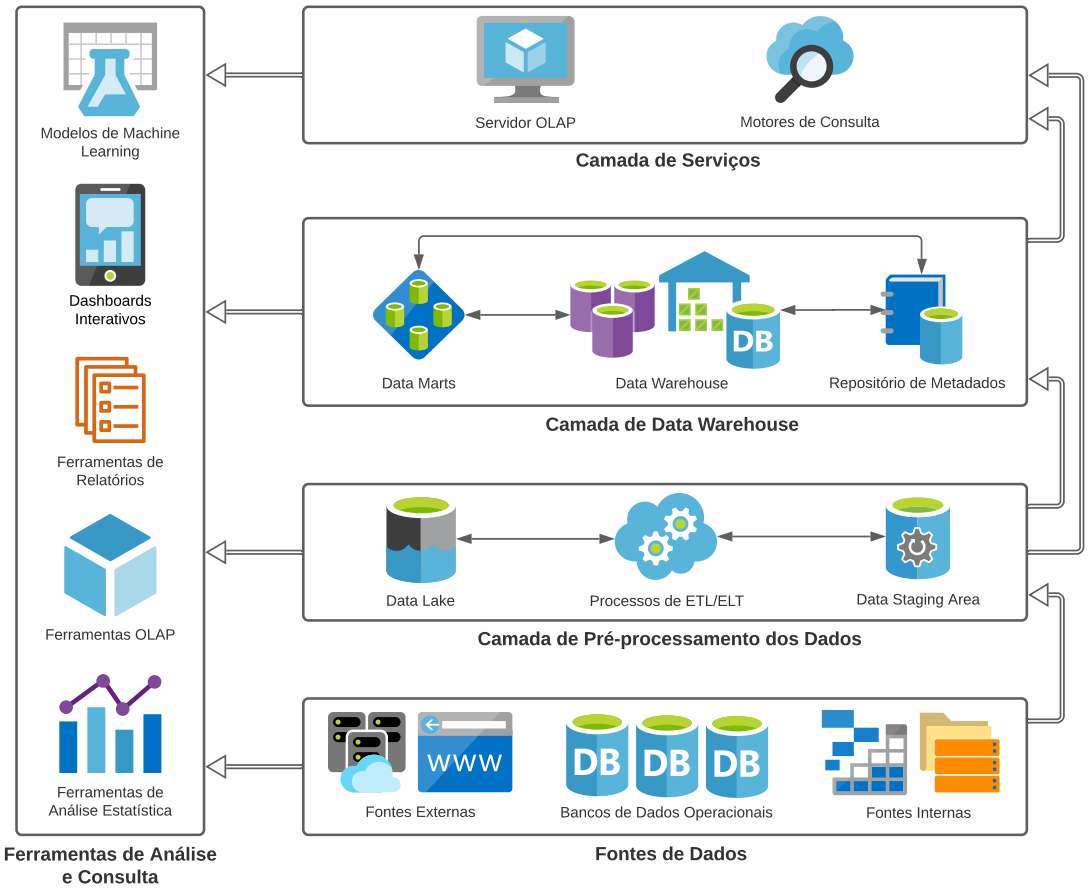

Figura 1: Visão geral da arquitetura do *data warehousing*.

1.1 FONTES DE DADOS

As **fontes de dados** contêm os dados operacionais. Elas são caracterizadas por serem autônomas, heterogêneas e distribuídas. O termo autônoma refere-se ao fato de que as fontes de dados foram desenvolvidas de forma independente, sem a perspectiva de fornecer seus dados ao *data warehousing*.

Ademais, o termo heterogênea indica que as fontes de dados podem podem possuir uma variedade de formatos e modelos. Exemplos de fontes heterogêneas de dados incluem:

- Sistemas gerenciadores de banco de dados (SGBDs) relacionais, orientados a objetos e objeto-relacionais.
- Bases de conhecimento e bases de dados NoSQL (*Not Only SQL*).
- Sistemas legados baseados em modelos hierárquicos e de rede.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

- Documentos HTML (*Hypertext Markup Language*) e SGML (*Standard Generalized Markup Language*).
- Planilhas e arquivos.

Quanto ao termo distribuída, ele indica que as fontes de dados encontram-se usualmente localizadas em diferentes ambientes computacionais ou servidores.

1.2 CAMADA DE PRÉ-PROCESSAMENTO DOS DADOS

A **camada de pré-processamento dos dados** tem como funcionalidade possibilitar a preparação dos dados para que eles possam ser posteriormente armazenados no *data warehouse* (DW). Componentes dessa camada são processos de ETL/ELT, *data staging area* e *data lake*.

No **processo ETL/ELT**, os dados relevantes das fontes de dados são extraídos, traduzidos, filtrados e integrados para serem posteriormente armazenados no DW. Esse processo também é responsável por realizar a atualização periódica do DW de forma a refletir as alterações nos dados das fontes e realizar a expiração de dados antigos armazenados no DW.

Considerando a evolução histórica de *data warehousing*, primeiramente surgiu o termo ETL (*extract*, *transform*, *load*), traduzido como extração, transformação e carga. Desde que o processo de ETL é demorado e custoso, a *data staging area* passou a ser usada como uma área de armazenamento intermediária. Essa área de armazenamento contém os dados das fontes de dados que vão passando por sucessivas modificações até que estejam prontos para serem carregados no DW [11].

