# Fundamentos de Big Data

UNIDADE 05

## Hadoop: interação com HDFS

Nas últimas quatro semanas mergulhamos nos fundamentos de *big data*. Quer dizer: falamos sobre a teoria, mas não necessariamente colocamos em prática estes conhecimentos. Confesso que pensei se seria melhor primeiro apresentarmos a teoria e depois a prática ou, quem sabe, trabalhar com a teoria e a prática ao mesmo tempo semanalmente.

Contudo, como você percebeu até o momento temos vários conceitos complexos *em paralelo*: termos como *big data,* Spark, Hadoop, HDFS, MapReduce, *data streaming* e arquitetura de dados. Alguns desses conceitos não podem ser facilmente tratados em

código ou em algum software, como é o próprio caso da arquitetura. Também existem casos em que é importante entendermos o porquê de termos estas soluções, como foi o caso da história do surgimento do Spark.

## Hora de colocar a mão na massa!

Dito isso, é o momento de colocarmos a mão na massa. Lembra do material que a gente preparou para você acessar a VM, lá na semana 2? Então, provavelmente, o seu acesso já deve estar funcionando. O que acha de testar novamente?

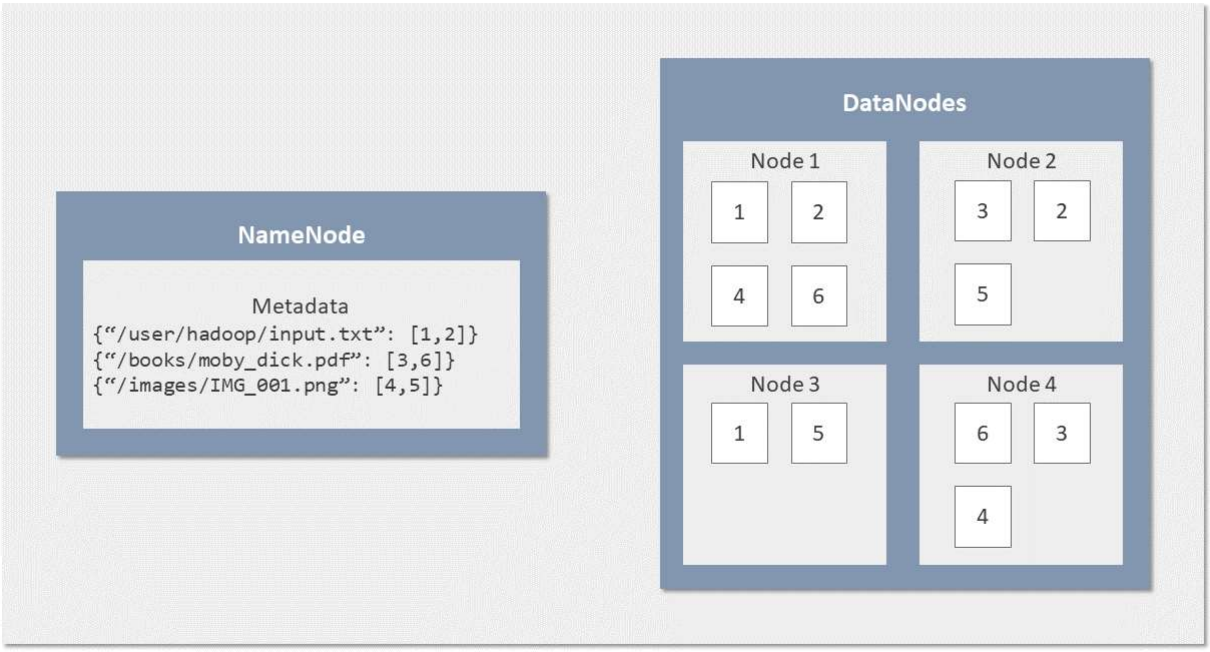
**EXPERIMENTE**

Aproveite este momento para acessar a máquina virtual (VM) da Universidade utilizando o guia que disponibilizamos para você durante a Semana 2. Isso será essencial para fazermos as atividades a partir de agora.

## Trabalhando com o Hadoop

Para o armazenamento de arquivos em massa no Hadoop, é disponibilizado o HDFS, um sistema distribuído de arquivos nativo dele. Para seu uso, é importante a preparação do ambiente, configurando-o adequadamente para que arquivos de dados possam ser transferidos e acessados no HDFS via comandos ou aplicações.

