

BIG DATA



Disciplina: Aplicações de Big Data com Hadoop

Tema da Aula: Introdução ao MapReduce

Coordenação:

Prof. Dr. Adolpho Walter
Pimazzi Canton

Profa. Dra. Alessandra de
Ávila Montini

Profa. Rosangela de Fátima Pereira

Junho de 2016

Curriculum

Formação

- Mestrado em Engenharia de Computação pela Escola Politécnica da Universidade de São Paulo (Poli-USP) ([em andamento](#))
- Especialização em Tecnologia Java pela Universidade Tecnológica Federal do Paraná (UTFPR) (2011)
- Tecnologia em Análise e Desenvolvimento de Sistemas pela UTFPR ([2011](#))
- Bacharelado em Administração de Empresas pela Universidade Estadual do Norte do Paraná (UENP) (2007)

Experiência

- Professora de Big Data Analytics em empresas e programas de MBA - FIA ([2013 - atual](#))
- Pesquisadora no Laboratório de Arquitetura e Redes de Computadores (LARC) – USP ([2013 - atual](#))
- Professora de cursos de engenharia na UTFPR ([2011 -2012](#))
- Analista de sistemas na BSI Tecnologia ([2009-2010](#))

LinkedIn: <https://br.linkedin.com/pub/rosangela-de-fatima-pereira/68/a10/b56>

Apaixonada por Big Data!

Objetivo da Aula

Apresentar ao aluno as características, arquitetura e funcionalidades do MapReduce

Apresentar exemplos práticos da implementação e execução de uma aplicação MapReduce

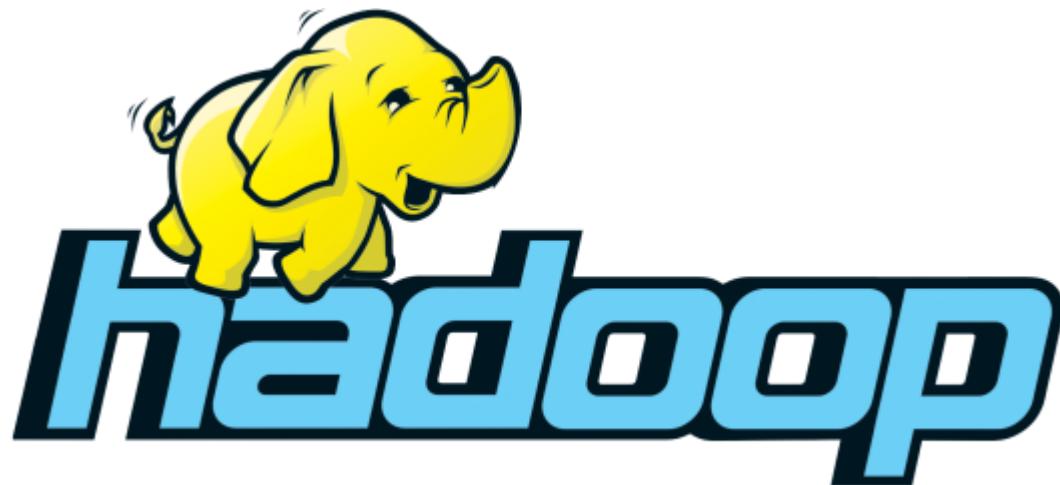
Conteúdo da Aula

- Revisão da aula anterior
- MapReduce
 - Características
 - Arquitetura MapReduce
 - Modelo de Programação MapReduce
- MapReduce na Prática
- Considerações finais

Conteúdo da Aula

- Revisão da aula anterior
- MapReduce
 - Características
 - Arquitetura MapReduce
 - Modelo de Programação MapReduce
- MapReduce na Prática
- Considerações finais

Aula anterior



Aula anterior

Arcabouço de software **open source** que permite a execução de aplicações utilizando milhares de máquinas

Oferece recursos de armazenamento, gerenciamento e processamento **distribuído** de dados (**SO de Big Data**)

