Tema 2

Hadoop e Armazenamento de Dados

É fundamental que o profissional de Tecnologia da Informação compreenda a tecnologia Hadoop para gerenciar uma infraestrutura complexa e desenvolver aplicações para uma das áreas que mais cresce na atualidade: Big Data.

Tempo total de leitura
126 min.

Créditos

F
Professor (a)
FERNANDO DURIER

Introdução

Você sabia que as aplicações de Big Data estão em praticamente todos os setores da sociedade moderna?

Nesse cenário de oportunidades e desafios, a Apache Foundation desenvolveu o framework Hadoop: a tecnologia que estudaremos aqui. Além de explorar conceitos, também vamos demonstrar alguns exemplos práticos que ajudarão você a entender o funcionamento do Hadoop.

Preparação

Para reproduzir os exemplos apresentados ao longo deste conteúdo, você deve utilizar o sistema operacional Windows e instalar o Java 8 em seu computador. Além disso, também precisará instalar o Hadoop.

Objetivos

Ao final desta aula, você será capaz de:

- Analisar a arquitetura da tecnologia de framework de software livre chamada Hadoop.
- Comparar os sistemas HDFS e RDBMS.
- Descrever a preparação do ambiente para uso prático do Hadoop.
- Aplicar um exemplo prático no Hadoop.

01. Arquitetura do Hadoop

Fundamentos da tecnologia de Hadoop

Características das aplicações de Big Data

Atualmente, há muitas aplicações que envolvem grandes volumes de dados, como as transações financeiras on-line, a produção e o compartilhamento de conteúdo nas redes sociais e os estudos nas áreas da biologia genética.

Esses são apenas alguns exemplos de nosso cotidiano que fazem parte do que conhecemos como **Big Data**. Essa expressão, que vem do inglês, foi incorporada ao nosso dia a dia para descrever um conjunto de tecnologias que gerenciam aplicações complexas.

Essas tecnologias que compõem as aplicações de Big Data são modernas e ainda estão se expandindo, como é o caso da computação em nuvem e da Internet das Coisas (IoT).

As características fundamentais de tais aplicações contemplam os 5 Vs do Big Data (ISHWARAPPA; ANURADHA, 2015). São eles:

Volume

Trata da quantidade de dados gerados e coletados pelas aplicações. Normalmente, uma aplicação é classificada como Big Data quando trabalha com um volume de dados da ordem de petabytes (PB): 1 PB = 1.024 terabytes. Aplicações que envolvem

dados não estruturados, como os arquivos de câmera de vigilância de grandes aeroportos, são um bom exemplo.

Variedade

Corresponde à diversidade de formatos dos dados. É bastante comum trabalharmos nesse tipo de aplicação com dados disponíveis em tabelas, arquivos texto e JSON, por exemplo.

Velocidade

Refere-se à velocidade com a qual os dados são gerados e processados. Um exemplo prático são os sistemas de monitoramento em tempo real.

Veracidade

Refere-se à questão fundamental da qualidade dos dados. Em especial, nesse tipo de aplicação com tantas variáveis para controlar, é muito importante aplicarmos técnicas e usarmos ferramentas para garantir a integridade e a qualidade dos dados e evitar processamentos desnecessários. Algoritmos que trabalham com a confiabilidade da fonte e a credibilidade da informação são exemplos de técnicas para garantir a veracidade dos dados.

Valor

Relaciona-se à recompensa que esperamos obter ao trabalhar com aplicações de Big Data. Dados em grandes volumes são muito úteis em estudos estatísticos para descobrir padrões e adquirir conhecimento. Podemos encontrar um ótimo exemplo de extração de valor em Big Data nos algoritmos de recomendação de vendas.

Framework Hadoop

Para que possamos lidar com toda a complexidade que envolve as aplicações de Big Data, precisamos de uma tecnologia que gerencie todos esses recursos computacionais de hardware e software. Uma das tecnologias que obteve mais sucesso com essa finalidade é o **framework Hadoop**.

O Hadoop é uma tecnologia de framework de software livre desenvolvida pela Apache Foundation, aplicada no armazenamento e no processamento de dados de grandes volumes, ou seja, em Big Data.

Além da distribuição livre da Apache, o Hadoop possui outras distribuições, como:

- Cloudera;
- Hortonworks;
- MapR;
- IBM;
- Microsoft Azure;
- Amazon Web Services Elastic MapReduce Hadoop Distribution.

Todos esses fornecedores têm suas soluções baseadas no Hadoop Apache.

Arquitetura básica do Hadoop

As grandes empresas da Internet, como Facebook, Yahoo, Google, Twitter e LinkedIn, usam o Hadoop pela natureza de suas aplicações, ou seja, pelos diferentes tipos de dados, que podem ser:

Estruturados

Como tabelas e planilhas;

Não estruturados

Como logs, corpo de e-mails e texto de blogs;

Semiestruturados

Como metadados de arquivos de mídia, XML e HTML.

Essas aplicações possuem todas as características dos 5 Vs da tecnologia Big Data, o que ajudou bastante na divulgação e no aperfeiçoamento do framework Hadoop ao

longo dos anos. De modo geral, muitas organizações de grande porte, como empresas governamentais e prestadores de serviços financeiros e de saúde, por exemplo, utilizam o Hadoop.

> De acordo com White (2015), a tecnologia Hadoop possui um sistema de cluster que funciona, basicamente, como uma arquitetura mestre-escravo: um único nó do tipo mestre (master) com vários nós escravos (slave). Essa estrutura permite armazenar e processar grandes volumes de dados em paralelo.

Essa arquitetura do Hadoop é formada por quatros componentes principais. São eles:

MapReduce

Modelo de programação paralela.

HDFS (Hadoop Distributed File System)

Sistema de arquivos distribuídos do Hadoop.

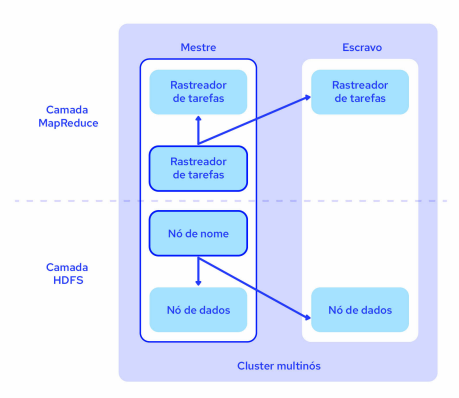
YARN (Yet Another Resource Negotiator)

Responsável pela coordenação, pelo gerenciamento dos recursos do Hadoop e pelo agendamento dos trabalhos que serão executados.

Hadoop Common

Utilitários comuns do Hadoop.

Ao longo deste conteúdo, analisaremos esses componentes com mais detalhes. A imagem a seguir apresenta uma visão geral da arquitetura do Hadoop:



Arquitetura do Hadoop.

Atividade

Questão 1

As aplicações de Big Data fazem parte de nosso cotidiano, e isso é muito fácil constatar. Até os usuários que não conhecem aspectos da tecnologia, mas que já consumiram algum serviço da Internet, mesmo sem saber, já utilizaram serviços de Big Data. O Hadoop é uma tecnologia voltada exatamente para esse tipo de aplicação. Nesse sentido, assinale a alternativa correta a respeito do Hadoop e das características dos projetos de Big Data.