Com o avanço tecnológico, principalmente advindo da manipulação de gigantescos volumes de dados relacionados ao conceito de *big data* [3], surgiu o termo ELT (*extract*, *load*, *transform*). Isso indica que primeiro os dados precisam ser extraídos e carregados no *data lake*, para depois continuarem a ser processados e finalmente armazenados no DW. Nesse sentido, o *data lake* atua como uma área de armazenamento que contém um grande volume de dados estruturados, semiestruturados e não estruturados e que são processados somente quando a informação precisa ser obtida [5]. Desde que o DW armazena dados estruturados, semiestruturados e não estruturados, costuma-se caracterizar que o *data lake* armazena dados em seu formato nativo. Esse termo, formato nativo, é usado ao longo deste texto. Entretanto, na prática, quando se tem dados tabulares ou aninhados, é usualmente comum transformá-los em um formato mais adequado para serem explorados no *data lake* e também para serem processados quando do armazenamento no DW.

1.3 CAMADA DE DATA WAREHOUSE

Na **camada de data warehouse** são armazenados os dados que já passaram pelos processos providos pela camada de pré-processamento, bem como os metadados associados. Componentes desta camada são DW, *data marts* e repositório de metadados.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

O DW, considerado o coração do ambiente de *data warehousing*, é um banco de dados voltado para o suporte aos processos de gerência e tomada de decisão. Ele armazena dados estruturados, os quais são organizados multidimensionalmente, ou seja, são organizados de acordo com as diferentes perspectivas de análise dos usuários de sistemas de suporte à decisão (usuários de SSD) [4].

Em adição ao DW, podem existir diversos *data marts*, cada um dos quais representando um pequeno DW que possui escopo limitado quando comparado ao DW propriamente dito. Os dados armazenados nos *data marts* compartilham as mesmas características que os dados do DW, ou seja, são dados organizados multidimensionalmente.

Outro componente desta camada é o **repositório de metadados**. Ele armazena os metadados de todos os dados e processos envolvidos no *data warehousing*. O conceito de metadados refere-se ao fato de que dados de nível mais baixo podem ser descritos por dados de nível mais alto. Metadados consistem em uma abstração que provê significado semântico aos dados. Por exemplo, metadados associados a uma sequência de 0s e 1s devem indicar se tais caracteres representam palavras, ou números, ou dados estatísticos sobre salários de empregados, ou ainda informações sobre a distribuição de cargos na empresa.

O armazenamento de metadados no *data warehousing* possui um nível de importância elevado, uma vez que é necessário conhecer a estrutura e o significado dos dados presentes em todos os processos envolvidos. Em outras palavras, metadados constituem-se no principal recurso para a administração dos dados no *data warehousing*. Portanto, uma grande variedade de metadados precisa ser armazenada, visando a utilização efetiva do *data warehousing*.

1.4 CAMADA DE SERVIÇOS

Componentes presentes na **camada de serviços** devem oferecer funcionalidades adequadas para facilitar o acesso aos dados com o objetivo de dar suporte à tomada de decisão estratégica.

Servidores OLAP (*on-line analytical processing*) proveem visões multidimensionais dos dados do DW ou dos *data marts*, independentemente da forma na qual os dados encontram-se armazenados [11]. Uma visão multidimensional permite a visualização dos dados sob diferentes perspectivas, de acordo com as necessidades dos usuários de suporte à decisão. É importante destacar que o servidor OLAP está relacionado ao conceito de hipercubo de dados multidimensional, ou seja, à metáfora do cubo de dados. Esse conceito está atrelado ao nível conceitual do modelo de dados multidimensional [4].

Os **motores de consulta** oferecem serviços que têm como funcionalidade executar consultas contra dados armazenados em bancos de dados. Por exemplo, um motor de consulta SQL (*structured query language*) executa consultas nessa linguagem de programação contra dados armazenados no modelo relacional.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

1.5 CAMADA DE FERRAMENTAS DE ANÁLISE E CONSULTA

O principal propósito de um *data warehousing* consiste em disponibilizar informação integrada aos usuários de SSD para a tomada de decisão estratégica. Esses usuários interagem com o ambiente por meio de **ferramentas de análise e consulta** dos dados, as quais devem oferecer facilidades de navegação e de visualização. Em especial, essas ferramentas devem permitir que informações relevantes ao contexto de tomada de decisão sejam derivadas a partir da detecção de análise de tendências, da monitoração de problemas e de análises competitiva e comparativa.

Dentre os principais tipos de ferramentas existentes, pode-se citar:

- **Ferramentas de consulta gerenciáveis e geradores de relatório**. São os tipos mais simples de ferramentas e, em geral, não são voltadas especificamente ao *data warehousing*. Geradores de relatório, como o próprio nome diz, têm como principal objetivo produzir relatórios periódicos. Já ferramentas de consulta gerenciáveis oferecem aos usuários visões de negócio específicas ao domínio dos dados armazenados e permitem que esses usuários realizem consultas independentemente da estrutura e/ou da linguagem de consulta oferecida pelo banco de dados. Por exemplo, uma ferramenta desse tipo poderia permitir a criação de comandos SQL por meio da utilização de um conjunto de opções. A saída destas ferramentas é geralmente na forma de um relatório.

- **Ferramentas de análise estatística**. Essas ferrramentas permitem que os usuários de SSD analisem os dados usando métodos estatísticos.

- **Dashboards interativos**. Um *dashboard* é uma ferramenta que reúne diversos dados e indicadores por meio de gráficos e tabelas, permitindo o monitoramento simultâneo de um grande número de informações, as quais são visualizadas com facilidade em um único ambiente. *Dashboards* possibilitam interações de detalhamento ou agrupamento de gráficos, indicadores e tabelas, dentre outros.

