FUNDAMENTOS DE BIG DATA

Sergio Eduardo Nunes

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Hadoop: ecossistema de processamento em *big data*

OBJETIVOS DE APRENDIZAGEM

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Explicar quais são as características desejadas de um ambiente de processamento para *big data*.
- Reconhecer o Hadoop e as suas principais características, inclusive a sua linha do tempo.
- Descrever as principais ferramentas do Hadoop.

Introdução

Neste capítulo, você vai identificar quais são os cenários potencialmente propícios à utilização e à exploração do *big data*. Para tanto, você vai estudar um ecossistema de processamento de dados conhecido por Apache Hadoop, verificando como os seus componentes são utilizados no processamento de dados em ambientes com aglomerado de máquinas (*cluster*). A partir desses conhecimentos, você desenvolver

uma capacidade técnica básica para a identificação de cenários aplicáveis do *big data* e vai conhecer as ferramentas do Hadoop e como elas podem ser utilizadas no processamento de massa de dados.

Hadoop

O gerenciamento de dados se tornou um grande desafio aos profissionais de tecnologia da informação (TI). Isso porque o advento de novas tecnologias e o acesso da população aos dispositivos do tipo *smart* (*smart* TV, *smartphone* etc.) e à internet móvel colaboraram para o surgimento de uma inundação de dados na rede mundial de computadores. Segundo Sammer (2012), as redes sociais, os sistemas *web*, os buscadores, entre outros meios, produzem diariamente um volume de dados em petabytes, o que representa um grande valor de mercado.

Para que você possa compreender como a mudança de comportamento dos usuários pode refletir no gerenciamento de TI nas empresas, vamos analisar o caso do Facebook. Quando a rede social foi lançada, o banco de dados utilizado era do tipo relacional, o que, na época, pelo número de usuários, era suficiente para atender à demanda. Porém, por volta de 2007, a quantidade de dados gerados já alcançava alguns terabytes. Isso tornou necessário um novo tipo de banco de dados para atender às características da rede social. Foi nesse ponto que o banco de dados não relacional entrou para atender às necessidades da empresa.

O *big data* e as suas tecnologias surgiram como ferramentas para a solução de aplicações com grande volume de dados. Porém, para resolver alguns grandes desafios computacionais, os engenheiros de desenvolvimento de sistemas de gerenciamento de banco de dados precisaram propor opções de arquiteturas, para que fossem atendido os mais diversos cenários. Com isso, segundo Silberschatz (2010), os bancos de dados não relacionais passaram a oferecer sistemas de gerenciamento de banco de dados com arquitetura paralela e distribuída (*cluster*).

Silberschatz (2010) define que, na **arquitetura paralela** (Figura 1), em um sistema de gerenciamento de banco de dados, é possível que muitas operações sejam executadas simultaneamente. Isso porque um servidor em arquitetura paralela possibilita que os dados sejam processados em

diferentes dispositivos e paralelamente. Isso faz com que a carga de processamento não necessariamente ocorra em um único servidor; o disco de armazenamento e a memória podem ser compartilhados, ou o sistema pode funcionar sem compartilhamento de *hardware*. Exemplos desse tipo de sistemas de gerenciamento de banco de dados são aqueles utilizados por redes bancárias, em que as operações podem ser feitas paralelamente por mais de um servidor.

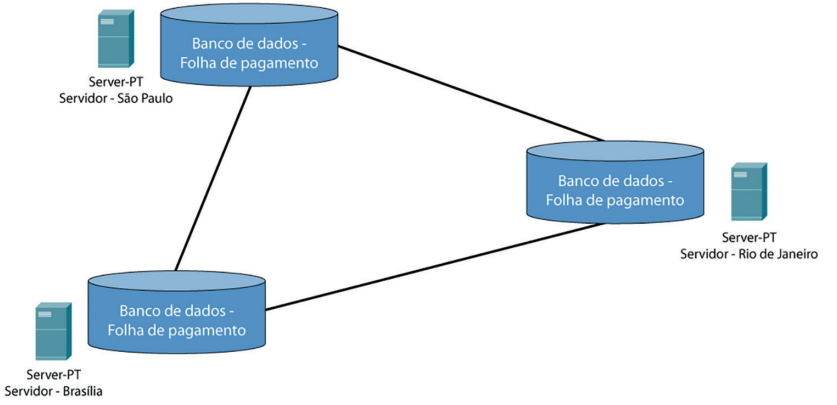

Figura 1. Arquitetura de banco de dados paralelo.

