

FUNDAMENTOS DE BIG DATA

Maycon Viana Bordin



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Casos de uso do Hadoop no MapReduce

OBJETIVOS DE APRENDIZAGEM

- > Reconhecer os casos de sucesso do Hadoop.
- > Determinar possíveis implementações dos casos de sucesso do MapReduce.
- > Discutir desafios para extração de informação no Hadoop.

Introdução

O Hadoop surgiu como uma implementação *open source* do MapReduce e, ao longo dos anos, se tornou a principal plataforma de *big data* do mercado, sendo utilizada pelas principais empresas de tecnologia do mundo. Assim, o Hadoop se transformou em um ecossistema composto por diversas ferramentas para a resolução de diversos problemas.

Neste capítulo, você vai estudar os principais casos de sucesso do Hadoop, compreendendo de que forma as empresas utilizaram essa ferramenta para processar grandes volumes de dados. Você também vai verificar algumas implementações de casos de sucesso do MapReduce e, por fim, vai compreender as dificuldades encontradas na extração de informações utilizando o Hadoop.

Casos de sucesso do Hadoop

O **MapReduce** foi desenvolvido pelo Google e consiste em um modelo de programação e uma implementação para processamento e geração de grandes

conjuntos de dados (*datasets*)(DEAN; GHEMAWAT, 2008). Programas escritos nesse modelo funcional são automaticamente paralelizados e podem ser executados em grandes *clusters* de máquinas *commodity*. A popularização do MapReduce ocorreu com a criação de uma implementação *open source*, chamada de **Hadoop**, que permitiu que qualquer indivíduo com acesso a recursos de computação *commodity* pudesse armazenar e processar grandes volumes de dados.

Ao longo dos anos, o Hadoop foi ganhando popularidade e passou a ser oferecido por empresas especializadas, contando com suporte e treinamento. Dessa forma, o Hadoop ganhou força no lado corporativo, oferecendo também segurança para que pudesse ser utilizado de forma confiável. Assim, o Hadoop se tornou um ecossistema de ferramentas e padrões interoperáveis, que permite a resolução de uma gama muito grande de problemas. Existem diversos casos de uso que se tornaram referência na implementação do Hadoop.

Caso de uso em saúde: Cerner

Um dos casos de sucesso da implementação do Hadoop é o da empresa de tecnologia focada em saúde Cerner, que precisava criar um índice de busca mais eficiente para registros médicos. Essa tarefa era desafiadora, pois envolvia o processamento de grandes volumes de documentos para extração de termos e resolução do relacionamento do termo com outros termos. Outras tecnologias haviam sido avaliadas, como uma arquitetura orientada a eventos, mas os requisitos de se reprocessar todos os documentos em intervalos de horas devido a melhorias constantes nos algoritmos de extração de termos levou à escolha do Hadoop como solução mais adequada (WHITE, 2015).

A princípio, a Cerner utilizou aplicações MapReduce para processar documentos médicos e indexá-los no Apache Solr. Mas, com o tempo, a empresa percebeu que existia uma grande complexidade nesses documentos, pois eles eram oriundos de diferentes sistemas, com formatos e padrões diversos. Isso se tornou um grande desafio, mais até do que o volume dos dados. No final, a Cerner escolheu o Apache Crunch, uma interface de programação de aplicações (API, do inglês *application programming interface*) de alto nível para a construção de aplicações que vão executar no MapReduce ou Apache Spark. Dessa forma, pode-se construir uma aplicação capaz de processar os diversos formatos de documentos médicos, padronizá-los e armazená-los no formato Avro. Por meio do Hadoop, a Cerner conseguiu escalar para grandes volumes de dados e, com as ferramentas de mais alto nível, conseguiu processar os mais diversos tipos e formatos de documentos, criando uma visão padronizada de registros médicos (WHITE, 2015).