Projetado para **processamento em lote** de grandes conjuntos de dados

Um dos **pioneiros** da geração de tecnologias de Big Data

Aula anterior

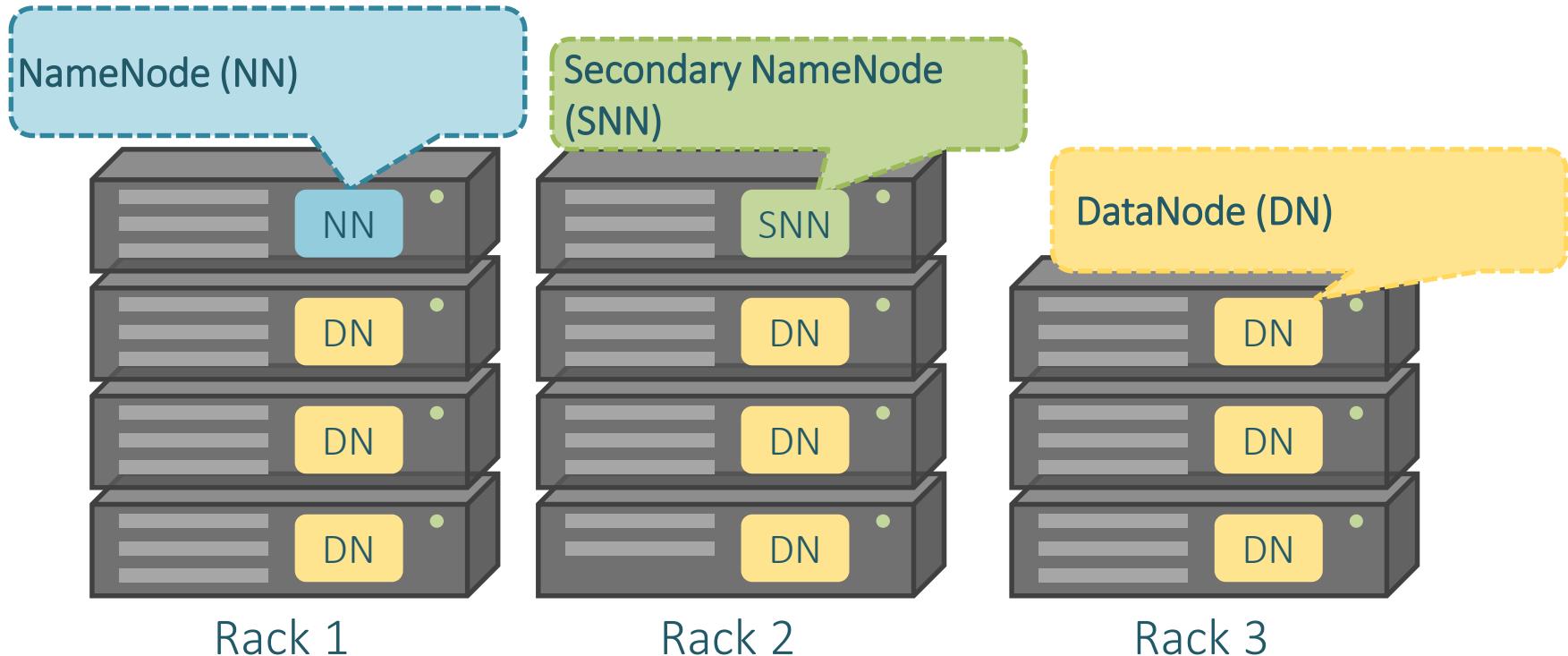


Aula anterior

HADOOP DISTRIBUTED FILE SYSTEM - HDFS

- Sistema de arquivos distribuído
- Executado em um sistema de arquivos nativo
- Otimizado para processamento de grande volume de dados (alta taxa de transferência)
- Abstrai questões de armazenamento distribuído dos dados
- Escalável e tolerante a falhas

Aula anterior



Conteúdo da Aula

- Revisão da aula anterior
- **MapReduce**
 - Características
 - Arquitetura MapReduce
 - Modelo de Programação MapReduce
- MapReduce na Prática
- Considerações finais

MapReduce - MR



MR - Características

O que é necessário implementar em uma aplicação distribuída?

MR - Características

Escalabilidade

Tolerância a falhas

Comunicação entre máquinas

Balanceamento de carga

Escalonamento de tarefas

Alocação de máquinas

Lógica do problema

Implementado pelo desenvolvedor

Implementado pelo MapReduce

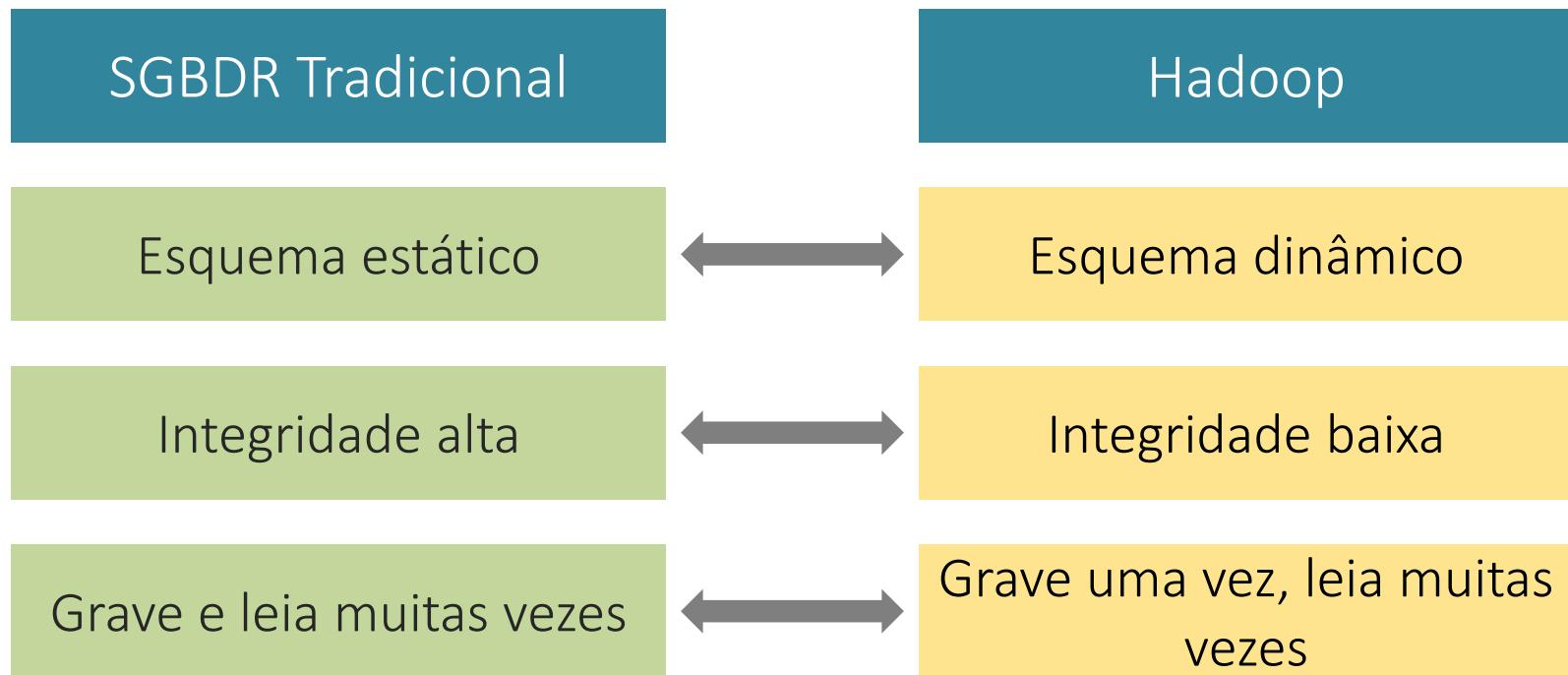
MR - Características

Arcabouço de software para facilitar a execução de aplicações que processam um grande volume de dados em um cluster de milhares de nós de hardware convencional de maneira tolerante a falhas.