Segundo Tanenbaum (1997), um sistema do banco de dados com **arquitetura distribuída** (Figura 2) possui uma dependência da infraestrutura de redes de computadores para que se alcance uma *performance* dentro do padrão de qualidade de serviço desejado. Silberschatz (2010) define que, nos bancos de dados distribuídos, os dados estão armazenados em diversos servidores, e cada sistema de gerenciamento de banco de dados tem gerenciamento independente dos demais. Essa arquitetura é considerada por alguns profissionais como uma junção de banco de dados e redes de computadores. Um exemplo desse tipo de arquitetura em banco de dados são os *sites* de busca de passagens aéreas, reservas de hotéis e demais serviços, em que existe uma busca em diversas bases de dados distribuídas, sendo que essas, a nível de redes de computadores, estão geograficamente distribuídas.

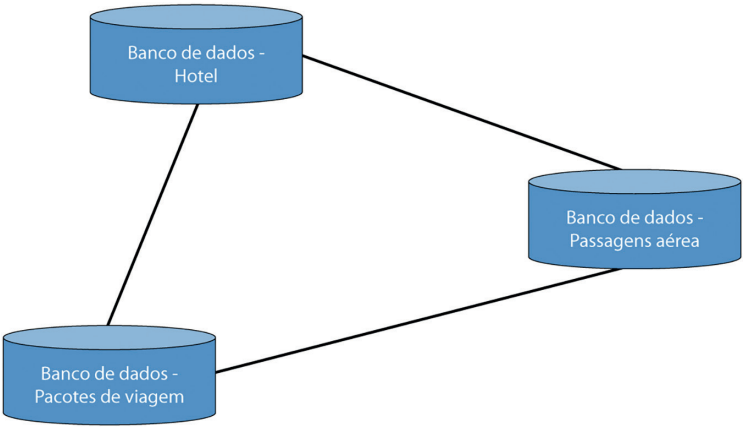

Figura 2. Arquitetura de banco de dados distribuído.

Dentro desse contexto, surge uma solução para diferentes necessidades de estrutura, capacidade de processamento massivo de dados e suporte a diferentes arquiteturas: o ecossistema **Hadoop**. Desenvolvido pelo projeto Apache Hadoop, o Hadoop é um ecossistema de *big data* que propõe soluções eficientes em múltiplas arquiteturas, com integridade dos dados, alta disponibilidade dos nós (servidores em *cluster*), processamento rápido em grande quantidade de dados, entre diversas outras vantagens.

Saiba mais

Para saber mais sobre o Hadoop, acesse o *site* da Apache Hadoop. O *site* oficial não possui versão em português — se você tem dificuldade na leitura em língua inglesa, é aconselhável utilizar o tradutor do navegador. Nesse *site*, você vai ter acesso a recursos como informações de novas versões, módulos do Hadoop, *download*, documentação completa, lista de discussões e demais repositórios de conhecimento. O projeto, desde a sua concepção até a presente data, apresentou uma evolução significativa (APACHE HADOOP, 2020b).

Linha do tempo do Hadoop

O Hadoop é um ecossistema que surgiu como uma importante solução para processamento de grande quantidade de dados. A primeira versão surgiu em outubro de 2006, sendo a 0.15.0; em 2020, foi disponibilizada a

versão 3.3.0. A seguir, com base no histórico de versões apresentado no *site* oficial da Apache Hadoop, serão apresentados alguns pontos marcantes de sua evolução.