- **Ferramentas OLAP**. São caracterizadas por permitir que usuários de SSD sofisticados analisem os dados usando visões multidimensionais complexas e elaboradas, e por oferecer navegação facilitada pelas diferentes visões. Assim, os usuários podem analisar os dados sob diferentes perspectivas e/ou determinar tendências por meio da navegação entre diferentes níveis de agregação. Tais ferramentas apresentam os dados de acordo com o modelo multidimensional [2, 7, 4], independentemente da forma na qual eles estão realmente armazenados.

- **Modelos de machine learning**. De maneira geral, as ferramentas de análise e consulta possuem duas funcionalidades básicas: facilitar o acesso aos dados do DW e permitir que

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

informações, tendências e padrões de negócio “escondidos” nesses dados sejam descobertos. Modelos de *machine learning* vislumbram a segunda funcionalidade. Nesse sentido, aplicações de *machine learing* processam os dados para descobrir esses padrões escondidos e mostrá-los aos usuários de SSD, os quais podem inferir conhecimento útil a partir destes. A qualidade do conhecimento descoberto é altamente dependente da aplicação e tem um aspecto subjetivo inerente.

Independentemente do tipo de ferramenta utilizada, um fator primordial a ser considerado refere-se à **visualização** dos resultados obtidos. Técnicas de visualização dos dados devem determinar a melhor forma de se exibir relacionamentos e padrões complexos, de modo que o problema inteiro e a solução sejam claramente visíveis. Por exemplo, padrões podem ser mais facilmente detectados se forem expressos graficamente, melhor do que por meio de simples tabelas. Em especial, técnicas de visualização devem oferecer interação com os usuários de SSD, os quais devem ser capazes de alterar tanto o tipo de informação sendo analisada quanto o método de apresentação sendo utilizado (como histogramas, mapas hierárquicos e gráficos de dispersão).

É importante destacar que a visualização dos dados pode não ser a atividade fim das análises realizadas sobre os dados do DW ou, principalmente, sobre os dados do *data lake*. Existe a possibilidade de que as métricas extraídas por meio dessas análises alimentem automaticamente as aplicações que atuam como fontes de dados. Por exemplo, suponha uma aplicação de *data warehousing* de *e-commerce*. Suponha também que os dados armazenados no DW ou no *data lake* tenham sido utilizados para treinar um modelo de aprendizado de máquina. Como resultado, o conhecimento obtido pode retornar como uma recomendação para a página de *e-commerce*.

2 DIFERENÇAS ENTRE LOCAIS DE ARMAZENAMENTO

Excluindo as fontes de dados, a arquitetura de *data warehousing* (Figura 1) pode englobar quatro locais de armazenamento de dados: DW, *data mart*, *data lake* e *data staging area*. A seguir são destacadas as diferenças existentes entre esses locais de armazenamento.

2.1 DATA WAREHOUSE E DATA MARTS

Como visto anteriormente, um *data mart* consiste na implementação de um DW no qual o escopo do dado é limitado quando comparado ao DW propriamente dito. Entretanto, os dados armazenados em *data marts* compartilham as mesmas características que os dados do DW, ou seja, são orientados a assunto, integrados, históricos e não voláteis, além de serem organizados em níveis de agregação.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

É necessário, portanto, discutir a importância dos *data marts* dentro do *data warehousing*. Em uma grande corporação, *data marts* tendem a ser utilizados como uma política de construção evolucionária do DW. Uma vez que o processo de construção de um DW sobre toda a organização é longo e complexo e os custos envolvidos são altos, *data marts* são construídos paulatinamente e, à medida que estes se consolidam, inicia-se a construção do DW corporativo. De maneira geral, tais *data marts* representam soluções fragmentadas de porções de negócio da empresa, sendo chamados de *data marts* independentes.

A utilização de *data marts* independentes tende, inicialmente, a reduzir problemas financeiros. Isso se deve ao fato de que a construção desses *data marts* exige recursos monetários inferiores do que os despendidos com a construção de um DW corporativo, fazendo com que os usuários de SSD sejam capazes de reconhecer o valor e a potencialidade da solução de *data warehousing* em um período menor de tempo. Entretanto, em longo prazo, a criação de *data marts* independentes pode conduzir a problemas de integração, caso um modelo de negócio completo não seja desenvolvido. Isso se deve ao fato de que cada *data mart* independente pode assumir formas diferentes de consolidar seus dados, gerando inconsistências.

2.2 DATA STAGING AREA E DATA LAKE

A *data staging area* contém dados extraídos das fontes de dados que vão passando por modificações sucessivas até que estejam prontos e que possam ser carregados no DW [11]. Ela consiste em uma área de armazenamento intermediária para a qual não se prevê acesso pelos componentes da camada de ferramentas de análise e consulta. Portanto, o fluxo de dados é no sentido *data staging area* $\rightarrow$ DW. Conforme descrito anteriormente, a *data staging area* é decorrente historicamente do processo de ETL.