A

Apesar da capacidade de tratar grandes volumes de dados, o framework Hadoop é limitado à infraestrutura de redes locais.

B

Uma das limitações do Hadoop é que ele não pode ser utilizado para trabalhar com tabelas tradicionais de bancos de dados.

C

O Hadoop é um framework cuja distribuição é gratuita e é voltado, especificamente, para tratar aplicações de grande volume de dados.

D

As apresentações de vídeos ao vivo nas redes sociais – conhecidas como lives – são uma das aplicações que podem ser tratadas por

D técnicas de Big Data.

E O Hadoop foi desenvolvido em um modelo flexível, em que todos os nós podem assumir papéis de mestre e escravo, dependendo do contexto da aplicação.

Parabéns! A alternativa D está correta.

O Hadoop Apache é um framework desenvolvido para trabalhar com aplicações de Big Data. Essas aplicações são caracterizadas pela complexidade que envolve seus dados. Um de seus aspectos básicos é a variedade de dados, que podem ser estruturados, não estruturados e semiestruturados. O Hadoop está preparado para trabalhar com todas essas situações. No caso de vídeos, trata-se de um caso de dados não estruturados.

MapReduce

O que é o mecanismo MapReduce?

Quando trabalhamos com aplicações de Big Data, não há como aplicar o processamento serial. Para realizar o processamento paralelo dos dados, o Hadoop utiliza um mecanismo chamado de **MapReduce**: a base fundamental para que o framework tenha um bom desempenho computacional.

O MapReduce é uma estrutura que trabalha com duas fases distintas. São elas:

Mapeamento dos dados

Nesta fase, os dados de entrada são coletados e convertidos em um conjunto de dados que podem ser processados como um par do tipo chave-valor.

×

Redução dos dados

Nesta fase, consome-se o resultado do mapeamento dos dados. Em seguida, eles são processados até atingir o objetivo da aplicação.

Vamos analisar com mais detalhes cada uma dessas fases.

Mapeamento dos dados

A entrada da fase de mapeamento é um conjunto de dados passados para a função Map, que divide esses blocos de dados em tuplas: pares do tipo chave-valor.

Esses pares são enviados para a fase de redução após o mapeamento, que é dividido nas seguintes subfases:

Leitor de registros (record reader)

Divide os dados de entrada em pares de chave-valor, de modo que sejam enviados como entrada para o mapeador. A chave possui informações de localização, e o valor contempla os dados associados a ela.

Mapeador

É uma função definida pelo usuário que faz o processamento das tuplas obtidas do leitor de registros. Essa função não gera nenhum par do tipo chave-valor.

Combinador (combiner)

É usado entre o mapeador e o redutor para diminuir o volume de transferência de dados entre ambos.

Particionador

Tem duas funções principais: buscar pares de chave-valor gerados na subfase do mapeador e gerar os fragmentos correspondentes a cada redutor.

Redução dos dados

A função Reduce combina as tuplas geradas pela fase de mapeamento e aplica as operações necessárias para processar os dados, que, em seguida, são enviados para o nó de saída final.

O processamento é sempre feito na fase de redução dos dados, que é dividida nas seguintes subfases:

Embaralhamento e classificação

O processo em que o mapeador gera os pares intermediários chave-valor e os transfere para a fase de redução é conhecido como embaralhamento (shuffle). Por meio da aplicação do processo de embaralhamento, o sistema pode classificar os dados com o uso de seus pares chave-valor. Aqui, perceba o benefício desse processo: à medida que algumas tarefas de mapeamento são concluídas, o embaralhamento já começa, ou seja, o processamento dos dados não espera pela conclusão da tarefa realizada pelo mapeador

Redução

Esta tarefa agrupa as tuplas geradas a partir do mapeamento e, em seguida, aplica os processos de classificação e agregação nesses pares chave-valor.

Gravação da saída

Quando todas as operações são executadas, os pares chave-valor são gravados em arquivo. Cada registro é gravado em uma nova linha, e a chave e o valor são separados por espaço.

Atividade

Questão 1

A tecnologia Hadoop é composta por diversos componentes. Um deles é o MapReduce, utilizado em aplicações de computação distribuída que envolve, por exemplo, cálculos. Assinale a alternativa correta a respeito do MapReduce do Hadoop.

A — A principal característica do MapReduce é a forma como armazena os dados.

B — O Hadoop utiliza o MapReduce para fornecer uma interface gráfica com os usuários.

C — O Hadoop obriga que os desenvolvedores utilizem o MapReduce.

D — O MapReduce otimiza o processamento das operações computacionais.

E — O MapReduce é indicado para realizar operações de consultas dos dados.

Parabéns! A alternativa D está correta.

O MapReduce é uma estratégia da computação distribuída para processar dados de forma eficiente. Ele é um dos principais componentes do Hadoop.

Componentes do Hadoop

HDFS

O Hadoop Distributed File System ou sistema de arquivos distribuídos do Hadoop é responsável pelo armazenamento de dados em um cluster desse framework. Ele foi projetado para trabalhar com grandes volumes de dados em hardware comum, ou seja, em dispositivos baratos, como computadores pessoais.

Trata-se de um sistema que tem tolerância a falhas. Além disso, fornece alta disponibilidade para a camada de armazenamento e outros dispositivos presentes nesse cluster do Hadoop.

A arquitetura do HDFS utiliza um sistema de arquivos distribuídos, por meio dos componentes NameNode e DataNode, que proporcionam alto desempenho no acesso aos dados em clusters Hadoop e podem ser expandidos, ou seja, são altamente escaláveis.

Vamos entender os componentes do HFDS com mais detalhes:

NameNode

Desempenha o papel de mestre em um cluster Hadoop que gerencia os DataNodes (nós escravos). Seu objetivo é armazenar os dados sobre os dados, ou seja, os metadados. Os NameNodes gerenciam os DataNodes por meio das operações de abrir, excluir, criar, replicar, renomear e fechar arquivos. Para ilustrar melhor, no NameNode, encontramos informações sobre o nome do arquivo e a quantidade de replicações e de identificação dos blocos.

Metadados

Contêm informações sobre os arquivos (nomes e tamanhos) e sobre a localização (número e ids do bloco) do DataNode em que o NameNode faz o armazenamento. Essas informações são úteis para encontrar o DataNode mais próximo. Como consequência, as operações de comunicação ficam mais rápidas. Os logs de transações que servem para rastrear a atividade do usuário em um cluster Hadoop são um exemplo de metadados.

DataNode

Desempenha o papel de escravo. O principal objetivo dos DataNodes é armazenar os dados em um cluster Hadoop. A quantidade de DataNodes pode ser muito grande, aumentando, assim, a capacidade de armazenamento que o cluster Hadoop pode realizar. Para entender melhor, o DataNode armazena os dados do HFDS em arquivos de sistema local.

YARN

Yet Another Resource Negotiator é o componente estrutural sobre o qual funciona o MapReduce. Ele realiza duas operações distintas. São elas:

Agendamento ou monitoramento de tarefas

Divide uma grande tarefa em tarefas menores, para que possam ser atribuídas a vários nós escravos em um cluster do Hadoop. Dessa forma, o desempenho do processamento será maximizado. É o agendador que faz o controle das prioridades de execução das tarefas, considerando aspectos como sua importância e dependências em relação às demais tarefas e quaisquer outras informações, como o tempo para conclusão do trabalho, por exemplo.