- Ano de 2003: a Google estava em busca de uma solução tecnológica para melhorar o sistema de busca. Com isso, foi desenvolvida uma tecnologia conhecida por MapReduce (apresentada em detalhes mais adiante neste capítulo), que visa a otimizar a indexação e a catalogação de dados em páginas e aplicações *web*. Ainda, nesse mesmo ano, foi lançada uma tecnologia conhecida como Google File System, que é basicamente um sistema de banco de dados de forma distribuída, que tem a capacidade de processamento de grande volume de dados por meio do MapReduce.
- Ano de 2006: essa é uma data muito importante para o projeto. Nesse ano, o Yahoo começava a enfrentar dificuldades de indexação de páginas, idênticas àquelas enfrentadas pela Google em 2003. A empresa, então, começou a investir no projeto denominado Hadoop. Quando o Yahoo entrou no projeto, havia um aglomerado de 1.000 servidores executando o Hadoop; anos depois, já contava com cerca de 40.000 servidores.
- Ano de 2008: o Hadoop se tornou o principal projeto da Apache. Em uma demonstração da potencialidade de aplicações com o Hadoop, por meio de um *cluster* com 900 servidores, o Hadoop conseguiu efetuar um processamento de ordenação de 1 terabyte de dados em apenas 209 segundos.
- Ano de 2009: a empresa Cloudera efetuou a redistribuição de uma versão comercial do Apache Hadoop, que faz a integração de diversos outros projetos do Hadoop em uma única solução de sistema de gerenciamento de banco de dados com tais características.
- Ano de 2011: finalmente, o Hadoop chegou na sua versão 1.0.0, e as grandes mudanças ocorreram no sistema de autenticação, no suporte a tabelas grandes e no suporte à escrita e leitura de dados por meio de uma interface HTTP (do inglês *Hypertext Transfer Protocol*, ou Protocolo de Transferência de Hipertexto).
- Ano de 2020: foi lançada a versão 3.3.0 do Hadoop, cujas principais mudanças ocorreram no suporte a nós no YARN (solução do Hadoop), nas políticas de armazenamento de dados, no suporte a Azure, entre outras atualizações.

Saiba mais

Quando o Yahoo entrou no projeto, a empresa contratou um grande profissional chamado Doug Cutting, sendo ele um dos responsáveis por fundar o projeto em código aberto batizado de Hadoop. O projeto foi denominado Hadoop em referência a um elefante amarelo de pelúcia com esse nome, que pertencia ao filho de Cutting.

O logotipo do Apache Hadoop é muito popular entre os desenvolvedores de soluções em *big data* e pode ser observado na figura abaixo.

Fonte: Apache Hadoop (2020a, documento *on-line*).

Conforme mencionado anteriormente, o Hadoop possui alguns subprojetos, que visam a criar soluções para diversos tipos de aplicações. Os subprojetos apresentados a seguir foram consultados no *site* oficial do Apache Hadoop:

- Hadoop Common: trata-se de um conjunto de ferramentas, e a sua estrutura é utilizada em muitos projetos. É utilizado para a manipulação de dados e possui uma interface desenvolvida pela empresa Amazon. Você pode encontrar esse recurso no *site* Maven Repository (em inglês) (APACHE HADOOP, 2020c).
- Hadoop MapReduce: é uma solução para programação voltada para o processamento de dados distribuído em um *cluster*. Quanto à sua forma de atuação, normalmente, o conjunto de dados é dividido em partes, que serão processadas de forma independente dentro do mapa. A estrutura classifica as saídas geradas pelo mapa, que são inseridas em um processo de redução e finalmente são armazenadas em sistemas de arquivos. A sua configuração e documentação está disponível no *site* do Hadoop (em inglês) (APACHE HADOOP, 2020d).
- Hadoop Distributed File System (HDFS): é um mecanismo de tolerância à falha que utiliza mecanismos de armazenamento e transmissão dos dados. Nas últimas versões, o HDFS é nativo do Apache Hadoop.
- Hbase: é uma solução desenvolvida pela empresa Google, que visa a dar suporte ao armazenamento estruturado e otimizado, quando

se tem a necessidade de ter tabelas grandes; é também conhecido como *BigTable*.
- ZooKeeper: é uma solução desenvolvida pelo Yahoo, que visa a promover soluções em aplicações distribuídas de altíssimo desempenho.

O Hadoop possui outros projetos e soluções, porém, não caberia aqui citar todos eles. Percebe-se que as maiores contribuições e interesses partiram das empresas Google e Yahoo, para suportar as suas respectivas demandas, a fim de garantir a qualidade e a disponibilidade de seus serviços. Salienta-se que, por se tratar de um *software* da Apache, o seu código é aberto, permitindo modificações e redistribuição.

Características do Hadoop

O Hadoop é um ecossistema de *big data* que possui diversos componentes que moldam as suas características técnicas. Sammer (2012) define as funcionalidades apresentadas a seguir.