O HDFS é disponibilizado com a instalação do ecossistema Hadoop. Só relembrando: para o HDFS instanciamos apenas um *NameNode*, responsável por conter os metadados (ex.: nome do arquivo, local em que ele está, permissões de leitura e escrita) referentes ao controle dos arquivos de dados que serão armazenados em um ou mais nós de dados (*DataNodes*), conforme figura a seguir. São os *DataNodes* que fazem o armazenamento dos blocos de dados que são partes dos arquivos originais a ser armazenados e que são **replicados de modo a haver recuperação em caso de falhas**. Para ficar mais claro, vejamos a figura abaixo:




Fonte: Radtka e Miner (2016, p. 3).

Vamos entender melhor o que essa figura quer dizer?

## NameNode

O **NameNode** é o cérebro do HDFS, gerenciando o *namespace* do sistema de arquivos e regulando o acesso aos arquivos pelos clientes. Ele armazena os metadados dos arquivos, como os nomes dos arquivos e a lista de blocos (e seus DataNodes correspondentes) que compõem cada

arquivo. Por exemplo, o arquivo "*/user/hadoop/input.txt*" é dividido entre os blocos 1 e 2.

### DataNodes

Já os **DataNodes** são os trabalhadores (*workers*) que armazenam e recuperam blocos de dados quando solicitados pelo NameNode ou pelos clientes. Cada DataNode reporta ao NameNode com uma lista de blocos que ele está armazenando. A distribuição dos blocos é feita de forma a garantir redundância e eficiência no acesso. Por exemplo, o bloco 1 está presente nos nós 1 e 3

### Processo de Armazenamento

Quando um arquivo é carregado no HDFS, ele é dividido em blocos de dados de tamanho fixo (exceto o último bloco, que pode ser menor). Cada bloco é armazenado em vários DataNodes para assegurar a tolerância a falhas.

### Leitura e Escrita

Para ler um arquivo, o cliente solicita ao NameNode a localização dos blocos. Para escrever, o cliente pede ao NameNode para preparar os DataNodes para receber os blocos de dados.

A ideia de tudo isso é a de garantir um **processamento paralelo** (o que significa *velocidade*) e **tolerância a falhas** (o que significa em *confiabilidade*).

Agora, eu gostaria que você segurasse um pouco estas informações em seu cérebro: testaremos o HDFS na prática a partir de agora. Contudo, eu preciso lhe apresentar uma outra ferramenta antes: o ***Jupyter***.

## Jupyter e databricks

Profissionais de dados (ex.: engenheiros de dados, cientistas de dados, analistas de dados e engenheiros de *machine learning*) estão bem acostumados a trabalhar com ***notebooks***. Você se lembra quando escreveu os seus códigos em Raciocínio Computacional? Eram *scripts* em Python: arquivos simples contendo somente código.

Agora, estes profissionais de dados não trabalham com estes scripts na maioria do tempo, mas sim com notebooks (não confunda isso com aqueles computadores portáteis: estamos falando de *outra coisa* aqui). Um notebook é um ambiente de desenvolvimento interativo que permite que os usuários criem e compartilhem documentos que contêm código ao vivo, equações, visualizações e texto explicativo.

Nestes notebooks, você pode escrever código em células individuais que podem ser executadas independentemente. Isso permite que você execute códigos em partes, o que ajuda bastante no processo de experimentação e *debug*. Além disso, os notebooks suportam a integração de texto rico usando a linguagem de marcação Markdown, permitindo que você crie documentos que combinem código, texto formatado, equações matemáticas e visualizações. Em outras palavras, isto faz com que o seu código também seja uma documentação para explicar o que você fez e o que descobriu durante o seu processo criativo.

Logo, usaremos ***notebooks*** nesta disciplina. Temos algumas soluções que permitem o seu uso, como o **Jupyter Notebook, JupyterLab, Google Colab** e o **Databricks**. Tanto o **Jupyter Notebook** quanto o **JupyterLab** são desenvolvidos pelo Projeto Jupyter, sendo o **JupyterLab** o mais atual. Ele permite que você trabalhe com documentos e atividades como notebooks Jupyter, editores de texto, terminais e visualizações de dados personalizadas, tudo em uma única interface com abas flexíveis e responsivas. É como se fosse uma IDE para ciência de dados.

Já o **Databricks** é uma solução de mercado. É uma plataforma baseada em nuvem que unifica análise de dados e inteligência artificial. Ele foi construído em cima do Apache Spark, e oferece um ambiente colaborativo e interativo, semelhante ao Jupyter, para trabalhar com Spark. O Databricks permite que você execute trabalhos complexos e pesados de análise de dados, treine modelos de IA em grande escala e muito mais, tudo isso sem se preocupar com a configuração e manutenção da infraestrutura.