O *data lake* contém um grande volume de dados extraídos das fontes de dados em seu formato nativo (*raw data*), incluindo dados estruturados, semiestruturados e não estruturados. Esses dados são processados somente quando a informação precisa ser obtida [5]. Existem dois fluxos de dados quando se trata de *data lake*. O primeiro deles é no sentido *data staging area* $\rightarrow$ DW, significando que o *data lake* pode atuar também como uma *data staging area* para a carga de dados no DW. O segundo fluxo de dados é no sentido *data lake* $\rightarrow$ componentes da camada de ferramentas de análise e consulta, indicando que os dados do *data lake* também podem ser usados para a descoberta de novas informações ou para a geração de valor a partir desses [8]. Isso significa que as consultas e análises podem ser realizadas diretamente sobre o *data lake*, sem a necessidade de se usar os dados do DW. Esse fluxo é necessário, por exemplo, quando se deseja analisar dados de *streaming*. Conforme descrito anteriormente, o *data lake* é decorrente historicamente do processo de ELT.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

2.3 DATA WAREHOUSE E DATA LAKE

Na Tabela 1 são contrastadas diferenças existentes entre o DW e o *data lake*, considerando os seguintes aspectos relacionados aos dados: característica, formato, pré-processamento, tipos de consulta, latência de disponibilidade dos dados, custo de geração dos dados e custo de análise dos dados [1]. Essas diferenças são detalhadas a seguir.

Tabela 1: Comparativo entre as características do DW e do *data lake*, considerando aspectos relacionados aos dados.

	Data Warehouse	Data Lake
Característica	consolidados, organizados e estruturados	estruturados, semiestruturados e não estruturados
Formato	esquema estruturado (formato bem definido)	formato nativo (diferentes formatos)
ETL/ELT	dados pré-processados antes de serem carregados	dados extraídos e carregados, sem sofrer transformações
Tipos de Consulta	OLAP	variado
Latência de disponibilidade dos dados	alta	baixa
Custo de Geração	maior	menor
Custo de Análise	menor	maior

Enquanto o DW contém dados consolidados e organizados que já passaram pelo processo de ETL e que são armazenados segundo um esquema estruturado bem definido, o *data lake* deve oferecer suporte a vários formatos de dados, facilitando a aquisição dos dados das fontes de dados para prover agilidade.

Para povoar o DW, os dados precisam primeiro passar pelo processo de ETL. Em contrapartida, os dados armazenados no *data lake* são decorrentes do processo de ELT, ou seja, eles são extraídos das fontes de dados e carregados no *data lake*, sem passar por processos de transformação, os quais ocorrem somente quando necessário. O esforço necessário para extrair e carregar os dados é reduzido porque os dados não são pré-processados.

Com relação aos tipos de consulta, o DW é um banco de dados especialmente projetado para oferecer suporte eficiente ao processamento de consultas analíticas, ou seja, consultas

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

OLAP (*on-line analytical processing*). O *data lake*, por sua vez, oferece suporte para os mais variados tipos de consulta.

A latência de disponibilidade dos dados, o custo de geração e o custo de análise são decorrentes do pré-processamento aplicado aos dados. Devido ao processo ETL, os dados do DW demoram para serem processados e demandam processos muito custosos. Portanto, possuem latência alta e maior custo de geração dos dados. Em contrapartida, requerem menor custo relacionado à análise dos dados, desde que os dados já encontram-se preparados para serem utilizados na tomada de decisão estratégica. No caso dos dados do *data lake*, a situação é inversa. Devido ao processo ELT, os dados são extraídos e já armazenados no *data lake*, garantindo uma latência baixa de disponibilidade dos dados e um custo de geração dos dados menor. Em contrapartida, os dados precisam ser transformados e integrados para serem usados nas análises dos usuários de SSD, requerendo custos maiores para o processamento de consultas analíticas.

3 BIG DATA

Conforme descrito na arquitetura de *data warehousing*, o processo ELT e o *data lake* surgiram em resposta aos gigantescos volumes de dados relacionados ao conceito de *big data*. Nesta seção são descritos a definição de *big data* e os desafios impostos por esses.

3.1 DEFINIÇÃO

As definições de *big data* mais usadas são baseadas no modelo de diferentes Vs, sendo que na literatura existem trabalhos que definem 3Vs [3], 4Vs [6], 5Vs [9] e 7Vs [12]. O modelo de 7Vs é definido da seguinte forma.

- Volume: gigantesca quantidade de dados, a qual atualmente varia de *terabytes* a *exabytes*.
- Velocidade: captura e disponibilidade de um gigantesco volume de dados em um pequeno intervalo de tempo.
- Variedade: dados podem ser de qualquer tipo, incluindo dados semiestruturados e não-estruturados como áudio, vídeo, páginas web e texto, além de dados estruturados.
- Veracidade: quão confiáveis são os dados, considerando aspectos como abrangência, consistência, precisão e atualidade.
- Valor: os dados devem ter importância dentro do contexto da aplicação, de forma que justifique a necessidade de manipulação desses dados.
- Variabilidade: os valores dos dados e seus significados podem variar constantemente.
- Visualização: exibição apropriada dos dados volumosos.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

3.2 DESAFIOS

A manipulação de *big data* introduz vários desafios [3]. O primeiro deles se refere ao uso de **ambientes computacionais com grande capacidade de armazenamento e processamento**, tais como *clusters* de computadores e ambientes de computação em nuvem (*cloud computing*). Esses ambientes exigem que diversos aspectos sejam considerados para que o processamento de grandes volumes de dados ocorra de forma otimizada, o que muitas vezes reflete em tarefas não triviais.

O segundo desafio consiste no uso de **frameworks de processamento paralelo e distribuído de dados**, os quais surgiram para simplificar a interação do usuário com os ambientes computacionais descritos anteriormente por meio do provimento de uma interface simplificada de programação de aplicações. Exemplos amplamente utilizados são Apache Hadoop[1] e Apache Spark[2].