Gerenciamento de recursos (ResourceManager)

Faz o controle de todos os recursos que são disponibilizados para execução de uma tarefa em um cluster do Hadoop. Relacionado ao gerenciador de recursos está o gerenciador de dados (NodeManager), que é o agente de estrutura por máquina responsável pelos containers por meio do monitoramento do uso de recursos (CPU, memória, disco, rede) e envio de relatórios de uso para o gerenciador de recursos.

A ideia fundamental do YARN é dividir as operações de agendamento ou monitoramento de tarefas e gerenciamento de recursos em daemons separados, ou seja, processos que executam em background.

Hadoop Common

Os utilitários comuns do Hadoop – também conhecidos como núcleo do Hadoop (Hadoop Core) – são as bibliotecas e aplicações que oferecem suporte a essa tecnologia. Eles são usados para executar as aplicações no cluster do Hadoop pelos seguintes componentes:

- HDFS;
- YARN;
- MapReduce.

Como nos demais módulos do Hadoop, os utilitários assumem que as falhas de hardware são comuns e que devem ser tratadas automaticamente no software pelo Hadoop framework.

O pacote de utilitários fornece serviços essenciais e processos básicos, como:

- Abstração do sistema operacional e de seu sistema de arquivos.
- Seu sistema de arquivos.

Esse pacote também contém os arquivos Java Archive (JAR) e os scripts que são necessários para iniciar o Hadoop.

Além disso, também podemos encontrar no pacote de utilitários: códigos-fontes, documentação e informações sobre as contribuições de diferentes projetos da comunidade Hadoop.

Atividade

Questão 1

O armazenamento dos dados é um dos elementos centrais do Hadoop. É por meio de uma estrutura específica que os dados são acessados e processados por aplicações no contexto do Hadoop. Nesse sentido, assinale a alternativa correta a respeito do armazenamento dos dados no Hadoop:

A | O NameNode é a principal forma de o Hadoop gerenciar os dados do sistema. .

B | O Hadoop utiliza o DataNode para garantir a redundância dos dados.

C | O Hadoop utiliza o HDFS para gerenciar os dados de forma eficiente.

D | O NameNode é outra forma de se referenciar para o HDFS.

E | O MapReduce utiliza os metadados para realizar computações sobre os dados.

Parabéns! A alternativa C está correta.

O HDFS é a forma como o Hadoop armazena os dados. As demais aplicações o utilizam para otimizar o processamento das operações, além de oferecer maior confiabilidade para o sistema, aumentando, assim, a tolerância a falhas.

Vantagens e desvantagens do Hadoop

Potenciais de aplicação

Entre as principais vantagens ou os principais potenciais de aplicação do Hadoop estão:

Escalabilidade

O Hadoop foi projetado desde o início para trabalhar com grandes volumes de dados. Para isso, os componentes de sua arquitetura lidam com distintos aspectos do armazenamento e processamento de dados distribuídos em diferentes nós da infraestrutura que aplicamos na solução.

Redução de custos

A distribuição Apache do Hadoop é de um software livre. Além disso, não demanda por uma infraestrutura de hardware especial, podendo utilizar equipamentos comuns.

Flexibilidade

O Hadoop é capacitado para trabalhar com diferentes tipos de dados: estruturados, semiestruturados e não estruturados. Dessa forma, as empresas podem aplicar suas estratégias para gerar valor a partir da composição e análise desses dados.

Velocidade

Os componentes da arquitetura do Hadoop, como o HDFS e o MapReduce, são projetados, respectivamente, para gerenciar e processar dados com a aplicação de

estruturas de dados e estratégias de algoritmos que otimizam a operação dos processos.

Tolerância a falhas

O Hadoop utiliza um processo de replicação dos dados entre os nós do cluster. Se ocorrer falha em algum nó, haverá outra cópia disponível para uso.

Limitações de aplicação

Entre as principais desvantagens ou limitações de aplicação do Hadoop estão:

Segurança

Devido à complexidade das aplicações de Big Data, os aspectos relacionados à segurança são um grande desafio. No caso do Hadoop, esse desafio está longe de ser trivial. Por exemplo, o modelo de segurança do Hadoop é desabilitado por padrão. Portanto, é da responsabilidade de quem vai gerenciar a infraestrutura da plataforma fazer a habilitação do módulo de segurança. Caso contrário, os dados correrão um grande risco. Também é necessário tratar explicitamente aspectos de criptografia dos dados.

Vulnerabilidade

O Hadoop foi desenvolvido na linguagem de programação Java. Existem diversos casos já catalogados de quebra de segurança do Hadoop, como escalonamento de privilégios e acesso não autorizado a senhas. Tudo isso ocorre devido à complexidade das aplicações de Big Data. O profissional que trabalha com os aspectos de segurança e controle de vulnerabilidades precisa conhecer muito bem a arquitetura do Hadoop e estudar constantemente os fóruns oficiais sobre esse tema, que é bastante dinâmico.

Tratamento de pequenos volumes de dados

O Hadoop foi projetado para trabalhar com grandes volumes de dados. Infelizmente, isso significa que ele não é uma boa opção para trabalhar com pequenos volumes. Parece ser contraditório, mas não é. Os componentes HDFS e MapReduce utilizam técnicas que são eficientes para manipular muitos dados. Isso significa que as estruturas de dados e os algoritmos são dimensionados com essa finalidade. Se essas técnicas forem usadas para trabalhar com pequenos volumes, serão ineficientes para soluções mais simples.

Estabilidade

O Hadoop está em constante evolução e tem versões distribuídas por vários fornecedores. Por isso, não é raro que ocorram problemas relacionados à estabilidade da plataforma. Mais uma vez, isso reforça a necessidade de um profissional focado em aspectos da arquitetura e infraestrutura do Hadoop, que se atualize constantemente nos canais oficiais e fóruns de usuários e analistas.

Atividade

Questão 1

As aplicações de Big Data são complexas devido à diversidade de tecnologias envolvidas para tratar situações que também são difíceis. Para o usuário final, o importante é que a aplicação funcione bem. O Hadoop tem como objetivo tratar esse tipo de situação e dar transparência a essa complexidade para os usuários consumidores. Mas, na prática, seu uso tem vantagens e desvantagens. Nesse sentido, assinale a alternativa correta a respeito dos potenciais e das limitações do Hadoop:

A

Projetos de Big Data precisam de muitos recursos computacionais e, portanto, podem ser muito caros. Devido à flexibilidade do Hadoop, não é necessário utilizar equipamentos especiais, o que ajuda a equilibrar a relação custo e benefício.

B

O fato de todas as distribuições do Hadoop terem licenças gratuitas reduz os custos dos projetos. Além disso, existe uma documentação com muitos exemplos que facilitam sua aplicação.

C

O Hadoop é um framework muito complexo. Apesar de ter algumas distribuições gratuitas, como o Apache, não é uma boa escolha para a implementação de projetos de Big Data. Seu valor é histórico, pois foi uma das primeiras ferramentas de Big Data aplicadas para uso comercial.

D

Uma das vantagens do Hadoop é seu sistema de armazenamento muito eficiente, conhecido como MapReduce, que é capaz de gerenciar dados nos mais variados formatos.

