

INF550 - Computação em Nuvem I

MapReduce

Islene Calciolari Garcia

Instituto de Computação - Unicamp

Junho de 2018

Um pouco sobre mim...

- ▶ Formação e filiação
 - ▶ Instituto de Computação—Unicamp
- ▶ Interesses de pesquisa
 - ▶ Sistemas distribuídos
 - ▶ Sistemas operacionais



SOFTWARE
LIVRE

15/06 **MapReduce** (Islene)

- ▶ Introdução a Computação em Nuvens e Big Data
- ▶ História
- ▶ HDFS
- ▶ MapReduce
- ▶ WordCount e outras aplicações
- ▶ Experimento prático (OpenStack e Hadoop)

16/06 **Virtualização** (Luiz)

24/06 **Computação em nuvens** (Luiz)

01/07 **Spark** (Islene)

Critério de avaliação: um experimento por aula, pesos iguais

Computação em Nuvem

O que você associa ao termo computação em nuvem?

- ▶ Pontos positivos?
- ▶ Pontos negativos?



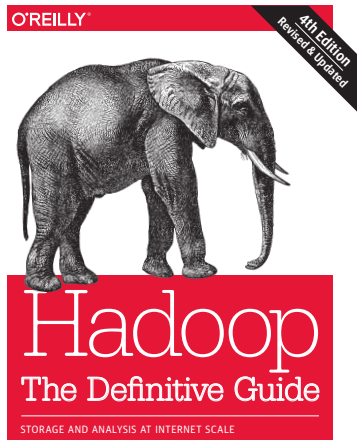
Fonte: https://upload.wikimedia.org/wikipedia/commons/thumb/1/12/Cloud_computing_icon.svg

Computação em Nuvem e Big Data

- Foco da aula de hoje:
Processamento de grandes massas de dados na nuvem



Hadoop e a importância de um framework



Tom White

Exemplo retirado do livro
Hadoop—The Definitive Guide

- ▶ Achar a temperatura máxima por ano em um conjunto de arquivos texto
- ▶ Codificar todo o trabalho em Unix...

Weather dataset

Dados crus, comentários ilustrativos

```
0057
332130  # USAF weather station identifier
99999   # WBAN weather station identifier
19500101 # observation date
0300    # observation time
4
+51317  # latitude (degrees x 1000)
+028783 # longitude (degrees x 1000)
FM-12
+0171   # elevation (meters)
99999
V020
320     # wind direction (degrees)
1       # quality code
N
0072
1
00450   # sky ceiling height (meters)
1       # quality code
C
N
010000  # visibility distance (meters)
1       # quality code
N
9
-0128   # air temperature (degrees Celsius x 10)
1       # quality code
-0139   # dew point temperature (degrees Celsius x 10)
```

Fonte: Hadoop—The Definitive Guide, *Tom White*

Weather dataset

Organização dos arquivos

```
% ls raw/1990 | head
010010-99999-1990.gz
010014-99999-1990.gz
010015-99999-1990.gz
010016-99999-1990.gz
010017-99999-1990.gz
010030-99999-1990.gz
010040-99999-1990.gz
010080-99999-1990.gz
010100-99999-1990.gz
010150-99999-1990.gz
```

Fonte: Hadoop—The Definitive Guide, *Tom White*

Weather dataset

Código em `awk` e saída

```
#!/usr/bin/env bash
for year in all/*
do
    echo -ne `basename $year .gz`"\t"
    gunzip -c $year | \
        awk '{ temp = substr($0, 88, 5) + 0;
              q = substr($0, 93, 1);
              if (temp !=9999 && q ~ /[01459]/ && temp > max) max = temp }
            END { print max }'
done
```

```
% ./max_temperature.sh
1901    317
1902    244
1903    289
1904    256
1905    283
...
```

Fonte: Hadoop—The Definitive Guide, *Tom White*

Weather dataset

Como paralelizar de maneira mais simples?