Caso de uso: Facebook

Outro caso de sucesso do Hadoop é o do Facebook — talvez um dos maiores casos, se levarmos em conta o volume de dados armazenado e a quantidade de dados ingeridos diariamente. O Facebook começou seus experimentos com o Hadoop por volta de 2007, buscando uma forma de armazenar e analisar dados históricos para melhorar a experiência dos usuários. Porém, a empresa percebeu que isso só seria possível se conseguisse empoderar seus engenheiros e analistas com ferramentas de fácil uso para minerar e manipular grandes volumes de dados (SARMA, 2008).

Atualmente, o Facebook possui um dos maiores *clusters* de Hadoop no mundo, com mais de 4 mil máquinas armazenando centenas de milhões de gigabytes de dados. O Hadoop fornece a infraestrutura comum para o Facebook operar com eficiência e confiabilidade, dando suporte para buscas, processamento de *logs*, sistemas de recomendação, *data warehousing* e análise de imagens e vídeo (CHAKRABORTY, 2017).

O Facebook chegou a desenvolver a primeira versão do Facebook Messenger sobre Apache HBase, um banco de dados NoSQL (No Structured Query Language) que funciona em conjunto com o ecossistema do Hadoop, e, ao longo dos anos, desenvolveu diversas ferramentas de *big data* integradas com o Hadoop. A mais significativa delas foi o Hive, uma ferramenta que permitiu a análise de dados armazenados no Hadoop por meio da linguagem SQL, traduzindo consultas em programas MapReduce (CHAKRABORTY, 2017).

Caso de uso: eBay

Outra empresa que depende fortemente do Hadoop é o eBay, um dos maiores *e-commerce* do mundo. Em 2015, o eBay possuía um *cluster* com mais de 10 mil máquinas, 150 mil cores e mais de 150 petabytes de dados armazenados (PADMANABHAN; CHANDRAMOHAN; KANTH, 2015).

No eBay, um dos usos do Hadoop é para a criação de índices para seu sistema de busca. Esse caso de uso possui acordos de nível de serviço (SLAs, do inglês *service level agreements*) rígidos (de segundos até alguns minutos), pois tem impacto direto na receita da empresa. A solução é capaz de processar 1,68 milhões de itens em menos de 3 minutos. O armazenamento é feito pelo Apache HBase, que é utilizado para a construção dos índices de busca. Ao longo dos anos, o eBay também criou diversas ferramentas para garantir o monitoramento e a manutenção dos seus *clusters* Hadoop, além de ter criado o Apache Kylin, uma ferramenta de OLAP (*online analytical processing*) com interface SQL sobre o Hadoop.

Caso de uso: Salesforce

Um último caso de sucesso do Hadoop que vamos abordar é o da Salesforce, um dos maiores sistemas de gestão de relacionamento com o cliente (CRM, do inglês *customer relationship management*) do mercado em computação nas nuvens. Com mais de 130 mil consumidores e milhões de usuários, são bilhões de transações efetuadas todos os dias nos sistemas da Salesforce, por meio de diversos canais de acesso (API, *web* e *mobile*). Todos esses eventos (transações, interações, *clickstreams*) são centralizados em um único lugar e são de grande valor tanto para uso interno como para casos de uso em produtos (BHARADWAJ, 2013).

No uso interno, um dos casos de uso é o de métricas de produtos. Elas permitem que gerentes de produto compreendam a utilização e a adoção de suas funcionalidades, além de permitirem que eles tomem decisões com base em tendências, seja para adicionar novas funcionalidades ou para saber quando matar funcionalidades impopulares (BHARADWAJ, 2013). A Salesforce usa intensamente tecnologias do ecossistema Hadoop, como o Apache Pig, além de ter criado o Apache Phoenix, um banco de dados relacional massivamente paralelo que utiliza o Apache HBase para armazenamento de dados.



Fique atento

O MapReduce foi pioneiro na popularização do processamento de grandes volumes de dados, dando origem ao Hadoop. Atualmente, existem dezenas de ferramentas utilizadas para o processamento de dados, como Hive, Presto, Spark, Impala e Drill.