E

O Hadoop é um framework muito seguro e fácil de ser configurado. Junto com os baixos custos de implantação e manutenção, essas vantagens o tornam uma das melhores escolhas para o desenvolvimento de projetos de Big Data.

Parabéns! A alternativa A está correta.

Projetos de Big Data são complexos por vários motivos, mas o valor que pode ser extraído desses projetos justifica o investimento em tecnologias que otimizem a

relação custo e benefício. Nesse sentido, o Hadoop contribui com uma arquitetura que é flexível e não exige a utilização de equipamentos especiais.

02. HDFS x RDBMS

Importância dos sistemas HDFS e RDBMS

Necessidade da tecnologia de Big Data

As aplicações de Big Data fazem parte de nosso cotidiano.

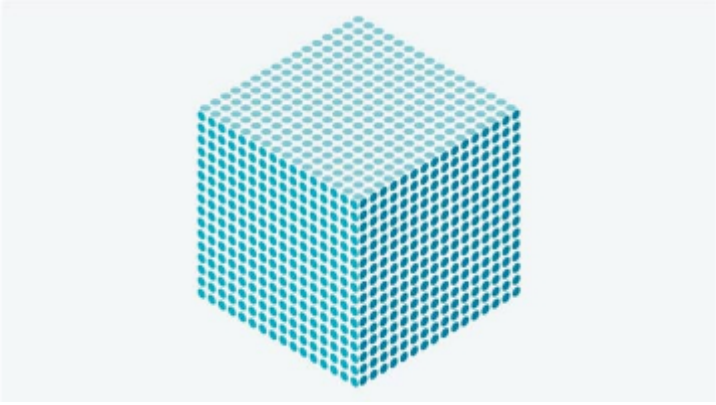
Exemplo

Há diversas situações práticas que demandam técnicas capazes de lidar com grandes volumes de dados. Esses dados podem crescer com muita velocidade e vir de diferentes fontes com formatos diversos.

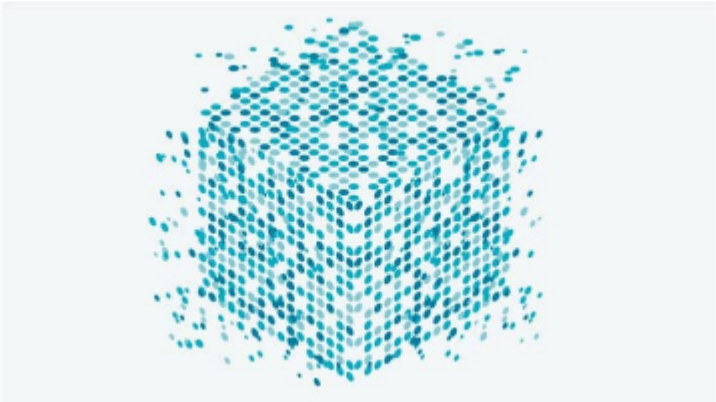
Essas aplicações devem agregar valor à solução, com a identificação de padrões e descoberta de conhecimento para dar suporte à tomada de decisão.

O Hadoop é uma tecnologia de Big Data que possui uma arquitetura projetada para lidar com essas situações. Como já vimos, um dos principais componentes dessa arquitetura é o HDFS ou sistema de gerenciamento de arquivos distribuídos.

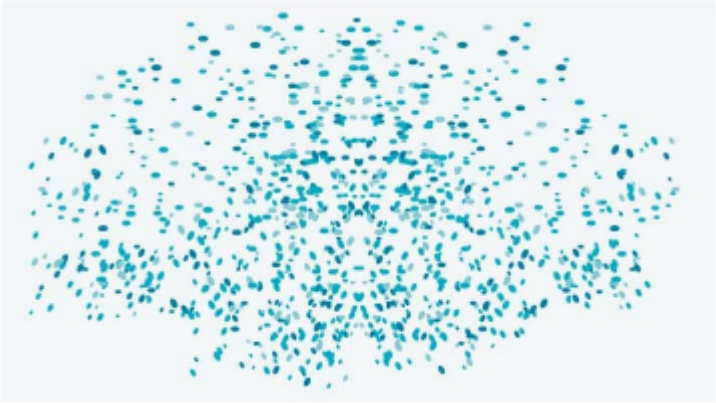
Esse sistema pode tratar de:



Dados estruturados



Dados semiestruturados



Dados não estruturados

Mas sabemos que a questão de segurança merece bastante cuidado, pois o sistema pode conter vulnerabilidades que o expõem a ataques de cibercriminosos.

Já o RDBMS (Relational Database Management System) ou sistemas tradicionais de gerenciamento de dados são mais restritos quanto ao gerenciamento do que o HDFS. Essa restrição limita as aplicações distribuídas, mas é bastante segura no contexto em que se aplica.

Elementos do ecossistema do Hadoop

Inúmeras aplicações ainda usam RDBMS, mas há muitas situações em que queremos migrar esses sistemas para um ambiente de Big Data, como Hadoop, que oferece recursos para realizar esse processamento.

Para isso, devemos conhecer a composição do ecossistema do Hadoop, que possui os seguintes componentes:

HDFS

Sistema de arquivos distribuídos do Hadoop.

YARN

Um negociador de recursos.

MapReduce

Processamento de dados que usa programação paralela.

Hadoop Common

Utilitários comuns do Hadoop.

Spark

Processamento de dados na memória.

PIG, HIVE

Serviços de processamento de dados que usam consulta (semelhante a SQL).

HBase

Banco de dados NoSQL.

Mahout, Spark MLlib

Aprendizado de máquina.

Apache Drill

SQL no Hadoop.

ZooKeeper

Gerenciamento de um cluster.

Oozie

Agendamento de trabalho.

Flume, Sqoop

Serviços de ingestão de dados.

Solr & Lucene

Pesquisa e indexação.

Ambari

Provisão, monitoramento e manutenção de um cluster.

Desses componentes, o Sqoop é o serviço do Hadoop para transferir dados de RDBMS para HDFS. Ele pode fazer a importação e a exportação de dados de RDBMS para HDFS. Ambos os processos são bem similares.

Na transferência dos dados de RDBMS para HDFS, quando realizamos o processamento do Sqoop, nossa tarefa principal é dividida em subtarefas, que, por sua vez, são tratadas internamente como tarefas de mapa individuais. A subtarefa Map Task (tarefa de mapeamento) importa os dados para o ecossistema Hadoop.

No caminho contrário, ou seja, na transferência dos dados de HDFS para RDBMS, o Sqoop é mapeado em tarefas que trazem o bloco de dados do HDFS. Esses blocos são exportados para um destino de dados estruturados. Esse processo é conhecido como ETL (Extract, Transform and Load) ou extrair, transformar e carregar.

Atividade

Questão 1

Os sistemas gerenciadores de bancos de dados relacionais (RDBMS) oferecem muitos recursos que garantem eficiência e confiabilidade dos dados. A partir da evolução das demandas por dados para os chamados problemas de Big Data, os RDBMS mostraram limitações. Dessa forma, foram necessárias outras soluções, como o Hadoop e o HDFS, por exemplo. Em relação às tecnologias de RDBMS e ao Hadoop/HDFS, assinale a alternativa correta:

A | Devido à flexibilidade do Hadoop, o framework pode trabalhar de forma equivalente com volumes de dados de grande e baixo porte, enquanto os RDBMS trabalham apenas com dados de baixo porte.