- ▶ Criar uma infraestrutura que gerencie
 - ▶ distribuição
 - ▶ escalabilidade
 - ▶ tolerância a falhas
- ▶ Criar um modelo genérico para *big data*
 - ▶ Conjuntos chave-valor
 - ▶ Operações *map* e *reduce*

Weather dataset

Dados crus e conjuntos chave-valor

```
00670119909999991950051507004...9999999N9+00001+9999999999...  
00430119909999991950051512004...9999999N9+00221+9999999999...  
00430119909999991950051518004...9999999N9-00111+9999999999...  
00430126509999991949032412004...0500001N9+01111+9999999999...  
00430126509999991949032418004...0500001N9+00781+9999999999...
```

```
(0, 00670119909999991950051507004...9999999N9+00001+9999999999...)  
(106, 00430119909999991950051512004...9999999N9+00221+9999999999...)  
(212, 00430119909999991950051518004...9999999N9-00111+9999999999...)  
(318, 00430126509999991949032412004...0500001N9+01111+9999999999...)  
(424, 00430126509999991949032418004...0500001N9+00781+9999999999...)
```

Fonte: Hadoop—The Definitive Guide, *Tom White*

Weather dataset

Função map

```
(0, 0067011990999991950051507004...9999999N9+00001+9999999999...)  
(106, 0043011990999991950051512004...9999999N9+00221+9999999999...)  
(212, 0043011990999991950051518004...9999999N9-00111+9999999999...)  
(318, 0043012650999991949032412004...0500001N9+01111+9999999999...)  
(424, 0043012650999991949032418004...0500001N9+00781+9999999999...)  
  
(1950, 0)  
(1950, 22)  
(1950, -11)  
(1949, 111)  
(1949, 78)
```

Fonte: Hadoop—The Definitive Guide, *Tom White*

Weather dataset

Pré-processamento e função reduce

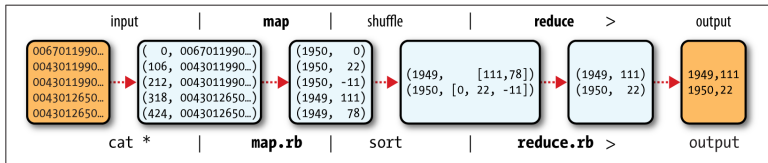
(1949, [111, 78])
(1950, [0, 22, -11])

(1949, 111)
(1950, 22)

Fonte: Hadoop—The Definitive Guide, *Tom White*

Weather dataset

Fluxo de dados



Fonte: Hadoop—The Definitive Guide, *Tom White*

Projeto Apache Hadoop

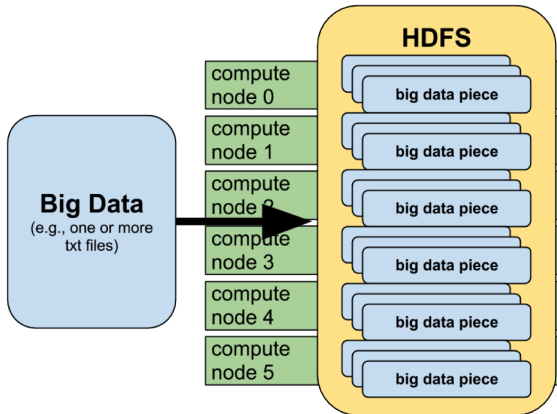


- ▶ Sistema real! Software livre!
- ▶ Big Data: *Volume, Velocity, Variety, Veracity*
- ▶ Computação distribuída escalável e confiável
- ▶ Altamente relevante: usado por empresas como Amazon, Facebook, LinkedIn e Yahoo! Veja mais em Powered by Apache Hadoop?

Um pouco da história do projeto Hadoop

- ▶ 2002-2004: Doug Cutting e Mike Cafarella trabalham no projeto Nutch.
 - ▶ Nutch deveria indexar a web e permitir buscas
 - ▶ Alternativa livre ao Google
- ▶ 2003-2004: Google publica artigo sobre o Google File System e MapReduce
- ▶ 2004: Doug Cutting adiciona o DFS e MapReduce ao projeto Nutch
- ▶ 2006: Doug Cutting começa a trabalhar no Yahoo!
- ▶ 2008: Hadoop se torna um projeto Apache

Arquitetura do HDFS



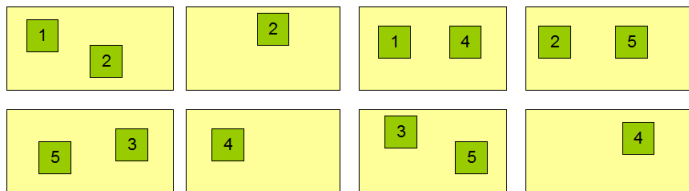
Fonte: <http://www.glennklockwood.com/data-intensive/hadoop/overview.html>

HDFS e réplicas

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}, ...