Implementações de casos de sucesso do MapReduce

O sucesso do Hadoop e de todo o seu ecossistema possibilitou muitos casos de sucesso, como foi visto na seção anterior. Nesta seção, vamos abordar alguns casos de sucesso que utilizaram o MapReduce e como esses casos foram implementados.

Um caso de sucesso de grande notoriedade do Hadoop é o da Uber, em que o crescimento constante no volume de dados forçou a empresa a mudar totalmente sua arquitetura, substituindo um *data warehouse* da Vertica por um ***data lake*** baseado em Hadoop (SHIFTEHRAR, 2018). A Figura 1 detalha a arquitetura adotada pela Uber, utilizando o Hadoop como base para seu *data lake*. O armazenamento e o processamento dos dados ficaram a cargo do Hadoop, sendo ainda utilizados o Apache Parquet como formato de armazenamento de arquivos, o Hive e o Presto para consultas *ad-hoc* sobre os dados armazenados no Hadoop, e o Spark como *framework* para processamento de dados.

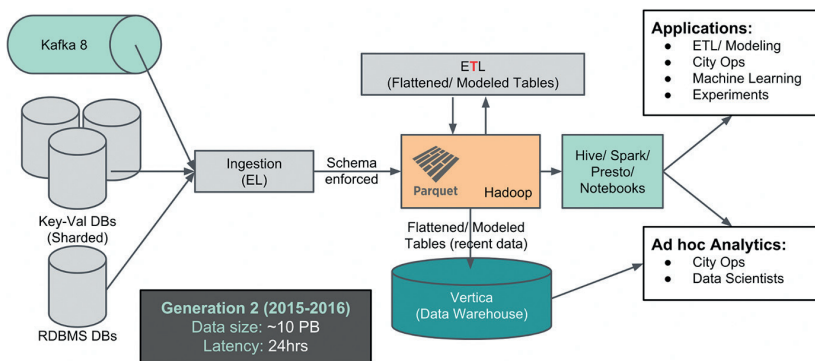


Figura 1. Arquitetura do *data lake* da Uber com Hadoop.

Fonte: Adaptada de SHIFTEHRAR (2018).

A arquitetura da Uber segue um padrão dentro das arquiteturas de dados para *data lakes*, com processos bem definidos para cada uma das etapas descritas a seguir.

- **Ingestão:** captura e transporte dos dados de suas respectivas origens para dentro do *data lake*, sejam as origens bancos de dados relacionais, NoSQL, sistemas de arquivos/objetos ou mensagens.
- **Armazenamento:** consiste em persistir de forma permanente os dados do *data lake*, permitindo o acesso por meio de diversas ferramentas. No Hadoop, a principal ferramenta de armazenamento é o Hadoop Distributed File System, um sistema de arquivos distribuídos, escalável e tolerante a falhas.

- **Transformação:** consiste na transformação dos dados armazenados no seu formato original para um formato mais amigável, realizando transformações nos dados e na sua estrutura. Nessa etapa, ferramentas como o MapReduce e o Spark são muito utilizadas.
- **Consulta/Entrega:** a última etapa é permitir a consulta dos dados, seja por consumidores finais, como analistas de dados ou cientistas de dados, seja por outros sistemas. Nesse caso, ferramentas de SQL sobre o Hadoop, como o Hive e o Presto, são ideais para consultas *ad-hoc*.

Outro caso de sucesso do MapReduce, o primeiro de todos, é o do Google, onde o modelo de programação MapReduce foi criado. Em 2007 o MapReduce chegou a ser a principal ferramenta para processamento de dados, com mais de 80% do processamento pesado de dados sendo executado nela (ALFONSECA, 2013). No Google, o MapReduce é utilizado para indexação da *web* em larga escala para o sistema de buscas, produção de relatórios para consultas populares do Google Trends, processamento de imagens de satélite, processamento de modelos de linguagem para tradução automática com uso de estatística, além de problemas de aprendizado de máquina em larga escala.