O terceiro desafio refere-se ao uso de **sistemas de arquivos distribuídos**, dentre os quais destaca-se o HDFS (*Hadoop Distributed File System*) [10]. Ele provê suporte para o armazenamento de grandes quantidades de dados e possui alta tolerância a falhas. Adicionalmente, HDFS é capaz de ser empregado também em equipamentos de *hardware* de baixo custo. Ambos Apache Hadoop e Apache Spark utilizam o HDFS como sistema de arquivos distribuídos padrão.

Por fim, em adição ao uso de alguns SGBDs relacionais, pode-se destacar o quarto desafio como a possibilidade de se usar **bases de dados NoSQL**. Eles são caracterizados por serem baseados em diferentes formatos, usualmente não relacionais, e por garantirem alta escalabilidade. Adicionalmente, eles introduzem flexibilidade no armazenamento de diferentes tipos de dados, como dados não estruturados, semiestruturados e estruturados.

O detalhamento desses desafios não é o objetivo do presente texto. Os desafios, bem como a definição de *big data*, foram introduzidos neste texto para dar suporte à discussão de arquiteturas instanciadas por meio de tecnologias. A descrição detalhada dos desafios será realizada posteriormente.

4 INSTANCIAÇÃO DA ARQUITETURA DE DATA WAREHOUSING

A arquitetura ilustrada na Figura 1 mostra todos os componentes de um *data warehousing*. Entretanto, nem todos precisam estar presentes no desenvolvimento de uma aplicação voltada à tomada de decisão estratégica. A escolha de quais componentes devem participar do *data warehousing* depende do propósito da aplicação: para o que ela serve, quais seus objetivos e quais

[1] https://hadoop.apache.org/
[2] https://spark.apache.org/

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

componentes são capazes de oferecer suporte para as demandas impostas pela aplicação.

Nesta seção são exemplificadas arquiteturas de *data warehousing* instanciadas por meio de tecnologias, as quais são usualmente chamadas no mercado de trabalho de *pipelines*. Note que o objetivo é mostrar exemplos, sem ser exaustivo. Ou seja, não são listadas todas as tecnologias existentes nem todas as possíveis instanciações que podem ocorrer. Também são considerados diferentes tipos de tecnologias, incluindo soluções de *software livre* e produtos pagos.

Em todas as arquiteturas ilustradas nesta seção, os dados são armazenados no DW para propósito ilustrativo. Porém, eles poderiam estar armazenados em um ou mais *data marts* de acordo com as discussões realizadas na seção 2.3.

4.1 PIPELINES PARA VOLUMES DE DADOS TRADICIONAIS

Na Figura 2 é ilustrado um *pipeline* para uma aplicação de *data warehousing* tradicional de processamento de dados em lotes. Como exemplos de fontes de dados, tem-se um SGBD relacional, uma base de dados no formato JSON (*JavaScript Object Notation*) e uma planilha Excel. Nessa proposta de *pipeline*, os dados das fontes operacionais são extraídos, transformados e carregados (ETL) na *data staging area* usando Pandas[3]. Uma vez na *data staging area*, que é construída no SGBD relacional MySQL[4], os dados passam por diversas transformações até se adequarem à forma de organização do DW. Na sequência, os dados são extraídos da *data staging area* e carregados no DW, também construído no MySQL. Desde que ambos *data staging area* e DW utilizam a tecnologia relacional para o armazenamento dos dados, utiliza-se a linguagem de programação SQL[5] para a movimentação dos dados entre esses locais de armazenamento. O *pipeline* inclui também duas tecnologias de análise e visualização dos dados do DW: a ferramenta de construção de *dashboards* interativos Metabase[6] e uma aplicação Python[7] para a geração de relatórios analíticos.

O *pipeline* ilustrado na Figura 3 mostra a implementação no ambiente de nuvem da AWS[8] (*Amazon Web Services*) para a mesma arquitetura do *pipeline* anterior (Figura 2). Para o processo de ETL utiliza-se o AWS Lambda[9], uma ferramenta de computação em nuvem sem servidor (*serverless*) de baixo custo que permite a execução de códigos em linguagens de programação como Python e JavaScript, dentre outras. Ambos *data staging area* e DW são construídos usando o serviço de provisionamento de bancos de dados relacionais em nuvem da AWS chamado de AWS RDS[10] (*Relational Database Service*). Por fim, a aplicação Python para geração de relatórios analíticos do *pipeline* utiliza o mesmo serviço do processo de ETL, ou seja, AWS

[3] https://pandas.pydata.org/
[4] https://www.mysql.com/
[5] https://www.iso.org/standard/63555.html
[6] https://www.metabase.com/docs/latest/users-guide/07-dashboards.html
[7] https://www.python.org/
[8] https://aws.amazon.com/
[9] https://aws.amazon.com/lambda/
[10] https://aws.amazon.com/rds/