**DICA**

Trabalharemos com o JupyterLab na disciplina. Ao conhecer o JupyterLab, você estará apto para poder desenvolver códigos no Google Colab, Databricks, e outras plataformas corporativas pagas.

Como comentamos, usaremos o JupyterLab para interagir com o Hadoop, Spark e demais frameworks desta disciplina. O que acha de entendermos melhor como ele funciona?

## Apresentando o JupyterLab

É o momento de conhecermos a plataforma que usaremos para desenvolver os nossos códigos: o JupyterLab. Vamos lá?

## PyArrow

Espero que neste momento você já saiba como funciona o JupyterLab, e que você já tenha configurado o seu próprio ambiente na máquina virtual. Agora é o momento de interagirmos com o HDFS usando o JupyterLab. Em específico, usaremos a linguagem Python para isso.

Conseguiremos esta conexão com o **PyArrow**, uma solução feita pela Apache. Você já deve ter ouvido falar sobre a Apache em alguns momentos – afinal, eles estão por trás de alguns projetos muito importantes em TI que você já deve ter ouvido falar como, por exemplo:

- Kafka
- OpenOffice
- Hadoop
- CouchDB
- Apache HTTP Server
- NetBeans
- Spark
- Cassandra
- Hive
- HBase
- Arrow
- Log4j
- Parquet
- Avro

A Apache Software Foundation (ASF) é uma organização sem fins lucrativos que foi criada para apoiar os projetos de software Apache. Ela serve como um lar para mais de 350 projetos de código aberto e iniciativas, incluindo Apache Hadoop, Apache Kafka e Apache Spark, apenas para citar alguns. Ela promove a colaboração e a comunidade aberta, fornecendo ferramentas, infraestrutura e suporte para ajudar as comunidades de desenvolvedores a colaborar e construir soluções de software de alta qualidade.

A importância da ASF é imensa, e é por isso que comento sobre ela aqui. Ela desempenha um papel crucial na promoção do software de código aberto (*open source*), permitindo que indivíduos e empresas contribuam para projetos de software que são usados por milhões de pessoas em todo o mundo.

Um desses projetos é o **Apache Arrow**. O Python pode usar o **PyArrow** (a implementação do Apache Arrow para Python) para ler e escrever dados no HDFS de um jeito padronizado. Isso permite que os cientistas de dados e engenheiros de software usem Python para processar grandes conjuntos de dados armazenados no HDFS, aproveitando a eficiência do Arrow para transferência e processamento de dados. Isso acontece porque o Arrow fornece um formato de memória padronizado que permite a transferência eficiente de dados entre sistemas e processos, e é justamente isso que precisamos.

Logo, usaremos o PyArrow para interagirmos com o HDFS escrevendo código em Python. O que acha de começarmos?

### VIDEOAULA: Vamos trabalhar com o HDFS?

Agora que já conhecemos o JupyterLab, é hora de criarmos o nosso algoritmo para manipular dados com o HDFS pelo PyArrow. Vamos lá?

## Conclusão

Você pode ter percebido que o conteúdo escrito desta semana ficou bem enxuto. Em compensação, temos vídeos maiores. Afinal, a ideia é a de praticarmos estes conceitos. Por isso, nesta unidade aprendemos como acessar o ambiente Hadoop em um laboratório virtual em Linux. Para isso, aprendemos ferramentas novas como o JupyterLab e o PyArrow. Neste caso, também vimos como podemos usar o PyArrow para interagir com arquivos que estão armazenados no HDFS usando Python.

## Referências

ODEDARA, P. How to install Hadoop on Windows. **Exit Condition**, 4 ago. 2018. Disponível em: https://exitcondition.com/install-hadoop-windows/. Acesso em: 22 jul. 2023.

RADTKA, Z.; MINER, D. **Hadoop with Python**. Sebastopol: O' Reilly, 2016.

SANTOS, R. R. *et al*. **Fundamentos de big data**. Porto Alegre: SAGAH, 2021.

VULTR. **Install and configure Apache Hadoop on Ubuntu 20.04**. 2023. Disponível em: https://www.vultr.com/docs/install-and-configure-apache-hadoop-on-ubuntu-20-04/?utm_source=performance-max-latam&utm_medium=paidmedia&obility_id=17096555207&utm_adgroup=&utm_campaign=&utm_term=&utm_content=&gclid=CjwKCAjwsvujBhAXEiwA_UXnAJoMGyZvOBuRATSjZY76O7e2GXz398hS4-u4bx7h-_D33v0lHF9gEhoC_x4QAvD_BwE. Acesso em: 8 jun. 2023.