B | O HDFS é o sistema de gerenciamento de arquivos do Hadoop que o capacita a trabalhar com dados dos mais variados formatos, diferentemente do RDBMS, que trabalha apenas com dados relacionais.

C | Os sistemas do tipo RDBMS são muito eficientes para trabalhar com dados semiestruturados, desde que estejam em um ambiente centralizado.

D | O sistema de gerenciamento de arquivos do Hadoop não está capacitado para trabalhar com tabelas relacionais. Caso um projeto demande por dados estruturados, deve usar um RDBMS.

E | Sistemas do tipo RDBMS são tão complexos como os sistemas de gerenciamento de arquivos do Hadoop. Basicamente, o que os diferencia é o escopo da aplicação.

Parabéns! A alternativa B está correta.

Os problemas de Big Data são caracterizados, entre outras propriedades, pela variedade de seus dados, que podem ser estruturados, semiestruturados e não estruturados. O HDFS, que é o sistema de gerenciamento de arquivos do Hadoop, foi projetado para trabalhar com essa variedade de dados. Já os sistemas do tipo RDBMS não têm essa característica, ficando limitados a trabalhar com tabelas em bancos de dados.

Diferenças entre os sistemas HDFS e RDBMS

Propriedades dos sistemas gerenciados de dados

Os RDBMS são sistemas de gerenciamento de banco de dados relacionais.

Exemplo

Temos Oracle, SQL Server da Microsoft, MySQL e PostgreSQL. Eles utilizam tabelas para fazer o armazenamento dos dados e regras de integridade, que servem para relacionar as tabelas entre si e restringir as ações realizadas sobre os dados.

Os sistemas RDBMS garantem as propriedades ACID das transações, que incluem:

Atomicidade

A execução das ações em um banco de dados, ou seja, as sequências de operações chamadas de transações, é indivisível.

Consistência

Um conceito fundamental em sistemas de banco de dados é o de integridade referencial, que trata das relações de dependências lógicas entre os dados. Ao final da execução de uma transação, o sistema deve manter a integridade dos dados, assim como respeitar todas as regras previamente definidas.

Isolamento

Esta propriedade faz a separação das execuções de transações, para que não interfiram entre si e possam levar o sistema a um estado de inconsistência.

Durabilidade

Enquanto os dados estão sendo tratados pelas transações em execução, permanecem em um estado transiente. Ao final da execução da transação, o sistema deve garantir que os dados sejam gravados no banco de dados, ou seja, que fiquem no estado de persistência.

Todas essas propriedades são fundamentais para um projeto de banco de dados. Portanto, podemos entender que os objetivos dos RDBMS são armazenar, gerenciar e recuperar os dados da forma mais rápida e confiável possível em um ambiente de arquitetura cliente-servidor.

No caso do HDFS, os dados estão contextualizados em um ambiente distribuído. Devido às características próprias das aplicações de Big Data, o gerenciamento desses dados é bem mais complexo. Por isso, há situações em que é mais adequado aplicar um modelo do que o outro, ou seja, **o HDFS não é uma substituição do RDBMS**.

Comparação entre HDFS e RDBMS

Vamos analisar algumas diferenças entre os dois sistemas que envolvem as seguintes características:

Volume de dados

O RDBMS funciona bem com o volume de dados na ordem de Gigabytes até Terabytes, enquanto o HDFS foi projetado para trabalhar com dados na ordem de Petabytes.

Taxa de transferência

Nada mais é do que o volume total de dados processados em determinado período. O RDBMS faz a operação de leitura muito rápida e a de escrita de forma lenta. Já o HDFS realiza as operações de leitura e escrita rapidamente.

Variedade de dados

O HDFS tem a capacidade de processar e armazenar dados estruturados, semiestruturados e não estruturados. Já o RDBMS é usado apenas para gerenciar dados estruturados e semiestruturados.

Tempo de resposta

Também conhecido como latência, está relacionado à velocidade para recuperar informações. Apesar de o HDFS ter um rendimento alto para acessar lotes de grandes conjuntos de dados, o RDBMS é comparativamente mais rápido na recuperação dessas informações.

Escalabilidade

Para escalar uma solução no RDBMS, é necessário aumentar os recursos da máquina, o que é chamado de escalabilidade vertical. No caso do HDFS, a escalabilidade é horizontal, ou seja, para expandir o sistema, basta adicionarmos mais computadores aos clusters existentes.

Processamento de dados

O HDFS suporta OLAP (On-line Analytical Processing), que envolve consultas e agregações complexas. Dependendo da quantidade de dados, a velocidade de processamento pode levar muito tempo. No RDBMS, as consultas e agregações são feitas por meio do OLTP (On-line Transaction Processing), que, comparativamente, é mais rápido do que o OLAP. Devemos ficar atentos para o fato de que, nas aplicações OLAP, as tabelas estão desnormalizadas, enquanto

na OLTP, elas estão normalizadas segundo as regras do modelo relacional de dados.

Custo

Além de ser um framework de software livre, o Hadoop Apache, que inclui o HDFS, permite aumentar a infraestrutura de um sistema sem a necessidade de equipamento especial. No RDBMS, em muitos casos, precisamos considerar questões relacionadas à licença de uso e aos custos de aquisição de recursos de hardware para a infraestrutura sobre a qual opera.

Atividade

Questão 1

Os sistemas de gerenciamento de dados têm evoluído ao longo do tempo. Parte dessa evolução pode ser atribuída ao surgimento de novas tecnologias que os capacitam à resolução de problemas até então não tratáveis. Um exemplo desse processo ocorre nos sistemas de gerenciamento de bancos de dados relacionais (RDBMS) e no sistema gerenciador de arquivos do Hadoop (HDFS). Em relação às tecnologias de RDBMS e HDFS, assinale a alternativa correta.

A | O RDBMS é considerado uma tecnologia antiga e deve ser substituído pelo HDFS sempre que possível.

B | Com a evolução da tecnologia, o RDBMS também evoluiu. Atualmente, ele não apresenta nenhuma desvantagem em relação ao HDFS, a não ser para o tratamento de dados não estruturados.

C

Uma das vantagens do HDFS em relação ao RDBMS é a capacidade de expandir a infraestrutura de uma aplicação sem a necessidade de equipamentos especiais.

D

Um aspecto muito importante em um sistema de gerenciamento de dados é a velocidade com que realiza as operações de escrita e leitura dos dados, que são feitas mais eficazmente no RDBMS do que no HDFS.

E

Os RDBMS e HDFS são sistemas equivalentes para o gerenciamento de dados, diferenciando-se apenas pelo uso de tecnologias proprietárias, mas que têm os mesmos objetivos.

Parabéns! A alternativa C está correta.

O HDFS é o sistema de gerenciamento de arquivos do Hadoop, que tem muitas vantagens em relação ao RDBMS, mas também desvantagens. Uma das vantagens está relacionada à capacidade de expansão de um sistema. No HDFS, esse processo – chamado de expansão horizontal – pode ser feito sem a necessidade de equipamentos especiais. Já no RDBMS, essa expansão é vertical, implicando maiores custos na aquisição de equipamentos.

Data lake

O que é um data lake?

Como você já sabe, as aplicações de Big Data são caracterizadas pelos 5 Vs (volume, variedade, velocidade, veracidade e valor).