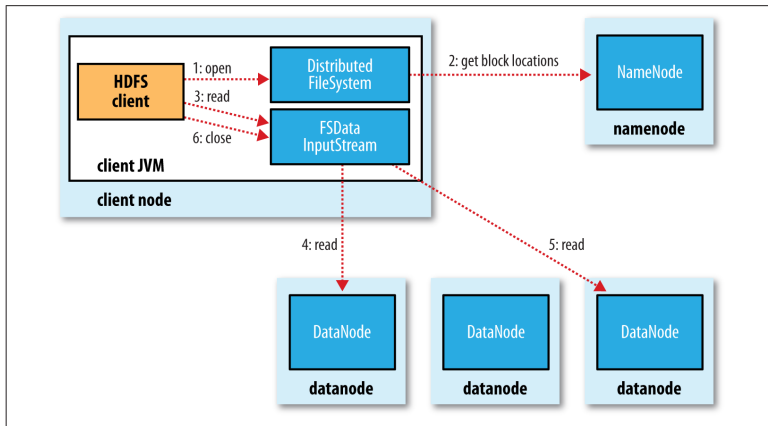
Datanodes



Fonte: <http://hadoop.apache.org>

HDFS

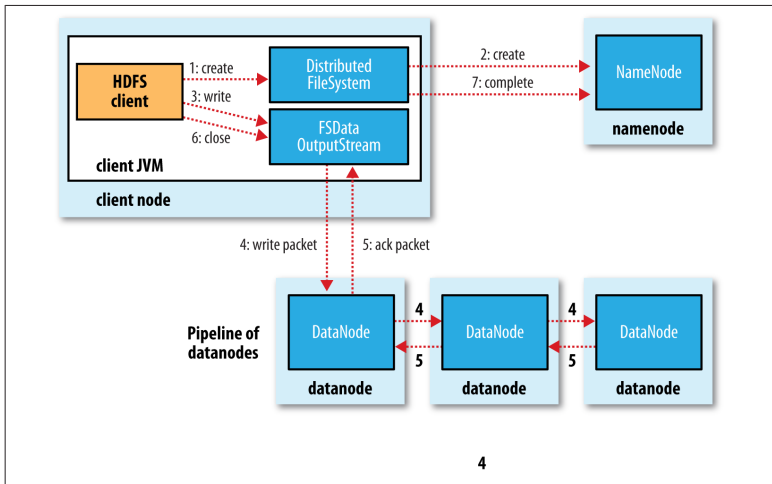
Leitura de arquivo



Fonte: Hadoop—The Definitive Guide, *Tom White*

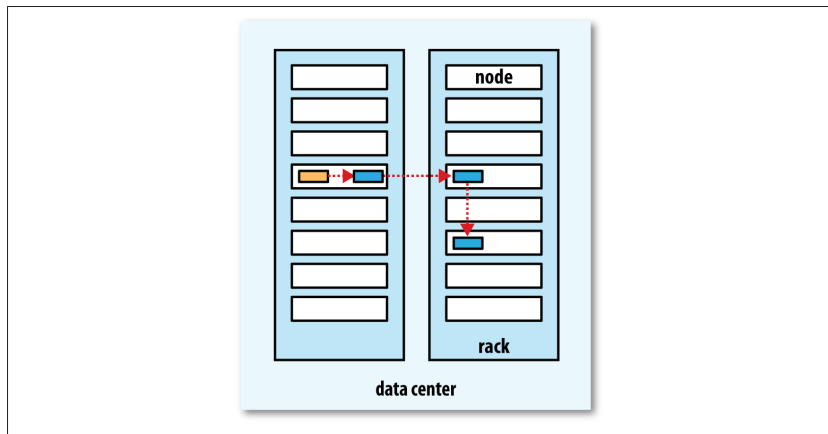
HDFS

Escrita em arquivo



HDFS

Pipeline



Fonte: Hadoop—The Definitive Guide, *Tom White*

HDFS

Tolerância a falhas

- ▶ Heartbeats
- ▶ Block reports
- ▶ Alta disponibilidade do NameNode
- ▶ Réplicas ou Erasure Coding?