MR - Características

1. Paradigma de programação distribuída
2. Engine de execução de aplicações distribuídas
3. Implementado em Java
4. Executa programas implementados em Java, Python, Ruby e C++

Hadoop MR vs SGBDR



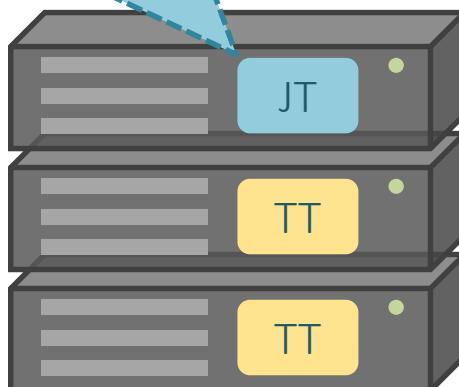
Conteúdo da Aula

- Revisão da aula anterior
- **MapReduce**
 - Características
 - Arquitetura MapReduce
 - Modelo de Programação MapReduce
- MapReduce na Prática
- Considerações finais

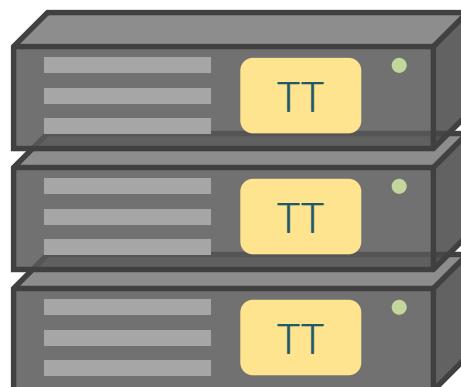
MR - Arquitetura

JobTracker (JT)

- Nô mestre
- Gerenciador de tarefas MapReduce



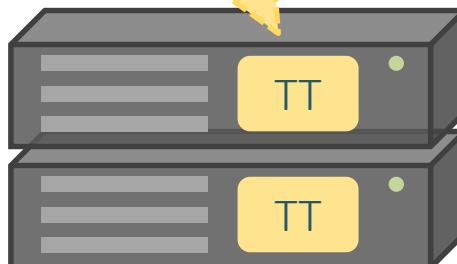
Rack 1



Rack 2

TaskTracker (TT)

- Executam as tarefas MapReduce



Rack 3

MR - Arquitetura

JOB

Um programa completo

Uma aplicação

MR - Arquitetura

TAREFA

Execução de um Mapper ou Reducer sobre uma fatia dos dados

MR - Arquitetura

Mova as tarefas, não os dados

1. Uma aplicação cliente submete um job ao JobTracker
2. JobTracker se comunica com o NameNode para determinar a localização dos dados
3. JobTracker localiza os nós TaskTrackers **próximos aos dados**
4. JobTracker submete as tarefas aos nós TaskTrackers

MR - Arquitetura

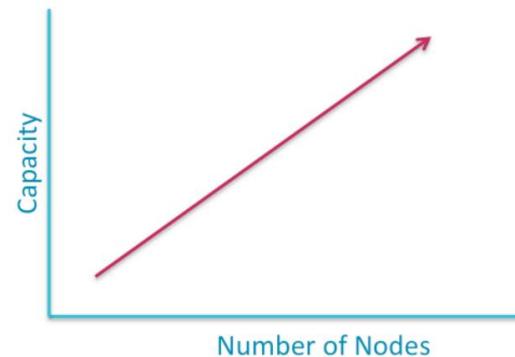
Tolerância a falhas

1. Em um grande cluster, as máquinas ficam lentas ou falham com frequência
2. MapReduce oferece **recuperação automática** de tarefas que falharam
3. MapReduce redireciona tarefas falhas para outros nós do cluster

MR - Arquitetura

Escalabilidade linear

“Escalabilidade é a capacidade do sistema de **manter o desempenho** com o aumento de carga, pela adição de mais recursos”



Conteúdo da Aula

- Revisão da aula anterior
- **MapReduce**
 - Características
 - Arquitetura MapReduce
 - **Modelo de Programação MapReduce**
- MapReduce na Prática
- Considerações finais

MR – Modelo de programação

Map

Atua **exclusivamente** sobre um conjunto de entrada com chaves e valores, produzindo uma lista de chaves e valores

Reduce

Atua sobre os valores intermediários produzidos pelo map para, normalmente, agrupar os valores e produzir uma saída