- ***NameNode*:** tem a função de mapear a localização, dividir os arquivos em blocos, encaminhar os dados aos nós escravos e gerenciar a localização das réplicas dos dados. Ainda, a sua função é integrar o HDFS (nó mestre) ao *JobTracker*, para garantir o desempenho.
- ***DataNode*:** é o responsável por gerenciar os blocos de arquivos. O HDFS tem o seu funcionamento em sistemas distribuídos, para que possam ser enviados para os nós escravos; então, o *DataNode* faz o gerenciamento dos blocos. Além disso, a sua função é transmitir informações constantemente ao *NameNode*, informando o *status* dos blocos.
- ***JobTracker*:** é o gerenciador de processamento do MapReduce. A sua função básica é designar o nó que deve gerenciar determinado dado e, ainda, verificar falhas, reenviar os dados em caso de falha, reiniciar um nó e trocar o nó que deve processar os dados.
- ***TaskTracker*:** é responsável por executar as tarefas do MapReduce dentro dos nós escravos. Essa funcionalidade é executada em máquina virtual; dessa forma, é possível criar mais de uma máquina virtual em

um mesmo servidor, para que alguns recursos possam ser mais bem aproveitados.

- ***SecondaryNameNode*:** auxilia o *NameNode* em seu funcionamento, fazendo as checagens e garantindo a sua recuperação em caso de falhas. Para isso, o *SecondaryNameNode* cria pontos de recuperação; assim, caso ocorra falha, o Hadoop volta ao último ponto sem falhas.

Para auxiliar na compreensão da forma como os componentes interagem dentro do Hadoop, observe a Figura 3.

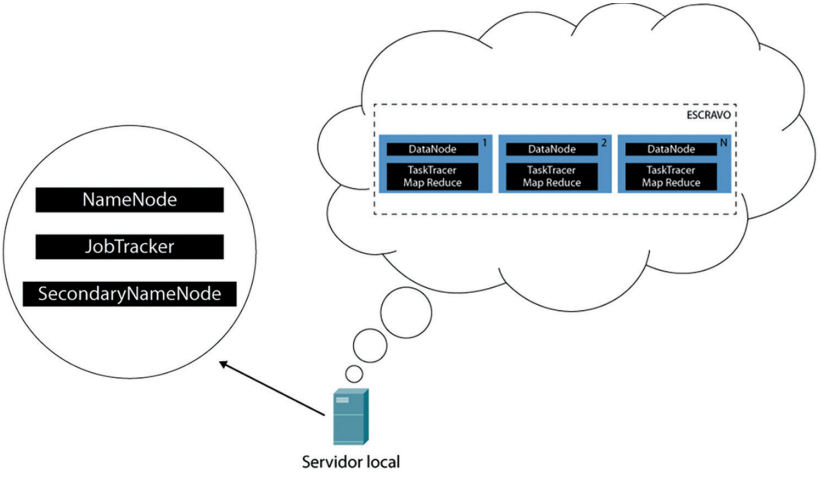

Figura 3. Processos de funcionamento dos componentes no Hadoop.

Os processos entre mestres e escravos são executados em camadas diferentes. Um cliente se conecta ao nó mestre, que solicita que os processos sejam executados. Nesse momento, o *NameNode* faz o gerenciamento das informações dos arquivos que estão sendo processados. Já no nó escravo, o *TaskReducer* executa as tarefas, como o MapReduce e o *DataNode*, ao mesmo tempo que o nó escravo atualiza o seu *status* junto ao nó mestre. Enquanto ocorrem esses processos, o *SecondaryNameNode* efetua pontos de checagem, para o caso de ocorrer falha e ser necessário fazer uma recuperação do Hadoop.