Ainda existem definições que incluem outros Vs, como as propriedades de Variabilidade e Visualização. Isso ocorre porque essas aplicações ainda estão em processo de evolução.

Devido às características dessas aplicações, muitas soluções de Big Data exigem novas abordagens. Um desses casos envolve o armazenamento dos dados. A solução aplicada pela tecnologia de Big Data nesse contexto é chamada de **data lake** (lago de dados).

> Trata-se de um local da aplicação de Big Data que centraliza um grande volume de dados no formato original: estruturados, semiestruturados e não estruturados. Esses dados são armazenados em objetos – conhecidos como object storage – que contêm tags de metadados e um identificador único.

Essa estrutura de entidade dos dados permite que possamos analisá-los e buscar por padrões, pois as consultas são realizadas com muita eficiência. Esses objetos de armazenamento podem ser consultados pelas demais aplicações de Big Data.

É natural compararmos aspectos da arquitetura do data lake com os modelos dos bancos de dados tradicionais, chamados de data warehouse (armazém de dados).

Vamos entender a diferença:

Data warehouse

É hierárquica, pois os dados são armazenados em arquivos ou pastas.

×

Data lake

Usa uma arquitetura plana por meio dos objetos de armazenamento de dados.

Benefícios do data lake

De acordo com Singh e Ahmad (2019), os benefícios de usar data lake incluem:

Escalabilidade

Para aumentar a capacidade de armazenamento do data lake, é necessário apenas acrescentar novos nós à infraestrutura, sem a necessidade de um hardware especial.

Alta velocidade

O data lake usa programas utilitários para adquirir dados com alta velocidade e integrá-los com grandes volumes de dados históricos.

Estruturação

O data lake utiliza os objetos de armazenamento para guardar diferentes tipos de dados, de modo que o acesso seja bem eficiente.

Acesso aos dados originais

Os dados armazenados no data lake estão no formato original. Isso é considerado uma boa característica no processamento analítico (OLAP) para tentar descobrir padrões.

Acessibilidade

Os dados do data lake podem ser utilizados pelos usuários do sistema por meio de programas utilitários.

Camadas do data lake

O data lake é composto por camadas e níveis. As camadas agrupam as funcionalidades comuns. São elas:

Governança e segurança de dados

Controla o acesso aos dados, utilizando mecanismos de segurança para que apenas usuários com perfis previamente mapeados tenham acesso e direito de manipulação dos dados.

Metadados

Marcam e identificam os dados com informações que auxiliem sobre a importância deles. Isso ajuda nos processos de análise para extração de valor. Essa camada trata de dados nos diversos formatos: estruturados, semiestruturados e não estruturados.

Gerenciamento do ciclo de vida da informação

Estabelece as regras de armazenamento dos dados e o período em que devem permanecer no sistema, pois, com o tempo, é normal que eles percam seu valor.

Níveis do data lake

Os níveis são uma forma didática de agrupar aspectos semelhantes de uma funcionalidade. Podemos pensar da seguinte maneira: nos níveis, os dados fluem sequencialmente, assim como ocorre quando passamos parâmetros para uma função.

Enquanto os dados se movem de camada para camada, os níveis realizam seu processamento nos dados em movimento. São eles:

Admissão

Também conhecido como nível de ingestão, possui todos os serviços para aquisição de dados de fontes externas. Esses dados podem vir em lotes (batch), microlotes (micro batch) e fluxo de dados de tempo real (real time streams).

Gerenciamento

Os dados são organizados por meio da aplicação de metadados e relacionamentos que auxiliem sua identificação e localização.

Consumo

Os dados do nível anterior são consumidos por aplicações relacionadas à análise de negócios.

Atividade

Questão 1

O data lake é um componente fundamental da arquitetura do Hadoop. Os dados das aplicações de Big Data podem ter muitas características complexas que precisam ser tratadas. Nesse sentido, assinale a alternativa correta a respeito do data lake.

A | Questões relacionadas à segurança da informação são tratadas por um programa utilitário específico do data lake, que fica na camada de gerenciamento de ciclo de vida da informação.

B

O data lake pode ter dados estruturados, semiestruturados e não estruturados, mas o tratamento para cada um deles é realizado em diferentes camadas.

C

A ingestão dos dados é responsável por fazer marcações deles, que, posteriormente, serão consumidos pelos programas utilitários do Hadoop de modo eficiente.

D

Um aspecto essencial em uma aplicação de Big Data é a consulta eficiente dos dados, que devem ser organizados para beneficiar as estruturas de dados de busca. Isso ocorre na camada de metadados.

E

As camadas e os níveis do data lake são equivalentes em termos do tratamento dos dados, com apenas duas exceções: a implementação da política de segurança e a identificação dos dados.

Parabéns! A alternativa D está correta.

Cada uma das camadas do data lake é responsável por funcionalidades que visam tratar a política de segurança de acesso, marcação e regras de permanência dos dados no sistema. Essas atribuições cabem, respectivamente, às camadas de governança, metadados e gerenciamento do ciclo de vida da informação.

03. Instalação, configuração e teste do Hadoop

Preparação do ambiente

Preparando o ambiente para uma aplicação Hadoop.

1. Verifique os pré-requisitos de instalação do Hadoop.

    1.1. Para utilizar o Hadoop, instale o Java 8 em seu computador. Caso não tenha o Java instalado, vá para a página oficial do Java, em Java Downloads. Escolha a versão adequada para seu sistema operacional, baixe o Java para sua máquina e faça a instalação.

    1.2. No final da instalação, não se esqueça de fazer a configuração da variável de ambiente `JAVA_HOME`, que deve apontar para a pasta bin do Java.

    1.3. Para verificar se a etapa anterior funcionou corretamente, abra um prompt de comando e execute a linha: `java -version`.

2. Faça o download do Hadoop no site oficial do Apache, escolhendo a versão 2.10.1, que pode ser baixada em Apache Hadoop Download.

    2.1. Depois de salvar o arquivo, descompacte-o.

    2.2. Copie o arquivo para a pasta raiz do Windows.

    2.3. Renomeie o arquivo para hadoop. O resultado do arquivo para hadoop.

3. Gere diretórios de dados, criando a pasta data dentro da pasta hadoop.

    3.1. Crie as pastas datanode (dentro da qual os arquivos do HDFS são hospedados) e namenode dentro da pasta data. O resultado da hierarquia da pasta data.

Atividade

Questão 1

Agora que você está com seu ambiente de trabalho pronto, é sua vez de praticar!

Depois de realizar os passos apresentados, localize a pasta em que está o arquivo hdfs.cmd. Esse exercício vai lhe ajudar a ganhar mais familiaridade com os arquivos do Hadoop.

Abrir solução

O arquivo hdfs.cmd está localizado na pasta hadoop/bin. Como você já sabe, o HDFS é o componente do Hadoop responsável pela organização dos dados, de modo que possam ser utilizados de forma eficiente. Aproveite para estudar os outros arquivos e relacioná-los com o que você já aprendeu.

Configuração de arquivos do Hadoop

Configurando os arquivos do Hadoop

1. Configure os arquivos do Hadoop, localizados na pasta `C:\hadoop\etc\hadoop`.

2. Abra o arquivo mapred-site.xml, que é uma lista de parâmetros para a configuração do MapReduce, e localize estas tags: `<configuration> </configuration>`.