Arquivo:

A	B
---	---

Réplicas simples:

A	A	B	B
---	---	---	---

Erasure coding:

A	B	$A+B$	$A+2*B$
---	---	-------	---------

Testando o HDFS

- ▶ Hadoop: Setting up a Single Node Cluster
- ▶ Interface web: `http://<ip>:50070/`
- ▶ Alguns comandos

```
$ bin/hdfs namenode -format
```

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

```
$ bin/hdfs dfs -put <arquivo_local> <arquivo_no_hdfs>
```

```
$ bin/hdfs dfs -get <arquivo_no_hdfs> <arquivo_local>
```

```
$ bin/hdfs dfs -ls <diretorio_no_hdfs>
```

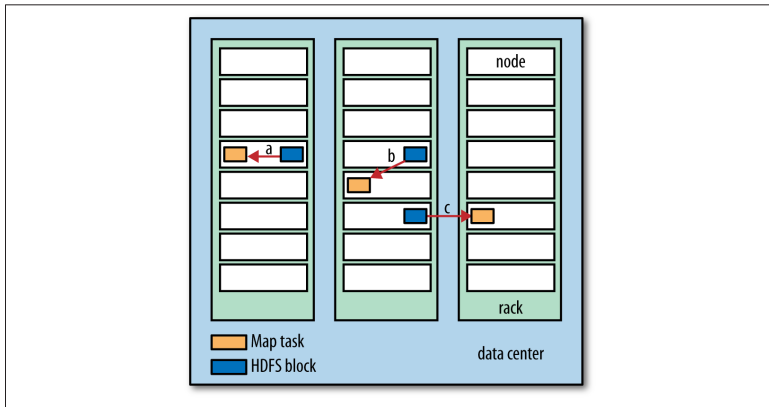
```
$ bin/hdfs dfs -rm <arquivo_no_hdfs>
```

```
$ bin/hdfs dfs -rm -r <diretorio_no_hdfs>
```

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


MapReduce

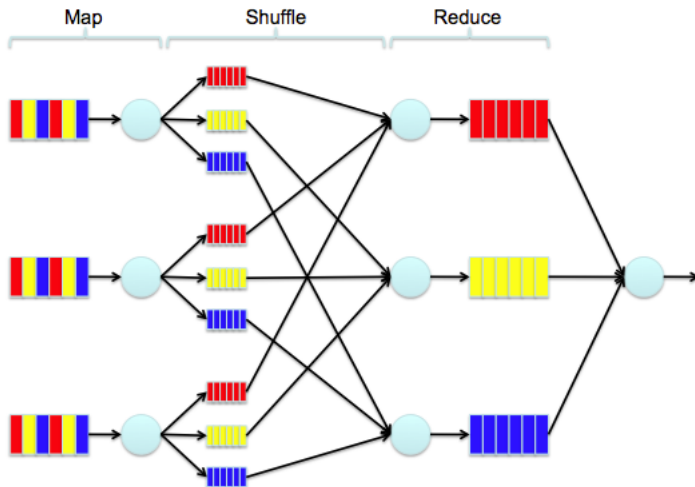
Processamento deve ficar perto dos dados...