MR – Modelo de programação

Exemplo tradicional do MapReduce: WordCount

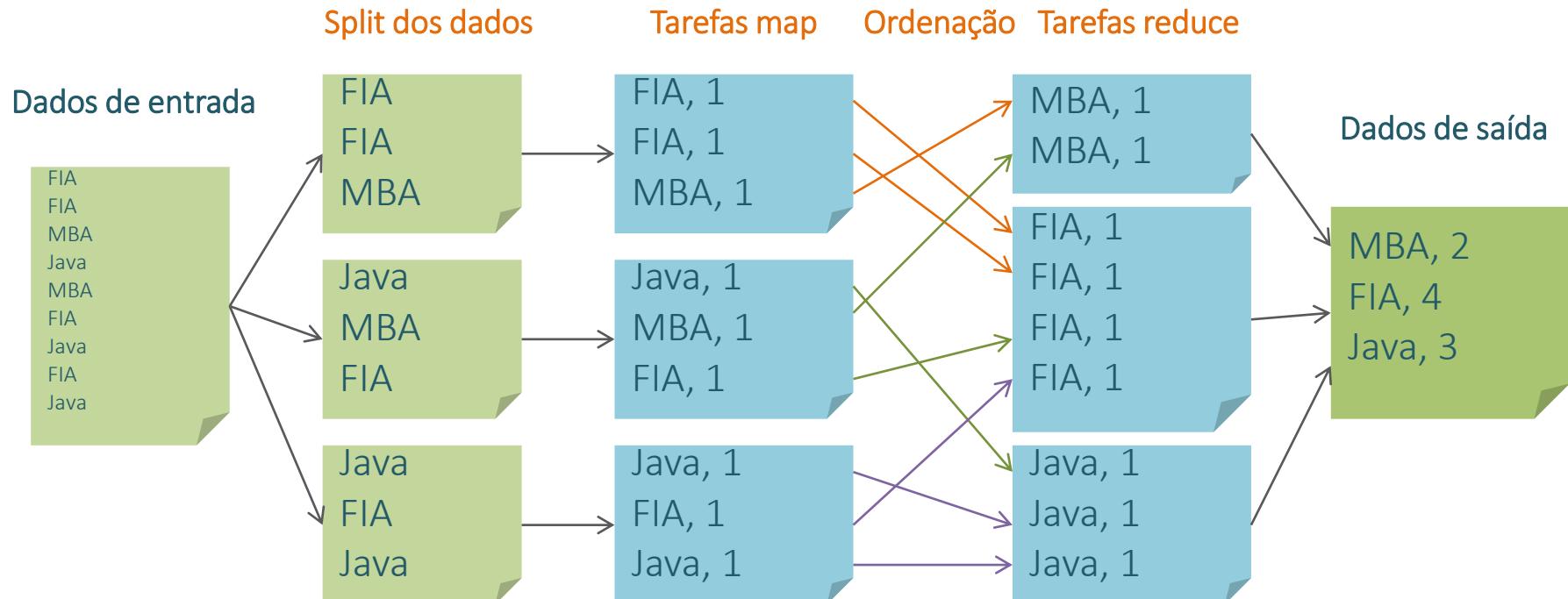
Dados de entrada: um ou mais arquivos de texto

Objetivo: contar o número de ocorrências de cada palavra encontrada no texto

Resultado: arquivo texto contendo uma lista em que cada linha apresenta uma palavra e seu respectivo número de ocorrências

Exemplo de aplicabilidade: *trending topics* do Twitter

MR – Modelo de programação



MR – Modelo de programação

Demo



MR – Modelo de programação

Exemplos de análises utilizando o modelo de programação MapReduce

Mineração de texto

Filtragem
colaborativa

Modelos preditivos

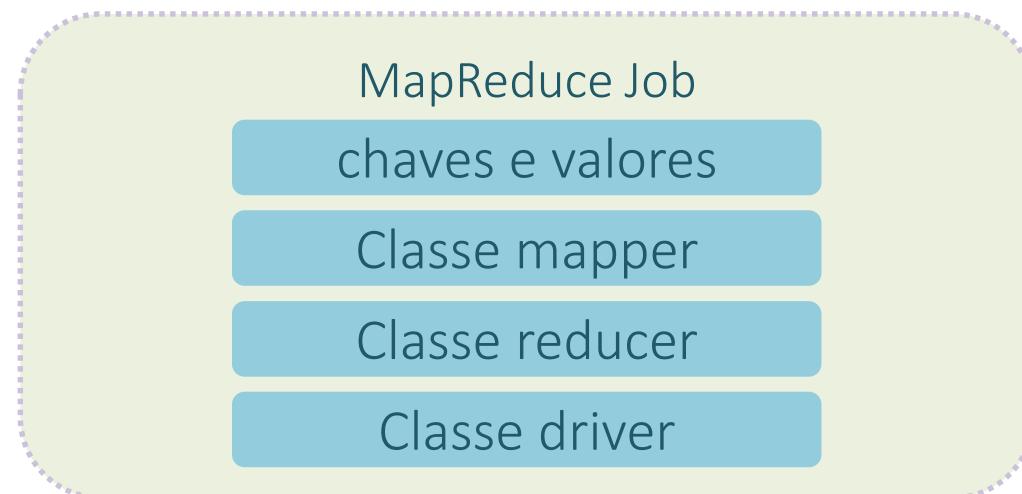
Reconhecimento de
padrões

Análise de
sentimento

Análise de risco

MR – Modelo de programação

Responsabilidade do **desenvolvedor**



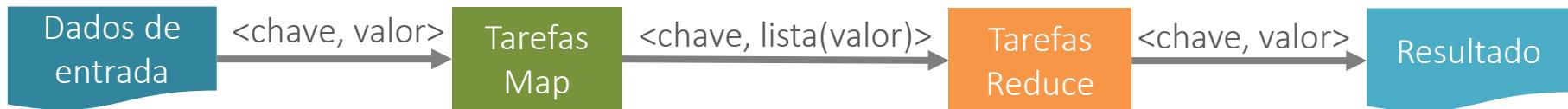
MR – Modelo de programação

Exemplos de chaves e valores

Um número de telefone (**chave**) com os registros das chamadas (**valores**)

Um usuário do facebook (**chave**) e suas informações do perfil (**valores**)