    2.1 Dentro das tags de configuração, copie o seguinte código:

```

<configuration>
  <property>
```

```
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

3. Abra o arquivo core-site.xml, com a configuração que substitui os parâmetros padrões do núcleo (core) do Hadoop, e localize estas tags: `<configuration> </configuration>`.

3.1 Dentro das tags de configuração, copie o seguinte código:

```

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

4. Abra o arquivo hdfs-site.xml, que contém a definição de configuração para processos do HDFS, além de especificar a replicação de bloco padrão e verificação de permissão nesse sistema, e localize estas tags: `<configuration> </configuration>`.

4.1 Dentro das tags de configuração, copie o seguinte código:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>PATH TO NAMENODE</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>PATH TO DATANODE</value>
    </property>
```

5. Abra o arquivo yarn-site.xml, que descreve as opções de configuração do YARN, e localize estas tags: `<configuration> </configuration>`.

5.1 Dentro das tags de configuração, copie o seguinte código:

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
```

```
        <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</na
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
</configuration>

```

6. Verifique como a variável de ambiente JAVA_HOME está configurada, considerando a versão 1.8.0_131 do Java que está instalada na pasta C:\Program Files\Java\jdk1.8.0_131.

7. Copie o nome do endereço e escreva uma das duas linhas no arquivo hadoop-env.cmd, que armazena as configurações globais usadas pelos comandos do Hadoop: `SET JAVA_HOME =% JAVA_HOME%` ou `SET JAVA_HOME = "C:\Progra~1\Java\jdk1.8.0_131\ "`.

Em nossa configuração, opte pela segunda linha, pois é necessário escrever “Program Files” como “Progra~1” para evitar problemas com espaços.

Atividade

Questão 1

Você acabou de realizar a configuração dos arquivos primários do Hadoop: um passo importante para trabalhar com suas aplicações. Sem essa configuração, simplesmente não é possível utilizar o Hadoop.

Para nossos exemplos, isso já é suficiente, mas há situações em que precisamos configurar outros arquivos. Então, é muito importante que você se familiarize com eles.

Como exercício, determine qual arquivo é responsável por registro de mensagens de log do Hadoop e indique onde ele está localizado.

O arquivo onde o Hadoop registra mensagens é o Log4j. Sua configuração é feita por meio do arquivo log4j.properties, o qual define o que deve ser registrado. Ele está localizado na pasta hadoop\etc\hadoop.

Agora que você já configurou os arquivos do Hadoop, poderá vê-lo funcionando.

Execução de uma aplicação no Hadoop

Executando uma aplicação no Hadoop

1. No painel de controle do Windows, crie e configure a variável de ambiente HADOOP_HOME para o endereço da pasta bin do Hadoop. Em nosso caso, o endereço da variável `HADOOP_HOME` vai apontar para: `c:\hadoop\bin`.

    1.1 Configure a variável de ambiente Path. Novamente, acesse o painel de controle e selecione a configuração de variáveis de ambiente.

    1.2 Depois de selecionar a variável Path, adicione, ao final dela, a seguinte linha de comando: `;%HADOOP_HOME%`.

    1.3 Para testar se as variáveis de ambientes estão devidamente configuradas – JAVA-HOME deve estar configurada desde o início do processo –, abra um terminal de comando, em qualquer pasta, e digite os seguintes comandos: `echo %JAVA_HOME%` e `echo %HADOOP_HOME%`.

2. Atualize a pasta bin do Hadoop, copiando os arquivos do winutils, que são binários utilitários do Windows para o Hadoop. Esses arquivos estão hospedados na página do winutils.

    2.1 Na página do winutils, baixe a versão do winutils compatível com nossa instalação do Hadoop. Em nosso caso, baixe a versão hadoop-2.9.2.

    2.2 Descompacte o pacote e copie todos os arquivos para a pasta bin do Hadoop.

3. Para verificar o funcionamento do Hadoop, faça os seguintes procedimentos:

3.1 Formate o namenode, abrindo um novo prompt de comando com o perfil de administrador e executando a seguinte linha de comando: `hadoop namenode -format`.

Como você já formatou todos os dados em namenode, use esse comando apenas no início do processo para evitar perda de dados.

3.2 Inicie o Hadoop, continuando no mesmo prompt de comando, com o perfil de Administrador, para evitar problemas de permissão. Para isso, execute a seguinte linha de comando: `start-all.cmd`. Se tudo funcionar corretamente, serão abertas automaticamente quatro janelas executando os seguintes daemons (programas que executam em background) do Hadoop: namenode; datanode; resourcemanager; nodemanager.

3.3 Monitore a execução dos serviços Hadoop.

3.3.1 Para monitorar o namenode, abra um browser e digite o endereço: **localhost:50070**.

3.3.2 Para monitorar o resourcemanager, abra um browser e digite o endereço: **localhost:8088**.

3.3.3 Para monitorar o datanode, abra um browser e digite o endereço: **localhost:50075**.

4. Se você obtiver erro na execução do Hadoop, faça os seguintes procedimentos:

4.1 Entre nas pastas do namenode e datanode e exclua o conteúdo delas.

4.2 Verifique a pasta tmp na raiz e elimine o conteúdo dela.

4.3 Verifique se o conteúdo dos arquivos de configuração está exatamente como o que apresentamos.

Atividade

Questão 1

Você já executou o Hadoop. Agora, na aba do Resource Manager (endereço localhost:8088), pressione a opção About. Qual é o resultado dessa operação?

Abrir solução

> Quando pressionamos a opção About no Resource Manager, obtemos informações sobre a utilização dos recursos pelo Hadoop, como as métricas de cluster e do scheduler.

04. Aplicação prática no Hadoop

Preparação para executar a aplicação

Preparando o ambiente para aplicação no Hadoop

1. Antes de iniciar a execução do Hadoop, entre na pasta: `c:\hadoop\data`.

Você vai encontrar duas pastas: datanode e namenode. Atenção: não exclua as pastas, apenas exclua o conteúdo de dentro delas. Caso você esqueça de fazer isso, sua aplicação não vai funcionar corretamente.

1.1 Abra um prompt de comando na pasta sbin, onde o Hadoop foi instalado. Por exemplo, em nosso caso, instalamos o Hadoop na pasta: `c:\hadoop`.

1.2 Execute a seguinte linha de comando: `hadoop namenode -format`.

1.3 Inicie a operação do Hadoop. Para isso, digite e execute o comando: `start-all.cmd`.

Veja com mais detalhes como realizar esse passo. Serão abertas quatro janelas. Não feche essas janelas, pois vamos precisar que esses daemons estejam em execução para realizar as próximas etapas.

2. Crie um diretório no HDFS, no qual vamos armazenar nosso arquivo. Para criar o diretório, use o mesmo prompt de comando para digitar e executar: `hadoop fs -mkdir /dir_entrada`.

3. Copie um arquivo para o diretório do HDFS que acabamos de criar no Hadoop. Para isso, crie um arquivo texto chamado de texto_exemplo.txt, que está no diretório C. O conteúdo desse arquivo é o seguinte:

Este texto tem palavras repetidas, como as palavras texto e muitas, por exemplo. Dessa forma, o processamento do texto nos ajuda a entender o pacote mrjob do python.