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

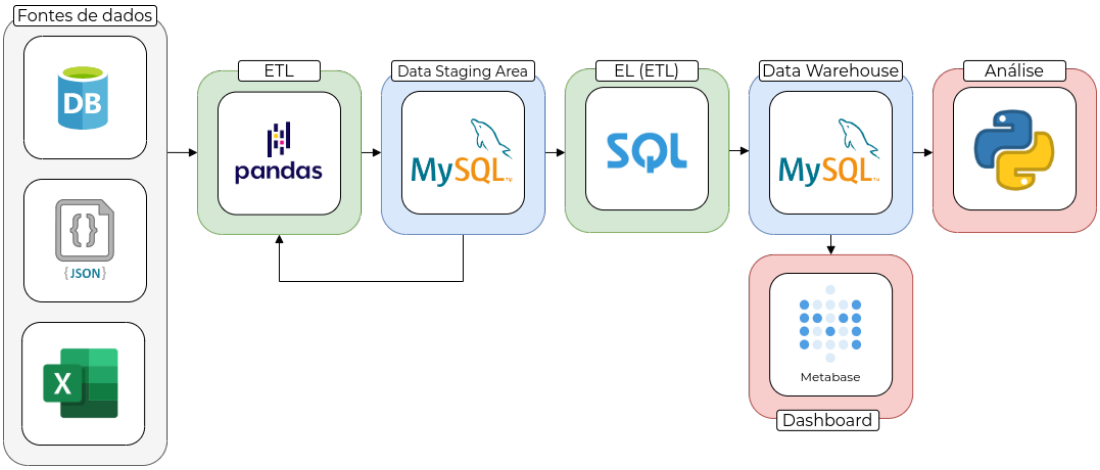

Figura 2: *Pipeline* de processamento de dados em lotes.

Lambda. Com relação à ferramenta geradora de *dashboards* interativos Tableau[11], esta usa seu próprio serviço de nuvem para visualizar os dados do DW no ambiente da AWS.

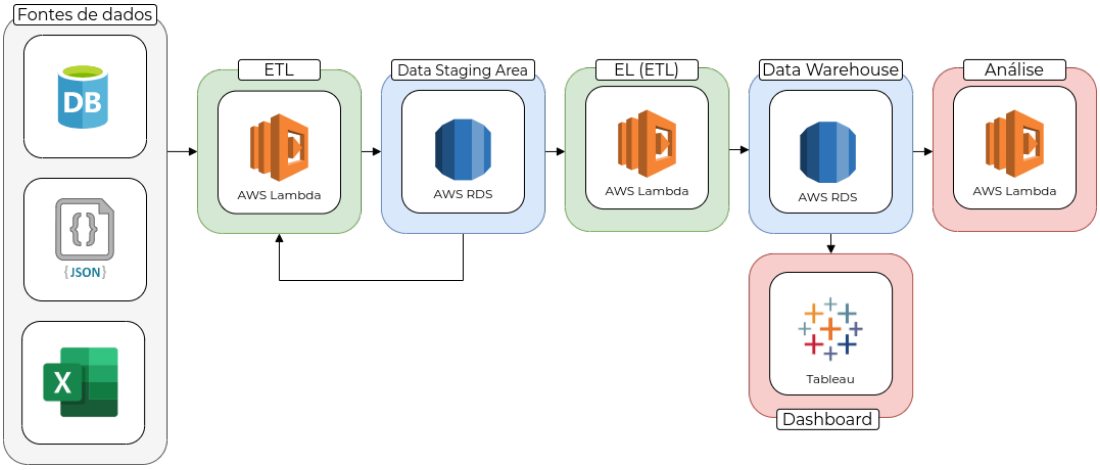

Figura 3: *Pipeline* de processamento de dados em lotes na nuvem.

4.2 PIPELINES PARA GIGANTESCOS VOLUMES DE DADOS

No contexto de *big data*, é proposto o *pipeline* ilustrado na Figura 4. Os dados de interesse das fontes de dados são extraídos e carregados em um *data lake* pelo Apache Spark. No *data lake*, os dados são explorados pelo motor de consulta Apache Hive[12]. Na sequencia, os dados passam por vários processos de transformação usando o Apache Spark até que se adequarem à forma de organização do DW. Quando prontos, esses dados são carregados no DW construído utilizando o Apache Druid[13], o qual também atua como um servidor OLAP no que tange às consultas analíticas. Por fim, usuários de SSD podem realizar análises e criar *dashboards* utilizando o Metabase.

[11] https://www.tableau.com/
[12] https://hive.apache.org/
[13] https://druid.apache.org/

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

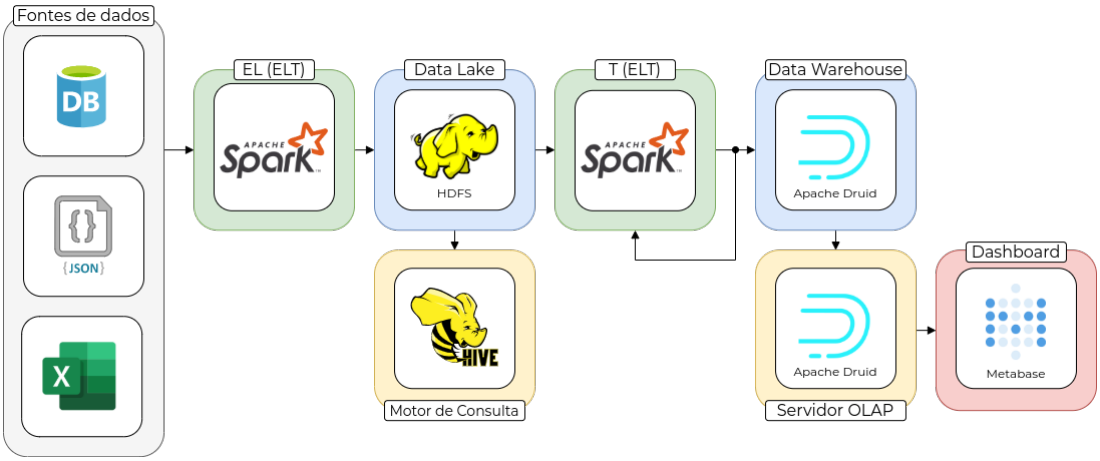

Figura 4: *Pipeline* de processamento de *big data* em lotes.