O número de um cartão de crédito (**chave**) e suas transações (**valores**)



Funções Map e Reduce

Classe Java da função
Map

Classe mapper

```
public static class Map extends Mapper <LongWritable, Text, Text, IntWritable> {  
    private final static IntWritable one = new IntWritable(1);  
    private Text word = new Text();  
  
    public void map (LongWritable key, Text value, Context context) {  
        String line = value.toString();  
        StringTokenizer tokenizer = new StringTokenizer(line);  
        while (tokenizer.hasMoreTokens()) {  
            word.set(tokenizer.nextToken());  
            context.write(word, one);  
        }  
    }  
}
```

Tipos de dados da chave e valor
da entrada, chave e valor da
saída

Declaração do método
map

Envio da chave e valor
para o reduce

Funções Map e Reduce

Classe Java da função
Reduce

Classe reducer

Tipos de dados da chave e valor
da entrada, chave e valor da
saída

```
public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
```

```
    public void reduce(Text key, Iterable<IntWritable> values, Context context) {  
        int sum = 0;  
        for (IntWritable val : values) {  
            sum += val.get();  
        }  
        context.write(key, new IntWritable(sum));  
    }  
}
```

Declaração do método
reduce

Envio da chave e valor
para o reduce

Funções Map e Reduce

Classe driver

```
public static void main(String[] args) throws Exception {  
    Configuration conf = new Configuration();  
    Job job = new Job(conf, "wordcount");  
    job.setOutputKeyClass(Text.class);  
    job.setOutputValueClass(IntWritable.class);  
    job.setMapperClass(Map.class);  
    job.setReducerClass(Reduce.class);  
    FileInputFormat.addInputPath(job, new Path(args[0]));  
    FileOutputFormat.setOutputPath(job, new Path(args[1]));  
    job.waitForCompletion(true);  
}
```

Criação do objeto job

Parâmetros de configuração do Job

Indicação do diretório de entrada e de saída

Inicialização do job

MR – Modelo de programação

Como utilizar as classes implementadas?

MR – Modelo de programação

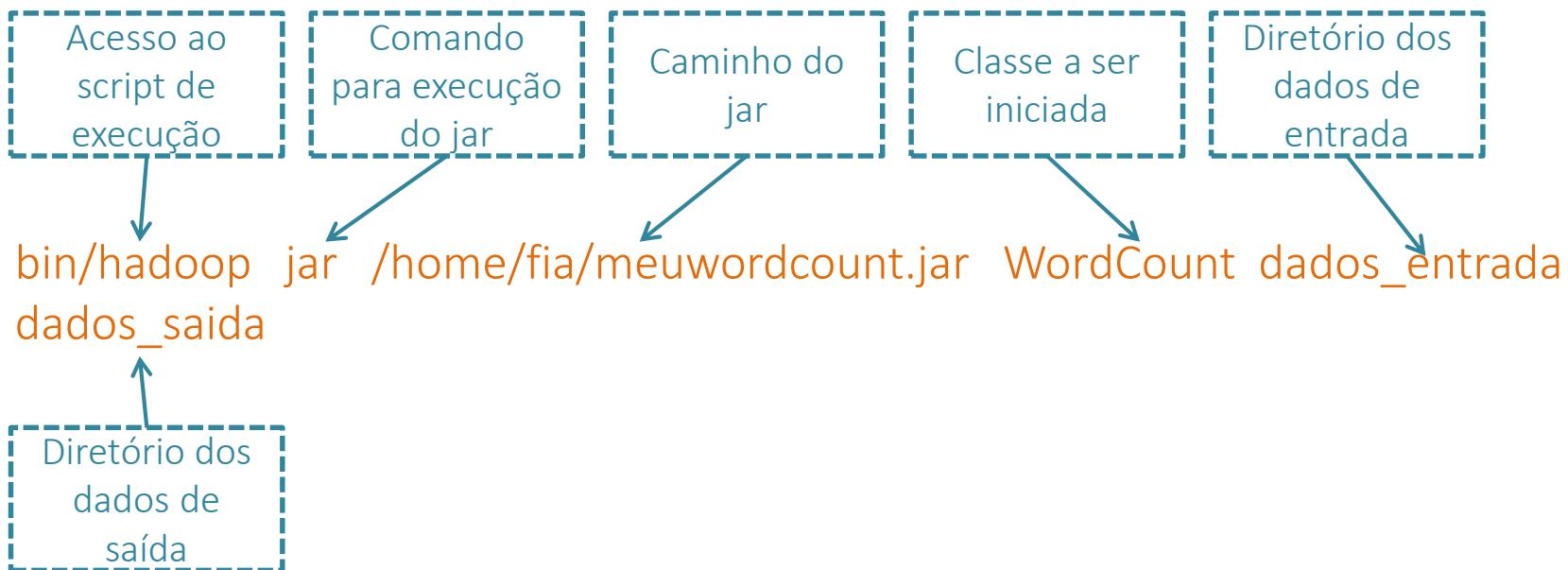
Como utilizar as classes implementadas?

1. Gerar um arquivo JAR com as classes implementadas
2. Submeter o JAR por meio da função “hadoop jar”

JAR (Java ARchive) é um arquivo compactado usado para distribuir um conjunto de classes Java. É usado para armazenar classes compiladas e metadados associados que podem constituir um programa.

MR – Modelo de programação

Chamada de um MapReduce job - Exemplo



MR – Modelo de programação

Mas como gerar uma aplicação MapReduce utilizando outras linguagens de programação?