MapReduce

Segundo Sammer (2012), o MapReduce é um paradigma de linguagem de programação, mas, independentemente da tecnologia em que ele será implementado, a sua ideia consiste em aplicar funções nas entradas de valores, de forma a reduzir a saída em um único valor. Para explicar e exemplificar o seu conceito, vamos nos apropriar da teoria dos conjuntos, conforme pode ser observado a seguir:

```
>> Map ({1, 2, 3, 4, 5}, * 10)
>> {10, 20, 30, 40, 50}
```

Nesse exemplo, foi enviado por meio do Map um conjunto numérico e um multiplicador de valor 10, que deve ser aplicado em todos os elementos do conjunto. Como saída, foi gerado o conjunto dos valores multiplicados por 10. Essa função de mapeamento de dado poderia ser chamada de "multiplicaDez", conforme pode ser observado:

```
>> Map ({1, 2, 3, 4, 5}, * multiplicaDez)
>> {10, 20, 30, 40, 50}
```

Sammer (2012) explica que a função Reduce vai receber essa lista e aplicar uma função qualquer, de forma que ocorra a redução para uma única saída gerada. Para a compreensão desse conceito, observe o exemplo demonstrado a seguir:

```
>> Reduce ({10, 20, 30, 40, 50}, mínimo)
>> 10
```

Nesse exemplo, foi aplicado o Reduce em uma lista, em que a função visava a encontrar um valor mínimo. Podem ainda ser utilizadas outras funções, como máximo, média etc. Porém, dentro desse paradigma, o Map e o Reduce devem ser utilizados em conjunto, formando, assim, o MapReduce. Para a compreensão desse conceito, observe o exemplo a seguir:

```
>> Reduce (Map({1, 2, 3, 4, 5}, triplo), média)
>> 9
```

Nesse exemplo, ocorreu a seguinte execução:

- o Map aplicou a função triplo na lista de entrada, resultando em {3, 6, 9, 12, 15};
- em seguida, o Reduce recebeu a lista gerada pelo Map;
- o Reduce aplicou a função média, que resultou em um valor único — nesse caso, o 9.

Fique atento

A função do Reduce visa a minimizar o resultado da consulta a um único valor, que é bem diferente de encontrar um valor mínimo em um conjunto de dados. Por exemplo, considere o conjunto numérico N = {2, 8, 2, 7, 5, 2, 3, 1}. Ao se minimizar o conjunto de dados aos valores que mais se repetem, o resultado seria o valor 2, que se repete três vezes. Isso é diferente de encontrar o valor mínimo, em que seria gerado o valor 1, pois é o menor valor encontrado na lista.

O MapReduce, no Hadoop, tem uma aplicação muito similar ao exemplo utilizado para a compreensão do seu mecanismo. Observe a sequência a seguir:

1. o Map utiliza os blocos de arquivos como entrada de dados;
2. com as saídas produzidas pelo Map, o Reduce aplica a sua função;
3. é gerado o resultado da busca, representado por um único valor.

Percebeu como o exemplo anterior e o exemplo do MapReduce apresentam ações idênticas? Pois bem, agora que você compreendeu o funcionamento do Hadoop, vamos discutir as formas como ele pode ser instalado para utilização, ou seja, como o servidor será configurado. Segundo Lam (2010), existem três formas para se utilizar o Hadoop, sendo elas: modo local (*localhost* ou *standalone mode*), modo pseudodistribuído (*pseudo-distributed mode*) e modo distribuído (*distributed mode*).

Modo local

O **modo local** é o modo mais comum de instalação para iniciar os estudos acerca do Hadoop. Embora tenha sido afirmado ao longo do texto que o Hadoop foi desenvolvido para atuar em um conjunto de máquinas, o projeto Apache entendeu que, em modo local, o desenvolvedor teria um ambiente

mais adequado para o desenvolvimento e os testes. Além do mais, a instalação e a configuração padrão do Hadoop é o modo local. Dessa forma, os arquivos não precisam ser alterados, bem como não é necessário instalar o HDFS, pois o Hadoop não será utilizado em processamento de dados distribuídos.

Para configurar um *cluster* de nó único, existe um manual desenvolvido pela Apache Hadoop, sendo possível seguir os passos para executar rapidamente operações simples usando o Hadoop. Lembre-se de que o Hadoop não possui documentação na língua portuguesa, então, caso você tenha dificuldade com a língua inglesa, utilize o tradutor de páginas do seu navegador. Outra característica é que toda a documentação orienta para a instalação e a configuração em ambiente Linux (APACHE HADOOP, 2020e).