Uma implementação interessante do MapReduce feita no Google é para unir imagens de satélite para gerar uma única imagem e exibi-la para os usuários, como pode ser visualizado na Figura 2. Esse caso de uso é um problema complexo, pois envolve imagens de diferentes provedores, com formatos diferentes, coberturas diferentes, tiradas em momentos diferentes, com resoluções e níveis de exposição e cor diferentes. Sem contar que são grandes volumes de dados que precisam ser processados.

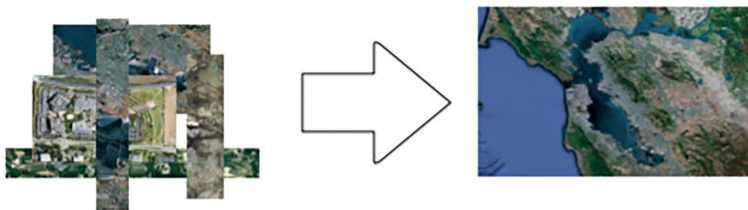


Figura 2. Imagens de satélite desconstruídas e uniformizadas.

A implementação desse caso de uso consiste em dividir o território em uma grade, com identificações fixas para cada localização. Então, divide-se cada imagem-fonte de acordo com as grades que a imagem cobre, e, para

cada pedaço da grade, unem-se as imagens das diversas fontes, com base na data da imagem e na resolução. A imagem final é servida para parte da grade, para que seja carregada e servida para os usuários finais.

O exemplo abaixo mostra a implementação da função de `map` em pseudocódigo do caso de uso das imagens de satélite do Google. A função recebe como chave o nome do arquivo da imagem e como valor os dados da imagem. Os dados da imagem são convertidos de acordo com o tipo de provedor da imagem; depois, a imagem é dividida em partes, de acordo com a divisão da grade. Cada parte é, então, enviada como valor, e a chave é a identificação da parte na grade.



Exemplo

```
função map(Texto chave, Texto valor):
    // chave: nome do arquivo de imagem

    // valor: dados da imagem
    imagem_completa = nulo

    para (pega_tipo_arquivo(chave)):
        se for provedor A:
            imagem_completa = converter_A(valor)
        se for provedor B:
            imagem_completa = converter_B(valor)
        ...

    // divide whole_image de acordo com a grade em partes
    para cada pedaço de imagem_completa:
        enviar(pedaço.ID_da_localização(), pedaço)
}
```

Já o método de `reduce`, mostrado no exemplo abaixo, recebe como chave a identificação da parte da grade e como valor a lista de imagens que correspondem a essa parte. As imagens são ordenadas de acordo com a resolução e a data em que a imagem foi tirada. O algoritmo percorre cada uma das imagens e preenche a imagem final com base na área que a imagem cobre. Ao final, toda a área da parte da grade deve ter sido preenchida. Então, a imagem é normalizada e enviada para ser armazenada e servida.



Exemplo

```
função reduce(Texto chave, Lista valores):
    // chave: ID da localização,

    // valores: pedaços de imagens de diferentes fontes
    // ordena os valores com base na resolução e data das
    imagens

    ordena valores por resolução e data em ordem decrescente

    imagem_unida = nulo

    para cada pedaço em valores:
        // une pixels da grade com base na cobertura de cada
        grade

        imagem_unida = juntaPixelsDaImagem(pedaço, imagem_unida)
        // suaviza imagem da grade unida

        imagem_unida = suavizarImagem(imagem_unida);
        enviar(chave, imagem_unida);
    }
}
```

Além do Google, o MapReduce foi utilizado também pelo Yahoo! para mapear a internet e para detecção de *spam*. No Facebook, ele também foi utilizado para criação de relatórios e aprendizado de máquina.

