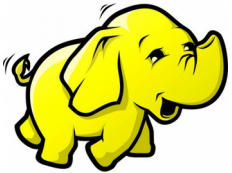


Cloudera Administrator Apache Hadoop

Parte 01-1 Introdução



Marco Reis
<http://marcoreis.net>

Agenda do curso



- Parte 01
 - Apache Hadoop
 - HDFS
 - MapReduce
 - Planejando o Cluster Hadoop
 - Instalação do Hadoop e configuração inicial
 - Instalação e configuração do Hive, Impala e Pig
- Parte 02
 - Clientes Hadoop
 - Inserindo dados no HDFS
- Parte 03
 - Cloudera Manager
 - Configuração avançada
- Parte 04
 - Segurança do Hadoop
 - Manutenção, Monitoramento e Troubleshooting
 - Gerenciamento de jobs

Agenda



- Apache Hadoop
- HDFS
- MapReduce

Apresentação



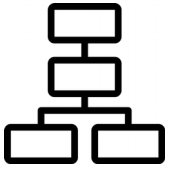
- Quem
 - Marco Reis
 - Mestre em Computação
 - Software Engineer | Big Data | Cloud Computing | Information Retrieval
 - Professor
 - Palestrante
 - **<http://marcoreis.net>**

Perfil do aluno



- Linux (Ubuntu)
- Rede
- SQL
- Java (opcional)
- Criatividade

O modelo tradicional



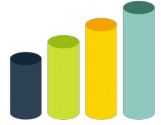
- Linguagem de alto nível (Java/.NET/Ruby)
- Banco de dados relacionais
- Padronização
 - Padrões de projeto de software
 - SQL ANSI
 - Servidor de aplicação/web
- Armazenamento e hardware barato
- Opções com software open source ou pago
- Nuvem pública
 - Amazon, Microsoft, Oracle
- Modelo maduro para desenvolvimento de software

Chega o big data



- Implementação mais conhecida: Hadoop
 - Não é a única, mas foi a responsável pela popularização do termo
- Complementação do modelo tradicional
 - Proposta de não substituição da arquitetura tradicional
- Pode ser integrado com outras ferramentas
 - Driver para banco de dados relacional
 - Messaging
 - API Java
 - API RESTful

Big Data



- Um termo novo para um problema antigo
 - Mais dados do que poder de processamento
- Processamento de grande volume de dados
- Os famosos 4 V's iniciais
 - Volume, variedade, velocidade e variabilidade
- Outras características
 - Veracidade, validade, volatilidade e valor
- Alternativa ao modelo tradicional
 - Escala (gigabytes / terabytes / petabytes / exabytes)
 - Velocidade (rede social, logs, integração)
 - Complexidade (dados estruturados, semi-estruturados e não estruturados)
- Não existe uma definição formal e consolidada para Big Data
 - Cada fabricante ou instituição tem a sua

Apache Hadoop

- Framework open source para processamento distribuído de grandes volumes de dados por meio de clusters de computadores de baixo custo
- Usado na computação em nuvem e aplicações big data
- Inicialmente, o Hadoop era baseado apenas no HDFS e no MapReduce
- O framework evoluiu para além do processamento batch, de forma que o termo Hadoop tem sido usado como sinônimo de um ecossistema de ferramentas de infraestrutura para computação distribuída
- Usado em situações diversas, como mecanismos distribuídos para consultas SQL interativas em grandes bases de dados, processamento iterativo em algoritmos de machine learning, processamento de dados em fluxo constante (streaming) e sistemas de busca textual
- Implementa características da computação distribuída
- Libera a TI para pensar na regra de negócio



Apache Hadoop (Módulos)

- Hadoop MapReduce
 - Um sistema para processamento paralelo de big data
- Hadoop Distributed File System (HDFS)
 - Sistema de arquivos distribuído de alta performance
- Hadoop Common
 - Programas utilitários e bibliotecas de suporte do projeto
- Hadoop YARN (Yet Another Resource Negotiator)
 - Framework para gerenciamento de recursos do cluster. Cada um desses módulos será descrito em detalhes nas próximas seções



Características do Hadoop

- Escalabilidade
- Alta disponibilidade / replicação
- Hardware “commodity”
- IO de alta performance
- Modelo simples de programação
- Data driven
- Modelagem diferente
- Moving Computation is Cheaper than Moving Data





Modos de operação

- Local
 - Sistema de arquivos local
 - Usado no máximo para desenvolvimento
- Pseudo distribuído
 - Simula um cluster em 1 servidor
 - Desenvolvimento ou homologação
- Totalmente distribuído
 - Implantado em mais de um servidor
 - Usado para homologação ou produção

Candidatos a usar Hadoop...

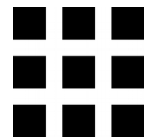


- Grande volume de dados (GB, TB, PB)
- Processamento em lote
 - Sistemas em tempo quase real
- Estrutura de dados complexa
- Processamento comum a um grupo de dados
- Problemas que podem ser convertidos em chave/valor
- Sumarização, agregação, agrupamento, filtros e junção de dados
- Dados que envolvem localização geográfica
- ETL
- Conversão de formatos

Candidatos a usar o ecossistema Hadoop



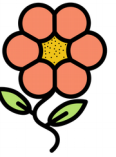
- Processamento de baixa latência
 - Sistemas em tempo real (real-time, streaming)
- Processamento com datasets pequenos
 - O custo de implementação é alto
- Buscar registros individuais
 - Use um banco NoSQL
- Programas que precisam compartilhar estado
 - As tarefas Map/Reduce são independentes e não conversam entre si



Hadoop é bom para...

- Processamento em lote
 - Só isso já resolve muitos problemas
- Armazenamento
- Redução de custos
- Análise de dados históricos

Nem tudo são flores...



- A codificação é simples, mas a manutenção do cluster não
- É uma tecnologia nova, ainda sob teste e avaliação
- A mudança no modelo de processamento exige treinamento para a equipe de tecnologia
- A configuração inicial pode ser trivial, mas a otimização não é...

Outras tecnologias não Hadoop

- Spark
- Google Dremel
- Cassandra
- MongoDB



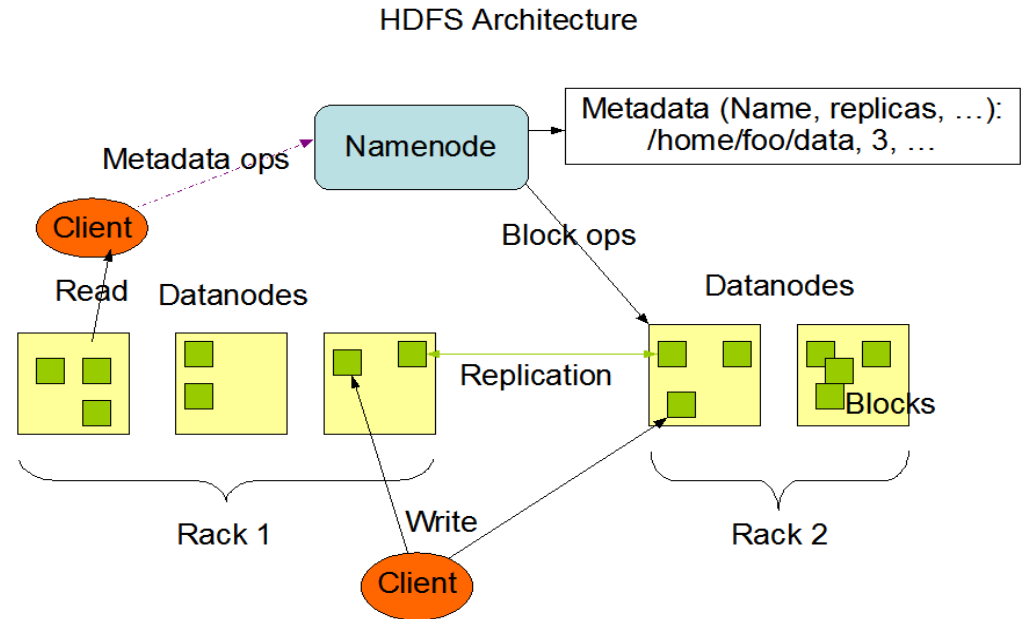
HDFS

- Hadoop Distributed File System
- Sistema de arquivos distribuído para armazenamento de big data
- Suporta alta tolerância a falhas executando em máquinas de baixo custo
- O HDFS considera que as falhas de hardware são inevitáveis e, por isso, replica partes dos arquivos em vários servidores diferentes para garantir a redundância dos arquivos
- Quando algum desses servidores de dados falha, o HDFS recupera automaticamente uma cópia dos arquivos em um servidor disponível que recebeu uma das réplicas daqueles dados
- Usa o conceito do write-once, read many times
 - O arquivo é escrito uma única vez e lido várias vezes com alta taxa de transferência
 - Um arquivo pode ser criado, escrito e fechado
 - Só pode ser alterado pelas operações de append ou truncate no final do arquivo
 - Alterações no meio dos arquivos não são permitidas.



Arquitetura

- Cliente solicita os dados ao NameNode.
- Cliente grava os dados obedecendo as regras de replicação



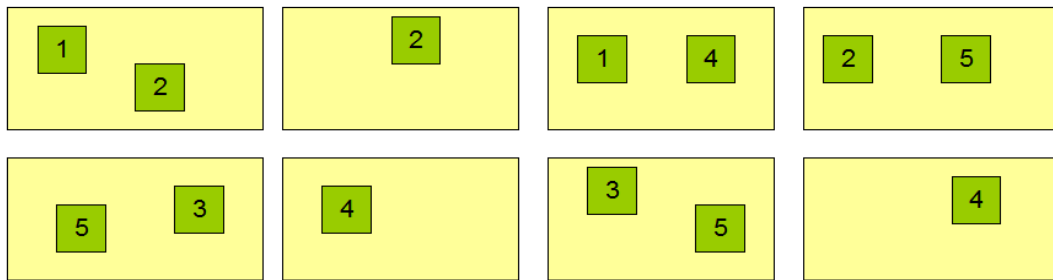
Replicação

- NameNode conhece os metadados e os DataNodes
 - nós disponíveis
 - nome dos arquivos
 - número de réplicas
 - localização dos blocos
- DataNode armazena os blocos

Block Replication

Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes



Espaço de armazenamento



- Considere 1 GB de dados
 - São ~8 blocos de 128 (ou 256) MB
- Replicação x 3
 - Cada bloco de dados terá 3 cópias
- Espaço total de armazenamento necessário
 - 3 GB

Disco rígido



- Por uma questão de custo X benefícios, usamos discos mecânicos
 - Se é mecânico, não pode ser muito rápido

Acesso a dados

- Sequencial
 - Máxima performance
- Randômica
 - Limitada pelo meio físico



MapReduce

- Sistema de processamento em batch e um modelo de programação para escrever aplicações distribuídas que acessam grande volumes de dados em paralelo
- Framework para processamento paralelo, com um modelo de programação simplificado, abstraindo a complexidade das operações do processamento distribuído
- Suporta clusters com tolerância a falhas rodando em milhares de nós
- Acessa os dados do HDFS (padrão) ou várias outras fontes de dados
- Usado pelo Google no começo da empresa

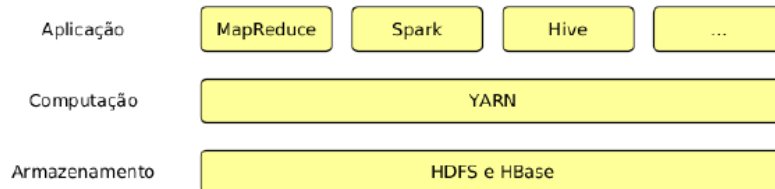


Comparativo SGBDR/MapReduce

Característica	SGBDR	MapReduce
Tamanho dos dados	<i>Gigabytes</i>	<i>Petabytes</i>
Acesso	Interativo e <i>batch</i>	<i>Batch</i>
Atualizações	Leitura e gravação repetidamente (<i>Read and write many times</i>)	Uma única gravação e várias leituras (<i>Write once, read many times</i>)
Transações	ACID	Não há
Estrutura	Definida antes da gravação (<i>Schema-on-write</i>)	Definida no momento da leitura (<i>Schema-on-read</i>)
Integridade	Alta	Baixa
Escalabilidade	Não-linear	Linear

YARN

- O YARN faz monitoramento e alocação dos processadores, disco e memória, para cada execução de programa no cluster Hadoop
- Uma aplicação Hadoop não faz acesso direto aos recursos computacionais do cluster, ela acessa o YARN
- As aplicações são escritas com um framework suportado pelo YARN, que pode ser MapReduce, Apache Pig, Apache Hive, Apache Spark, Apache Tez etc
- O framework, que está na camada de aplicação, é responsável pelo acesso ao YARN, que por sua vez tem interface com os recursos computacionais e o sistema de arquivos
- Dias versões disponíveis:
 - MRv1 e MRv2 (ou NextGen MapReduce)



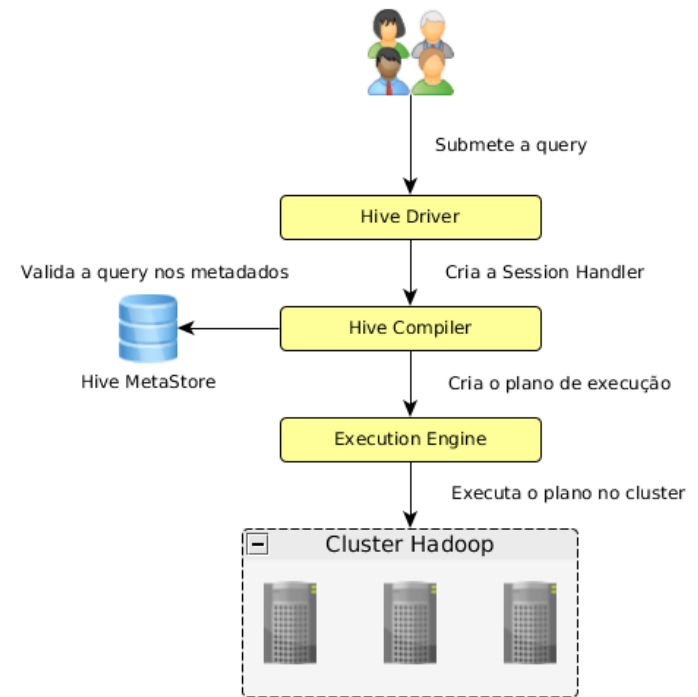
SQL on Hadoop



- A programação em MapReduce é simples, mas demanda tempo e pessoal especializado em programação
- Alternativa: SQL on Hadoop
- Diversas ferramentas foram criadas para aproveitar o conhecimento de SQL da equipe de TI
 - Hive, Impala, Drill e Spark SQL
- Suporte limitado a ANSI SQL
- Diversos desafios para otimização e latência
- Multiplicidade de formatos (Texto, ORC, Avro, Parquet, JSON, XML)

Hive

- O Hive usa o HDFS para armazenamento e YARN ou Spark para o processamento
- Composto por MetaStore, Driver, Compiler, Execution Engine
- MetaStore
 - Grava os metadados, como a definição dos dados armazenados, com a descrição das bases de dados, tabelas, partições, colunas e tipos de dados
 - Os metadados da MetaStore são armazenados em um SGBDR (Postgres, MySQL, Oracle etc)
- Driver
 - Recebe a query, envia para o plano de execução para o mecanismo de processamento e recebe o resultado
- Compiler
 - Cria o plano de execução, traduzindo a consulta SQL em um job YARN/Spark a partir dos metadados da MetaStore
- Execution Engine
 - Usa o plano de execução para agendar os jobs no YARN e recuperar os resultados

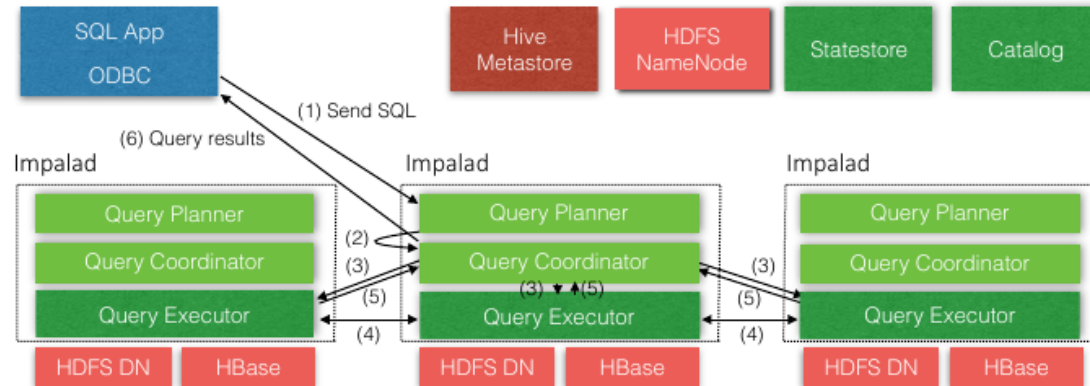


Impala

- O Cloudera Impala é um mecanismo open source para consultas SQL de baixa latência e alta concorrência baseado em MPP (Massively Parallel Processing)
- Usado para análise de dados e BI no Hadoop, acessa a MetaStore do Hive, o HDFS, YARN, HBase etc.
- Sua arquitetura é baseada em 3 componentes:
 - Impala Daemon (impalad): serviço que roda em cada datanode e executa as consultas dividindo as tarefas entre os nós;
 - StateStore Daemon (statestored): serviço de atualização de metadados para os processos do Impala em todo o cluster;
 - Catalog Daemon (catalogd): repositório central dos metadados do Impala, que é atualizado pelo statestored.

Arquitetura do Impala

- A arquitetura é semelhante à do YARN/HDFS: o cliente envia a consulta para o Query Planner que será o coordenador da execução e irá dividir as tarefas entre os datanodes que de fato contém os dados
 - Imagem do artigo: A Modern, Open-Source SQL Engine for Hadoop. Marcel Kornacker, Alexander Behm, Victor Bittorf, Taras Bobrovytsky, Casey Ching, Alan Choi, Justin Erickson, Martin Grund, Daniel Hecht, Matthew Jacobs, Ishaan Joshi, Lenni Kuff, Dileep Kumar, Alex Leblang, Nong Li, Ippokratis Pandis, Henry Robinson, David Rorke, Silvius Rus, John Russell, Dimitris Tsirogiannis, Skye Wanderman-Milne, Michael Yoder



Dúvidas?

Marco Reis
<http://marcoreis.net>