Na Figura 5 é ilustrada a implementação do *pipeline* da arquitetura anterior (Figura 4) no ambiente de nuvem da AWS usando as seguintes tecnologias: (i) AWS EMR[14] (*Elastic Map Reduce*) para a extração e o carregamento dos dados das fontes de dados no *data lake* e para a transformação dos dados do *data lake* para o DW; (ii) AWS S3[15] (*Simple Storage Service*) para o armazenamento dos dados no *data lake*; (iii) AWS Athena[16] como motor de consulta para a exploração dos dados armazenados no *data lake*; e (iv) AWS Redshift [17] como uma solução de armazenamento dos dados no DW e como servidor OLAP para esses dados. Usuários de SSD utilizam o Tableau, em sua própria infraestrutura em nuvem, como ferramenta de análise e criação de *dashboards* interativos.

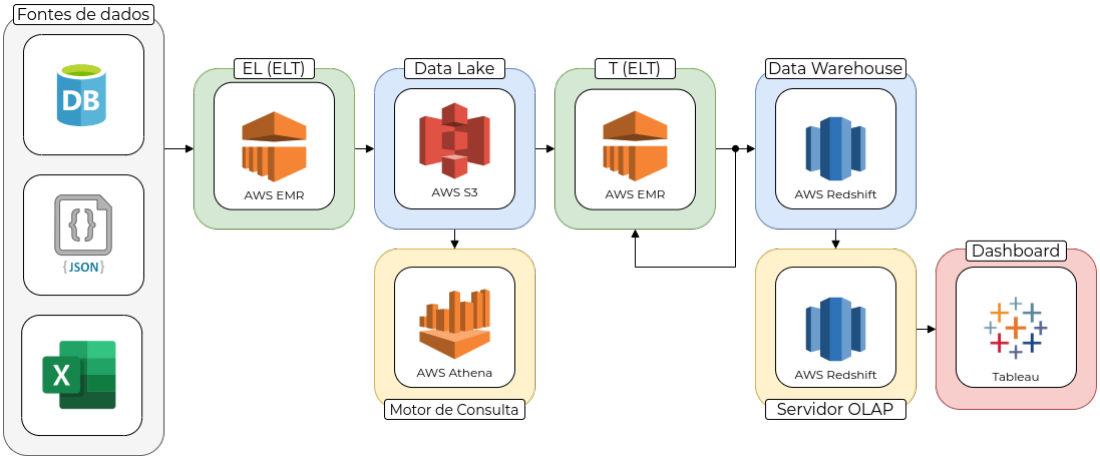

Figura 5: *Pipeline* de processamento de *big data* em lotes na nuvem.

[14] https://aws.amazon.com/emr/
[15] https://aws.amazon.com/s3/
[16] https://aws.amazon.com/athena/
[17] https://aws.amazon.com/redshift/

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

4.3 PIPELINES PARA DATA STREAMING DE GIGANTESCOS VOLUMES DE DADOS

Existe um conjunto de fontes de dados que produzem dados de maneira contínua (*data stream*) e que necessitam de monitoramento e extração de conhecimento em tempo (quase) real. Neste contexto, uma arquitetura de processamento de dados em lotes, como as ilustradas nas Figuras 2 a 5, pode não ser adequada. Por exemplo, considere um modelo de aprendizado de máquina que realiza uma recomendação a um usuário de um *e-commerce* enquanto ele navega pelas suas páginas. A aplicação de *data warehousing* que oferece suporte para esse modelo deve ser capaz de extrair e prover dados rapidamente. Por exemplo, no primeiro semestre de 2019, o tempo médio de navegação de um usuário dessa categoria de *website* foi de 4 minutos e 12 segundos[18]. Portanto, os dados precisam navegar por todo o *pipeline* e retornar a recomendação durante este intervalo de tempo para atender à demanda dessa aplicação.

Na Figura 6 é introduzido um exemplo de *pipeline* de processamento de dados de *data streaming* de *big data*. Como exemplos de fontes de dados, tem-se um *website*, uma aplicação executando em um aparelho *mobile* e dados de IoT (*Internet of Things*). Os dados das fontes de dados são enviados em fluxo contínuo para o Apache Kafka[19] e são extraídos, transformados e carregados no DW, em tempo (quase) real, usando a solução de *streaming* do Apache Spark. Os dados do DW são armazenados no Apache Druid. Por fim, a combinação do servidor OLAP do Apache Druid com a ferramenta de análise Metabase proporciona aos usuários de SSD *dashboard* interativo com métricas em tempo (quase) real dos dados do DW.

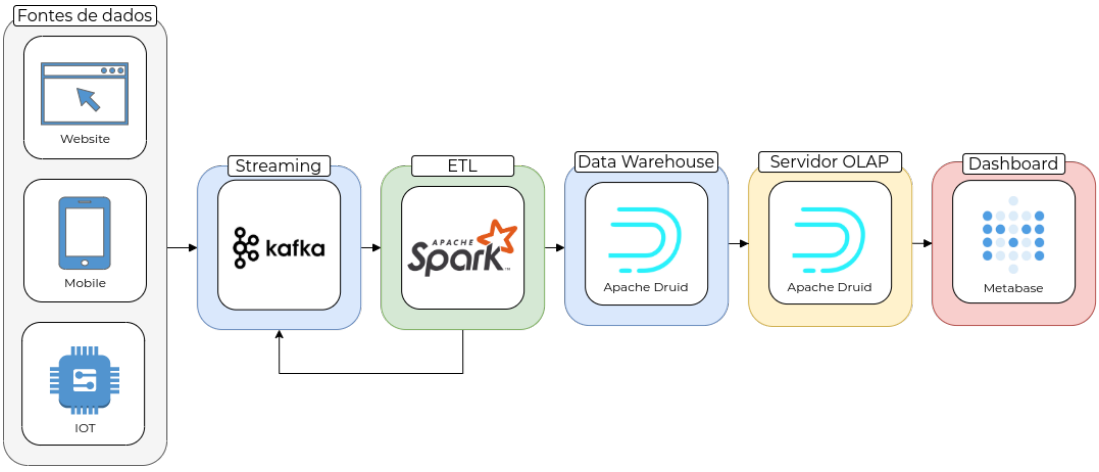

Figura 6: *Pipeline* de processamento de *data streaming* de *big data*.