Exemplo

Quando estamos desenvolvendo alguma aplicação *web*, no primeiro momento, é instalado um servidor de banco de dados e um servidor HTTP. Um exemplo é a ferramenta WAMP, para o sistema Windows, em que o servidor HTTP é o Apache, o servidor de banco de dados é o MySQL, e o servidor é PHP (*hypertext preprocessor*) *back-end*. Isso tudo funciona em *localhost*, ou seja, em uma máquina local, assim como acontece com o modo local do Hadoop.

Modo pseudodistribuído

Em um segundo momento, quando você já se sentir apto a avançar nos estudos, o **modo pseudodistribuído** é o ideal. O processo de instalação vai seguir os mesmos processos para o *cluster*, assim como a sua configuração, mas o servidor vai agir como um *cluster* de uma máquina só; ou seja, as execuções serão emuladas dentro do servidor — daí o nome pseudodistribuído. Dessa forma, os componentes como *NameNode*, *DataNode*, *SecondaryNameNode* etc. vão funcionar e fazer o processamento dos dados de forma distribuída.

Para auxiliar a compreensão da forma de agir do Hadoop no modo pseudodistribuído, observe a Figura 4.

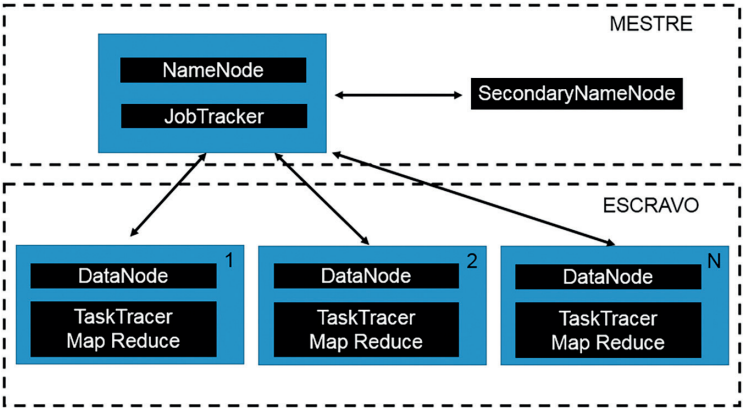

Figura 4. Hadoop em modo pseudodistribuído.

Observe que, no servidor local, estão instalados e configurados os componentes para processamento de dados em *cluster*; por sua vez, os escravos estão em um modo simulado. Porém, tudo isso está ocorrendo em uma única máquina (servidor local). Por isso, alguns profissionais da área apelidaram esse modo de *cluster* de uma única máquina.

Modo distribuído

Para utilizar o **modo distribuído**, você já deve ter uma boa familiarização com o Hadoop. Além disso, é necessário ter habilidades relacionadas a redes de computadores, visto que as máquinas estarão dispostas em uma topologia que pode possuir os mais diferentes dispositivos, tecnologias, equipamentos intermediários e protocolos de comunicação em rede.

Saiba mais

Para poder instalar e configurar o Hadoop no modo distribuído, é necessário que se tenha alguns conhecimentos básicos acerca de redes de computadores. Nesse contexto, a Teleco é um centro de pesquisa de redes de computadores e outros assuntos relacionados. Acesse o *site* da Teleco e saiba mais sobre como instalar e configurar o Hadoop.

O processo de instalação deve ocorrer em todas as máquinas que vão compor o *cluster*, sendo atribuídas as devidas configurações para o servidor

Master e as demais para as máquinas *Slave*. As configurações dos componentes são completamente diferentes do modo pseudodistribuído, uma vez que os nós escravos serão máquinas físicas, e não simuladas. Observe um exemplo de uma topologia com um *cluster* de forma distribuída geograficamente na Figura 5.

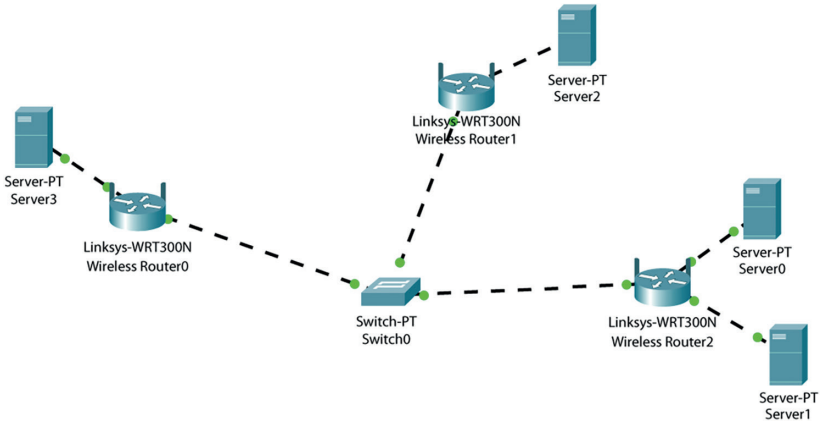

Figura 5. Topologia para o modo distribuído.