MR – Modelo de programação

Hadoop Streaming API

- Permite a execução de tarefas map e reduce com diversas linguagens (desde que a linguagem possa ler o input e output padrão)
- A entrada dos dados são recebidas do `stdin`
- A saída dos dados são gravadas no `stdout`
- A execução é oferecida por meio do componente `hadoop-streaming.jar`

MR – Modelo de programação

Hadoop Streaming API – Exemplo

```
bin/hadoop jar /usr/local/hadoop/hadoop-streaming.jar \
    -input "/user/dados_entrada.txt" \
    -output "/user/out" \
    -mapper "meuMapper.py" \
    -reducer "meuReducer.py"
```

Conteúdo da Aula

- Revisão da aula anterior
- MapReduce
 - Características
 - Arquitetura MapReduce
 - Modelo de Programação MapReduce
- **MapReduce na prática**
- Considerações finais

MapReduce na Prática

Software: [VMware](#)

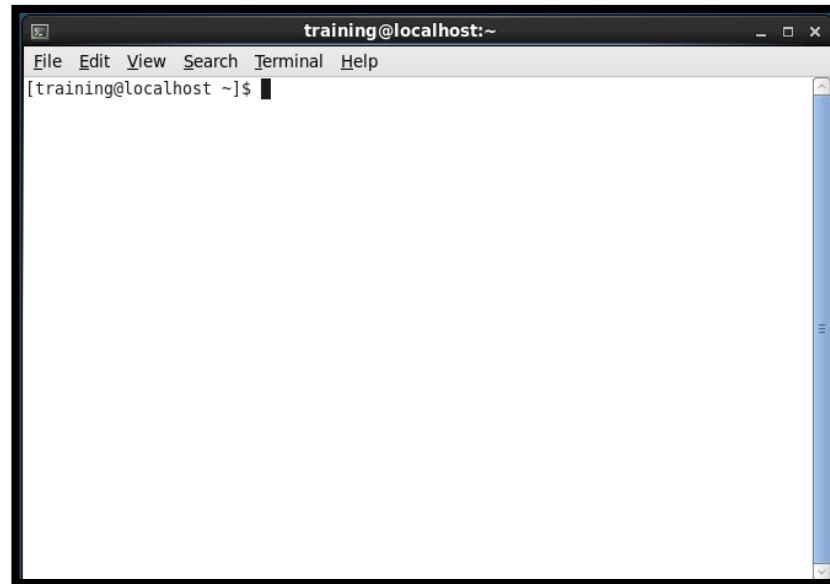
Máquina virtual: [Cloudera-training-analyst](#)



cloudera

MapReduce na Prática

Abrir um terminal



MapReduce na Prática

Executando uma aplicação MapReduce em Java

MapReduce na Prática

Base de dados:

Mensagens da rede social Twitter sobre os temas: Big Data, data visualization, data privacy e Internet of Things

Exemplo

How can data visualization be used in the sales process? Learn how it can yield great insights. <http://t.co/Lq2lcpe7d1>

RT @Mona_Mourshed: "When Big Data Meets the Blackboard" <http://t.co/f3lr4yQW9A> via @TheAtlantic

Big data: reduce privacy risks - REPUTATION PROTECT <http://t.co/DrAb6LzFJ>

Absolute privacy in handling #mhealth data. Patient security comes first."

RT @jose_garde: RT @MarshaCollier The Internet of Things and the Currency of Privacy <http://t.co/eXwmluX8sL>

MapReduce na Prática

Serão executadas 3 aplicações:

1. WordCount: conta palavras em uma base de dados do twitter

MapReduce na Prática

Serão executadas 3 aplicações:

1. WordCount: conta palavras em uma base de dados do twitter
2. ContadorHashTag: conta a quantidade de *hashtags* na base de dados

MapReduce na Prática

Serão executadas 3 aplicações:

1. WordCount: conta palavras em uma base de dados do twitter
2. ContadorHashTag: conta a quantidade de *hashtags* na base de dados
3. TopNHashTag: filtra as top n *hashtags* mais citadas

MapReduce na Prática

Verificar os processos do MapReduce

```
[training@localhost ~]$ sudo jps
```

```
8452 JobTracker
```

```
1951 TaskTracker
```

```
3817 Jps
```

```
2015 DataNode
```

```
8849 NameNode
```

MapReduce na Prática

Submeter base de dados para o HDFS

```
#criar diretório
```

```
[training@localhost ~]$ hadoop fs -mkdir bases_dados
```

MapReduce na Prática

Submeter base de dados para o HDFS

```
#submeter arquivo para o diretório
```

```
[training@localhost ~]$ hadoop fs -put ~/bases/base_tw.txt bases_dados
```

MapReduce na Prática

Submeter base de dados para o HDFS

```
#visualizar conteúdo do diretório
```

```
[training@localhost ~]$ hadoop fs -ls bases_dados
```

```
Found 1 items
```

```
-rw-rw-rw- 1 training supergroup 982302 2016-04-05 14:59  
bases_dados/base_tw.txt
```

MapReduce na Prática

Acessar a interface WEB do HDFS

#acessar a interface web do HDFS e do MapReduce

Abrir o navegador Firefox

Selecionar a opção **JobTracker**

MapReduce na Prática

Execução do WordCount

```
[training@localhost ~]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount bases_dados saida/resultado1
16/04/06 09:09:06 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
16/04/06 09:09:07 INFO input.FileInputFormat: Total input paths to process : 1
16/04/06 09:09:07 INFO mapred.JobClient: Running job: job_201602181111_0016
16/04/06 09:09:08 INFO mapred.JobClient: map 0% reduce 0%
16/04/06 09:09:19 INFO mapred.JobClient: map 100% reduce 0%
16/04/06 09:09:25 INFO mapred.JobClient: map 100% reduce 100%
16/04/06 09:09:28 INFO mapred.JobClient: Job complete: job_201602181111_0016
...
```