3.1 Digite e execute no prompt de comando:
`hadoop fs -put C:/texto_exemplo.txt /dir_entrada`.

4. Após copiar o arquivo para o diretório dir_entrada no HDFS, verifique o conteúdo do diretório por meio da execução do comando:
`hadoop fs -ls /dir_entrada`.

5. Para verificar o conteúdo do arquivo dentro do diretório, digite e execute o comando:
`hadoop dfs -cat /dir_entrada/texto_exemplo.txt`.

6. Verifique o conteúdo do arquivo de saída.

Atividade

Questão 1

Agora que você está com seu ambiente de trabalho pronto, é sua vez de praticar!

No início da prática apresentada, você precisou excluir os dados das pastas namenode e datanode. Agora, acesse as duas pastas e verifique o que aconteceu com elas.

Abrir solução

Tanto na pasta namenode quanto na pasta do datanode, foram criados alguns arquivos. Isso aconteceu, porque fizemos uma nova execução do Hadoop. O NameNode é o nó principal do HDFS. Já o DataNode é um nó escravo no sistema de arquivos distribuídos do Hadoop. É ele que armazena os dados gerenciados pelo NameNode.

Agora que você já preparou o ambiente do Hadoop para trabalhar, poderá executar um exemplo de uma aplicação de MapReduce.

Execução da aplicação MapReduce

Executando uma aplicação de MapReduce no Hadoop

1. Execute a aplicação MapReduce no Hadoop. Para isso, entre no diretório: `C:\hadoop\share\hadoop\mapreduce`.

    1.1 Copie o arquivo hadoop-mapreduce-examples-2.10.1.jar para a pasta C.

    1.2 Execute esta linha de comando: `hadoop jar C:/hadoop-mapreduce-examples-2.10.1.jar wordcount /dir_entrada /dir_saida`.

2. Verifique o conteúdo do arquivo de saída, ou seja, os detalhes do processamento do arquivo, executando a seguinte linha de comando: `hadoop dfs -cat /dir_saida/*`

A contagem de cada palavra do arquivo de entrada: a palavra processamento tem apenas uma ocorrência, enquanto a palavra texto aparece três vezes.

3. Agora que você conseguiu criar um diretório no HDFS do Hadoop, copiar um arquivo para esse diretório e executar um programa para processá-lo, faça os seguintes procedimentos, executando os comandos no prompt de comando.

    3.1 Saia do Hadoop no modo de segurança: hadoop dfsadmin -safemode leave.

    3.2 Exclua o arquivo do diretório no HDFS: hadoop fs -rm -r /dir_entrada/texto_exemplo.txt

    3.3 Exclua o diretório: hadoop fs -rm -r /dir_entrada.

4. Como você precisa estar atento a muitas aplicações do Hadoop executadas de forma simultânea, além das diversas parametrizações dos arquivos, ao iniciar a execução de um exemplo, observe com cuidado as seguintes situações:

    4.1 Verifique as versões do Hadoop e do Java que estão instaladas em seu sistema.

4.2 Verifique se as variáveis de ambiente JAVA_HOME e HADOOP_HOME estão configuradas corretamente.

4.3 Garanta que o conteúdo dos diretórios datanode e namenode esteja vazio. Atenção: só exclua esses dados se não for usá-los em outro momento, ou seja, se eles não forem realmente necessários.

4.4 Verifique o conteúdo do diretório tmp que está na raiz de diretórios. Atenção: se o conteúdo não for realmente necessário, deverá ser excluído.

4.5 Verifique os arquivos jar, para aplicações de mapreduce, que, normalmente, estão no endereço: C:\hadoop\share\hadoop\mapreduce.

4.6 No começo da execução de um exemplo, lembre-se de executar a sequência de passos: `hadoop namenode -format` e `start-all.cmd`.

4.7 Nunca perca a paciência! Se ainda assim tiver problemas, verifique se os arquivos estão configurados corretamente.

Atividade

Questão 1

Parabéns por ter concluído esse trabalho! Você conseguiu executar uma aplicação prática no Hadoop.

Agora, você vai executar mais um exemplo prático. Para isso, siga todos os passos apresentados anteriormente para iniciar o Hadoop. Em seguida, copie o arquivo hadoop-mapreduce-examples-2.10.1.jar para o diretório C:/teste. Execute o programa no prompt de comando, conforme a seguinte linha:

```
hadoop jar C:/teste/hadoop-mapreduce-examples-2.10.1.jar
```

Qual foi a saída da execução do programa?

Abrir solução

A saída do programa vai exibir diversas informações sobre os parâmetros que podemos passar para o programa hadoop-mapreduce-examples-2.10.1.jar. Veja alguns desses parâmetros e uma breve descrição da sua funcionalidade:

• aggregatewordcount – programa de mapeamento e redução que conta as palavras nos arquivos de entrada.

• aggregatewordhist – programa de mapeamento e redução baseado em agregação, que calcula o histograma das palavras nos arquivos de entrada.

• dbcount – programa que faz as contagens das visualizações de uma página de um banco de dados.

• grep – programa de mapeamento e redução que conta as correspondências de um regex na entrada.

• join – programa que efetua uma junção em conjuntos de dados classificados e igualmente particionados.

• multifilewc – programa que conta palavras de vários arquivos.

• randomtextwriter – programa que grava 10 GB de dados textuais aleatórios por nó.

• sudoku – solucionador de sudoku.

• wordcount – programa que conta as palavras nos arquivos de entrada.

• wordmean – programa que calcula o comprimento médio das palavras nos arquivos de entrada.

• wordmedian – programa que calcula a mediana do comprimento das palavras nos arquivos de entrada.

• wordstandarddeviation – programa que calcula o desvio padrão do comprimento das palavras nos arquivos de entrada.

Parabéns por ter realizado esses exercícios práticos! Agora, você já sabe como utilizar o Hadoop. Aproveite para aprofundar seus conhecimentos e fazer pesquisas no mercado sobre as oportunidades para os profissionais dessa área. Bons estudos!

Conclusão

O que você aprendeu neste conteúdo?

- O conceito de arquitetura do Hadoop;
- As diferenças entre os sistemas HDFS e RDBMS;
- Os aspectos de um data lake;
- Os passos para preparação do ambiente ao uso do Hadoop na prática;
- A execução de um exemplo prático no Hadoop.

Explore +

- Acesse o site oficial do **Hadoop** e aprofunde seu conhecimento nessa tecnologia.
- Acesse o site da **CVE** (Common Vulnerabilities and Exposures) e pesquise sobre vulnerabilidades já catalogadas do Hadoop.

Referências bibliográficas

ISHWARAPPA, K.; ANURADHA, J.. **A brief introduction on Big Data 5Vs characteristics and Hadoop technology.** Procedia Computer Science 48, 319 – 324, 2015. International Conference on Computer, Communication and Convergence (ICCC 2015).

JETBRAINS. **PyCharm.** Consultado na Internet em: 28 set. 2021.

ORACLE. **Java Downloads.** Consultado na Internet em: 28 set. 2021.

SINGH, A.; AHMAD, S. **Architecture of data lake.** International Journal of Scientific Research in Computer Science, Engineering and Information Technology, v. 5, n. 2, p. 411-414, 2019.

THE APACHE SOFTWARE FOUNDATION. **Apache Hadoop Download.** Consultado na Internet em: 28 set. 2021.

WHITE, T. **Hadoop:** the definitive guide. O'Reilly Media, Yahoo! Press, April 2015.