De maneira análoga, o *pipeline* da Figura 7 propõe a implementação do *pipeline* anterior (Figura 6) no ambiente de nuvem da AWS. Os dados são extraídos, transformados e carregados no DW, armazendo no AWS Redshift, em fluxo contínuo pelo par de tecnologias AWS Kinesis[20]

[18] https://www.salesforce.com/solutions/industries/retail/shopping-index/
[19] https://kafka.apache.org/
[20] https://aws.amazon.com/kinesis/

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

e AWS EMR. Já o servidor OLAP do AWS Redshift e o Tableau (este em sua própria infraestrutura de nuvem) proporcionam aos usuários de SSD um *dashboard* interativo com métricas em tempo (quase) real dos dados do DW.

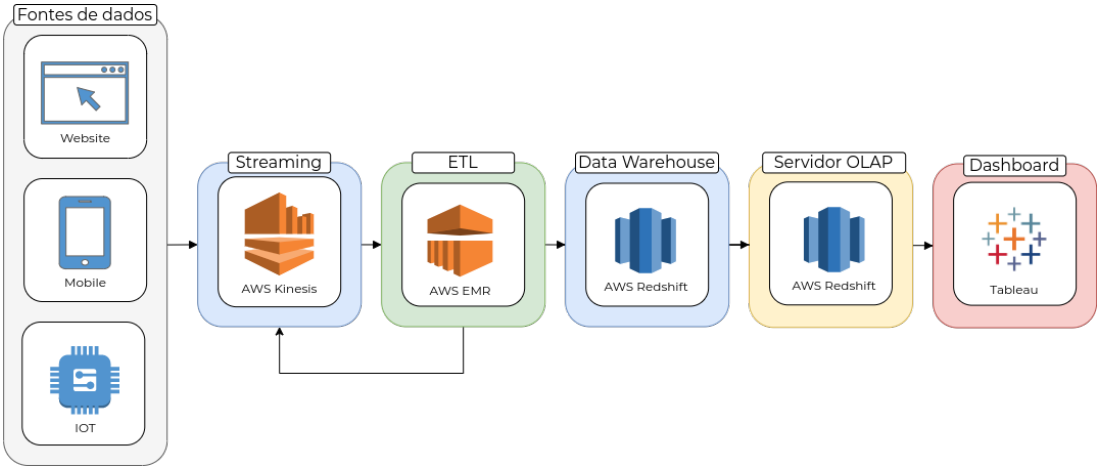

Figura 7: *Pipeline* de processamento de *data streaming* de *big data* na nuvem.

5 CONCLUSÃO

Neste texto, foram descritos os seguintes conceitos e aspectos relacionados:

- Arquitetura de *data warehousing*: fontes de dados, camada de pré-processamento dos dados, camada de DW, camada de servidores de aplicação e camada de ferramentas de análise e consulta.
- Diferenças entre os locais de armazenamento de dados: DW e *data marts*, *data staging area* e *data lake* e DW e *data lake*.
- *Big data*: definição e principais desafios.
- Instanciação da arquitetura de *data warehousing*: exemplificação de diferentes *pipelines* considerando aplicações de *data warehousing* tradicionais, no contexto de *big data* e no contexto de *data streaming*.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

Referências

[1] J. J. Brito. *Data Warehouses in the era of Big Data: efficient processing of Star Joins in Hadoop*. PhD thesis, Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e da Computação, Universidade de São Paulo, 2017.

[2] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1):65–74, 1997.

[3] M. Chen, S. Mao, , and Y. Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[4] C. D. A. Ciferri, R. R. Ciferri, L. Gómez, M. Schneider, A. Vaisman, and E. Zimányi. Cube algebra: A generic user-centric model and query language for OLAP cubes. *Journal of Data Warehousing and Mining*, 9(2):39–65, 2013.

[5] J. Couto, O. Borges, D. Ruiz, S. Marczak, and R. Prikladnicki. A mapping study about data lakes: An improved definition and possible architectures. In *Proceedings of the 31st International Conference on Software Engineering and Knowledge Engineering*, pages 453–458, 2019.

[6] X. L. Dong and D. Srivastava. Big data integration. *Proceedings of the VLDB Endowment*, 6(11):1188–1189, 2013.

[7] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley, 2nd edition, 2002.

[8] C. Mathis. Data lakes. *Datenbank-Spektrum*, 17(3):289–293, 2017.

[9] S. Sharma and V. Mangat. Technology and trends to handle big data: Survey. In *Proceedings of the Fifth International Conference on Advanced Computing & Communication Technologies*, pages 266–271, 2015.

[10] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The Hadoop distributed file. In *Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies*, pages 1–10, 2015.

[11] A. Vaisman and E. Zimányi. *Data Warehouse Systems: Design and Implementation*. Springer, 2014.

[12] R. Wrembel. Novel big data integration techniques: Painel discussion at BigNovelTI 2017@ADBIS2017. 2017.

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.