MapReduce na Prática

Visualizar resultado de execução do WordCount

```
[training@localhost ~]$ hadoop fs -ls saida/resultado1
```

Found 3 items

-rw-rw-rw-	1	training	supergroup	0	2016-04-06 09:09	saida/resultado1/_SUCCESS
drwxrwxrwx	-	training	supergroup	0	2016-04-06 09:09	saida/resultado1/_logs
-rw-rw-rw-	1	training	supergroup	187407	2016-04-06 09:09	saida/resultado1/part-r-00000

MapReduce na Prática

Visualizar resultado de execução do WordCount

```
[training@localhost ~]$ hadoop fs -cat saida/resultado1/part-r-00000
threaten    4
threatens   4
three       2
thriller    5
thrive      2
```

MapReduce na Prática

Execução da aplicação ContadorHashTag

```
[training@localhost ~]$ hadoop jar ~/bases/LabHadoop.jar  
com.hadoop.ContadorHashTag bases_dados saida/resultado2
```

```
16/04/06 09:13:11 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement  
Tool for the same.
```

```
16/04/06 09:13:11 INFO input.FileInputFormat: Total input paths to process : 1  
16/04/06 09:13:12 INFO mapred.JobClient: Running job: job_201602181111_0017  
16/04/06 09:13:13 INFO mapred.JobClient: map 0% reduce 0%  
16/04/06 09:13:23 INFO mapred.JobClient: map 100% reduce 0%  
16/04/06 09:13:28 INFO mapred.JobClient: map 100% reduce 100%  
16/04/06 09:13:32 INFO mapred.JobClient: Job complete: job_201602181111_0017  
16/04/06 09:13:32 INFO mapred.JobClient: Counters: 32
```

```
...
```

MapReduce na Prática

Visualizar resultado de execução da aplicação ContadorHashTag

```
[training@localhost ~]$ hadoop fs -ls saida/resultado2
Found 3 items
-rw-rw-rw- 1 training supergroup          0 2016-04-06 09:13 saida/resultado2/_SUCCESS
drwxrwxrwx - training supergroup          0 2016-04-06 09:13 saida/resultado2/_logs
-rw-rw-rw- 1 training supergroup 12511 2016-04-06 09:13 saida/resultado2/part-r-00000
```

MapReduce na Prática

Visualizar resultado de execução da aplicação ContadorHashTag

```
training@localhost ~]$ hadoop fs -text saida/resultado2/part-r-00000
#AI      7
#AMC2015 1
#API     36
#APR360   2
#ARM15    9
#AdobeEDEX      1
#Adwords   4
```

MapReduce na Prática

Execução da aplicação TopNHashTag

```
[training@localhost ~]$ hadoop jar ~/bases/LabHadoop.jar  
com.hadoop.TopNHashTag saida/resultado2/part-r-00000  
saida/resultado3 10
```

Obs: A base de dados de entrada desse job é a base de saída do job anterior

MapReduce na Prática

Visualizar resultado de execução da aplicação TopNHashTag

```
[training@localhost ~]$ hadoop fs -ls saida/resultado3
Found 3 items
-rw-rw-rw- 1 training supergroup      0 2016-04-06 09:18 saida/resultado3/_SUCCESS
drwxrwxrwx - training supergroup      0 2016-04-06 09:18 saida/resultado3/_logs
-rw-rw-rw- 1 training supergroup 139 2016-04-06 09:18 saida/resultado3/part-r-00000
```

MapReduce na Prática

Visualizar resultado de execução da aplicação TopNHashTag

```
[training@localhost ~]$ hadoop fs -text saida/resultado3/part-r-00000
#DataScience          1454
#BigData      1038
#datascience      834
#Analytics      488
#dataviz        417
#bigdata        402
#data          391
#privacy        382
#analytics      255
```

MapReduce na Prática

Pergunta: quais outras aplicações poderiam ser feitas a partir dessa base de dados?

MapReduce na Prática

Pergunta: quais outras aplicações poderiam ser feitas a partir dessa base de dados?

- Análise de sentimento
- Identificação de pandemias
- Comparação entre marcas
- Identificação de usuários influentes

Conteúdo da Aula

- Revisão da aula anterior
- MapReduce
 - Características
 - Arquitetura MapReduce
 - Modelo de Programação MapReduce
- MapReduce na prática
- Considerações finais

MR – Considerações finais

Em quais casos MapReduce não é a melhor escolha?

- Consultas que necessitam de baixa latência
 - Exemplo:
 - Sistemas de tempo-real
 - Consultas em um website
- Processamento de pequenas tarefas
 - Overhead para gerenciamento das tarefas

MR - Considerações finais

O Hadoop...

... é um framework **open source** desenvolvido em Java

MR - Considerações finais

O Hadoop...

... é um framework **open source** desenvolvido em Java

... é projetado para manipular **grande volume** de dados

MR - Considerações finais

O Hadoop...

- ... é um framework **open source** desenvolvido em Java
- ... é projetado para manipular **grande volume de dados**
- ... é projetado para ser **escalável** em milhares de máquinas

MR - Considerações finais

O Hadoop...

- ... é um framework **open source** desenvolvido em Java
- ... é projetado para manipular **grande volume de dados**
- ... é projetado para ser **escalável** em milhares de máquinas
- ... é projetado para ser executado em hardware de **baixo custo**

MR - Considerações finais

O Hadoop...

- ... é um framework **open source** desenvolvido em Java
- ... é projetado para manipular **grande volume de dados**
- ... é projetado para ser **escalável** em milhares de máquinas
- ... é projetado para ser executado em hardware de **baixo custo**
- ... oferece resiliência por meio da **replicação** de dados

MR - Considerações finais

O Hadoop...

- ... é um framework **open source** desenvolvido em Java
- ... é projetado para manipular **grande volume de dados**
- ... é projetado para ser **escalável** em milhares de máquinas
- ... é projetado para ser executado em hardware de **baixo custo**
- ... oferece resiliência por meio da **replicação** de dados
- ... oferece **recuperação automática** do processo em caso de falha

MR - Considerações finais

O Hadoop...

- ... é um framework **open source** desenvolvido em Java
- ... é projetado para manipular **grande volume de dados**
- ... é projetado para ser **escalável** em milhares de máquinas
- ... é projetado para ser executado em hardware de **baixo custo**
- ... oferece resiliência por meio da **replicação** de dados
- ... oferece **recuperação automática** do processo em caso de falha
- ... faz **distribuição** automática dos dados no cluster

MR - Considerações finais

O Hadoop...

- ... é um framework **open source** desenvolvido em Java
- ... é projetado para manipular **grande volume de dados**
- ... é projetado para ser **escalável** em milhares de máquinas
- ... é projetado para ser executado em hardware de **baixo custo**
- ... oferece resiliência por meio da **replicação** de dados
- ... oferece **recuperação automática** do processo em caso de falha
- ... faz **distribuição** automática dos dados no cluster
- ... projetado para **levar o processamento** para o dado

MR - Considerações finais

Dicas para adoção do Hadoop

- Faça uma expansão do cluster na medida em que novas aplicações de interesse forem identificadas
- Construa uma plataforma centralizada dos dados
- Propague o conhecimento de Hadoop e NoSQL com outros grupos
- Dê ao seu time tempo para fazer experimentos
- Planeje a existência de aplicações novas, tradicionais e legadas

MR - Considerações finais

Sugestão de leitura

Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades

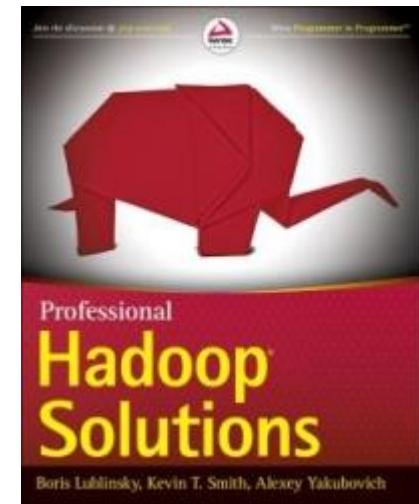
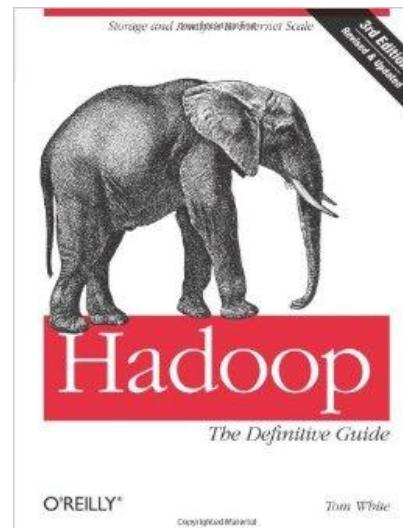
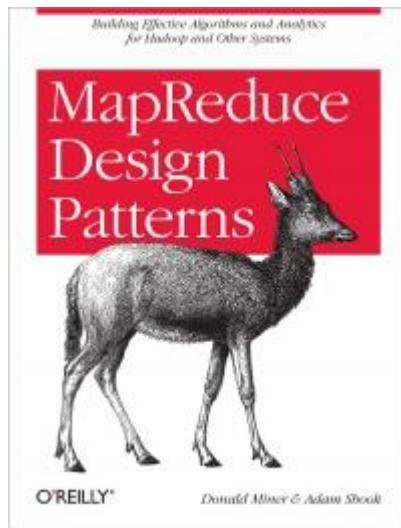
Alfredo Goldman, Fabio Kon, Francisco Pereira Junior,
Ivanilton Polato, Rosangela de Fátima Pereira

XXXI JAI

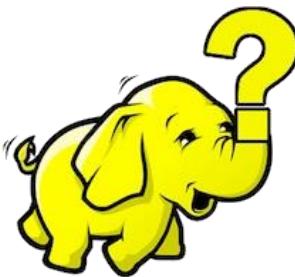
<http://www.ime.usp.br/~ipolato/JAI2012-Hadoop.pdf>

MR - Considerações finais

Indicações de livros sobre Hadoop



Perguntas



rpereira@larc.usp.br

Referências Bibliográficas

WHITE, Tom. **Hadoop: The definitive guide.** " O'Reilly Media, Inc.", 2012.

DEAN, Jeffrey; GHEMAWAT, Sanjay. **MapReduce: simplified data processing on large clusters.** Communications of the ACM, v. 51, n. 1, p. 107-113, 2008.

GHEMAWAT, Sanjay; GOBIOFF, Howard; LEUNG, Shun-Tak. **The Google file system.** In: ACM SIGOPS Operating Systems Review. ACM, 2003. p. 29-43.

VAVILAPALLI, Vinod Kumar et al. **Apache hadoop yarn: Yet another resource negotiator.** In: Proceedings of the 4th annual Symposium on Cloud Computing. ACM, 2013. p. 5.

SHVACHKO, Konstantin et al. **The hadoop distributed file system.** In: Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010. p. 1-10.

Referências Bibliográficas

- GOLDMAN, Alfredo et al. **Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades**. XXXI Jornadas de atualizações em informática, p. 88-136, 2012.
- VENNER, Jason. **Pro Hadoop**. Apress, 2009.