Desafios para extração de informações no Hadoop

O MapReduce é influenciado pela programação funcional e, como o nome sugere, é composto por duas fases: a fase de *map* e a fase de *reduce*. Ambas as fases possuem como entrada e saída pares de chave/valor. Fica a cargo do desenvolvedor especificar as funções de *map* e de *reduce*, bem como o tipo de dados para os pares de chave/valor em cada fase.

A Figura 3 mostra uma aplicação de *wordcount*, uma aplicação com propósito demonstrativo muito utilizada em *frameworks* de *big data*. Ela consiste

em contar o número de ocorrências de cada palavra contida em um *dataset* textual. Na aplicação de *wordcount*, a entrada de dados é um *dataset* com texto. A etapa de *map* vai receber pares de chave/valor compostos pelo número da linha como chave e o texto da linha como valor. A função *map* vai quebrar o texto em palavras e enviar como saída pares de chave/valor compostos pela palavra como chave e um contador com valor 1.

Após a etapa de *map* e antes da etapa de *reduce*, ocorre uma operação chamada de *shuffle*. Essa etapa, de acordo com White (2015), é responsável por ordenar as saídas do *map* para que elas possam ser entregues para a etapa de *reduce*. O processo em si é mais complexo, envolve o particionamento das saídas do *map* para cada tarefa de *reduce*, bem como a cópia desses dados para as máquinas onde as tarefas de *reduce* vão executar e o ordenamento desses dados. Quando os dados de todas as tarefas de *map* forem copiados, a tarefa de *reduce* pode iniciar.

Na etapa de *reduce*, a entrada são pares de chave/valor compostos pela palavra como chave e a lista de valores como valor. A função de *reduce* então soma os valores e emite um novo par com a palavra como chave e a soma dos valores como valor. No final, a saída será a frequência de todas as palavras do texto. O exemplo da Figura 3 mostra a complexidade de se desenvolver uma aplicação com MapReduce, apesar de o *wordcount* ser uma aplicação relativamente simples.

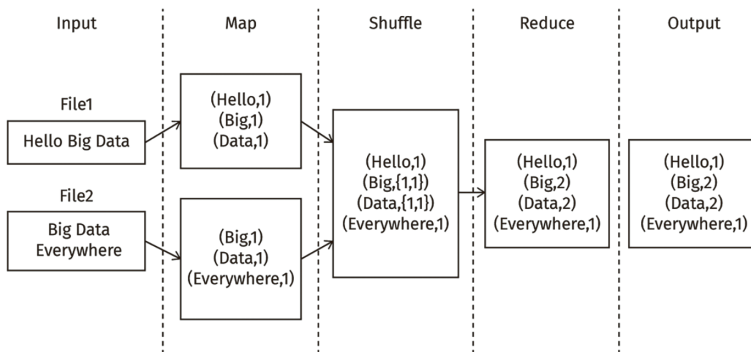


Figura 3. Exemplo do MapReduce para aplicação de *wordcount*.

Fonte: Adaptada de SIDDESH; HIRIYANNAIAH; SRINIVASA (2014).

Programas em MapReduce são simples, mas não o suficiente para permitir a expressão de aplicações úteis. Em outras palavras, para desenvolver uma aplicação de análise de dados que seja útil, muito provavelmente será preciso

criar diversos programas em MapReduce encadeados, cada um se alimentando do resultado do programa anterior. Esse processo é chamado de *workflow* e pode ser desenvolvido utilizando-se a API do MapReduce no Hadoop.

Existem casos mais complexos, em que as dependências entre *jobs* formam um grafo acíclico dirigido, e é preciso informar as dependências existentes entre os *jobs*. Nesses casos mais complexos, é possível utilizar a API JobControl do Hadoop ou utilizar orquestradores externos, como o Apache Oozie ou o Apache Airflow. No intuito de facilitar o uso do poder do MapReduce com o Hadoop, diversas ferramentas foram desenvolvidas para abstrair as complexidades de programas MapReduce por meio de linguagens de alto nível. Dentre essas ferramentas, as mais notáveis são Hive, Pig, Cascading, Crunch e Spark.

Os desafios na utilização do Hadoop começam com a instalação da plataforma em si, tarefa que exige tempo para a instalação dos requisitos e preparação das máquinas para que possam ser utilizadas. Existem diversas ferramentas criadas com o intuito de facilitar e automatizar esses processos, como Terraform, Ansible, Chef e Puppet. Essas ferramentas permitem que o processo todo de instalação e preparação de um ambiente Hadoop seja automatizado. Além disso, provedores de nuvem geralmente possuem ambientes Hadoop prontos, bastando apenas alguns cliques para serem utilizados.

Ultrapassado o desafio de subir o ambiente Hadoop, surgem os desafios de ingerir, processar, armazenar e analisar os dados no Hadoop. Para cada uma dessas tarefas, existem ferramentas específicas criadas para trabalhar dentro do ambiente Hadoop. O desafio está em saber escolher a ferramenta correta para cada tarefa e cada etapa do processo, desde a ingestão de dados, o armazenamento, a transformação e a consulta dos dados armazenados.

Referências

ALFONSECA, E. *MapReduce and its use for indexing: the programming model and practice*. 2013. Disponível em: http://www.ugr.es/~essir2013/slides/ESSIR_MapReduce_for_Indexing.pdf. Acesso em: 10 set. 2020.

BHARADWAJ, N. Hadoop and big data: use cases at salesforce.com. *Salesforce Developers Blog*, San Francisco, 6 Mar. 2013. Disponível em: <https://developer.salesforce.com/blogs/engineering/2013/03/hadoop-use-cases-at-salesforce-com.html>. Acesso em: 10 set. 2020.

CHAKRABORTY, S. How is Facebook deploying big data? *Big Data Zone*, New York, 6 Feb. 2017. Disponível em: <https://dzone.com/articles/how-is-facebook-deploying-big-data>. Acesso em: 10 set. 2020.

DEAN, J.; GHEMAWAT, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, New York, v. 51, n. 1, p. 107-113, Jan. 2008.

PADMANABHAN, A. M.; CHANDRAMOHAN, R.; KANTH, V. *Hadoop at ebay*. 2015. Disponível em: <https://pt.slideshare.net/aroomp/hadoop-at-ebay-53095161>. Acesso em: 10 set. 2020.

SARMA, J. S. Hadoop. *Facebook Engineering* : Facebook, 5 Jun. 2008. Disponível em: <https://www.facebook.com/notes/facebook-engineering/hadoop/16121578919/>. Acesso em: 10 set. 2020.

SHIFTEHRAR, R. Uber's Big Data Platform: 100+ Petabytes with minute latency. *Uber Engineering*, 17 Oct. 2018. Disponível em: <https://eng.uber.com/uber-big-data-platform/>. Acesso em: 10 set. 2020.

SIDDESH, G. M.; HIRIYANNAIAH, S.; SRINIVASA, K. G. Driving big data with Hadoop technologies. In: DEKA, G. C.; RAJ, P. *Handbook of research on cloud infrastructures for big data analytics*. Hershey: IGI Global, 2014. p. 232-262.

WHITE, T. *Hadoop: the definitive guide*. 4th ed. Sebastopol, CA: O'Reilly, 2015.

Leituras recomendadas

GROVER, M. *et al.* *Hadoop application architectures: designing real-world big data applications*. Sebastopol, CA: O'Reilly, 2015.

KUNIGK, J. *et al.* *Architecting modern data platforms: a guide to enterprise hadoop at scale*. Sebastopol, CA: O'Reilly, 2018.

SINGH, C.; KUMAR, M. *Mastering Hadoop 3: big data processing at scale to unlock unique business insights*. Birmingham: Packt Publishing, 2019.



Fique atento

Os *links* para sites da web fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS