



Faculdade

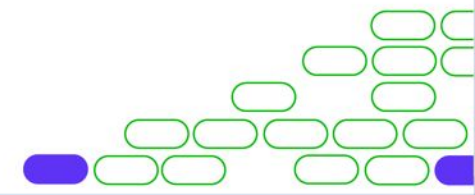


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 1. Introdução ao ecossistema Hadoop

Aula 1.1. Introdução ao Hadoop

Prof. Silas Liu



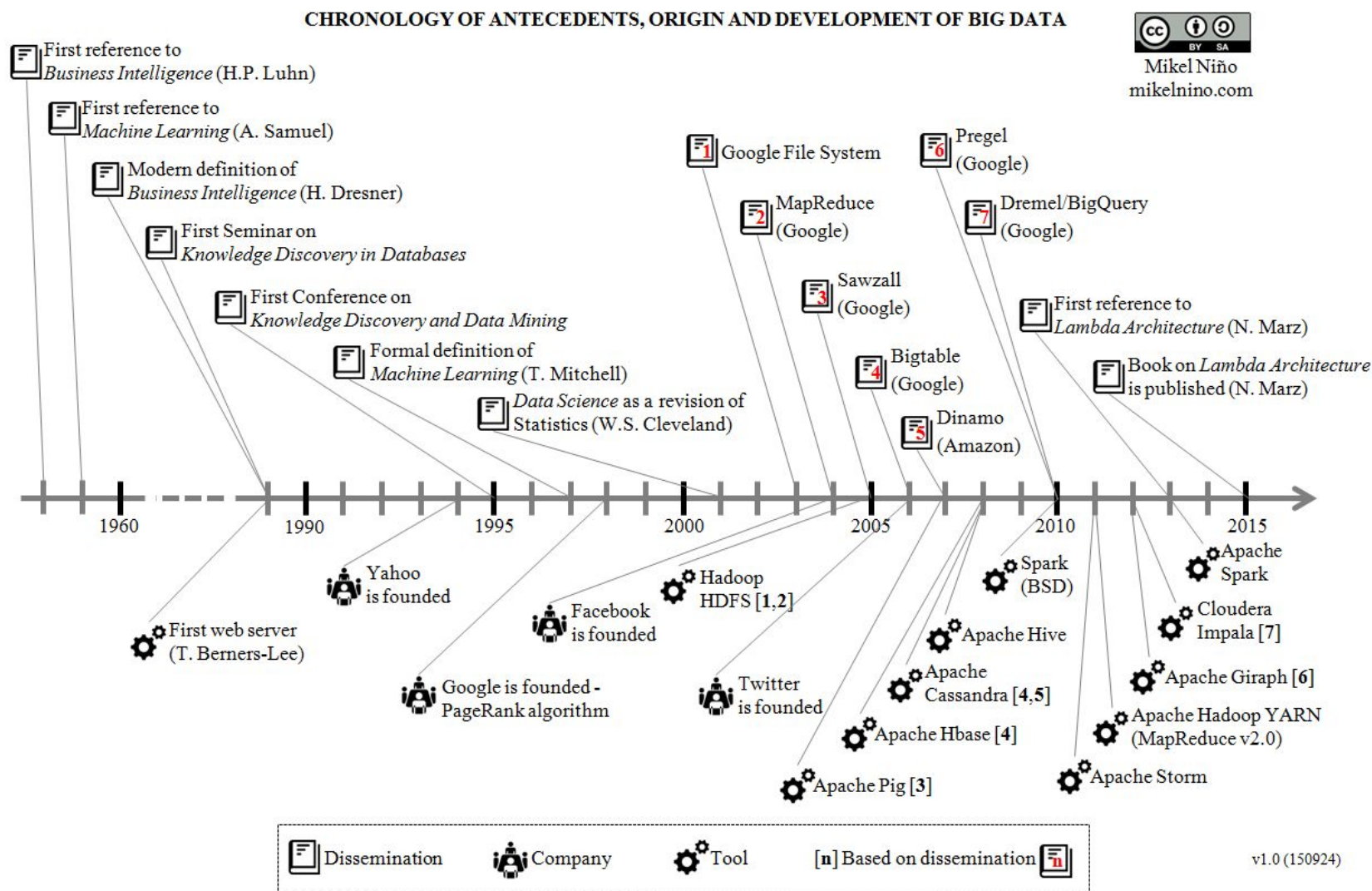
Nesta aula

- ☐ Linha do tempo Big Data.
- ☐ Introdução ao ecossistema Hadoop.
- ☐ Distribuições Hadoop.
- ☐ Escalabilidade.
- ☐ Clusters.
- ☐ Hadoop Core.
- ☐ Frameworks Hadoop.





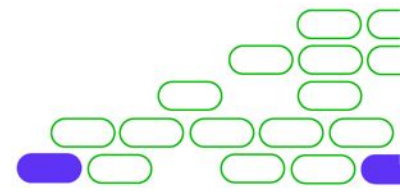
Linha do tempo Big Data



Introdução ao ecossistema Hadoop

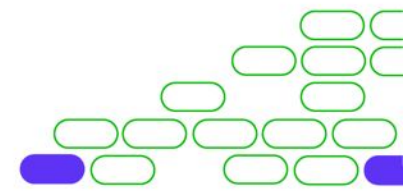


- Hadoop é um **framework**, de código aberto, para leitura, armazenamento e processamento de big-data.
- Seu código foi implementado em **Java**.
- Se caracteriza como processamento **confiável**, **escalável**, e **distribuído**.
- Pode ser rodado em hardwares **comuns**.
- É projetado para detectar e ser **tolerante** a falhas.

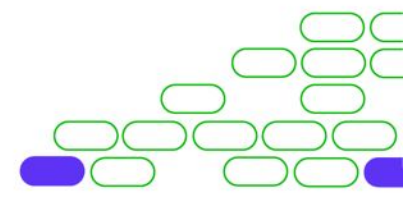
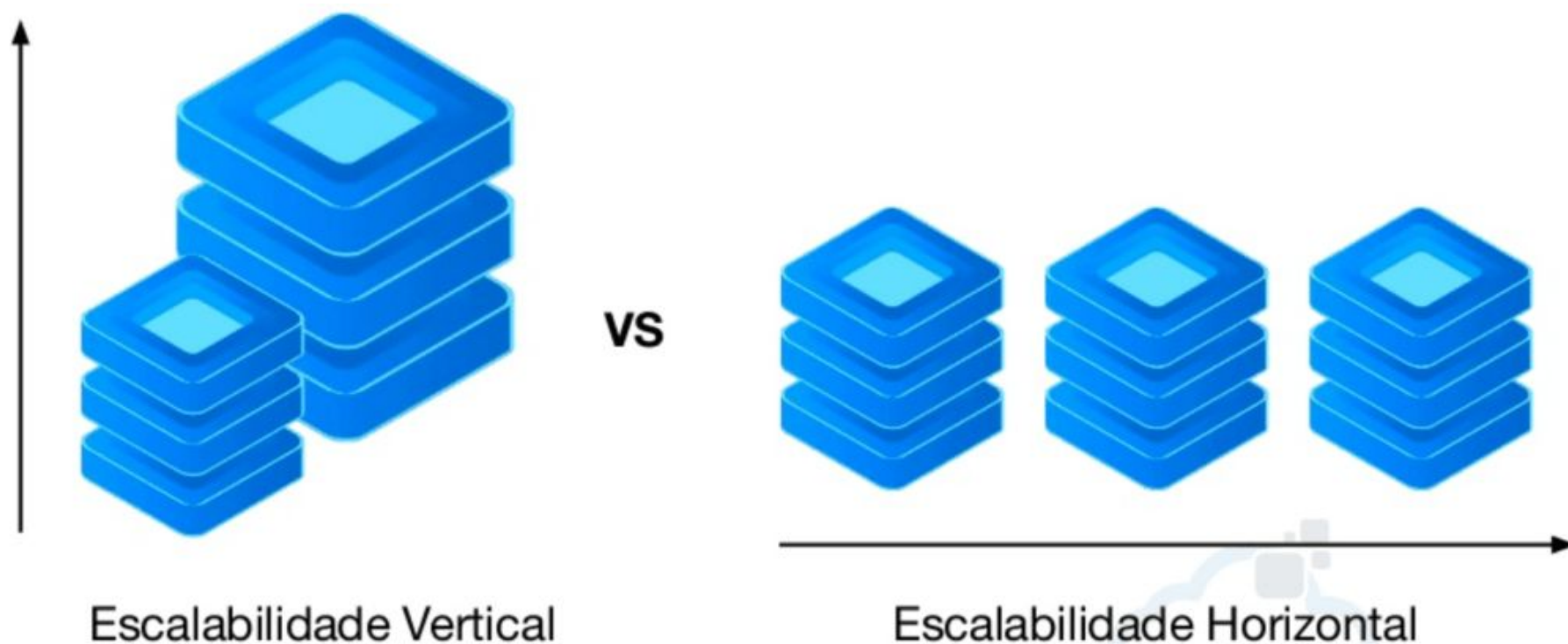


Distribuições Hadoop

- Distro código-aberto:
 - Apache Hadoop.
- Distros comerciais:
 - Cloudera Hadoop.
 - Hortonworks Hadoop.
 - MapR Hadoop.
 - AWS Elastic MapReduce (EMR).
 - Microsoft Azure HDInsight.
 - Google Cloud Dataproc.

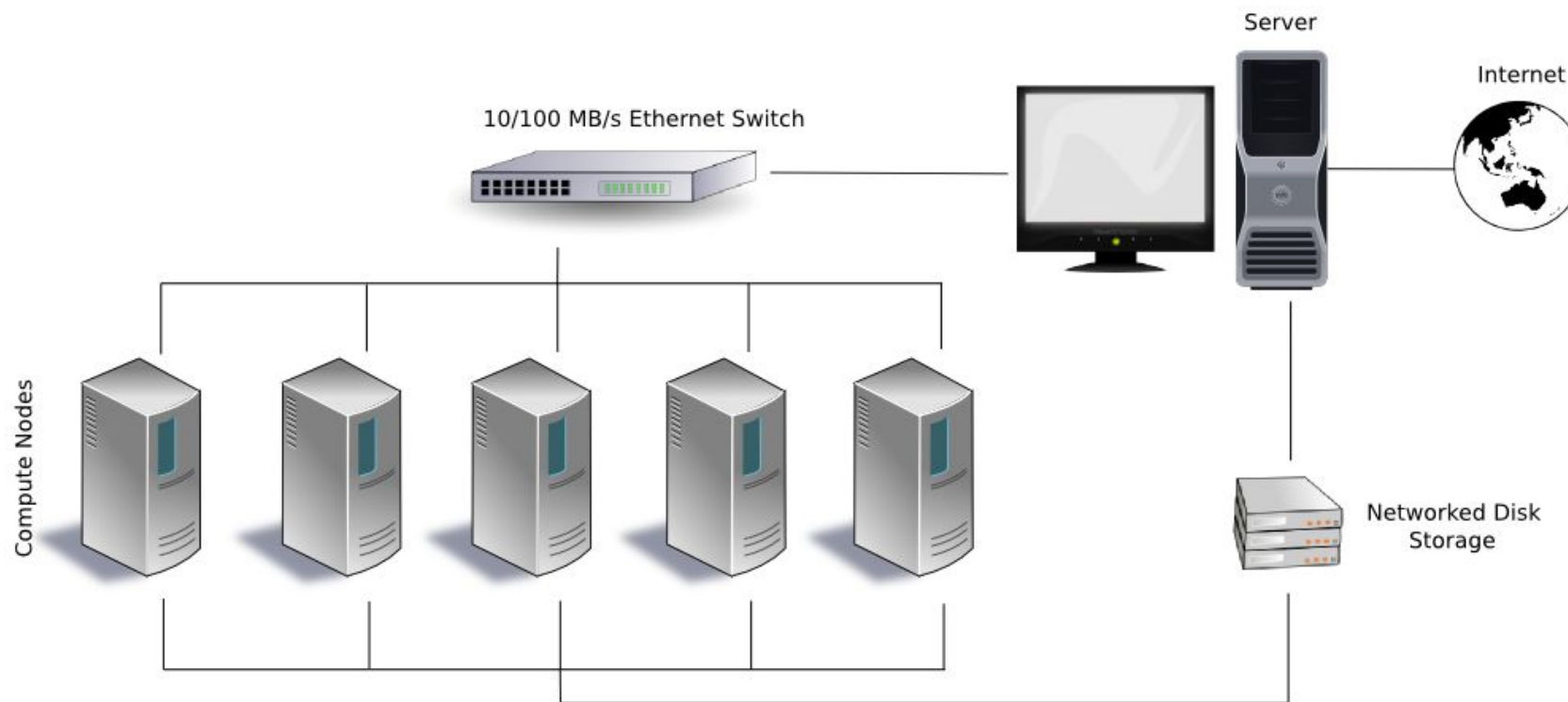


Escalabilidade



Clusters

Agrupamentos de computadores que trabalham juntos. Podem ser gerenciados por um único computador e provêm armazenamento, processamento e intercâmbio de recursos.



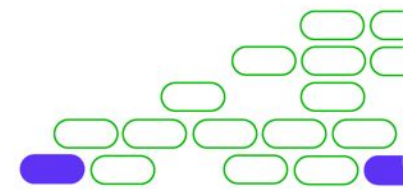
Clusters

- Nodo ou nó:

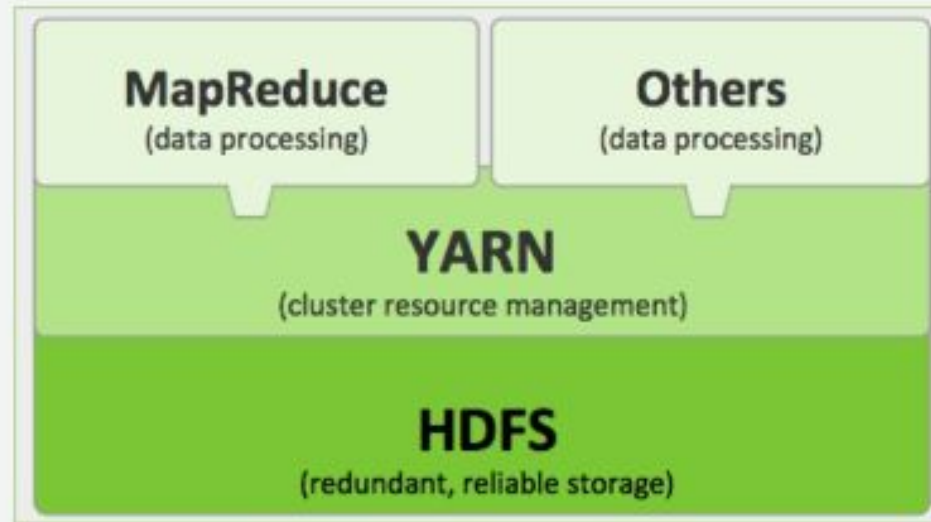
Nodo ou nó é o nome que se dá a um computador individual, dentro de um cluster. Existe o nó master (ou driver), que gerencia e distribui o trabalho entre os demais nós, chamados de nós workers (ou slaves).

- Daemon:

Se dá o nome de daemon ao programa (job ou serviço), que é executado em um nó. Esses daemons podem ser dos mais diversos possíveis.



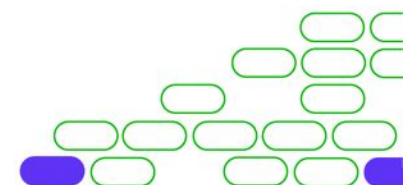
Hadoop Core



Frameworks Hadoop



- **Hive:** abstrai a complexidade de códigos Java em SQL.
- **Hbase:** banco de dados NOSQL.
- **Spark:** executa processamento massivo de dados em memória RAM.
- **Sqoop:** possibilita import e export do HDFS.
- **ZooKeeper:** serviço de coordenação de serviços distribuídos.
- **Ambari:** gerenciamento de clusters Hadoop.
- **Ranger:** gerencia a segurança de clusters Hadoop.
- **Atlas:** serviço de governança.
- **Nifi:** serviço para automatizar fluxos de dados.
- **Kafka:** serviço de mensagens para fluxo de dados em streaming.
- **Flink:** serviço para processamento de dados em tempo real.



Conclusão

- ❑ Hadoop é um framework completo, para leitura, armazenamento e manipulação de big data.
- ❑ O Hadoop se aproveita de clusterização, para prover escalabilidade horizontal.
- ❑ Sua estrutura o torna confiável e tolerante a falhas.



Próxima aula

- ☐ Hadoop Core: HDFS.
- ☐ O que é o HDFS?
- ☐ Estrutura do HDFS.
- ☐ Como o HDFS funciona?





Faculdade

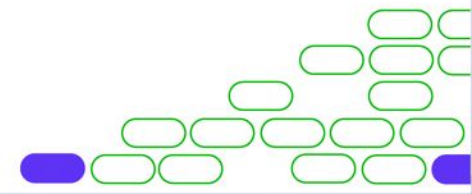


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 1. Introdução ao ecossistema Hadoop

Aula 1.2. Introdução ao HDFS

Prof. Silas Liu

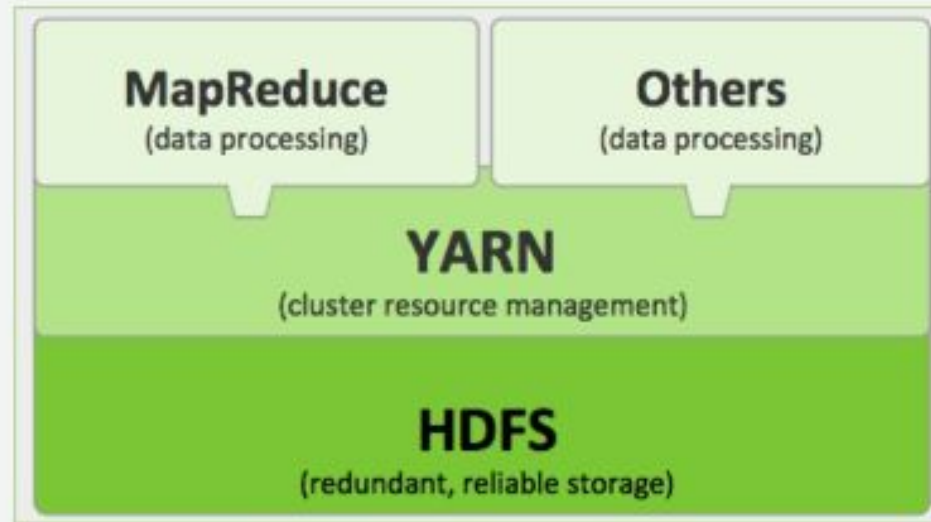


Nesta aula

- ☐ Hadoop Core: HDFS.
- ☐ O que é o HDFS?
- ☐ Estrutura do HDFS.
- ☐ Como o HDFS funciona?



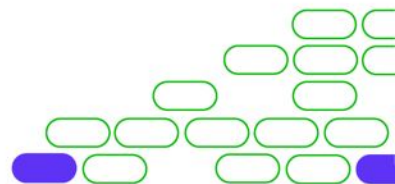
Hadoop Core: HDFS



O que é o HDFS?

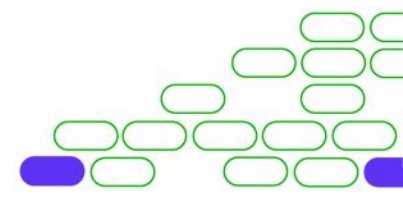
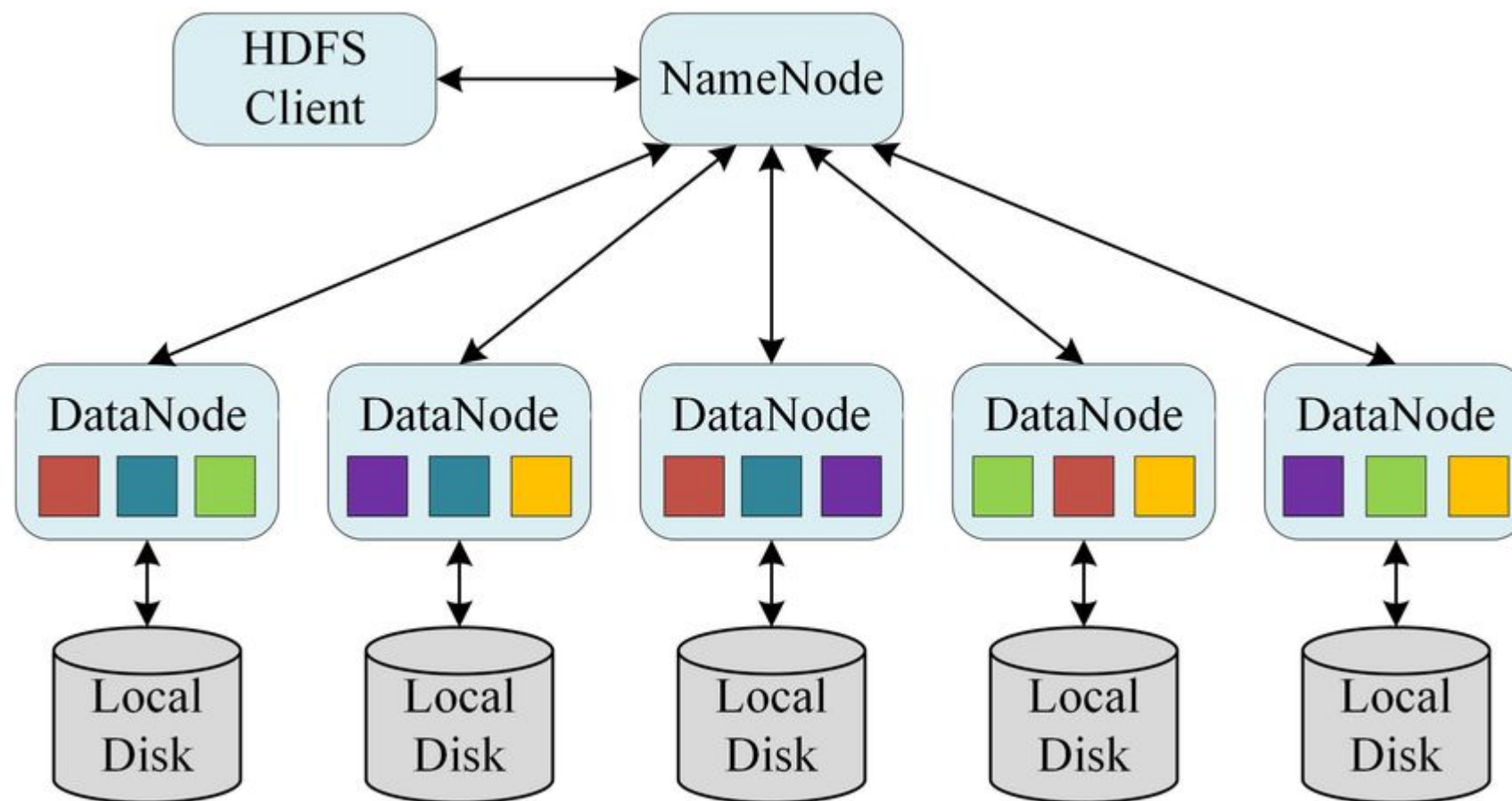


- HDFS: Hadoop Distributed File System.
- Baseado no Google File System (GFS).
- Sistema de armazenamento:
 - Escalável.
 - Tolerante a falhas.
 - Aceita dados tabulares e NOSQL.
- Possui fator de replicação (padrão 3).

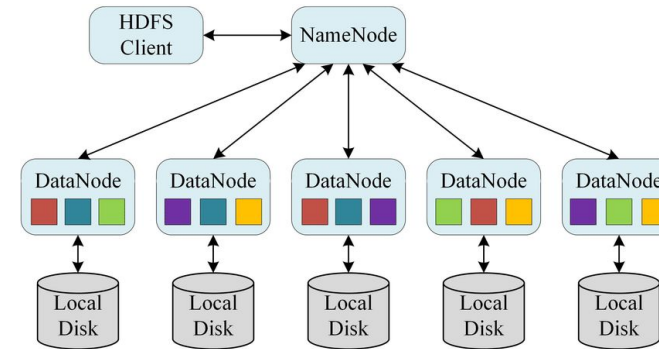




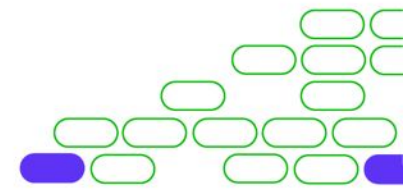
Estrutura do HDFS



Estrutura do HDFS

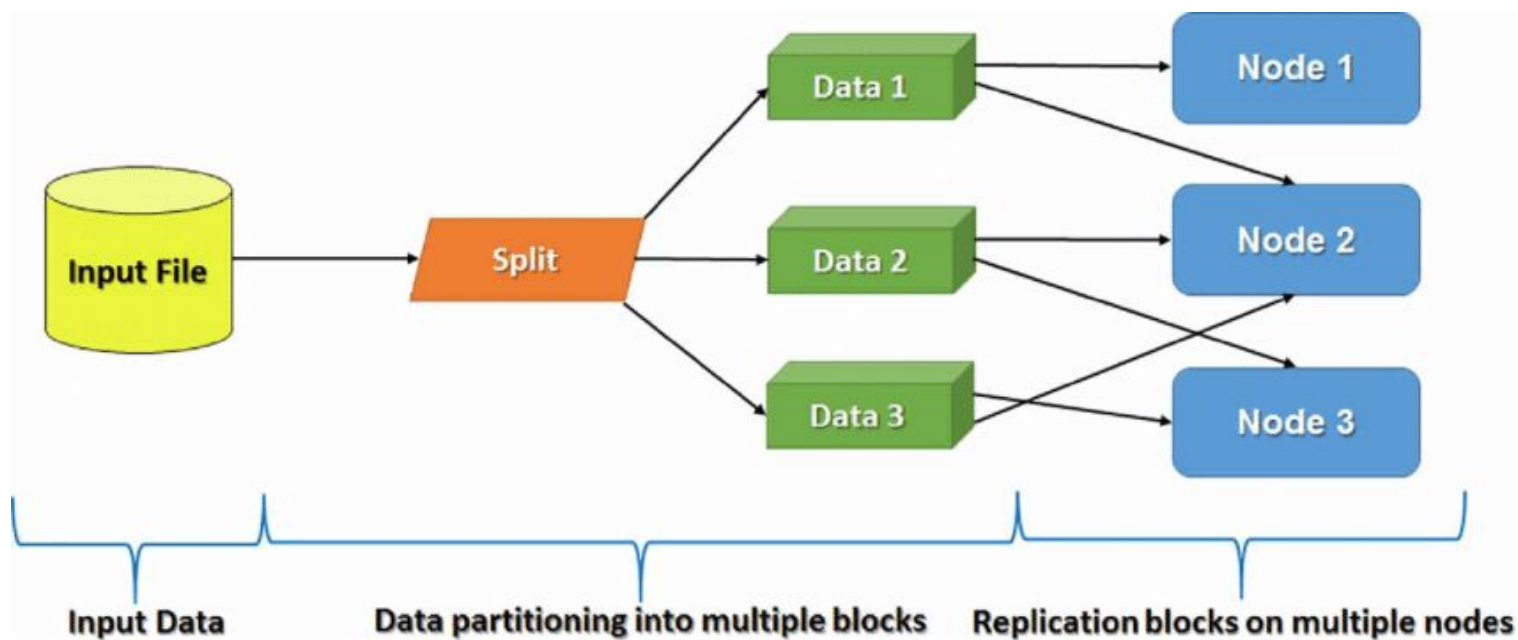


- NameNode:
 - Gerencia o Namespace.
 - Responsável por armazenar o metadado dos demais clusters.
- Secondary NameNode:
 - Disponibiliza pontos de verificação e manutenção do NameNode.
- DataNode:
 - Responsável por armazenar os blocos de arquivos.

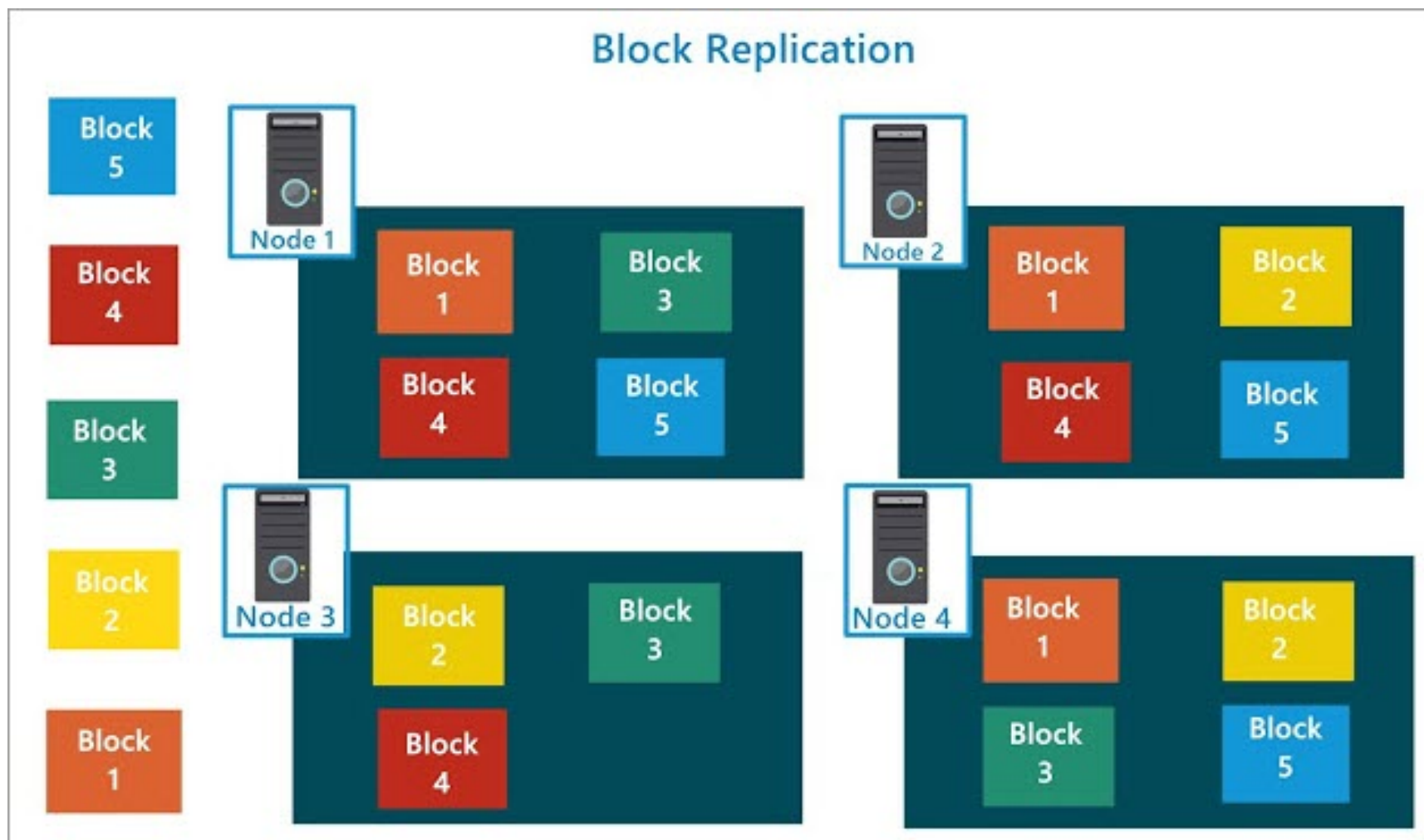


Como o HDFS funciona?

- Os dados são separados em blocos.
(tamanho padrão de 128 MB por bloco)
- Cada bloco é replicado e armazenado em um DataNode.
- O NameNode armazena os metadados correspondentes.



Como o HDFS funciona?



Conclusão

- ❑ HDFS é o método de armazenando distribuído utilizado pelo Hadoop.
- ❑ HDFS é constituído pelo NameNode e DataNodes.
- ❑ HDFS oferece escalabilidade.
- ❑ HDFS é tolerante a falhas.



Próxima aula

- ☐ Hadoop Core: YARN.
- ☐ O que é o YARN?
- ☐ Estrutura do YARN.





Faculdade

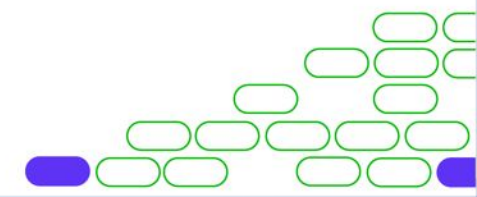


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 1. Introdução ao ecossistema Hadoop

Aula 1.3. Introdução ao YARN

Prof. Silas Liu

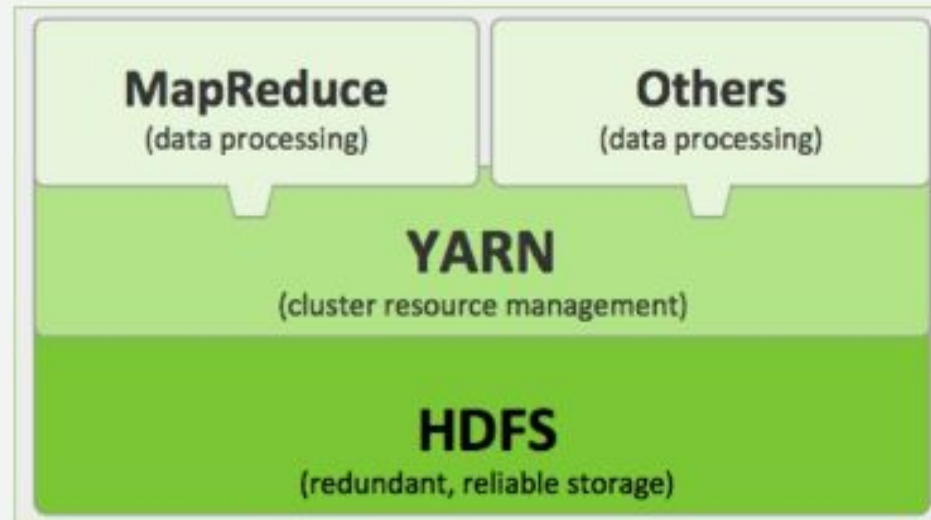


Nesta aula

- ☐ Hadoop Core: YARN.
- ☐ O que é o YARN?
- ☐ Estrutura do YARN.



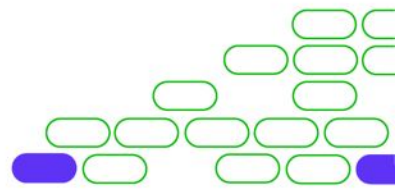
Hadoop Core: YARN



O que é o YARN?



- YARN: Yet Another Resource Negotiator.
- Foi desenvolvido a partir da versão 2.0 do Hadoop.
- Realiza o gerenciamento dos recursos, otimizado para o processamento paralelo.
- Gerencia e monitora os jobs (serviços) dos nós.
- Provê os recursos aos nós apenas quando necessário, sob requisição.

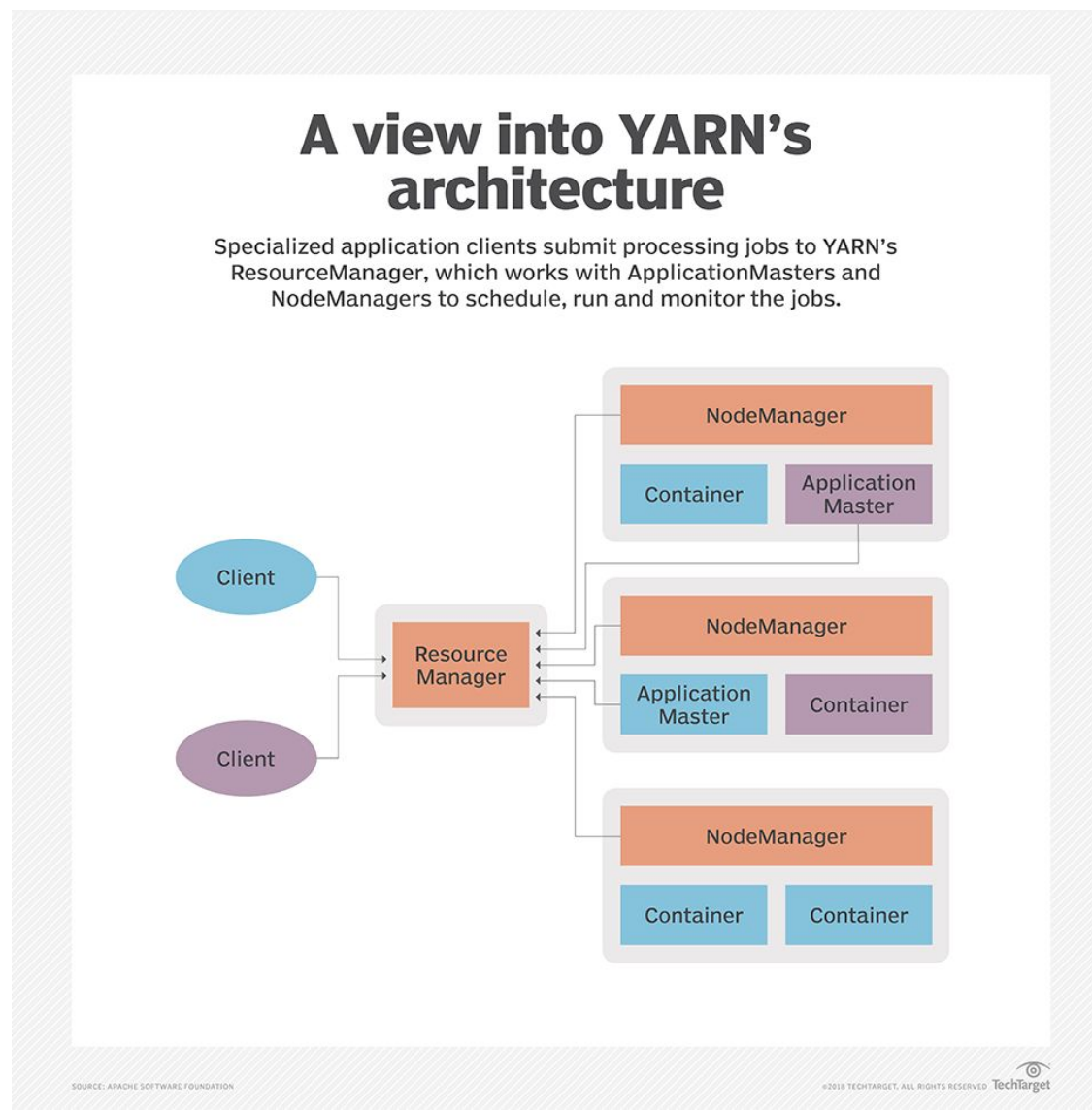


Estrutura do YARN

- Resource Manager: gerenciador global dos jobs e recursos.
- Node Manager: é inicializado em cada nó, monitora os recursos do container e reporta ao Resource Manager.
- Application Master: é inicializado um para cada job, ele gerencia a tarefa negociando recursos com o Node Manager.
- Container: unidade de alocação de recursos (memória, processamento), controlado pelo Node Manager.



Estrutura do YARN



Conclusão

- ❑ YARN é o gerenciador de recursos e atividades do Hadoop.
- ❑ Um Resource Manager gerencia todos os recursos.
- ❑ Um Node Manager por nó gerencia os containers.
- ❑ Um Application Master gerencia cada atividade.



Próxima aula

- ☐ Hadoop Core: MapReduce.
- ☐ O que é o MapReduce?
- ☐ Funcionamento do MapReduce.





Faculdade

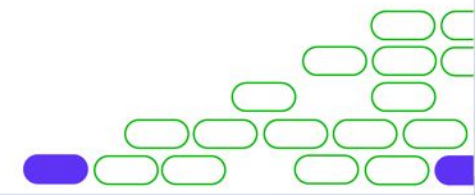


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 1. Introdução ao ecossistema Hadoop

Aula 1.4. Introdução ao MapReduce

Prof. Silas Liu

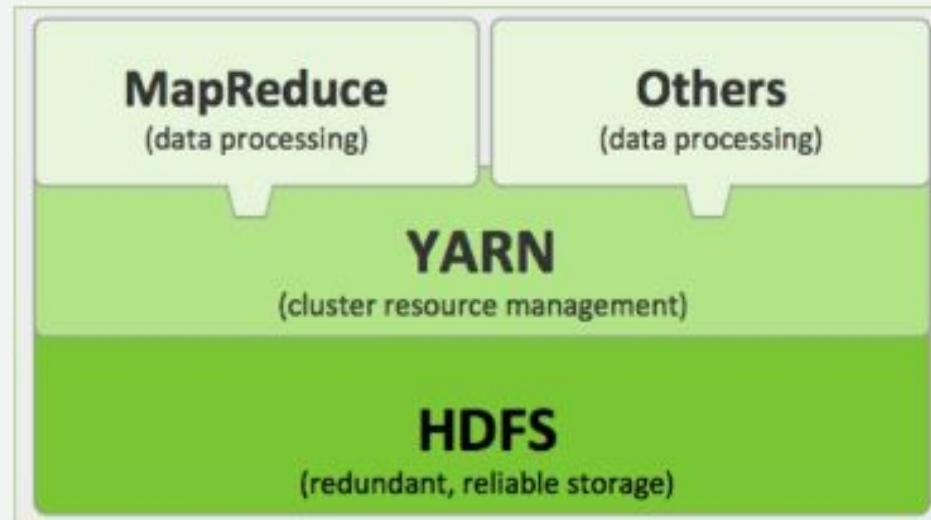


Nesta aula

- ☐ Hadoop Core: MapReduce.
- ☐ O que é o MapReduce?
- ☐ Funcionamento do MapReduce.



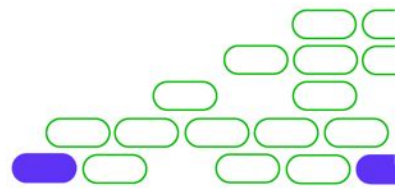
Hadoop Core: MapReduce



O que é o MapReduce?



- MapReduce: modelo de programação para o processamento massivo de dados, de forma paralela e distribuída.
- Projetado para funcionar independente da quantidade e como os dados estejam distribuídos entre os clusters.
- Incremento considerável na velocidade de processamento de big data utilizando a arquitetura convencional.



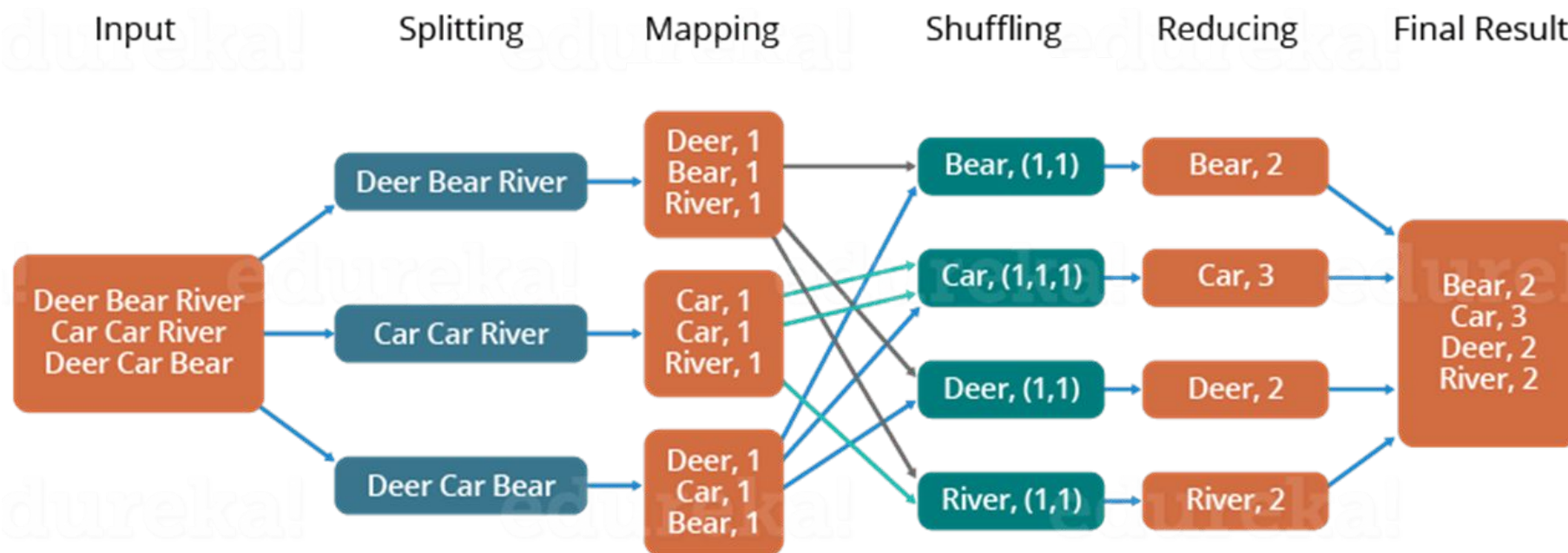
Funcionamento do MapReduce

- Splitting (separação): os dados são divididos entre os clusters inerente à estrutura HDFS.
- Mapping (mapeamento): na fase de mapeamento cada cluster faz uma tarefa individual em seus elementos.
- Shuffling (embaralhamento): nessa fase o resultado de Mapping de todos os clusters são reunidos e ordenados em ordem alfabética.
- Reducing (redução): nesta fase, com as chaves já ordenadas, realiza-se outra tarefa para extrair o resultado.



Funcionamento do MapReduce

The Overall MapReduce Word Count Process



Conclusão

- ❑ O processo de MapReduce funciona, independente do tamanho e quantidade de clusters.
- ❑ Cada cluster realiza o processo de forma independente e em paralelo, durante a fase de Mapping.
- ❑ É de suma importância na fase de Shuffling a ordenação de todos os resultados. Isso é feito de forma automática pelo MapReduce.
- ❑ É papel do operador implementar os processos Mapping e Reducing. O Hadoop permite programar nas mais diversas linguagens: Java, Python, R etc.



Próxima aula

- ☐ O que é uma Máquina Virtual?
- ☐ Virtual Machine Softwares.
- ☐ Oracle VM VirtualBox.





Faculdade

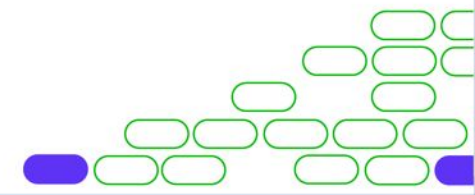


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 2. Apresentação do Ambiente

Aula 2.1. Máquina Virtual

Prof. Silas Liu

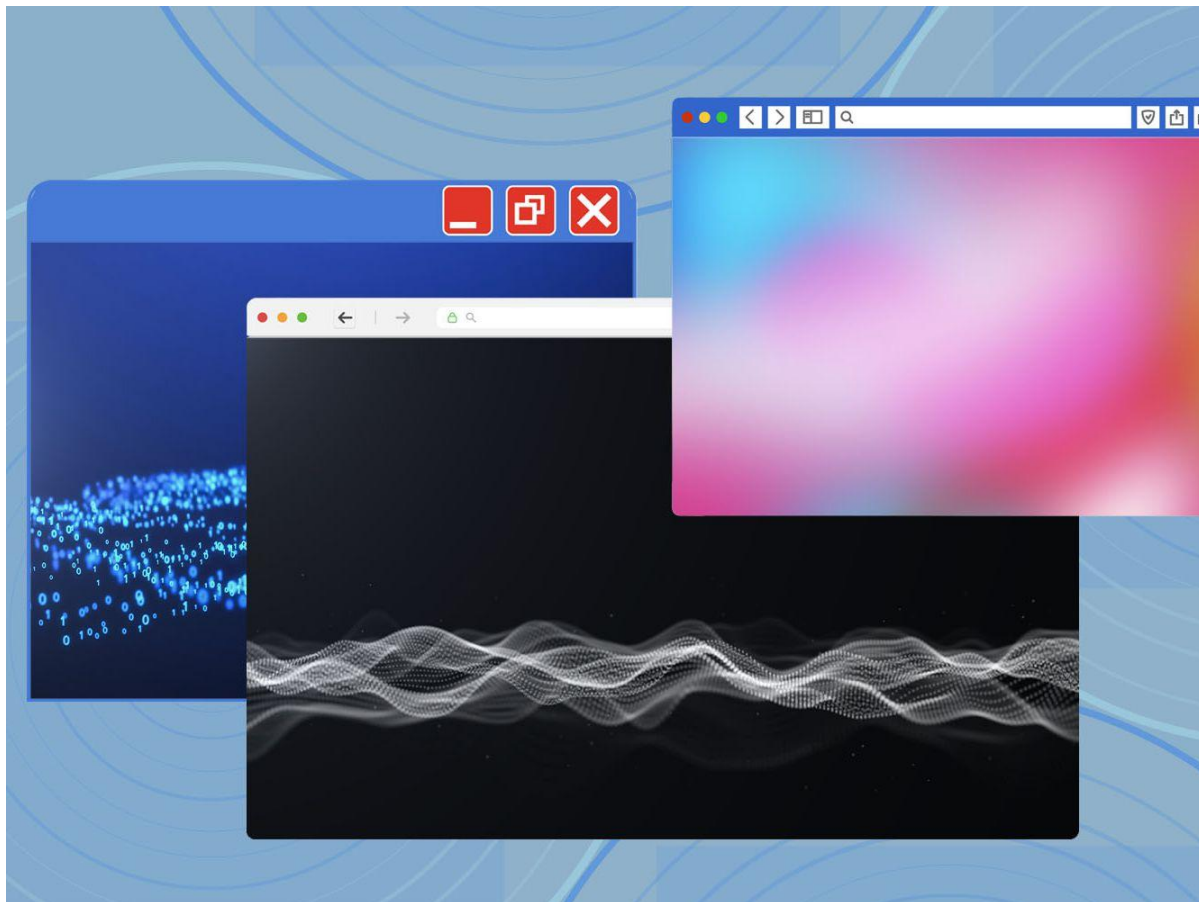


Nesta aula

- ☐ O que é uma Máquina Virtual?
- ☐ Virtual Machine Softwares.
- ☐ Oracle VM VirtualBox.

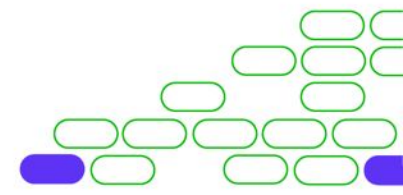


O que é uma Máquina Virtual?



- Máquina Virtual
(Virtual Machine):

É uma ferramenta para simular um sistema operacional dentro do seu computador local.



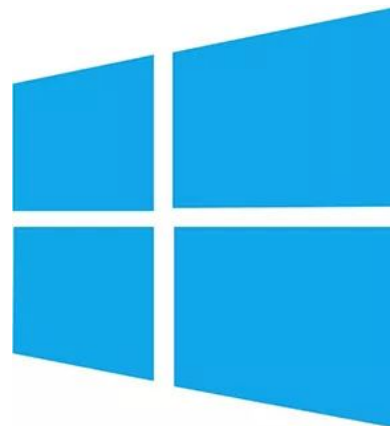
Virtual Machine Softwares



vmware®
Workstation

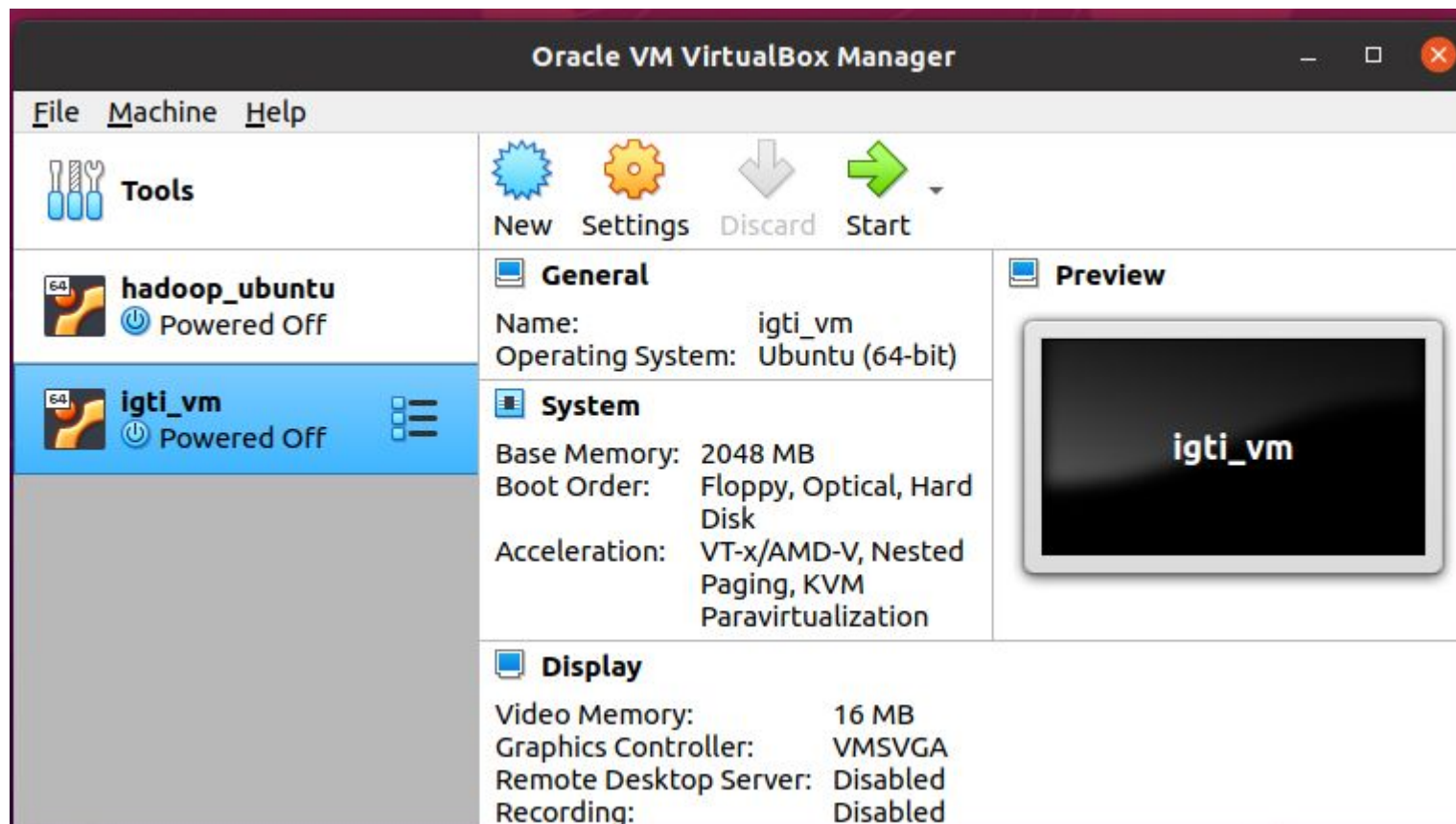


VirtualBox

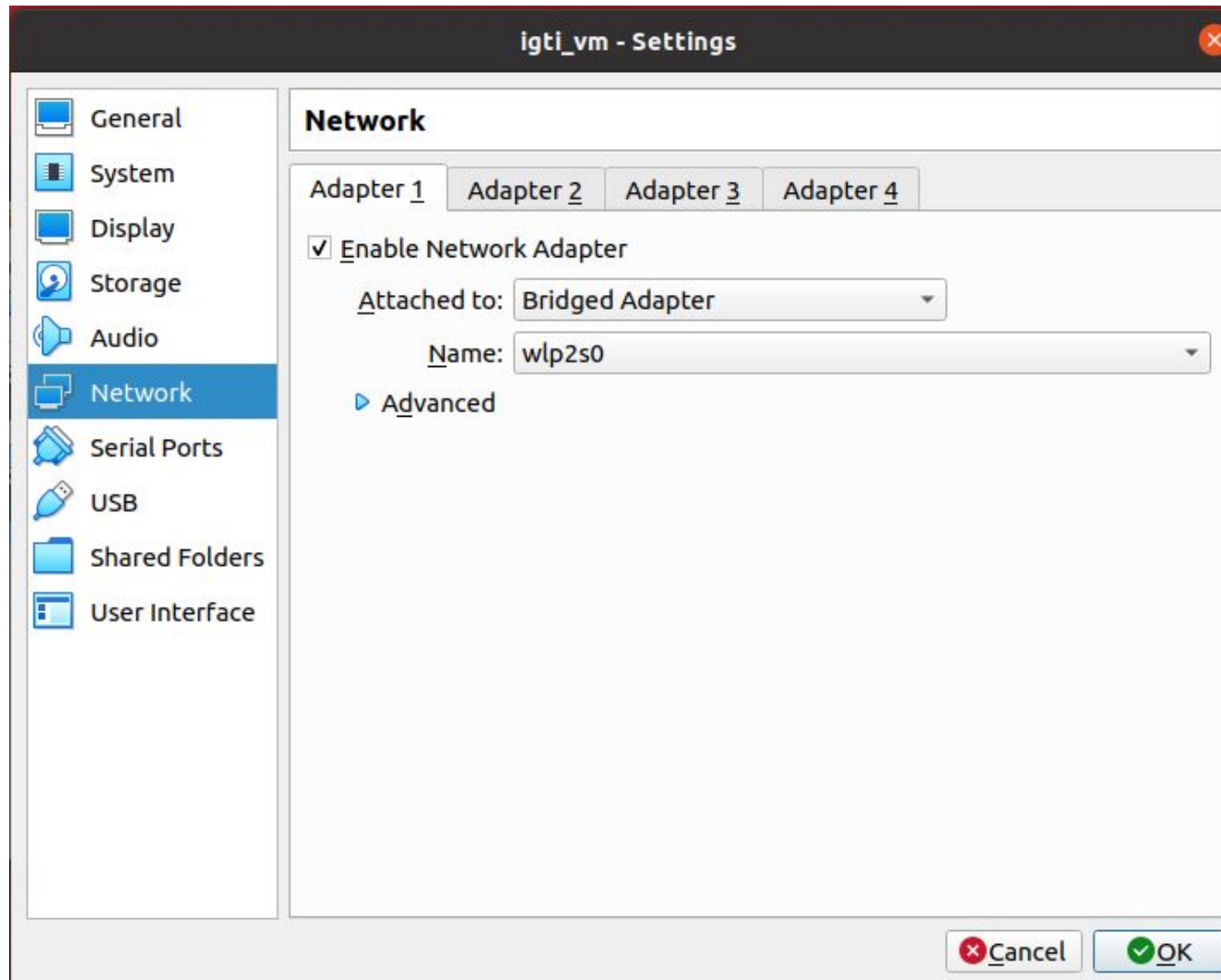


Microsoft
Hyper-V

Oracle VM VirtualBox



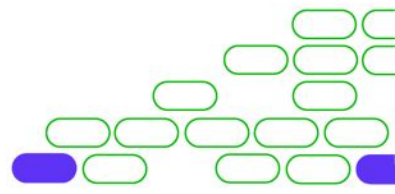
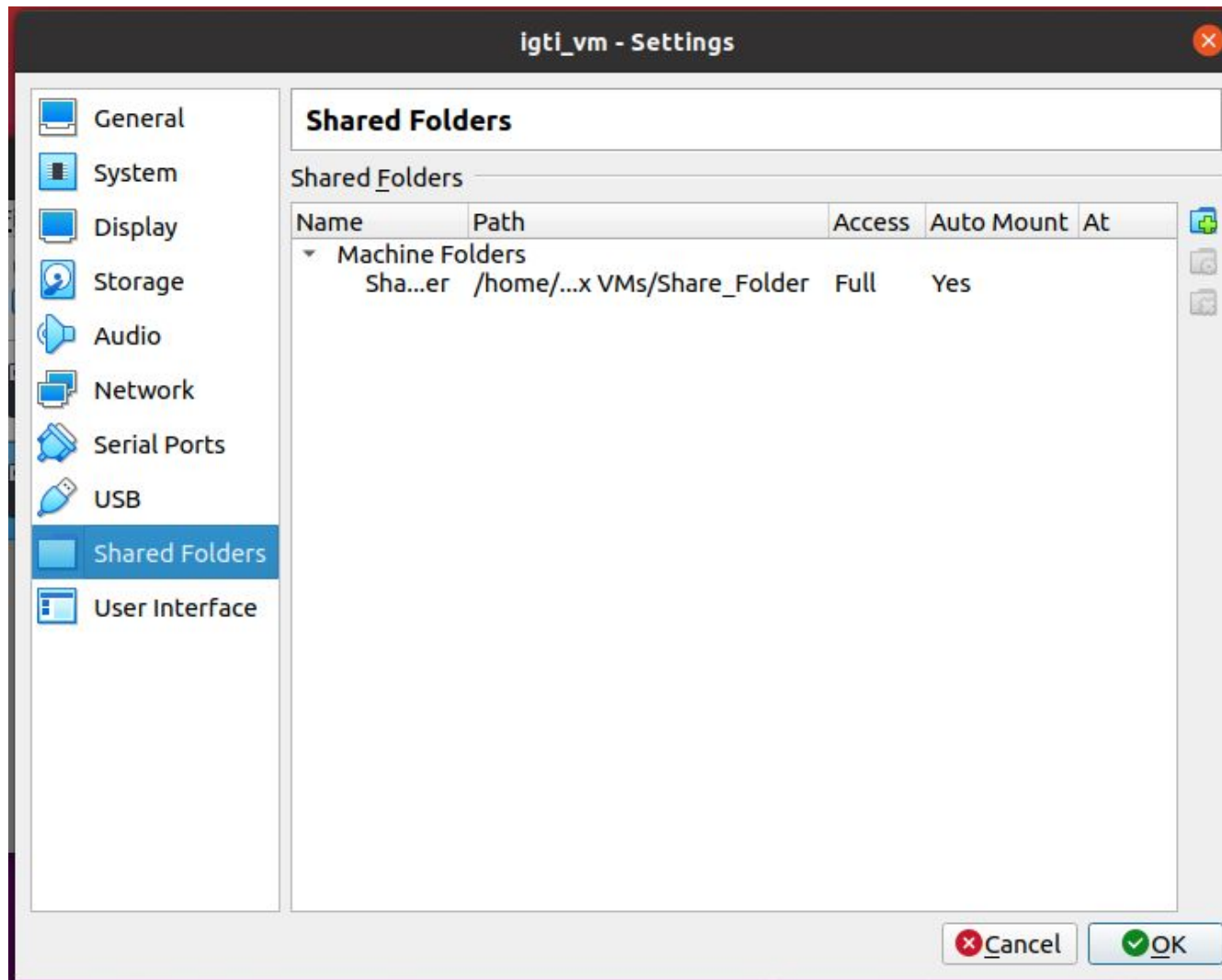
Oracle VM VirtualBox



Oracle VM VirtualBox



XPe



Conclusão

- ❑ Máquinas Virtuais (Virtual Machines) são softwares que emulam outros sistemas operacionais.
- ❑ Com eles é possível rodar um ambiente Linux em um ambiente Windows ou vice-versa.
- ❑ Ideal para estudar outros SOs e aplicar testes em ambientes fechados.
- ❑ É preciso ter o ISO do SO a instalar, como em um PC real.
- ❑ Utilizaremos o Oracle VM VirtualBox.



Próxima aula

- ☐ Terminal Linux.
- ☐ Comandos CLI Linux.
- ☐ Configurações do SO.





Faculdade

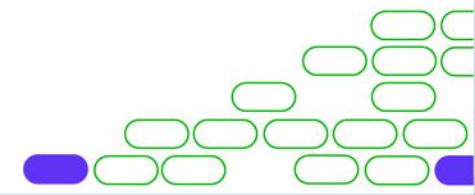


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 2. Apresentação do Ambiente

Aula 2.2. Comandos CLI no Ambiente Linux

Prof. Silas Liu

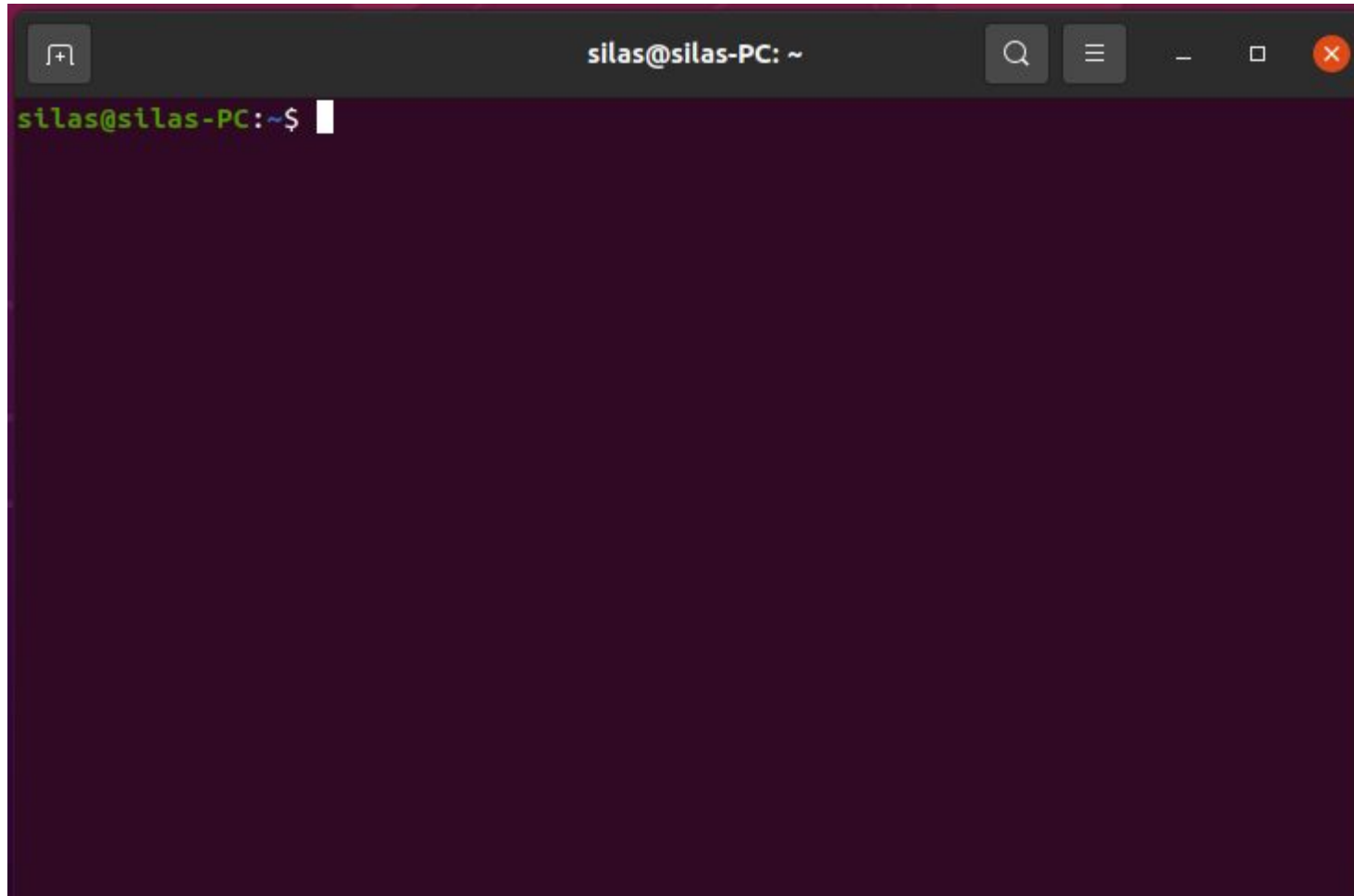


Nesta aula

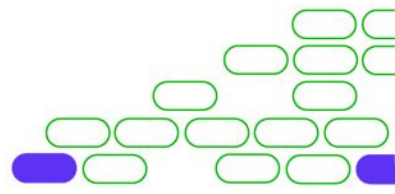
- ☐ Terminal Linux.
- ☐ Comandos CLI Linux.
- ☐ Configurações do SO.



Terminal Linux



XPe



Comandos CLI (Command Line Interface) Linux

\$ cd	mudar de diretório (para a pasta pessoal ~)
\$ cd ~	mudar para a pasta pessoal (~)
\$ cd /	mudar para o diretório raiz (/)
\$ cd dir	mudar para o diretório dir
\$ cd ..	subir um diretório
\$ ls	conteúdo do diretório atual
\$ cp orig dest	copia origem ao destino
\$ mv orig dest	move origem ao destino
\$ rm file	deleta o arquivo
\$ rm -R dir	deleta recursivo o diretório e todo seu conteúdo
\$ cat file	imprime o conteúdo do arquivo
\$ mkdir dir	cria um diretório
\$ wget url	download de arquivo do url no diretório atual
\$ nano file	abre o arquivo no editor de texto nano



Configurações do SO

\$ sudo apt-get update update do SO
\$ sudo apt-get upgrade upgrade do SO

\$ sudo apt-get install virtualbox-guest-utils instalar o bloco virtualbox
\$ sudo adduser \$USER vboxsf
\$ Sudo reboot



Conclusão

- ❑ Através do Terminal podemos entrar os comandos CLI.
- ❑ Comandos CLI representam Command Line Interface.
- ❑ É importante saber os principais comandos CLI do Linux, já que os comandos no Hadoop são semelhantes aos CLI do Linux.



Próxima aula

- ☐ Instalação do Hadoop.
- ☐ Ativando o Hadoop.
- ☐ Acessando o Hadoop.





Faculdade

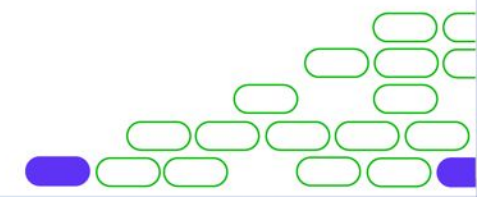


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 3. Hadoop na Prática

Aula 3.1. Instalação do Hadoop

Prof. Silas Liu



Nesta aula

- ☐ Instalação do Hadoop.
- ☐ Ativando o Hadoop.
- ☐ Acessando o Hadoop.



Instalação do Hadoop

- Instalação do Java.
- Download do Hadoop (binary).
- Instalação do Hadoop.
- Configuração do Hadoop.
- Acesso SSH.
- Ativando o Hadoop.
- Acessando o Hadoop.



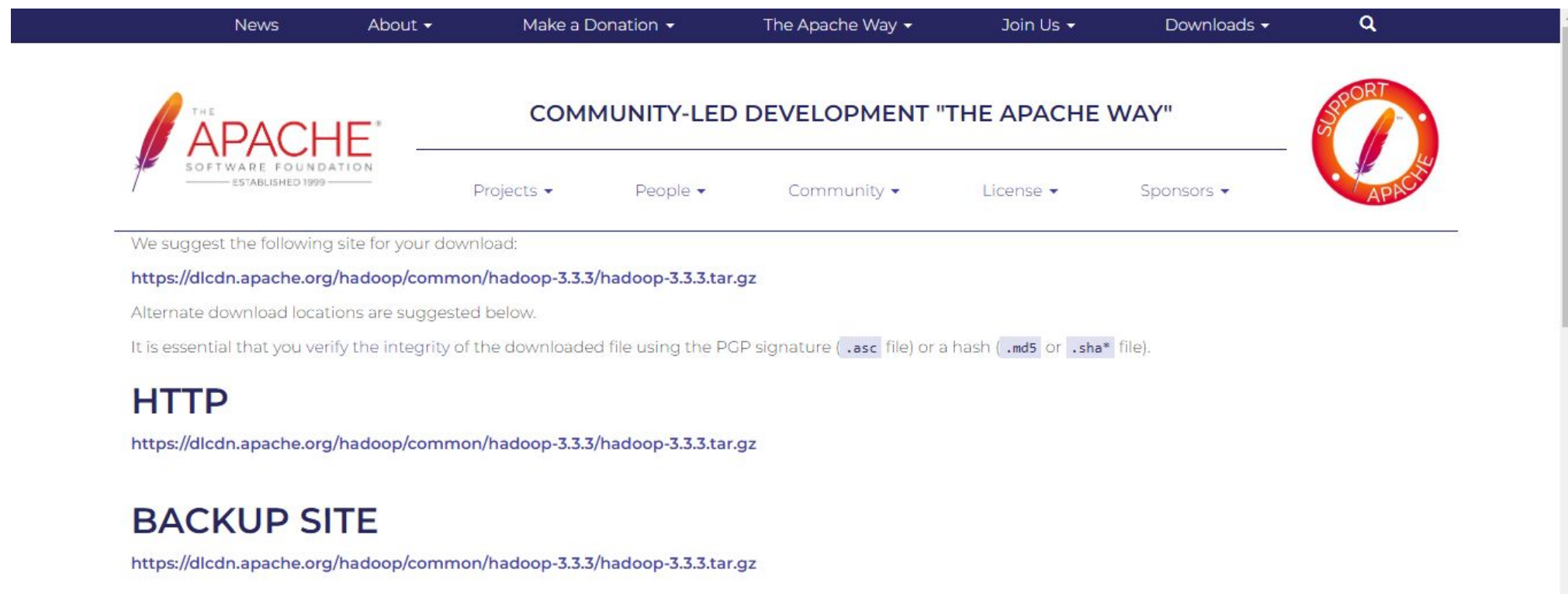
Download do Hadoop (binary)

- <https://hadoop.apache.org/>

A screenshot of the Apache Hadoop website's download page. The page has a dark navigation bar at the top with the Apache Hadoop logo and links for Download, Documentation, Community, Development, and Help. The main content area is titled 'Download' and contains a paragraph explaining that Hadoop is released as source code tarballs with corresponding binary tarballs. Below this is a table with five columns: Version, Release date, Source download, Binary download, and Release notes. The table lists three versions: 2.10.2, 3.3.3, and 3.2.3. In the 3.3.3 row, the word 'binary' in the 'Binary download' column is circled in red.



Version	Release date	Source download	Binary download	Release notes
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement
3.3.3	2022 May 17	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.2.3	2022 Mar 28	source (checksum signature)	binary (checksum signature)	Announcement

Download do Hadoop (binary)



The screenshot shows the Apache Hadoop download page. At the top is a dark blue navigation bar with links: News, About, Make a Donation, The Apache Way, Join Us, Downloads, and a search icon. Below the navigation bar is the Apache Software Foundation logo on the left and a red circular 'SUPPORT APACHE' logo on the right. In the center, the text 'COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"' is displayed. Below this text are links for Projects, People, Community, License, and Sponsors. The main content area states: 'We suggest the following site for your download: <https://dlcdn.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz>'. It then mentions 'Alternate download locations are suggested below.' and provides instructions on verifying file integrity using PGP signatures (.asc) or hashes (.md5 or .sha*). The page is divided into sections for HTTP and BACKUP SITE, both providing the same download URL.

News About ▾ Make a Donation ▾ The Apache Way ▾ Join Us ▾ Downloads ▾ 🔍

 **COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"** 

Projects ▾ People ▾ Community ▾ License ▾ Sponsors ▾

We suggest the following site for your download:

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz>

Alternate download locations are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

HTTP

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz>

BACKUP SITE

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz>

Conclusão

- ❑ Para instalar o Hadoop, baixamos a versão binária (compactada em tar.gz).
- ❑ Após fazer o download e descompactar, ainda é necessário configurarmos corretamente o sistema, bem como os diversos arquivos xml.
- ❑ A conexão é feita através de chave SSH.





Próxima aula

- ☐ Dados: Netflix TV Shows and Movies.
- ☐ Principais comandos HDFS.

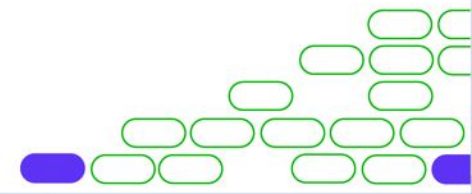


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 3. Hadoop na Prática

Aula 3.2. Manipulando o HDFS

Prof. Silas Liu




Nesta aula

- ❑ Dados: Netflix TV Shows and Movies.
- ❑ Principais comandos HDFS.



Dados: Netflix TV Shows and Movies

<https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies>



+

Create

Home

Competitions

Datasets

Code

Discussions

Courses

More

Your Work

RECENTLY VIEWED

View Active Events

Search

Netflix TV Shows and Movies

Data

Code (47)

Discussion (4)

Metadata

420

New Notebook

Download (2 MiB)

titles.csv (2.02 MiB)

Detail

Compact

Column

10 of 15 columns

About this file

Movies and TV Shows on Netflix dataset.

id	title	type	description	# releases
Movie or Show ID	Name of the Movie or Show	Movie or Show	Description of the Movie or Show	Rele or SI
5806 unique values	5752 unique values	MOVIE 65% SHOW 35%	5786 unique values	1945

Data Explorer

Version 1 (5.81 MiB)

credits.csv

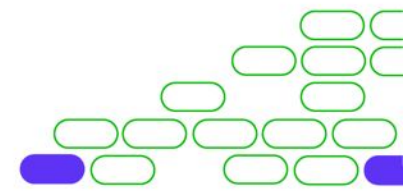
titles.csv



Dados: Netflix TV Shows and Movies

<https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies>

- Os dados constituem-se de dois arquivos:
 - credits.csv: 77213 x 5
 - titles.csv: 5976 x 15
- Vamos trabalhar primeiro com o titles.csv, manipulando ele no HDFS.



Principais comandos HDFS

\$ hdfs dfs -<COMMAND> todos os comandos começam com hdfs dfs -

O HDFS aceita a maioria dos comandos normais CLI do Linux:

\$ hdfs dfs -put local dir-hdfs copia o arquivo local para o destino hdfs

\$ hdfs dfs -get dir-hdfs copia o arquivo do hdfs para a pasta local

\$ hdfs dfs -ls / mostra o conteúdo do diretório raiz hdfs

\$ hdfs dfs -ls /dir-hdfs mostra o conteúdo do diretório hdfs

\$ hdfs dfs -help chama o arquivo de ajuda do hdfs

\$ hdfs dfs -mkdir /dir-hdfs cria diretório no hdfs

\$ hdfs dfs -cp origem destino copia o arquivo origem para destino, no hdfs

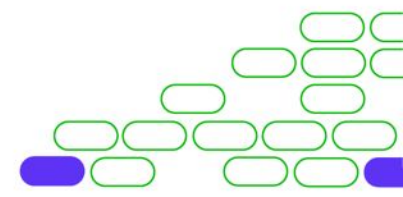
\$ hdfs dfs -cat /file imprime o arquivo

\$ hdfs dfs -head /file imprime as primeiras linhas do arquivo

\$ hdfs dfs -touchz /file cria um arquivo vazio

\$ hdfs dfs -rm /file apaga o arquivo

\$ hdfs dfs -rm -R /dir-hdfs apaga o diretório e todo seu conteúdo



Conclusão

- ❑ Kaggle é uma comunidade famosa de dados, com bastantes dados, que podemos usar para nossos estudos.
- ❑ O HDFS utiliza os mesmos comandos CLI do Linux, precedidos de 'hdfs dfs -'.
- ❑ Da mesma forma que o Linux, seu diretório raiz se localiza no /.



Próxima aula

- ☐ Estrutura do YARN.
- ☐ Inicialização dos nodos.
- ☐ Encerramento dos nodos.





Faculdade

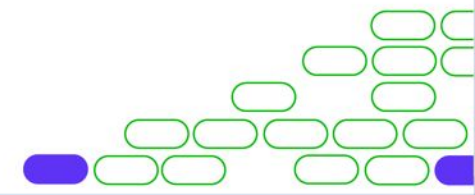


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 3. Hadoop na Prática

Aula 3.3. Comandos com o YARN

Prof. Silas Liu

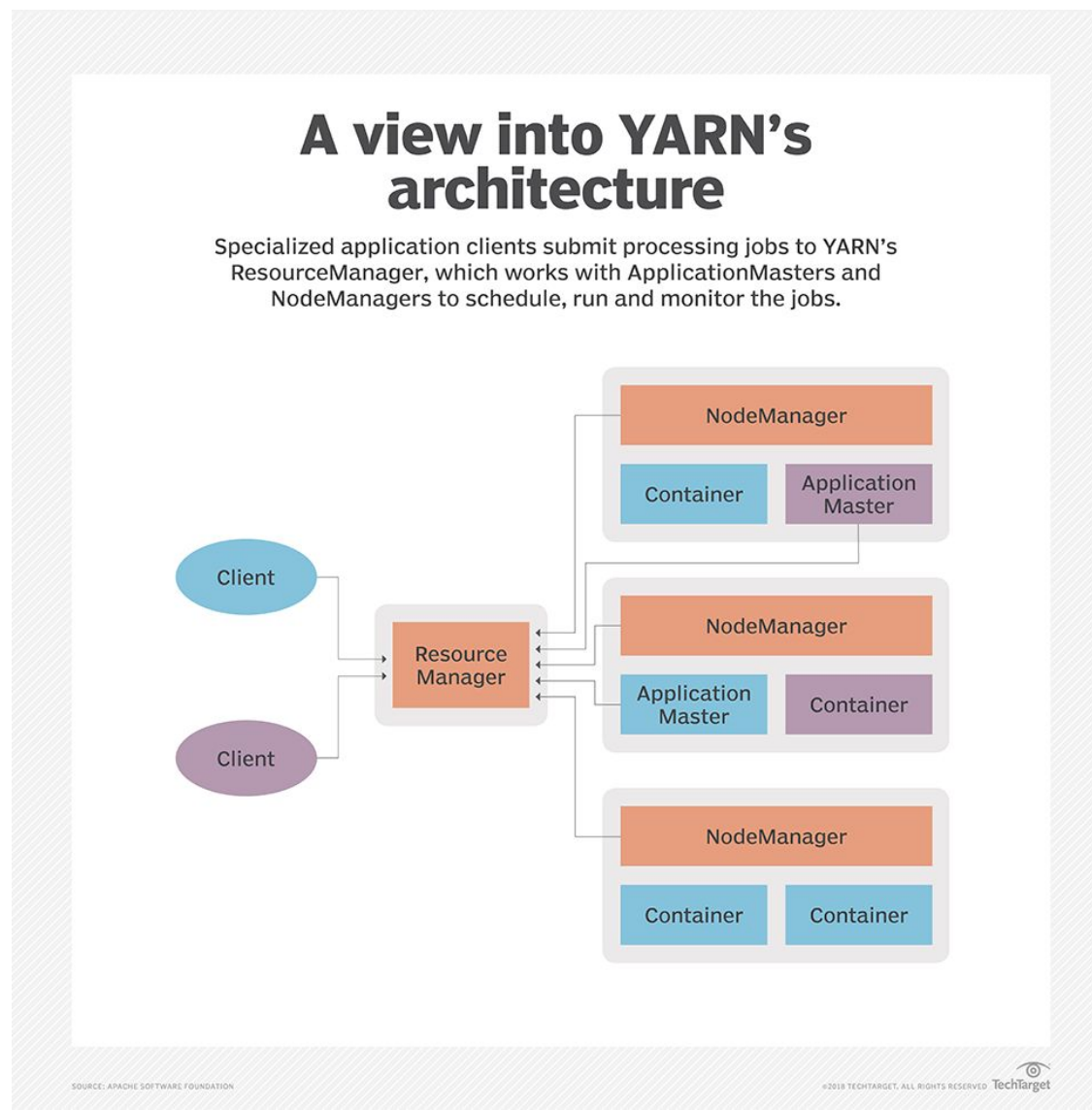


Nesta aula

- ☐ Estrutura do YARN.
- ☐ Inicialização dos nodos.
- ☐ Encerramento dos nodos.



Estrutura do YARN



Inicialização dos nodos

O YARN é quem controla os nodos:

```
$ cd ~/hadoop/sbin
```

```
$ ./start-dfs.sh
```

```
$ ./start-yarn.sh
```

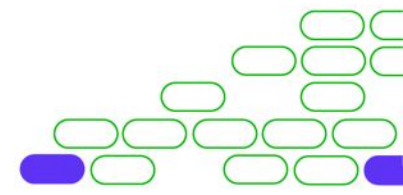
```
$ mapred --daemon start historyserver
```

Pode-se inicializar tudo por um comando só, mas não é recomendado para produção:

```
$ ./start-all.sh
```

Para verificar:

```
$ jps
```



Encerramento dos nodos

Os comandos para terminar os nodos são análogos:

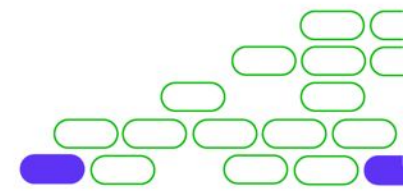
```
$ ./stop-dfs.sh
```

```
$ ./stop-yarn.sh
```

```
$ mapred --daemon stop historyserver
```

Pode-se terminar tudo por um comando só, mas não é recomendado para produção:

```
$ ./stop-all.sh
```



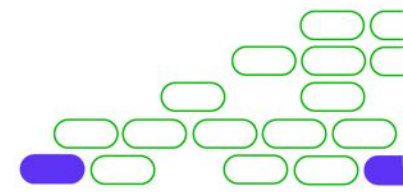
Acompanhamento de Logs

Pelo YARN é possível observar os logs de aplicativos, durante sua execução (cada aplicação tem um número, aqui representados por X):

```
$ yarn logs -applicationId application_XXXXXXXXXXXXX_XXXX
```

Ou salvar em um arquivo .log:

```
$ yarn logs -applicationId application_XXXXXXXXXXXXX_XXXX > processo.log
```



Conclusão

- ❑ O YARN é o responsável por gerenciar os recursos no cluster.
- ❑ Cada nó worker possui um Node Manager.
- ❑ O nó master possui um Resource Manager, para controlar todos.



Próxima aula

- ☐ Funcionamento do MapReduce.
- ☐ Hadoop Streaming.
- ☐ Mapper.
- ☐ Reducer.





Faculdade

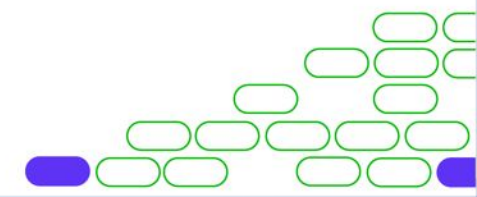


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 3. Hadoop na Prática

Aula 3.4. Aplicando o Map Reduce

Prof. Silas Liu



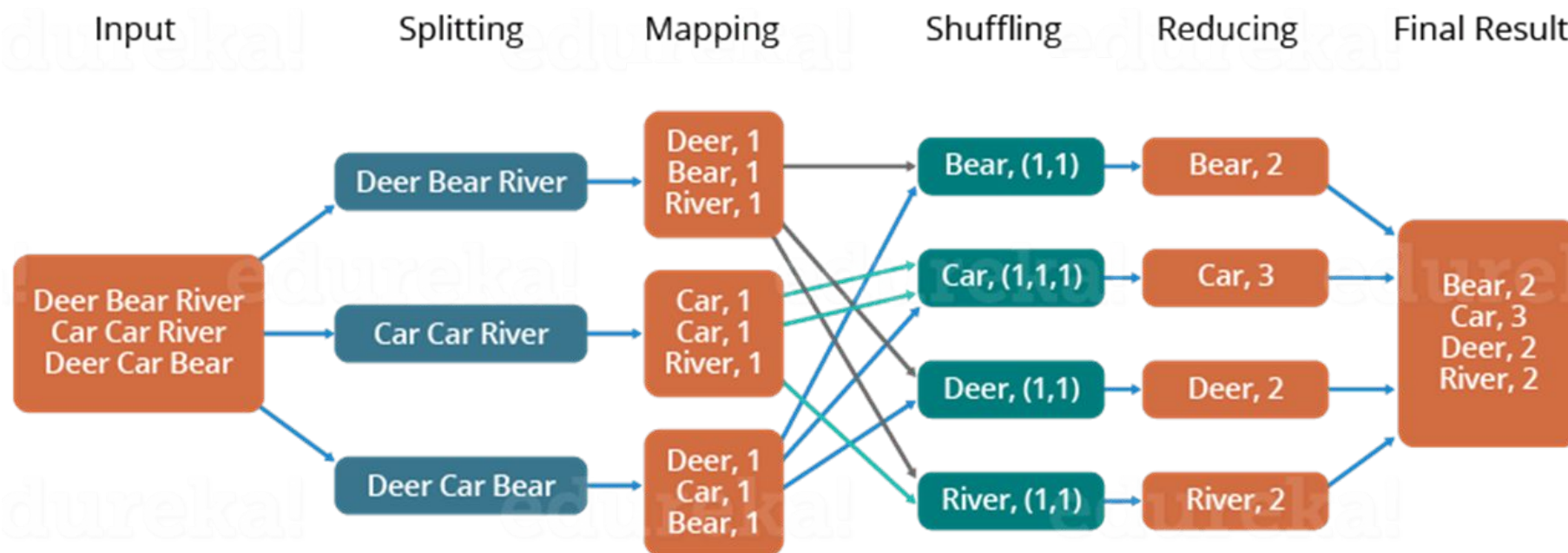
Nesta aula

- ☐ Funcionamento do MapReduce.
- ☐ Hadoop Streaming.
- ☐ Mapper.
- ☐ Reducer.



Funcionamento do MapReduce

The Overall MapReduce Word Count Process

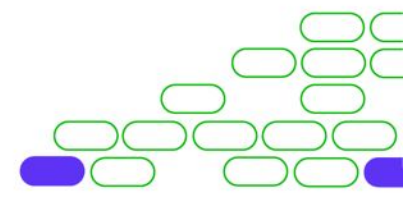


Hadoop Streaming

O Hadoop Streaming nos dá a flexibilidade de rodar qualquer script como mapper e reducer em nossos processos. A liberdade é que podemos empregar qualquer linguagem de programação:

```
$ mapred streaming -input <INPUT_DIR>  
    -output <OUTPUT_DIR>  
    -mapper <MAPPER.SCRIPT>  
    -reducer <REDUCER.SCRIPT>
```

Atenção, o diretório de saída (output) não pode existir!



Mapper

Exemplo de mapper contador de palavras em Python: (mapper.py)

```
1  #!/usr/bin/env python3
2
3  import sys
4
5  try:
6      for line in sys.stdin:
7          words = line.split()
8          for word in words:
9              print('{0}\t{1}'.format(word, 1))
10 except Exception as e:
11     raise(e)
```

Reducer

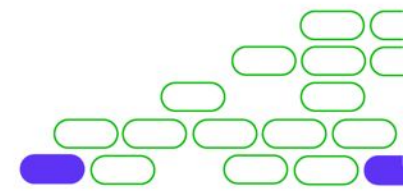
Reducer contador em Python: (reducer.py)

```
1  #!/usr/bin/env python3
2
3  import sys
4
5  curr_key = None
6  curr_count = 0
7
8  try:
9      for line in sys.stdin:
10         key, count = line.split("\t", 1)
11         count = int(count)
12         if key == curr_key:
13             curr_count += count
14         else:
15             if curr_key:
16                 print('{0}\t{1}'.format(curr_key, curr_count))
17                 curr_count = count
18                 curr_key = key
19             if curr_key == key:
20                 print('{0}\t{1}'.format(curr_key, curr_count))
21 except Exception as e:
22     raise(e)
```

Mapper

Exemplo de mapper contador (para tabela) em Python: (mapper_counter.py)

```
1  #!/usr/bin/env python3
2
3  import sys
4
5  try:
6      for line in sys.stdin:
7          data = line.split(",")
8          print('{0}\t{1}'.format(data[4], 1))
9  except Exception as e:
10     raise(e)
11
```



Conclusão

- Hadoop Streaming possibilita que os scripts de Mapper e Reducer sejam escritos em qualquer linguagem de programação.
- O processo de Mapper e Reducer seguem a lógica de um pipe de processos.
- Os procedimentos de Mapper e Reducer são sempre referentes a um campo chave.
- É papel do desenvolvedor escrever os arquivos Mapper e Reducer, mas estes acabam se tornando trabalhosos.



Próxima aula

- ☐ O que é o Hive?
- ☐ Pontos importantes HQL.





Faculdade

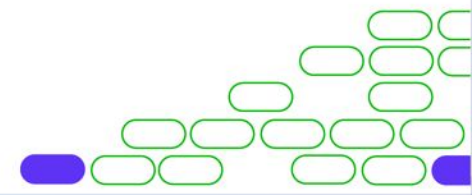


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 4. Framework Hive

Aula 4.1. Introdução ao Hive

Prof. Silas Liu



Nesta aula

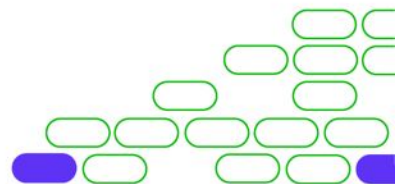
- ❑ O que é o Hive?
- ❑ Pontos importantes HQL.



O que é o Hive?



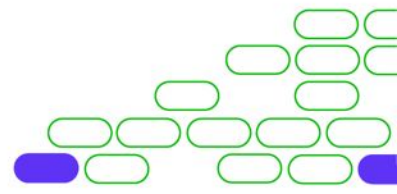
- Hive é um framework construído em cima do Hadoop, para abstrair e facilitar o uso do MapReduce.
- Foi desenvolvido em 2007 pelo Facebook, mas em 2008 se tornou um projeto Open Source.
- Os comandos empregam a linguagem HQL (Hive Query Language), que é bem semelhante ao SQL.



O que é o Hive?



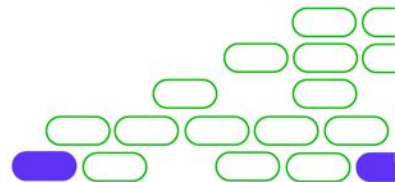
- Os comandos HQL são traduzidos em jobs MapReduce (ou outros como Spark, segundo configuração).
- As consultas e manipulação dos dados seguem o padrão de banco de dados relacional.
- O usuário não precisa mais programar as funções MapReduce, bastando apenas usar os comandos HQL.



Pontos importantes HQL



- Não se diferencia maiúscula de minúscula nos comandos, mas nomes de variáveis e classes Java diferenciam.
- Os comandos devem ser terminados com ‘;’.
- Atenção ao uso de espaço, vários comandos separam comandos e argumentos pelo espaço.
- Utilização de comandos usuais de SQL, tais como: CREATE, DROP, SELECT, WHERE.



Conclusão

- ❑ Hive é um framework para facilitar o uso do MapReduce.
- ❑ Ele utiliza a linguagem HQL, semelhante ao SQL.
- ❑ Abstrai a dificuldade de programar os algoritmos Mapper e Reducer.



Próxima aula

- ☐ Metastore.
- ☐ Modelagem dos Dados.
- ☐ Tipos de Tabelas.





Faculdade

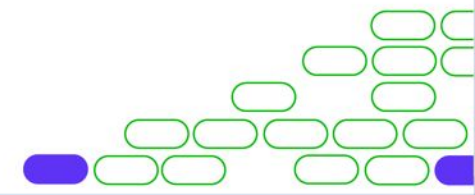


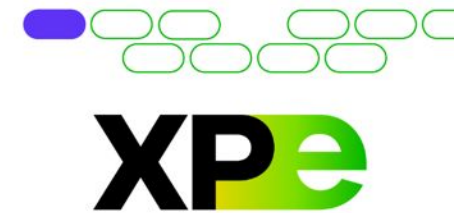
Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 4. Framework Hive

Aula 4.2. Metastore e modelagem dos dados

Prof. Silas Liu





Nesta aula

- ☐ Metastore.
- ☐ Modelagem dos Dados.
- ☐ Tipos de Tabelas.



Metastore

- O Hive trabalha com Metastore: a estrutura de banco de dados que será utilizada. Esse Metastore segue a estrutura de tabelas relacionais e pode ser configurada: MySQL, PostgreSQL, Derby, etc.
- Os metadados armazenam as informações das tabelas.
- Dentro do HDFS, os bancos de dados são armazenados como diretórios.
Caminho padrão: /user/hive/warehouse/<database>.
- Dentro de cada banco de dados (database), ficam as tabelas (tables):
Caminho padrão: /user/hive/warehouse/<database>/<table>.



Modelagem dos Dados

Hive aceita uma grande quantidade de tipos de dados:

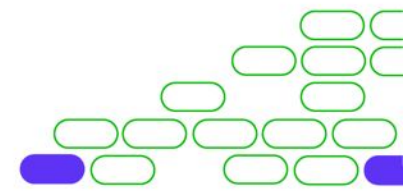
- Numéricos:
 - Tinyint.
 - Smallint.
 - Int / Integer.
 - Bigint.
 - Float.



Modelagem dos Dados

Hive aceita uma grande quantidade de tipos de dados:

- Datas:
 - Timestamp.
 - Date.
 - Interval.
- Misc:
 - Boolean.
 - Binary



Tipos de Tabelas

- **EXTERNAL:**
 - A forma usual de se empregar tabelas.
 - A tabela é um metadado dos dados armazenados no HDFS.
 - Se a tabela for dropada, os dados continuam.
- **MANAGED:**
 - Útil em ocasiões especiais, como tabelas temporárias.
 - A tabela corresponde aos próprios dados no HDFS.
 - Se a tabela for dropada, os dados são apagados.



Conclusão

- ❑ Os dados são armazenados como tabelas relacionais, no HDFS.
- ❑ Pode-se acessar os bancos de dados e as tabelas, como diretórios.
- ❑ Para cada coluna deve-se especificar o tipo de variável a armazenar.
- ❑ Hive oferece uma gama de tipos de variáveis para otimizar o armazenamento.
- ❑ Usualmente empregamos tabelas do tipo EXTERNAL.



Próxima aula

- ☐ Instalação do Hive.
- ☐ Instalação do Derby.
- ☐ Inicializar o Derby.
- ☐ Checar versão do Hive.

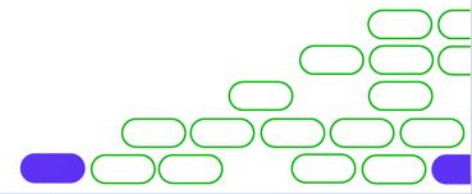


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 4. Framework Hive

Aula 4.3. Instalação do Hive

Prof. Silas Liu



Nesta aula

- ☐ Instalação do Hive.
- ☐ Instalação do Derby.
- ☐ Inicializar o Derby.
- ☐ Checar versão do Hive.



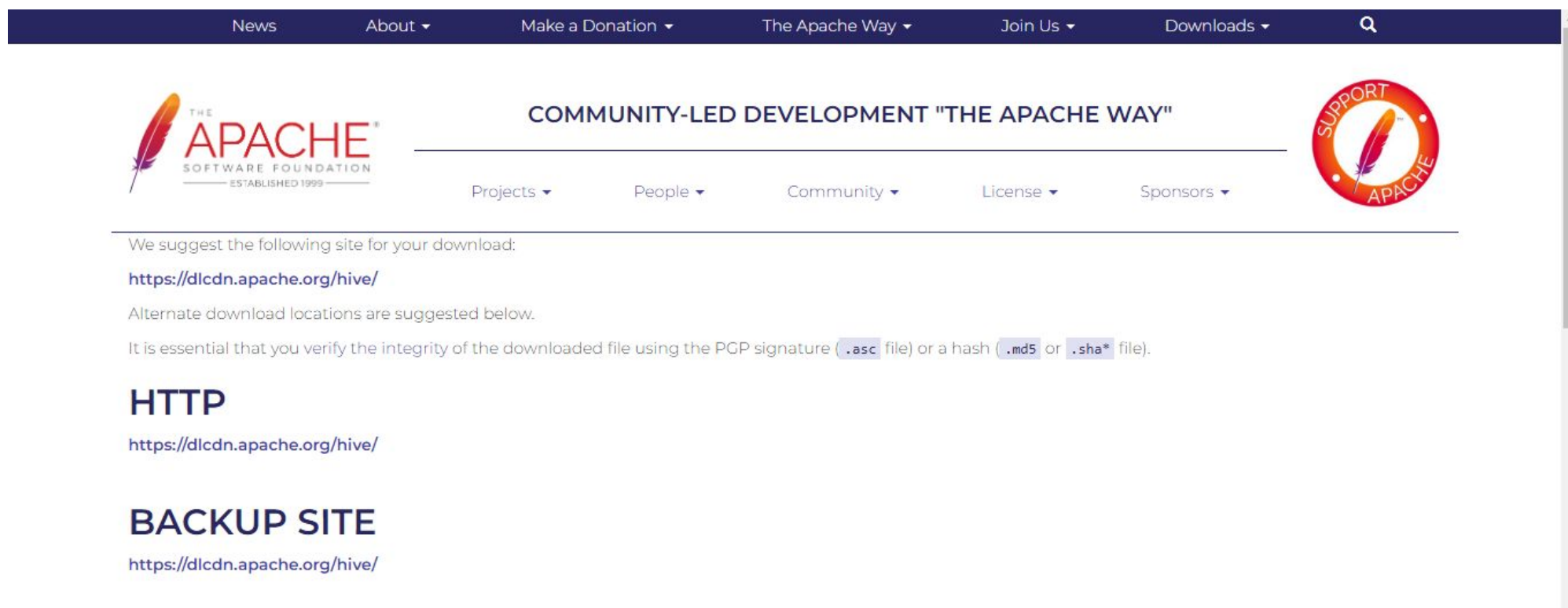
Instalação do Hive

- Download do Hive (binary).
- Download do Derby (binary).
- Instalação do Hive.
- Instalação do Derby.
- Inicializar o Derby.
- Checar versão do Hive.



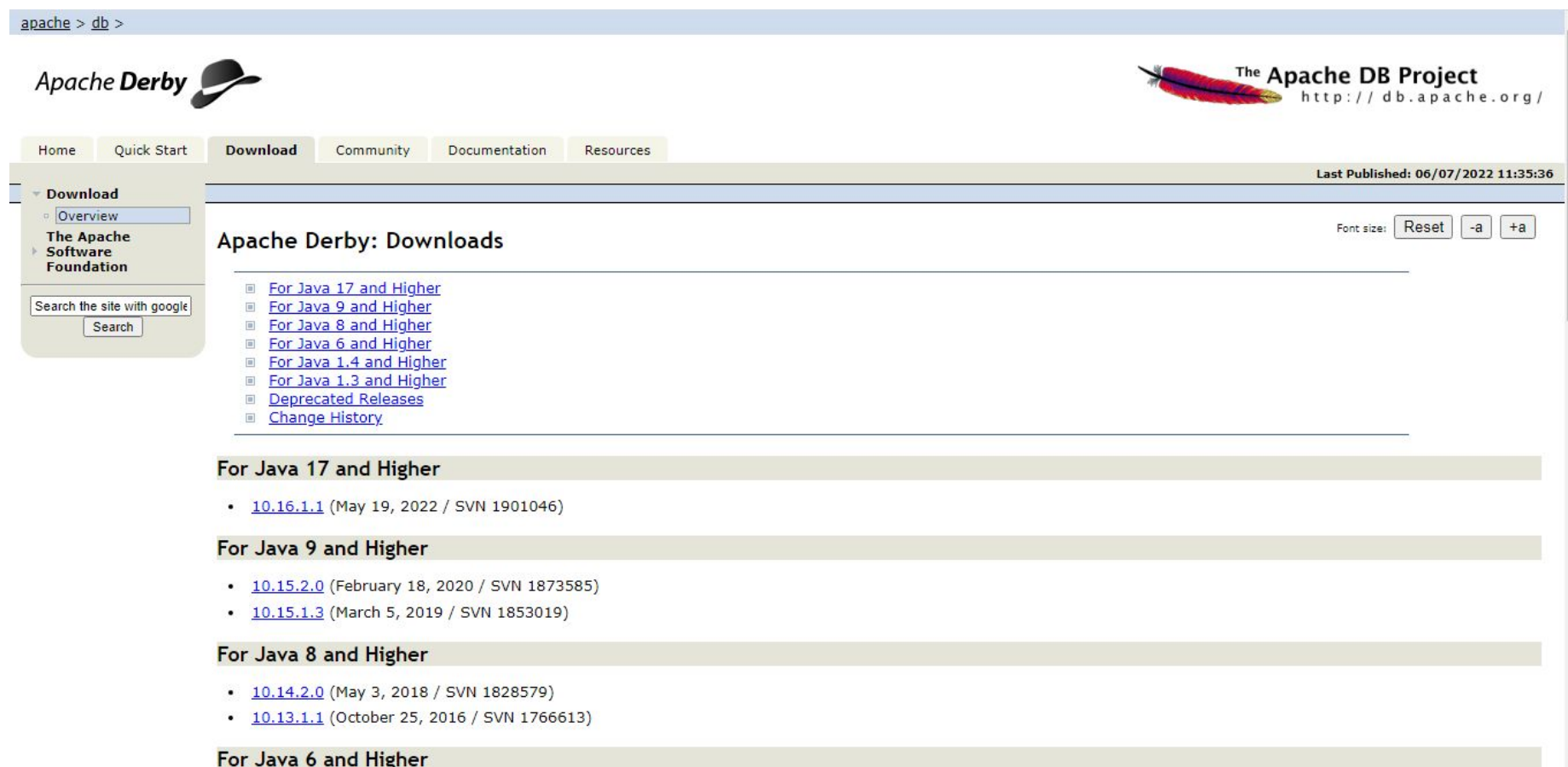
Download do Hive (binary)

- <https://hive.apache.org/>

A screenshot of the Apache Software Foundation website. The top navigation bar is dark blue with links: News, About, Make a Donation, The Apache Way, Join Us, Downloads, and a search icon. Below the navigation bar is a white section with the Apache logo on the left, the text 'COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"' in the center, and a 'SUPPORT APACHE' logo on the right. Under the center text are links: Projects, People, Community, License, and Sponsors. Below this is a section titled 'We suggest the following site for your download:' with the URL 'https://dlcdn.apache.org/hive/'. It also mentions 'Alternate download locations are suggested below.' and provides instructions on how to verify the integrity of the downloaded file using PGP signatures or hashes. Finally, it lists 'HTTP' and 'BACKUP SITE' with the same URL 'https://dlcdn.apache.org/hive/'.

Download do Derby (binary)

- <https://db.apache.org/derby/>

The screenshot shows the Apache Derby website's download page. At the top, there's a navigation bar with links to Home, Quick Start, Download (selected), Community, Documentation, and Resources. Below this, a sidebar on the left contains a 'Download' section with a link to 'Overview' and a search bar. The main content area is titled 'Apache Derby: Downloads' and lists several download links for different Java versions: 'For Java 17 and Higher', 'For Java 9 and Higher', 'For Java 8 and Higher', 'For Java 6 and Higher', 'For Java 1.4 and Higher', 'For Java 1.3 and Higher', 'Deprecated Releases', and 'Change History'. Below these links, there are three sections: 'For Java 17 and Higher' with a link to '10.16.1.1' (May 19, 2022 / SVN 1901046), 'For Java 9 and Higher' with links to '10.15.2.0' (February 18, 2020 / SVN 1873585) and '10.15.1.3' (March 5, 2019 / SVN 1853019), and 'For Java 8 and Higher' with links to '10.14.2.0' (May 3, 2018 / SVN 1828579) and '10.13.1.1' (October 25, 2016 / SVN 1766613). The 'For Java 6 and Higher' section is partially visible at the bottom. The page also includes a 'Last Published' timestamp of 06/07/2022 11:35:36 and font size controls.

Conclusão

- ❑ Para instalar o Hive, utilizamos o binário dele e do metastore a se utilizar, no caso o Derby.
- ❑ Após a instalação de ambos é necessário inicializar uma vez o metastore, para criação do schema dos bancos de dados.



Próxima aula

- ☐ Comandos HQL.
- ☐ Descrição.
- ☐ SELECT.
- ☐ JOIN.





Faculdade

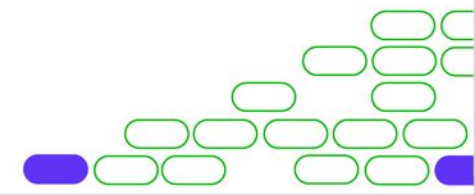


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 4. Framework Hive

Aula 4.4. Hive na prática

Prof. Silas Liu



Nesta aula

- ☐ Comandos HQL.
- ☐ Descrição.
- ☐ SELECT.
- ☐ JOIN.



Comandos básicos HQL

- \$ hive inicializa o Hive, a partir daí usamos comandos HQL
- > SHOW DATABASES; mostra os bancos de dados disponíveis
 - > CREATE DATABASE teste_db; cria banco de dados teste_db
 - > USE teste_db; seleciona um banco de dados
 - > SET hive.cli.print.current.db=true; mostrar o banco de dados ativo
 - > SHOW TABLES; mostra as tabelas disponíveis
 - > CREATE TABLE teste_db.alunos_tb
 (id INT COMMENT 'código id do aluno',
 nome STRING COMMENT 'nome do aluno',
 nota FLOAT COMMENT 'nota do aluno'
) COMMENT 'tabela dos alunos'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';'; cria schema da tabela



Comandos básicos HQL

- \$ `hdfs dfs -ls /user/hive/warehouse` mostra o conteúdo do HDFS
- \$ `hdfs dfs -ls /user/hive/warehouse/teste_db` mostra o conteúdo do HDFS

- > `DESC alunos_tb;` descrição das colunas e tipos
- > `INSERT INTO TABLE alunos_tb VALUES(9,'Pedro',8.8);` insere dados na tabela
- > `SELECT * FROM alunos_tb;` mostra toda a tabela
- > `SHOW CREATE TABLE alunos_tb;` mostra os comandos utilizados pelo Hive



Comandos básicos HQL

- > `CREATE DATABASE imdb_db;` cria banco de dados `imdb_db`
- > `USE imdb_db;` seleciona um banco de dados
- > `CREATE EXTERNAL TABLE credits_tb`
(`person_id INT`,
`id STRING`,
`name STRING`,
`character STRING`,
`role STRING`)
`ROW FORMAT DELIMITED FIELDS TERMINATED BY ','`
`LOCATION '/user/hive/warehouse/imdb_db'`
`TBLPROPERTIES ("skip.header.line.count"="1");` cria schema da tabela
- > `LOAD DATA INPATH '/credits.csv' OVERWRITE INTO TABLE credits_tb;`
carrega o arquivo na tabela
- > `SELECT * FROM credits_tb LIMIT 5;` mostra 5 linhas da tabela
- > `SELECT COUNT(*) FROM credits_tb;` conta número de linhas

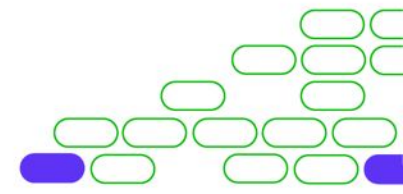


Comandos HQL - Descrição

- > `CREATE EXTERNAL TABLE titles_tb`
(id STRING,title STRING,type STRING,description STRING,release_year INT,
age_certification STRING,runtime INT,genres STRING,production_countries STRING,
seasons FLOAT,imdb_id STRING,imdb_score FLOAT,imdb_votes FLOAT,
tmdb_popularity FLOAT,tmdb_score FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'; cria schema da tabela
- > `LOAD DATA INPATH '/titles_fixed.csv' OVERWRITE INTO TABLE titles_tb;`
carrega o arquivo na tabela

Comandos de descrição:

- > `DESC titles_tb;`
- > `DESCRIBE titles_tb;`
- > `DESCRIBE FORMATTED titles_tb;`



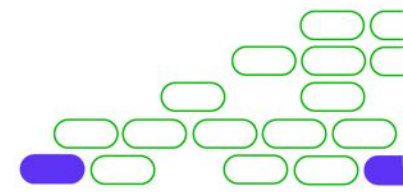
Comandos HQL - SELECT

Com o comando SELECT, em conjunto com outros comandos, podemos criar filtros avançados:

- > `SELECT * FROM titles_tb LIMIT 5;` mostra 5 linhas da tabela
- > `SELECT COUNT(*) FROM titles_tb;` conta número de linhas
- > `SELECT COUNT(*) FROM titles_tb WHERE release_year = '2022';`
- > `SELECT release_year, COUNT(1) AS total FROM titles_tb GROUP BY release_year ORDER BY total DESC;`

Podemos também calcular média, máximo, mínimo etc.:

- > `SELECT AVG(imdb_score), MAX(imdb_score), MIN(imdb_score) FROM titles_tb;`



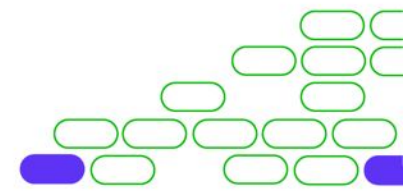
Comandos HQL - JOIN

Em Hive existem três JOINS disponíveis:

- JOIN / FULL OUTER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN

Exemplo de um JOIN entre as duas tabelas usando o Id de ambos:

```
> SELECT tab01.name,tab01.character,tab02.title  
FROM credits_tb tab01 JOIN titles_tb tab02  
ON (tab01.id = tab02.id)  
LIMIT 10;
```



Conclusão

- ❑ Hive abstrai toda a dificuldade de se programar os algoritmos Mapper e Reducer, ao trabalhar com o HDFS.
- ❑ Os comandos são realizados direto em HQL, linguagem semelhante ao SQL.
- ❑ Ganhamos liberdade de criar expressões e manipulações mais complexas.





Próxima aula

- ☐ Formatos de Arquivos.
- ☐ Aplicando no HQL.

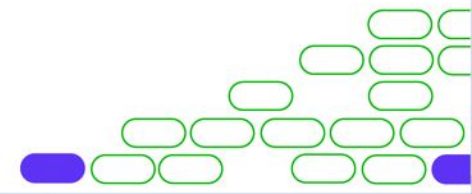


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 4. Framework Hive

Aula 4.5. Formatos de arquivos

Prof. Silas Liu



Nesta aula

- ☐ Formatos de Arquivos.
- ☐ Aplicando no HQL.



Formatos de Arquivos

- Na criação da tabela, é possível determinar o formato de arquivo que ele terá.
- O formato pode melhorar a compressão, velocidade de acesso ou transferência de dados, de acordo com a aplicação.
- Se tornam mais necessários com o crescimento do tamanho dos dados.
- Os algoritmos que realizam as compressões para esses formatos se chama codec. Ex: LZ4 e Snappy.



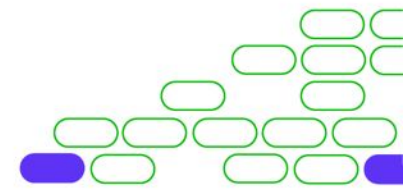
Formatos de Arquivos



Apache
orc™



- Parquet:
 - Formato colunar desenvolvido pela Cloudera e Twitter.
 - Reduz o espaço de armazenamento.
 - Aumenta a performance.
 - Mais eficiente com inserção de dados em batch (grande volume).
 - Ideal para dados colunares.
- Avro:
 - Armazenamento em formato de linhas.
 - Salva os dados em formato JSON.
 - Os dados são salvos em binário, otimizando a compactação.
 - Funciona com dados semiestruturados;.
- Apache ORC:
 - Otimizado para armazenamento eficiente de dado.
 - Sua estrutura armazena diversas linhas em colunas.



Formatos de Arquivos

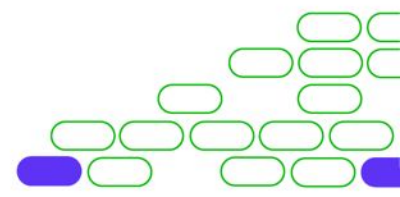


Apache
orc™



AVRO x PARQUET:

- AVRO é baseado em armazenamento por linha, enquanto o PARQUET é baseado em armazenamento por coluna.
- PARQUET é melhor para busca analítica (leitura) dos dados.
- AVRO é melhor para escrita dos dados.
- AVRO possui mais liberdade de schema, já que pode ser modificado livremente. PARQUET possui uma estrutura fechada e só possibilita append.
- PARQUET é ideal para busca por um subset de colunas em tabelas com múltiplas colunas. AVRO é ideal para operações gerais como busca por todas as colunas.



Formatos de Arquivos



Apache
orc™



ORC x PARQUET:

- PARQUET é melhor otimizado para armazenar dados aninhados.
- ORC é melhor otimizado para Spark Filter / predicate pushdown (uma espécie de busca otimizada, mais eficiente que o normal).
- ORC é mais eficiente para compressão dos dados.

Aplicando no HQL

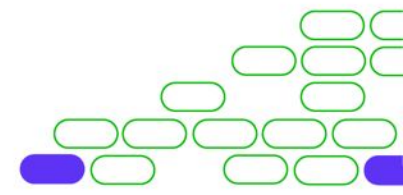
Na criação do schema da tabela, adiciona-se o formato em que se quer salvar.

Podemos também escolher o codec a utilizar:

- > CREATE EXTERNAL TABLE credits_parquet_tb
(person_id INT,id STRING,name STRING,character STRING,role STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS PARQUET
TBLPROPERTIES ("skip.header.line.count"="1", "parquet.compression"="SNAPPY");
- > LOAD DATA INPATH '/credits.csv' OVERWRITE INTO TABLE credits_parquet_tb;

Visualizar o arquivo parquet:

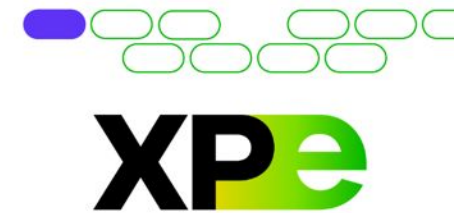
- \$ hdfs dfs -ls /user/hive/warehouse/imdb_db/credits_parquet_tb
- \$ hdfs dfs -get /user/hive/warehouse/imdb_db/credits_parquet_tb/000000_0
- \$ parquet-tools schema 000000_0



Conclusão

- Existem diferentes formatos de arquivos, obtidos por compressão com codecs.
- Cada formato é otimizado para diferentes aplicações e formatos de armazenamento de dados.





Próxima aula

- ☐ Terminal Linux.
- ☐ Comandos CLI Linux.





Faculdade

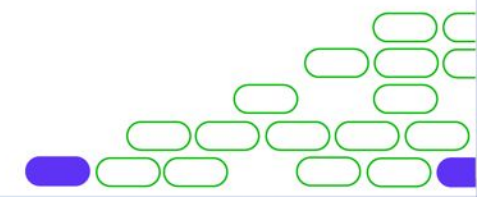


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 4. Framework Hive

Aula 4.6. Particionamento

Prof. Silas Liu



Nesta aula

- ❑ Formatos de arquivos.
- ❑ Aplicando no HQL.



Particionamento

- Podemos aplicar particionamento nas tabelas, para melhorar a organização dos dados.
- Os particionamentos funcionam como diretórios dentro das tabelas.
- Deve-se tomar cuidado para não particionar demais. Isso sobrecarregará o namenode do cluster, que precisará acessar um número grande de diretórios toda vez.



Aplicando no HQL

Criação de tabela com partição:

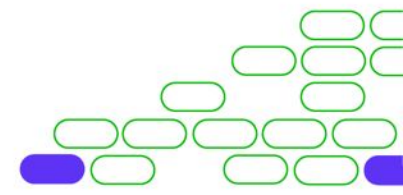
```
> CREATE EXTERNAL TABLE credits_part_tb  
  (person_id INT,id STRING,name STRING,character STRING,role STRING)  
  ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
  PARTITIONED BY (data STRING)  
  TBLPROPERTIES ("skip.header.line.count"="1");
```

Inserindo valor na tabela com partição:

```
> INSERT INTO TABLE credits_part_tb  
  PARTITION (data='01-01-2022')  
  VALUES(1,'A','Andre','Bob','Ator');
```

Adicionando um arquivo à tabela com partição:

```
> LOAD DATA INPATH '/credits.csv' OVERWRITE INTO TABLE credits_part_tb  
  PARTITION (data='01-02-2022');
```



Aplicando no HQL

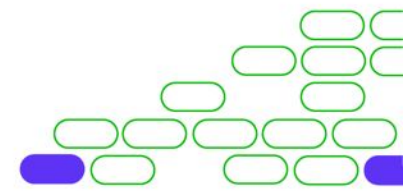
Verificar as partições no HDFS:

```
$ hdfs dfs -ls /user/hive/warehouse/imd_db
```

```
$ hdfs dfs -ls /user/hive/warehouse/imd_db/credits_part_tb
```

```
$ hdfs dfs -ls /user/hive/warehouse/imd_db/credits_part_tb/data=01-01-2022
```

```
$ hdfs dfs -ls /user/hive/warehouse/imd_db/credits_part_tb/data=01-02-2022
```



Conclusão

- ❑ Pode-se utilizar o recurso de partições para melhor organizar os dados.
- ❑ As partições funcionam como uma coluna extra, mas que funciona como diretório para separar os dados.
- ❑ Não se deve exagerar e criar partições demais.



Próxima aula

- ☐ Introdução ao Impala.
- ☐ Diferenças entre Impala e Hive.
- ☐ Comandos Impala.





Faculdade

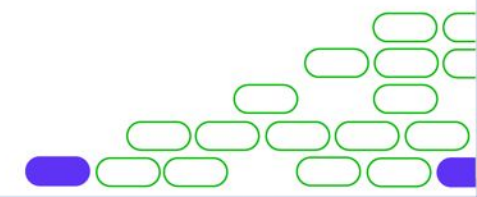


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 5. Framework Impala

Aula 5.1. Framework Impala

Prof. Silas Liu



Nesta aula

- ☐ Introdução ao Impala.
- ☐ Diferenças entre Impala e Hive.
- ☐ Comandos Impala.



Introdução ao Impala

- Descrição: engine MPP (massive parallel processing) open source, que atua no HDFS.
- Também emprega a linguagem SQL.
- Motor de SQL de alta performance.
- Inspirado no projeto Dremel do Google.
- Atua diretamente nos daemons, sem a necessidade de implementar o MapReduce.
- Ideal para processos em tempo real ou baixa latência.
- Muitas vezes Hive e Impala são comparados por ambos empregarem comandos SQL.



Diferenças entre Impala e Hive

- Características Hive:
 - Hive salva os estágios intermediários dos processos em disco. Com isso, ele se torna mais confiável e tolerante a falhas.
 - Recomendado para operações que exijam integridade dos dados e operações.
- Características Impala:
 - Impala é o mais recomendado para processamento em tempo real e consultas ad-hoc (é entre 5x a 50x mais rápido que MapReduce).
 - Impala consome e exige muita memória. A máquina precisa ter pelo menos 4 GB de RAM, mas o recomendado é pelo menos 8 GB de RAM para dados extremamente grandes, pois podem travar o sistema.
 - No caso de travamento, perde-se toda a operação e resultados.



Comandos Impala

- Impala roda com comandos SQL.
- Todos os comandos vistos com Hive se aplicam ao IMPALA, tais como:
 - CREATE.
 - DROP.
 - SELECT.
 - WHERE.
 - COUNT.
 - JOIN.



Conclusão

- ❑ Impala é um framework alternativo, que também utiliza comandos SQL para lidar com o HDFS.
- ❑ Impala é bem mais rápido que o MapReduce, sendo ideal para processamentos em tempo real.
- ❑ Entretanto, o Impala não oferece garantia de integridade e segurança das operações.
- ❑ Impala utiliza muita memória das máquinas.



Próxima aula

- ☐ Plataformas nuvem.
- ☐ Serviços Hadoop.
- ☐ Máquinas para Clusters.





Faculdade

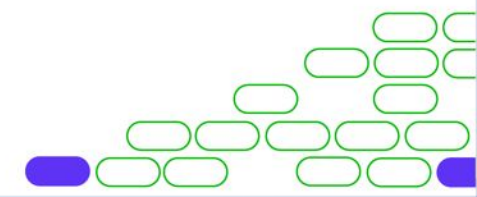


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 6. Introdução às Plataformas Nuvem

Aula 6.1. Apresentação das plataformas nuvem

Prof. Silas Liu



Nesta aula

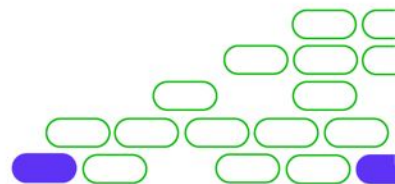
- ☐ Plataformas nuvem.
- ☐ Serviços Hadoop.
- ☐ Máquinas para Clusters.



Plataformas nuvem



Google Cloud



Serviços Hadoop

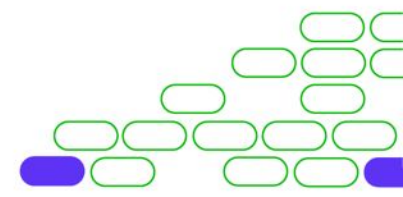


Hadoop:

Elastic MapReduce

Dataproc

HDInsight

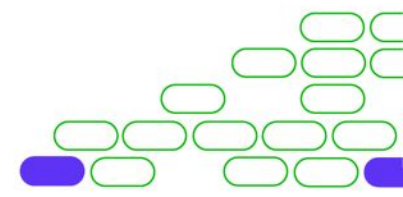


Máquinas para Clusters



Hadoop: Elastic MapReduce Dataproc HDInsight

Máquinas: EC2 Compute Engine Máquina Virtual Azure
(Elastic Compute Cloud)



Conclusão

- ❑ Serviços de nuvem oferecem testes gratuitos.
- ❑ Pode-se testar sistemas Hadoop com vários nodos.
- ❑ As plataformas nuvem oferecem serviço Hadoop, mas com nomes diferentes.
- ❑ É preciso realizar o mesmo procedimento da máquina virtual individual: instalar o Hadoop, configurar o Hadoop e configurar a conexão SSH.



Próxima aula

- ☐ Plataforma Databricks.
- ☐ Criação de conta gratuita.
- ☐ Databricks Community Edition.
- ☐ Notebooks Jupyter / Databricks.

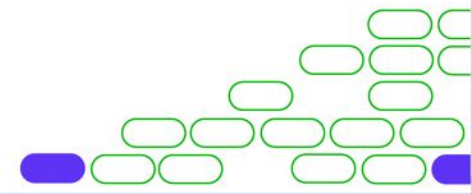


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 6. Introdução às Plataformas Nuvem

Aula 6.2. Plataforma Databricks

Prof. Silas Liu



Nesta aula

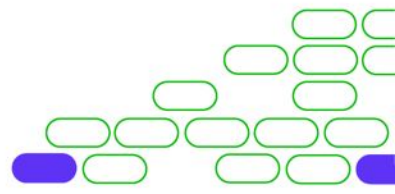
- ☐ Plataforma Databricks.
- ☐ Criação de conta gratuita.
- ☐ Databricks Community Edition.
- ☐ Notebooks Jupyter / Databricks.



Plataforma Databricks

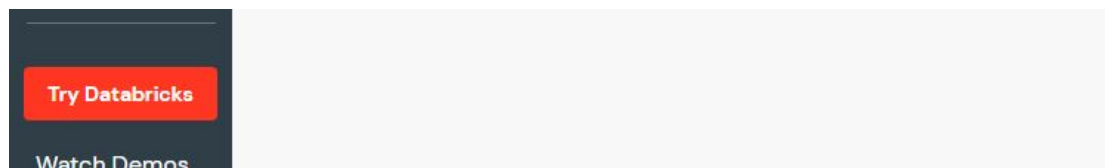


- Databricks é uma plataforma que engloba arquitetura Data Lake, com processamento Spark e notebooks próprios, semelhante ao Jupyter.
- É uma plataforma nuvem especializada em big data e sistemas distribuídos.
- Ele oferece o Databricks Community Edition: uma versão gratuita para utilizar seus recursos mais básicos.
- É ótimo para estudar o Spark (que iremos estudar no próximo capítulo), uma vez que ele abstrai a parte de instalação e configuração.



Criação de conta gratuita

- <https://databricks.com/>.



By Clicking "Get Started For Free", you agree to the [Privacy Policy](#).

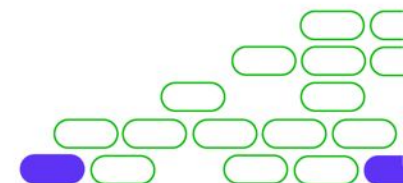
GET STARTED FOR FREE

Don't have a cloud account?

Community Edition is a limited Databricks environment for personal use and training.

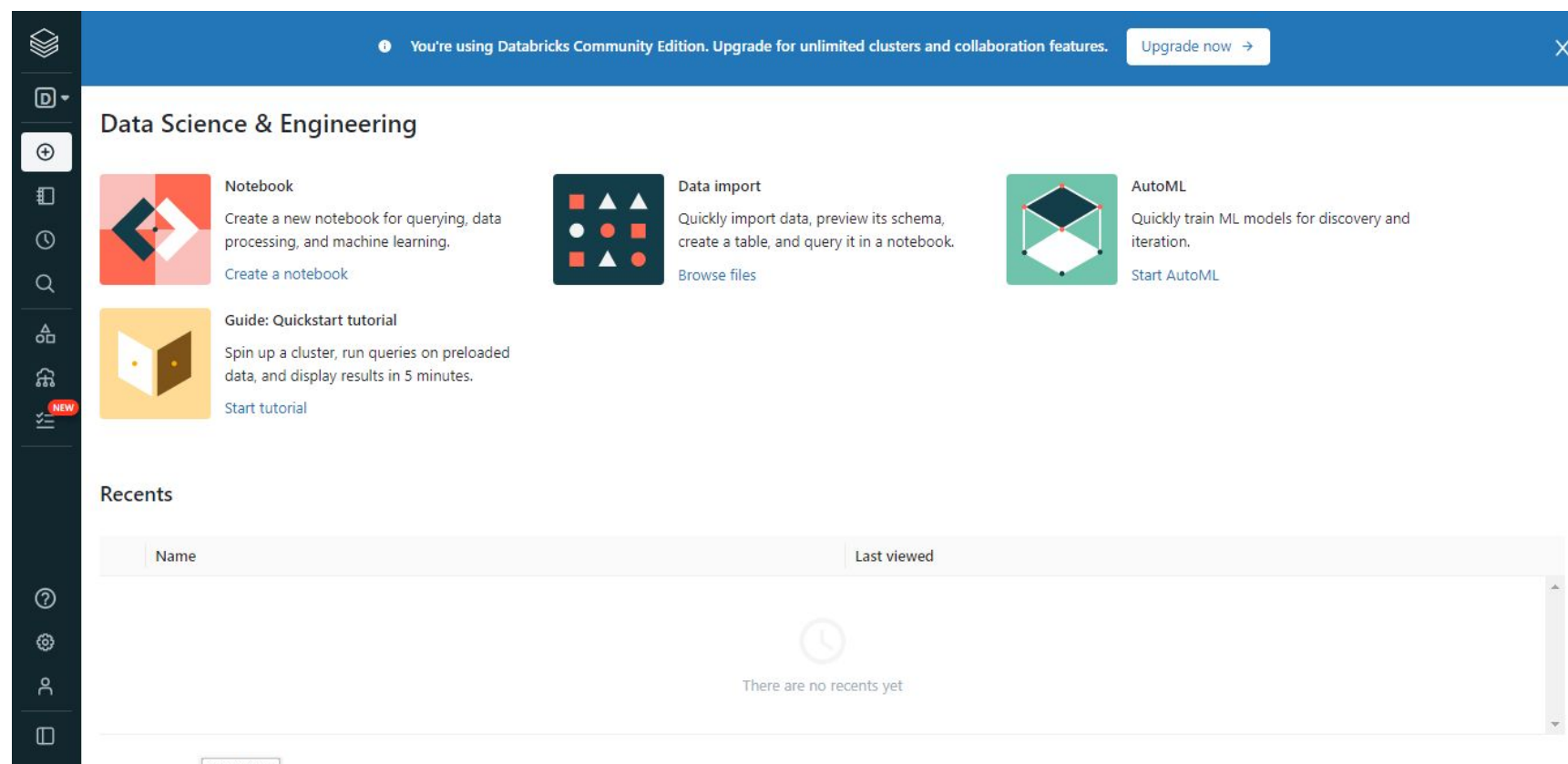
Get started with Community Edition

By clicking "Get started with Community Edition", you agree to the [Privacy Policy](#) and [Community Edition Terms of Service](#)

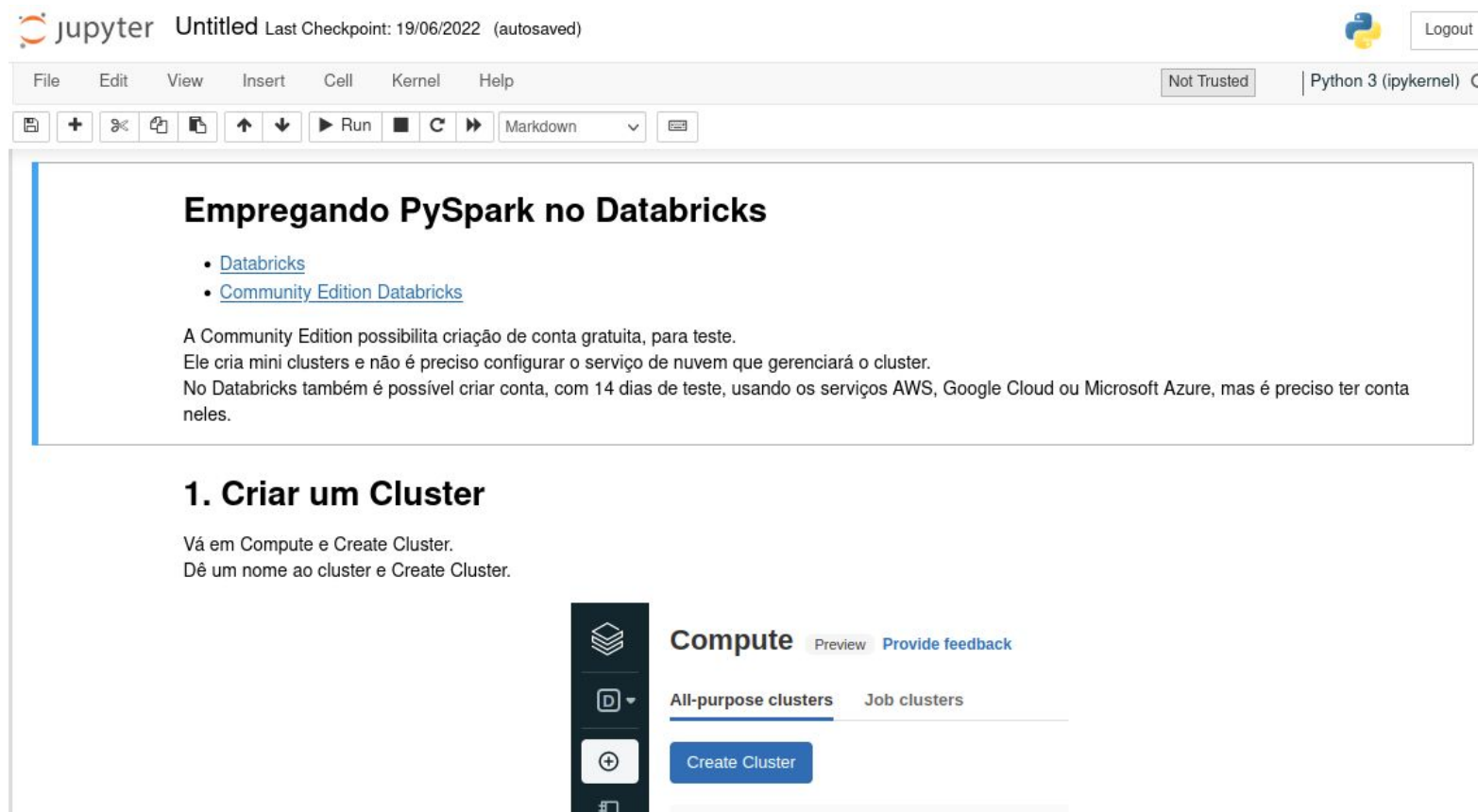


Databricks Community Edition

- <https://community.cloud.databricks.com/>.

The image shows the Databricks Community Edition dashboard. At the top, a blue banner contains a message: "You're using Databricks Community Edition. Upgrade for unlimited clusters and collaboration features." with an "Upgrade now" button. Below the banner, the "Data Science & Engineering" section features three main cards: "Notebook" (with a red and white icon), "Data import" (with a dark blue icon), and "AutoML" (with a green icon). Each card includes a brief description and a link to start the respective feature. Below this section is a "Recents" table with columns for "Name" and "Last viewed". The table is currently empty, displaying a clock icon and the text "There are no recents yet". A vertical sidebar on the left contains various navigation icons, including a "NEW" badge next to the cluster icon.

Notebooks Jupyter



The screenshot shows a Jupyter Notebook interface. At the top, the title bar says "jupyter Untitled Last Checkpoint: 19/06/2022 (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. On the right, there's a "Logout" button and a status bar indicating "Not Trusted" and "Python 3 (ipykernel)". The toolbar contains icons for saving, adding, deleting, and running cells, along with a dropdown menu set to "Markdown".


Empregando PySpark no Databricks

- [Databricks](#)
- [Community Edition Databricks](#)

A Community Edition possibilita criação de conta gratuita, para teste.
Ele cria mini clusters e não é preciso configurar o serviço de nuvem que gerenciará o cluster.
No Databricks também é possível criar conta, com 14 dias de teste, usando os serviços AWS, Google Cloud ou Microsoft Azure, mas é preciso ter conta neles.

1. Criar um Cluster

Vá em Compute e Create Cluster.
Dê um nome ao cluster e Create Cluster.



Compute

Preview [Provide feedback](#)

All-purpose clusters

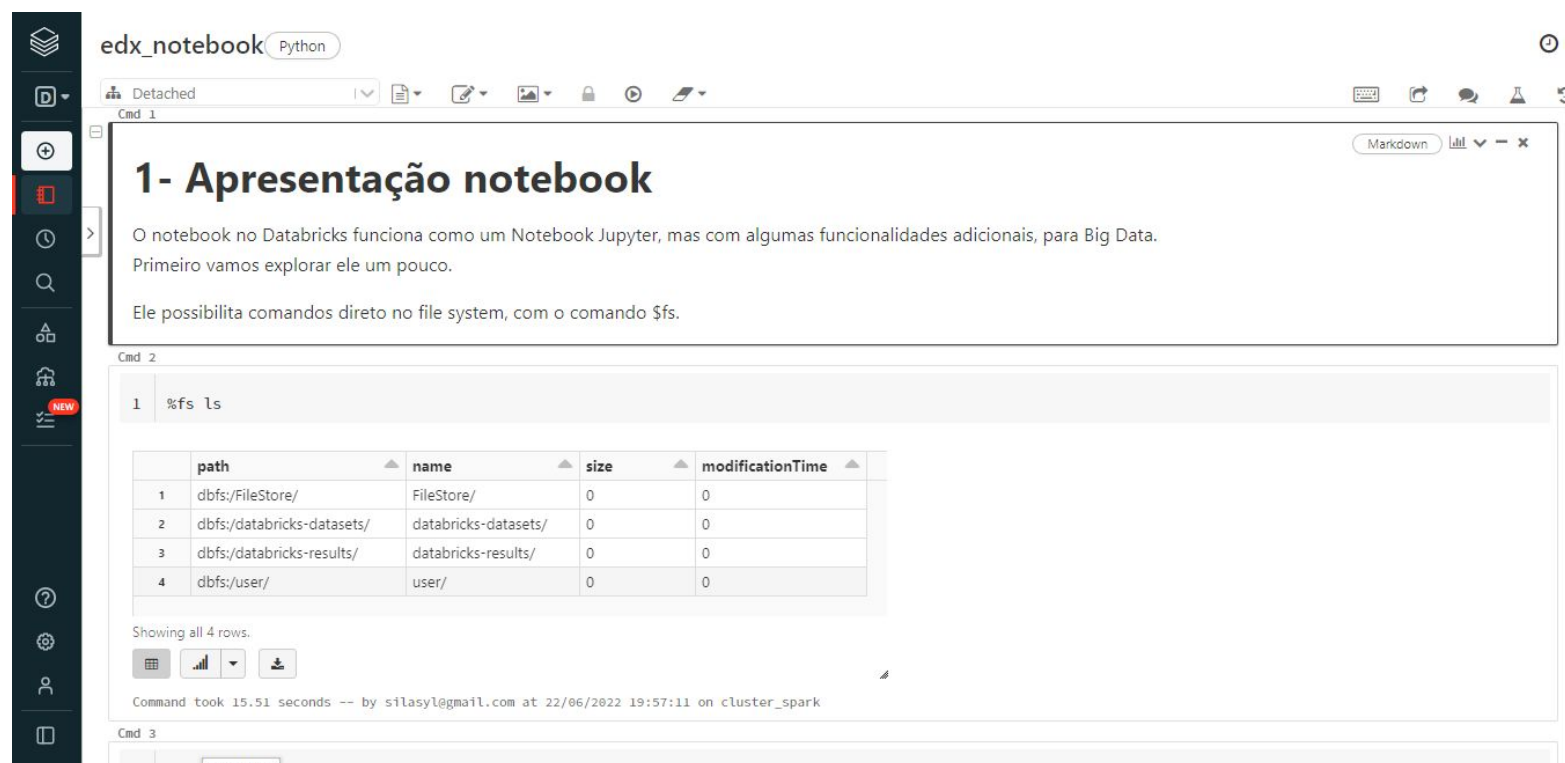
Job clusters

Create Cluster



Notebooks Databricks

- Além das linguagens Scala, Python, R, SQL, aceita comandos file system começando por %fs.

The screenshot shows a Databricks notebook interface. The top bar indicates the notebook is named 'edx_notebook' and is in 'Python' mode. The main content area displays a markdown cell titled '1- Apresentação notebook' with text explaining that Databricks notebooks function like Jupyter notebooks but include additional features for Big Data, such as direct file system commands using %fs. Below the markdown cell, a command prompt shows the execution of '%fs ls', which returns a table listing the contents of the file system. The table has columns for path, name, size, and modificationTime. The command took 15.51 seconds to execute.

edx_notebook Python

Detached

Cmd 1

1- Apresentação notebook

O notebook no Databricks funciona como um Notebook Jupyter, mas com algumas funcionalidades adicionais, para Big Data. Primeiro vamos explorar ele um pouco.

Ele possibilita comandos direto no file system, com o comando %fs.

Cmd 2

```
1 %fs ls
```

	path	name	size	modificationTime
1	dbfs:/FileStore/	FileStore/	0	0
2	dbfs:/databricks-datasets/	databricks-datasets/	0	0
3	dbfs:/databricks-results/	databricks-results/	0	0
4	dbfs:/user/	user/	0	0

Showing all 4 rows.

Command took 15.51 seconds -- by silasy@gmail.com at 22/06/2022 19:57:11 on cluster_spark

Cmd 3

Conclusão

- ❑ Databricks é uma plataforma nuvem especializada em big data e sistemas distribuídos.
- ❑ Conta com notebooks e instalação automática do Spark, pronto para usar.
- ❑ Ideal para se testar e aprender sobre o uso do Spark.



Próxima aula

- ☐ Apresentação ao Spark e PySpark.
- ☐ Bibliotecas do Spark.
- ☐ SparkContext.
- ☐ SparkSession.





Faculdade

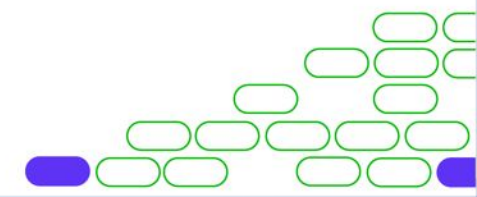


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 7. Framework Spark

Aula 7.1. Introdução ao Spark e PySpark

Prof. Silas Liu



Nesta aula

- ☐ Apresentação ao Spark e PySpark.
- ☐ Bibliotecas do Spark.
- ☐ SparkContext.
- ☐ SparkSession.



Apresentação ao Spark e PySpark

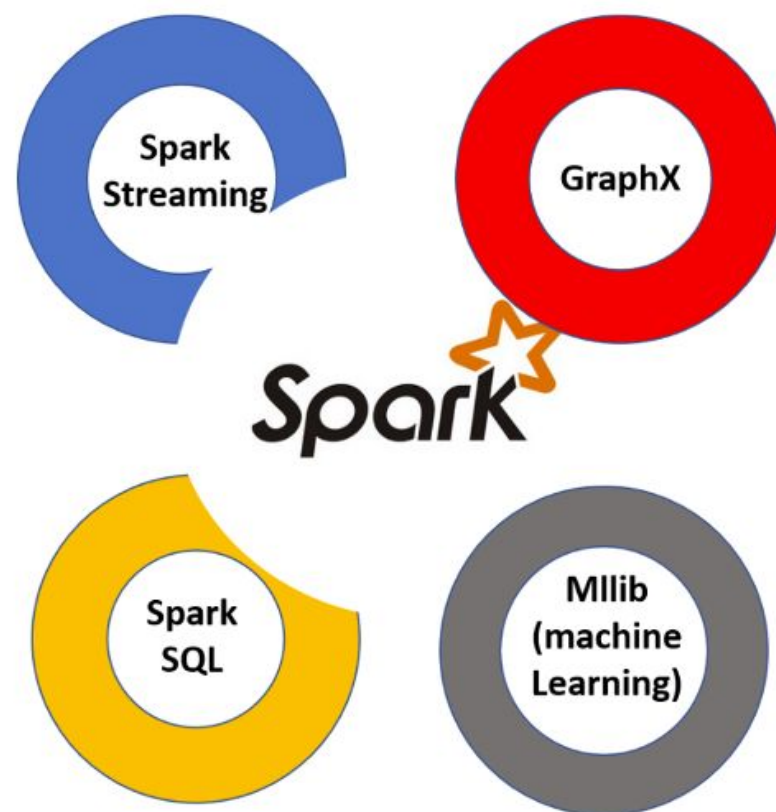


Apresentação ao Spark e PySpark

- Engine baseada em RDDs (Resilient Distributed Datasets).
- Realiza operações de forma extremamente rápidas, utilizando a memória RAM (até 100x mais rápida que MapReduce).
- É desenvolvida em linguagem Scala.
- Aceita linguagens Scala, Java, Python, R, SQL.
- Pode ser incorporado a notebooks, como o Jupyter.
- PySpark é a biblioteca que iremos utilizar, que suporta Python.



Bibliotecas do Spark



SparkContext

- O SparkContext realiza a conexão com o nodo máster.
- Esse objeto faz a comunicação entre o programa e o ambiente.
- Ele armazena as propriedades e configurações aplicadas.
- Esse passo já é realizado no Databricks, não é preciso executá-lo.
- Pode-se observar o SparkContext.

```
1  sc

SparkContext
Spark UI
Version
  v3.2.1
Master
  local[8]
AppName
  Databricks Shell

Command took 0.76 seconds -- by silasyl@gmail.com at 22/06/2022 20:04:10 on cluster_spark
```

SparkSession

- O SparkSession estabelece a interface ao nodo.
- Esse passo também é realizado pelo Databricks, não é preciso executá-lo.
- Pode-se observar o SparkSession.

```
1 # SparkSession criado pelo Databricks
2 spark
```

SparkSession - hive

SparkContext

[Spark UI](#)

Version

v3.2.1

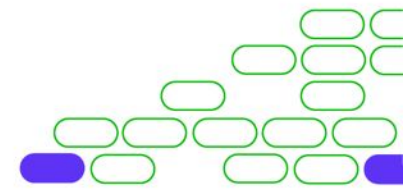
Master

local[8]

AppName

Databricks Shell

Command took 0.86 seconds -- by silasyl@gmail.com at 22/06/2022 20:05:35 on cluster_spark



Conclusão

- ❑ Spark é uma engine RDD extremamente rápida para análise e manipulação de dados muito grandes.
- ❑ É otimizado para operar em memória RAM da máquina.
- ❑ Pode ser incorporado a notebooks como o Jupyter.
- ❑ Aceita várias linguagens como Scala, Java, Python, R e SQL.
- ❑ Para inicializar um sistema Spark, deve-se estabelecer conexão com o SparkContext e SparkSession. No Databricks isso já é feito automaticamente.



Próxima aula

- ☐ Spark Dataframe.
- ☐ Leitura de dados.
- ☐ Visualizar os dados.
- ☐ Operações com o Dataframe.

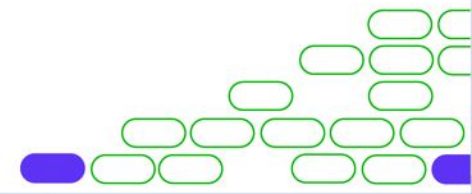


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 7. Framework Spark

Aula 7.2. Spark DataFrames

Prof. Silas Liu



Nesta aula

- ☐ Spark Dataframe.
- ☐ Leitura de dados.
- ☐ Visualizar os dados.
- ☐ Operações com o Dataframe.



Spark Dataframe

- É um objeto de alto nível, que implementa tabelas no nível RDD.
- É uma tabela local, visível apenas à máquina que a criou.
- Aceita linguagens Python ou SQL.
- Toda operação em RDD retorna um Spark Dataframe, estes são ideais para operações.
- Como o RDD é imutável, a cada operação cria-se um novo Dataframe.
- É semelhante ao Pandas Dataframe, mas com suas próprias implementações e funções.



Leitura de dados

- O spark pode ler diversos formatos de arquivos, todos a partir da função `spark.read`. Eles possuem diversos parâmetros, aqui ilustrados alguns possíveis:

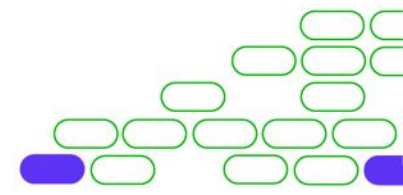
```
df = spark.read.csv("file:///home/name.csv", header="true", inferSchema="true")
df = spark.read.json("file:///home/name.json")
df = spark.read.format("json").load("file:///home.name.json")
df = spark.read.format("csv").
    option("sep", ",").
    option("header", "true").
    load("file:///home/name.csv")
```

Visualizar os dados

- É preciso usar o comando certo para mostrar o conteúdo de um Spark Dataframe, não basta chamá-lo. Podemos ainda visualizar seu schema:

```
df.show()
```

```
df.printSchema()
```



Operações com o Dataframe

- O Spark Dataframe suporta uma série de operações. Vamos analisá-los no notebook, no Databricks.



Conclusão

- ❑ Spark Dataframe é a estrutura mais usada pelo Spark.
- ❑ Ela é uma abstração em alto nível das tabelas armazenadas no RDD.
- ❑ Cada operação retorna um novo Spark Dataframe.
- ❑ O Dataframe oferece uma gama de funções e comandos semelhante ao Pandas Dataframe.



Próxima aula

- ☐ Spark Table.
- ☐ Spark Catalog.
- ☐ Operações com o Table.





Faculdade

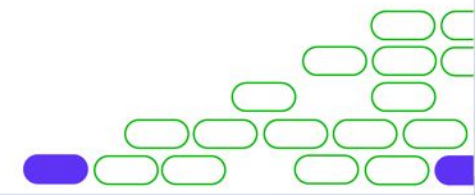


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 7. Framework Spark

Aula 7.3. Spark Tables

Prof. Silas Liu



Nesta aula

- ☐ Spark Table.
- ☐ Spark Catalog.
- ☐ Operações com o Table.



Spark Table

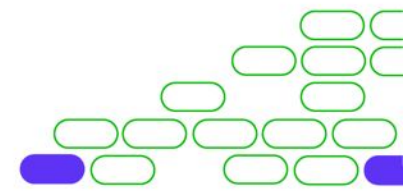
- O Spark Table corresponde às tabelas, também chamado de views, em baixo nível, no RDD.
- Podemos manipular esses views através de comandos SQL.
- As tabelas são a única maneira de compartilhar dados com outros nodos, em um cluster.
- Há dois tipos de tabelas: temporárias que apagam quando o cluster é desligado e as externas, que permanecem salvas.



Spark Catalog

- As views ficam salvas no catálogo do Spark. Podemos acessar seu conteúdo com o seguinte comando:

```
spark.catalog.listTables()
```



Operações com o Table

- O Spark Table suporta uma série de operações, com a função `spark.sql()`.
- Qualquer operação em Tables retornam sempre Dataframes.
- Pode-se usar o Spark Table para criar um Spark Dataframe ou vice-versa. Também é possível passar dados de um Table para um Pandas Dataframe ou ainda salvar um Pandas Dataframe em Spark Dataframe.
- Vamos analisá-los no notebook, no Databricks.



Conclusão

- ❑ Spark Table é a estrutura das tabelas em RDD.
- ❑ Qualquer operação em uma tabela retorna um Dataframe.
- ❑ Podemos manipular e trabalhar com as tabelas utilizando comandos em SQL.

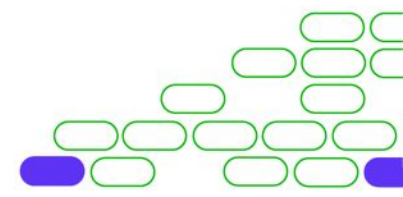


Próxima aula

- Streaming de dados.



XPe





Faculdade

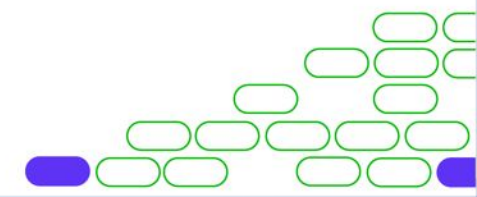


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 8. Streaming de dados

Aula 8.1. Streaming de dados

Prof. Silas Liu

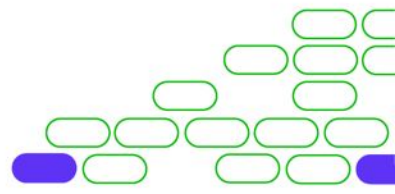


Nesta aula

- ❑ Streaming de dados.



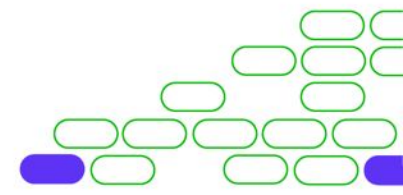
XPe



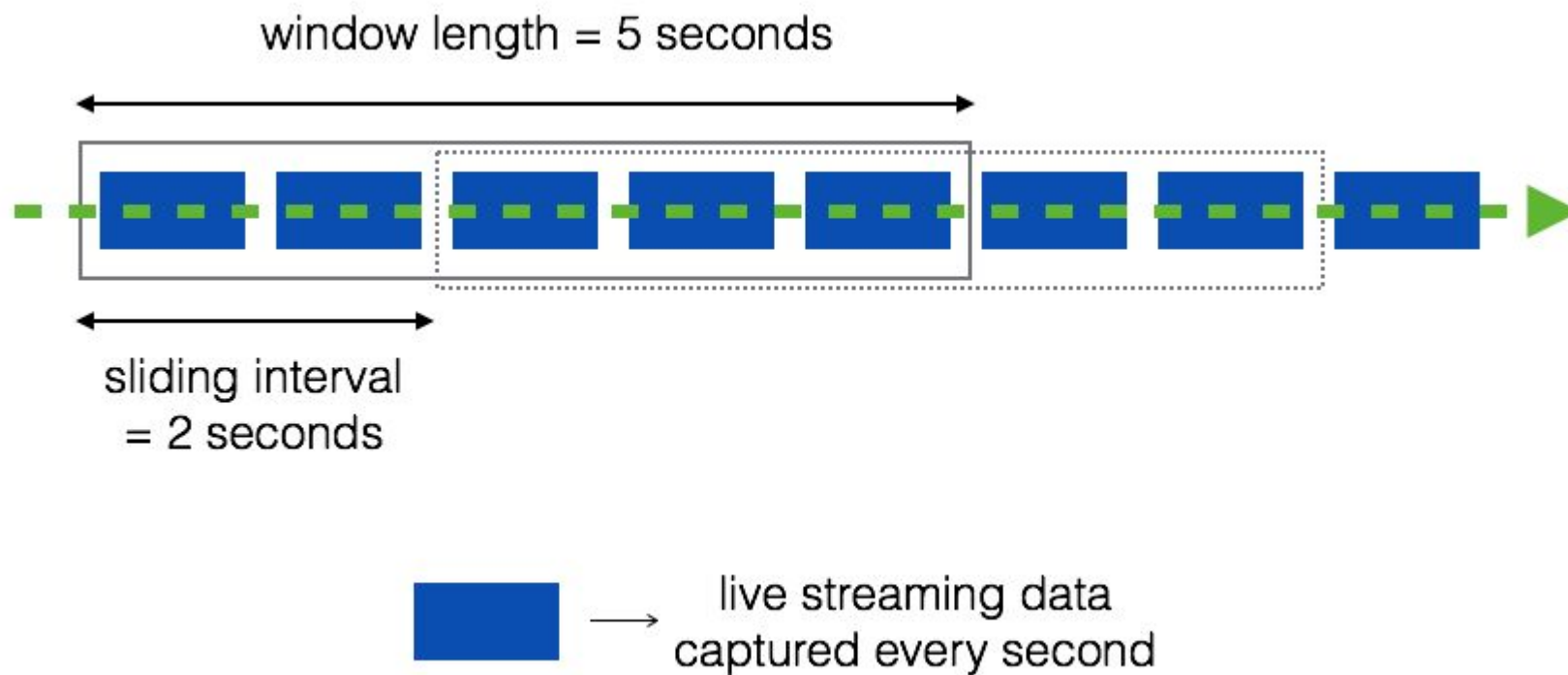
Aplicações Streaming de Dados

Alguns exemplos para processamento de dados em tempo real:

- Sensores médicos;
- Aparelhos de comunicação;
- Aparelhos de transmissão de dados;
- Sensores de monitoramento;
- GPS para localização de veículos/celulares;
- Sistemas de vídeo-vigilância.



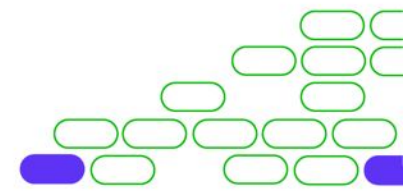
Janelas Temporais



Facilidades de Plataformas Nuvem

Plataformas nuvem oferecem serviços adicionais, tais como:

- Segurança na comunicação;
- Redundância e disponibilidade dos dados;
- Dados críticos em checkpoints;
- Adequação dos dados entre janelas;
- Governança dos processos paralelos.



Conclusão

- Streaming de dados se torna cada vez mais necessário nos dias atuais;
- As ferramentas devem acompanhar o avanço da tecnologia e serem capazes de manipular cada vez mais dados, sem perda de tempo;
- As plataformas nuvem vieram para auxiliar no streaming de dados.



Próxima aula

- ☐ Introdução ao Spark Streaming.
- ☐ StreamingContext.
- ☐ Processo Spark Streaming.





Faculdade

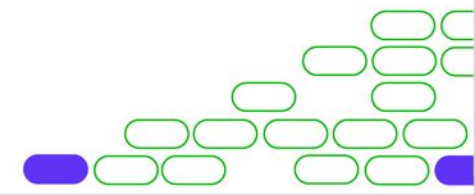


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 8. Streaming de dados

Aula 8.2. Spark Streaming

Prof. Silas Liu



Nesta aula

- ☐ Introdução ao Spark Streaming.
- ☐ StreamingContext.
- ☐ Processo Spark Streaming.



Introdução ao Spark Streaming

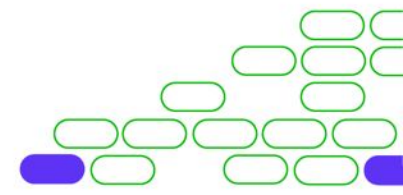
- Biblioteca construída em cima do Spark, para processamento em tempo real;
- Possui uma estrutura semelhante ao Spark convencional:
 - StreamingContext no lugar do SparkContext;
 - DStreams (discretized streams) no lugar do RDD;
- O Spark Streaming realiza operações em janelas temporais, que reúnem amostras de dados em intervalos definidos;
- O Spark Streaming ainda precisa do SparkContext para gerenciar o processo completo.



StreamingContext

O StreamingContext possui as configurações do Spark Streaming, tais como:

- Tamanho da janela temporal;
- Forma de comunicação para obtenção dos dados:
 - APIs de streaming, como do Twitter;
 - Soquetes TCP;
 - Sistemas de mensagens.



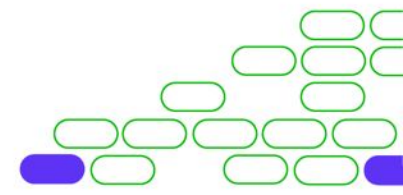
StreamingContext

A comunicação por soquete TCP, por exemplo, ocorre conforme:

```
StreamingContext.socketTextStream(hostname, port,  
storageLevel)
```

Exemplo:

```
from pyspark.streaming import StreamingContext  
ssc = StreamingContext(sc, 1)  
lines = ssc.socketTextStream('localhost', 9999)
```



Processo Spark Streaming

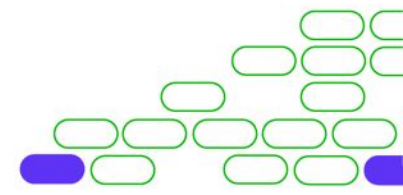
- Após fazer a conexão pelo StreamingContext, definimos as operações;
- Semelhante à declaração de Mapper e Reducer no Hadoop Streaming;
- Abaixo, código para contagem de palavras no DStreams:

```
counts = lines.flatMap(lambda line: line.split(" ")).
```

```
    map(lambda word: (word, 1)).
```

```
    reduceByKey(lambda a, b: a + b)
```

```
counts.pprint()
```

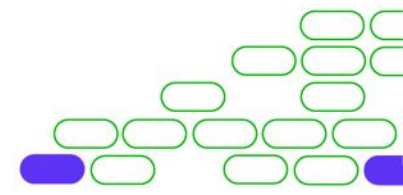


Processo Spark Streaming

- Após a declaração das operações, iniciamos o streaming através do comando `ssc.start()`;
- Para finalizar, podemos programar pelo comando `ssc.stop()` ou esperar pelo fim do processamento com `ssc.awaitTermination()`.

`ssc.start()`

`ssc.awaitTermination()`



Conclusão

- ❑ O Spark Streaming possui semelhanças em sua estrutura com o Spark convencional;
- ❑ O Spark Streaming utiliza o rápido e potente processamento em memória RAM dos RDDs para processar DStreams em streaming de dados;
- ❑ Entretanto, a declaração de suas operações não é tão flexível.



Próxima aula

- ☐ Introdução ao Structured Streaming.
- ☐ Leitura de dados.
- ☐ Operações.

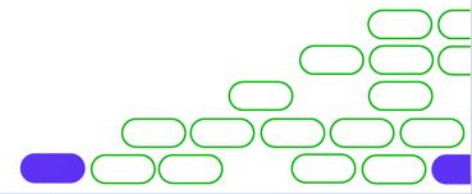


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 8. Streaming de dados

Aula 8.3. Structured Streaming

Prof. Silas Liu



Nesta aula

- ☐ Introdução ao Structured Streaming.
- ☐ Leitura de dados.
- ☐ Operações.



Introdução ao Structured Streaming

- A partir da versão 2.2 do Spark, lançado em 2021, a equipe desenvolveu o Structured Streaming;
- Engine baseada no Spark SQL, aplicada ao streaming de dados;
- Une mais velocidade em processamento a mais facilidade de programar;
- Do contrário do Spark Streaming que realizava as operações em batches, nas janelas temporais, o Structured Streaming realiza operações por linha;
- Spark Streaming e Structured Streaming, ambos ainda são muito utilizados.



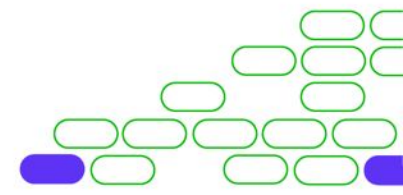
Leitura de dados

- Leitura Spark comum:

```
static_df = spark.read.  
    schema(jsonSchema).  
    json(input_data)
```

- Leitura Structured Streaming:

```
stream_df = spark.readStream.  
    schema(jsonSchema).  
    option("maxFilesPerTrigger",1).  
    json(input_data)
```



Operações

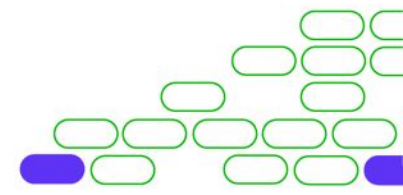
Realizamos operações no Structured Streaming de forma bem simples, semelhante aos comandos no Spark Dataframes:

```
df.select("col")
```

```
df.filter("col > 0")
```

```
df.where("col = 0")
```

```
df.withColumn("col_nova", valor_col_nova)
```

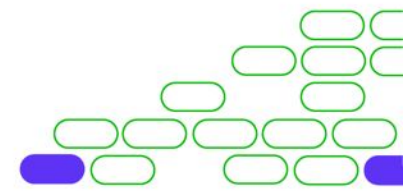


Operações

- Vamos analisar um streaming em tempo real, utilizando o Structured Streaming no Databricks.



XPe



Conclusão

- ❑ O Structured Streaming possui muito mais flexibilidade e é mais fácil de se programar as operações;
- ❑ O Structured Streaming também apresenta um ganho maior ainda no tempo de execução;
- ❑ Por ser uma biblioteca muito recente, ainda está em constantes melhorias e modificações.



Próxima aula

- Introdução ao MLlib.





Faculdade

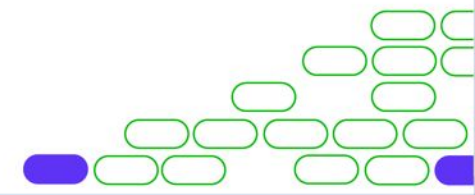


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 9. Spark MLlib

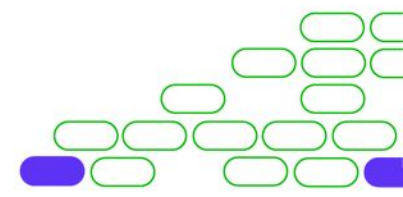
Aula 9.1. Introdução ao MLlib

Prof. Silas Liu



Nesta aula

- Introdução ao MLlib.

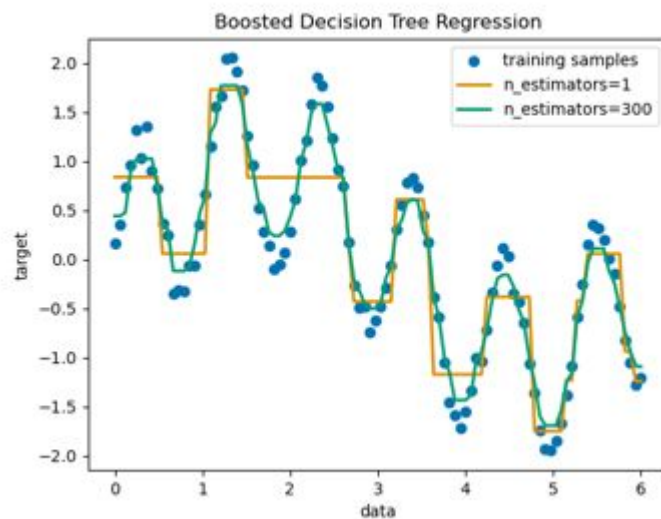
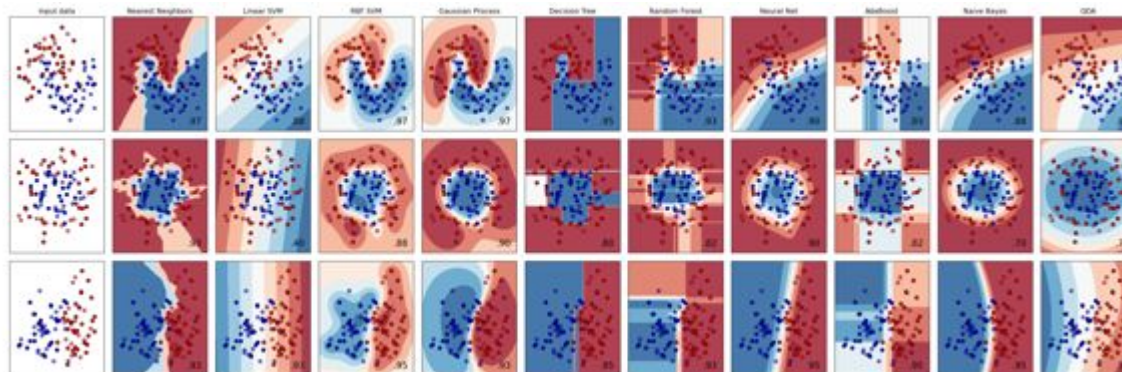


Introdução ao MLlib

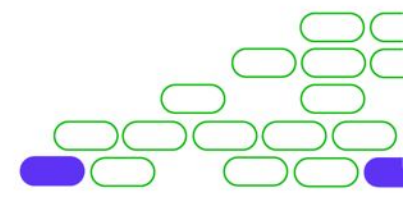
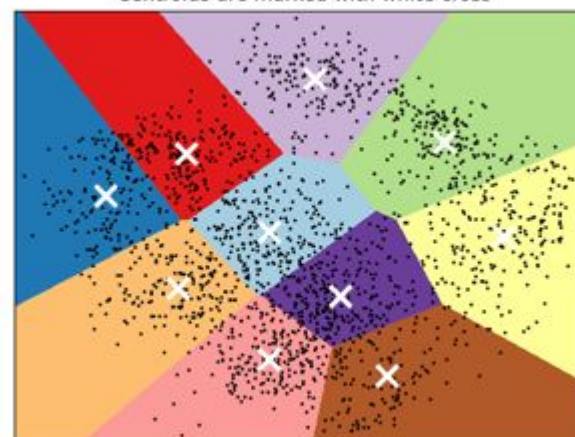
- MLlib é a biblioteca do Spark com implementações para Machine Learning;
- Contém diversos modelos para:
 - Classificação;
 - Regressão;
 - Clusterização;
 - Modelagem;
 - Decomposição;
 - Testes estatísticos;
 - Redes neurais.



Introdução ao MLlib



K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross



Introdução ao MLlib

- O MLlib se aproveita da capacidade e rápido processamento, em memória RAM, e de forma distribuída, para aplicar o Spark direto nos modelos de Machine Learning;
- Precisamos aprender como adequar os dados, vindos do RDD ou Dataframes, para alimentar modelos no MLlib.



Conclusão

- ❑ O MLlib é uma poderosa ferramenta, especializada para ciência de dados e Machine Learning, utilizando o Spark;
- ❑ É preciso adequar os dados previamente, para aplicar aos modelos do MLlib.



Próxima aula

- ☐ Preparação dos dados.
- ☐ Classes de modelos.
- ☐ Mudança de tipo de variável.
- ☐ Booleano para numérico.
- ☐ Texto para numérico.

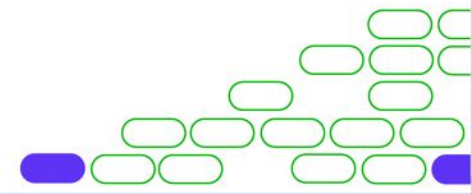


Processamento de Dados Utilizando o Ecossistema Hadoop

Capítulo 9. Spark MLlib

Aula 9.2. Preparação dos dados

Prof. Silas Liu



Nesta aula

- ☐ Preparação dos dados.
- ☐ Classes de modelos.
- ☐ Mudança de tipo de variável.
- ☐ Booleano para numérico.
- ☐ Texto para numérico.



Preparação dos dados

- Os modelos do MLlib lidam apenas com valores numéricos;
- Isso significa que temos de transformar todas as variáveis (string, boolean, categóricos) em inteiros (integer) ou fracionários (double);
- Desta forma, os modelos são ainda mais rápidos, eles não precisam inferir o tipo de variáveis e são otimizados para lidar com números.



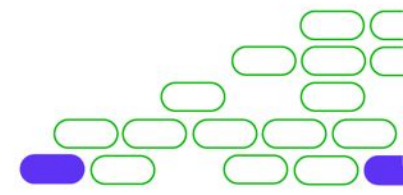
Classes de modelos

O pyspark.ml utiliza duas classes principais:

- Transformer: possui o método `.transform()` e realiza alguma operação em um Spark Dataframe, retornando um novo Dataframe. Exemplos:
 - Bucketizer, que cria colunas discretas de valores contínuos;
 - PCA, que reduz a dimensionalidade e retorna novas colunas;
- Estimator: possui o método `.fit()` e realiza operações de um Spark Dataframe, retornando um objeto do tipo modelo.

Exemplos:

- Treinamento de modelos, para obter os parâmetros.



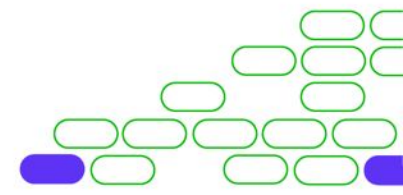
Mudança de tipo de variável

O comando `.cast()` transforma o tipo de variável:

- `.cast("integer")`
- `.cast("double")`

Exemplo:

```
df = df.withColumn("col_numerico",  
df.col_string.cast("integer"))
```

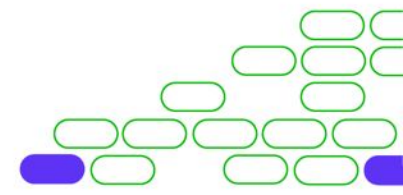


Booleano para numérico

Podemos aplicar lógicas de comparação de true/false ou de valor:

```
df = df.withColumn("res_bool", df.col > 10)
```

```
df = df.withColumn("bool", df.res_bool.cast('integer'))
```



Texto para numérico

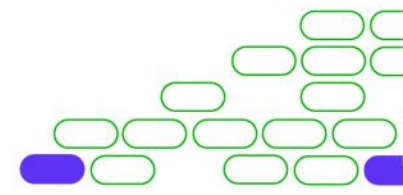
Consiste em duas etapas:

- Criar um Dataframe de números, correspondendo um número a cada palavra:

StringIndexer: estimator seguido de transformer

- Criar o One-Hot Vector, um vetor de zeros e um apenas na posição da palavra:

OneHotEncoder: estimator seguido de transformer

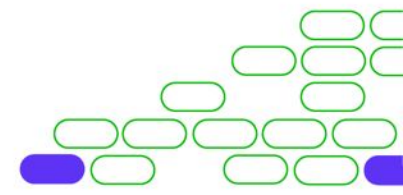


Texto para numérico

Para encapsular o One-Hot Vector, empregamos o **Vector Assembler**, que separa as colunas de features dos dados da saída do modelo.

A seguir juntamos todos os componentes através do **Pipeline**, que leva em conta todas as estruturas das transformações.

Por fim, chamamos o Pipeline através dos comandos `fit()` e `transform()`, passando como argumento o Dataframe que queremos transformar.



Texto para numérico

Exemplo de um processo:

```
from pyspark.ml import feature
```

```
from pyspark.ml import Pipeline
```

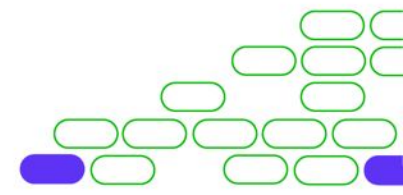
```
indexer = feature.StringIndexer(inputCol='col', outputCol='index')
```

```
encoder = feature.OneHotEncoder(inputCol='index', outputCol='factor')
```

```
assembler = feature.VectorAssembler(inputCols=['col1', 'col2'], outputCol='label')
```

```
pipe = Pipeline(stages=[indexer, encoder, assembler])
```

```
new_data = pipe.fit(df).transform(df)
```



Conclusão

- ❑ O MLlib é uma poderosa ferramenta, mas precisa que todos seus dados sejam transformados em numéricos;
- ❑ O próprio MLlib possui bibliotecas e funções para executar as mudanças de variáveis para numérico;
- ❑ Para transformar texto em numérico, empregamos o StringIndexer, OneHotEncoder, VectorAssembler e Pipeline, seguidos de fit() e transform().





Faculdade

XPe