Fonte: Hadoop—The Definitive Guide, *Tom White*

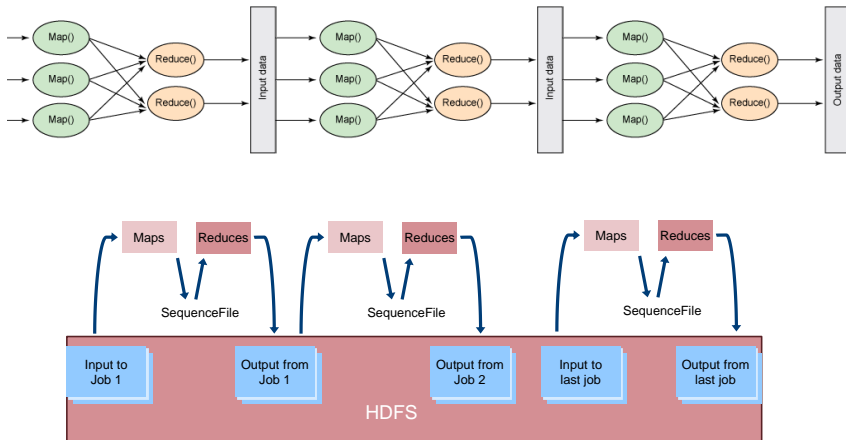
MapReduce

Visão colorida

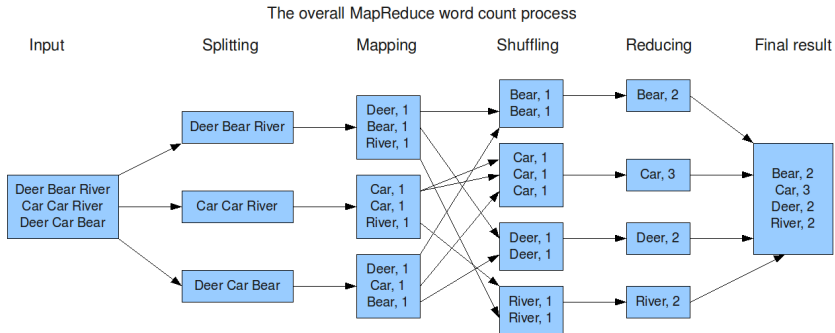


Mapreduce

Várias fases



Word Count



<http://www.cs.uml.edu/~jlu1/doc/source/report/img/MapReduceExample.png>

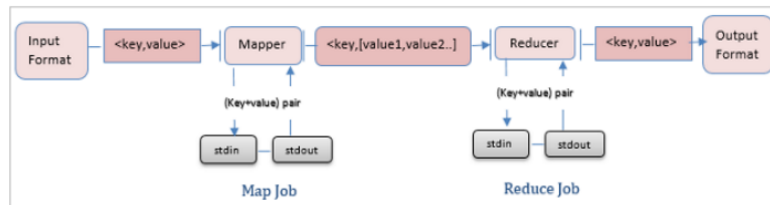
Testando o MapReduce

Pacote de exemplos prontos.

```
$ bin/hadoop dfs -put <dir_local_input> <dir_hdfs_input>
$ bin/hadoop jar \
  share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.4.jar \
  wordcount <dir_hdfs_input> <dir_hdfs_output>
$ bin/hadoop dfs -get <dir_hdfs_output> <dir_local_output>
```


Hadoop Streaming

```
cat input.txt | ./mapper.py | sort | ./reducer.py > output.txt
```



Fonte: <https://acadgild.com/blog/writing-mapreduce-in-python-using-hadoop-streaming/>

```
$ bin/hdfs dfs -put input /input
$ bin/hadoop jar \
  share/hadoop/tools/lib/hadoop-streaming-2.8.4.jar \
  -mapper ../wc-python/mapper.py \
  -reducer ../wc-python/reducer.py \
  -input input.txt -output output
```

Experimento

Familiarização com o ambiente Hadoop

Descrição detalhada em <http://www.ic.unicamp.br/~islene/2018-inf550/explorando-mapreduce.html>

- ▶ Parte fixa: (i) instanciar máquina virtual com o Hadoop (ii) testar HDFS e a Hadoop Streaming com Python.
- ▶ Parte livre: procurar um tema para a base de dados (futebol, música, etc), fazer uma pequena alteração no mapper e/ou reducer.
- ▶ Escrever um relatório sobre o experimento, comentando resultados e eventuais dificuldades encontradas.
- ▶ O trabalho pode ser feito em duplas; apenas uma pessoa precisa entregá-lo via Moodle.
- ▶ Em caso de fraude, poderá ser atribuída nota zero à disciplina.

- ▶ MapReduce
 - ▶ Grande revolução
 - ▶ Pontos fracos foram surgindo
- ▶ Spark: busca por melhor desempenho
- ▶ Necessidade de camadas mais altas de abstração

Principais referências

- ▶ Projeto Apache Hadoop
- ▶ Hadoop: The Definitive Guide, Tom White, 4th Edition, O'Reilly Media