Observe que o *switch* (Switch0) é o nó central da rede, que se liga aos roteadores (Wireless Router0, Wireless Router1 e Wireless Router2), que estão em suas respectivas subredes — ou seja, geograficamente distribuídos. Em cada um dos roteadores, existe ao menos um servidor conectado (*Server0*, *Server1*, *Server2* e *Server3*). Nesses servidores, está instalado e configurado o Hadoop. Mas, para que possa ocorrer o processamento de forma distribuída, deve ocorrer uma estruturação hierárquica entre os servidores. Poderíamos supor, nesse exemplo, que a máquina *Server0* assumiria como Mestre, e as demais seriam escravos — isto é, o *Server1*, o *Server2* e o *Server3*.

Você deve ter percebido quanta flexibilidade o Hadoop permite aos desenvolvedores; mas não é apenas essa vantagem que pode ser encontrada. Veja a seguir outras vantagens desse ecossistema.

- Código aberto: o projeto Apache surgiu com o intuito de permitir que desenvolvedores possam evoluir as suas ferramentas, com um trabalho

no formato colaborativo. Essa liberdade não está somente no código aberto, mas também no acesso a arquivos, tutoriais e documentações.

- Licença: existem empresas que desenvolvem soluções com base no Hadoop e possuem licenças de uso e/ou redistribuição. Porém, os componentes básicos apresentados ao longo deste capítulo não requerem nenhum tipo de licença, independentemente do tipo de uso.
- Comunidade: a comunidade disponibiliza diversas listas de discussões, e qualquer um pode se inscrever para ter acesso a soluções e dicas (APACHE HADOOP, 2020f).
- Economia: esse ponto está relacionado à potencialidade do Hadoop de processar grande volume de dados em um curto espaço de tempo, sem que seja necessário possuir máquinas de grande capacidade de *hardware*.
- Escalabilidade: o processo para se adicionar outros nós é relativamente simples e repetitivo. A Amazon possui uma solução (Elastic MapReduce) pela qual é possível alugar um aglomerado de máquinas em nuvem e processar dados em nuvem.
- Capacidade: é nesse ponto que o projeto Hadoop se coloca à frente de outros produtos similares disponíveis no mercado. A sua capacidade de processamento, mesmo em máquinas mais simples, se mostra de grande potencialidade no processamento de massa de dados.
- Simplicidade: a ideia central do Hadoop gira em torno das técnicas do mapeamento de dados (Map) e do processo de redução (Reduce), assim, tirando da função do desenvolvedor algumas atividades, como balanceamento de carga e gerenciamento de falhas.

Esses são alguns dos pontos mais destacados por especialistas e demais profissionais que trabalham com o processamento de grande massa de dados. Porém, toda e qualquer tecnologia, apresenta potencialidades e fraquezas, e com o Hadoop não é diferente. Veja a seguir algumas desvantagens encontradas no Hadoop.

- Limitação do modo local: o modo local auxilia o desenvolvedor a compreender os aspectos básicos relacionados à instalação, à configuração e aos testes de processamento de dados. Porém, não é possível criar pontos de checagem para testes de recuperação de falhas. Ainda, esse modo não é escalável, uma vez que possui apenas uma máquina no nó.
- Gerenciamento de nós escravos: essa desvantagem diz respeito ao nível de dificuldade para a configuração dos componentes de gerenciamento de nós escravos. Apesar de bem documentados, são muitos passos a

serem seguidos, sendo necessário, ainda, efetuar o mapeamento dos dispositivos disponíveis na rede de computadores que compõem o *cluster*.

- Quantidade de dados: por incrível que pareça, o Hadoop não se dá bem com muitas funções na configuração do MapReduce em um conjunto de dados de tamanho reduzido. Isso prova que o Hadoop foi desenvolvido para processar grandes volumes de dados.

Perceba que, apesar de algumas desvantagens, as vantagens do Hadoop se sobressaem. Essa é a percepção que a grande maioria dos profissionais de análise de dados tem a respeito do Hadoop. Não é à toa que empresas como Adobe, Facebook, LinkedIn, Twitter, Yahoo, entre outras, passaram a utilizar em sua infraestrutura ferramentas do Hadoop, para efetuar análises de dados de forma eficiente.

Com base no exposto, entende-se que o Hadoop é um ecossistema para análise de dados que, apesar de apresentar um nível de dificuldade mais acentuado nos modos mais avançados, demonstra simplicidade na forma de funcionamento, o que faz com que seja muito eficiente e prático quando em ação. Os componentes possuem o seu funcionamento bem definido, fazendo com que a sua forma de agir seja facilmente compreendida. O projeto da Apache tem, em sua essência, a ideia de criar soluções corporativas muito bem documentadas. Qualquer pessoa pode, por meio do *site* oficial, fazer *download*, consultar as documentações e utilizar os fóruns para buscar respostas e soluções. Tudo isso se refletiu em uma grande popularidade, fazendo com que muitas empresas aderissem ao Hadoop (APACHE HADDOP, 2019).

Fique atento

A certificação é um recurso que o profissional de TI pode utilizar para mostrar que tem conhecimento técnico para desenvolver em determinada tecnologia. Por exemplo, a Cloudera (2020a; 2020b; 2020c), que desenvolve soluções em Hadoop, oferece uma carreira muito interessante relacionada a esse ecossistema, apresentando as seguintes certificações:

- Cloudera Certified Professional — *Data Engineer*;
- Cloudera Certified Associate — *Data Analyst*;
- Cloudera Certified Associate — *Administrator.*

Referências

APACHE HADOOP. *Apache Hadoop Common*. Maryland: Apache Software Foundation, 2020c. Disponível em: https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-common/. Acesso em: 18 ago. 2020.

APACHE HADOOP. *Apache Hadoop 3.3.0*. Maryland: Apache Software Foundation, 2020b. Disponível em: https://hadoop.apache.org/docs/current/. Acesso em: 18 ago. 2020.

APACHE HADOOP. *Download*. Maryland: Apache Software Foundation, 2019. Disponível em: https://hadoop.apache.org/releases.html. Acesso em: 18 ago. 2020.

APACHE HADOOP. *Hadoop mailing lists*. Maryland: Apache Software Foundation, 2020f. Disponível em: https://hadoop.apache.org/mailing_lists.html. Acesso em: 18 ago. 2020.

APACHE HADOOP. *Hadoop: setting up a single node cluster*. Maryland: Apache Software Foundation, 2020e. Disponível em: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html. Acesso em: 18 ago. 2020.

APACHE HADOOP. *MapReduce Tutorial*. Maryland: Apache Software Foundation, 2020d. Disponível em: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html. Acesso em: 18 ago. 2020.

APACHE HADOOP. Maryland: Apache Software Foundation, 2020a. Disponível em: https://hadoop.apache.org/. Acesso em: 18 ago. 2020.

CLOUDERA. *CCA administrator certification*. California: Cloudera, 2020a. Disponível em: https://www.cloudera.com/about/training/certification/cca-admin.html. Acesso em: 18 ago. 2020.

CLOUDERA. *CCA data analyst*. California: Cloudera, 2020b. Disponível em: https://www.cloudera.com/about/training/certification/cca-data-analyst.html. Acesso em: 18 ago. 2020.

CLOUDERA. *CCP data engineer*. California: Cloudera, 2020c. Disponível em: https://www.cloudera.com/about/training/certification/ccp-data-engineer.html. Acesso em: 18 ago. 2020.

LAM, C. *Hadoop in Action*. Nova York: Manning Publications, 2010.

SAMMER, E. *Hadoop operations*. California: O'Reilly Media, 2012.

SILBERSCHATZ, A. *Sistemas de banco de dados*. 5. ed. Rio de Janeiro: Elsevier, 2010.

TANENBAUM, A. S. *Redes de computadores*. 4. ed. São Paulo: Campus, 1997.

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:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS