

Processamento paralelo é distribuído com Big Data

Dr. Kleber Vieira



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Big Data

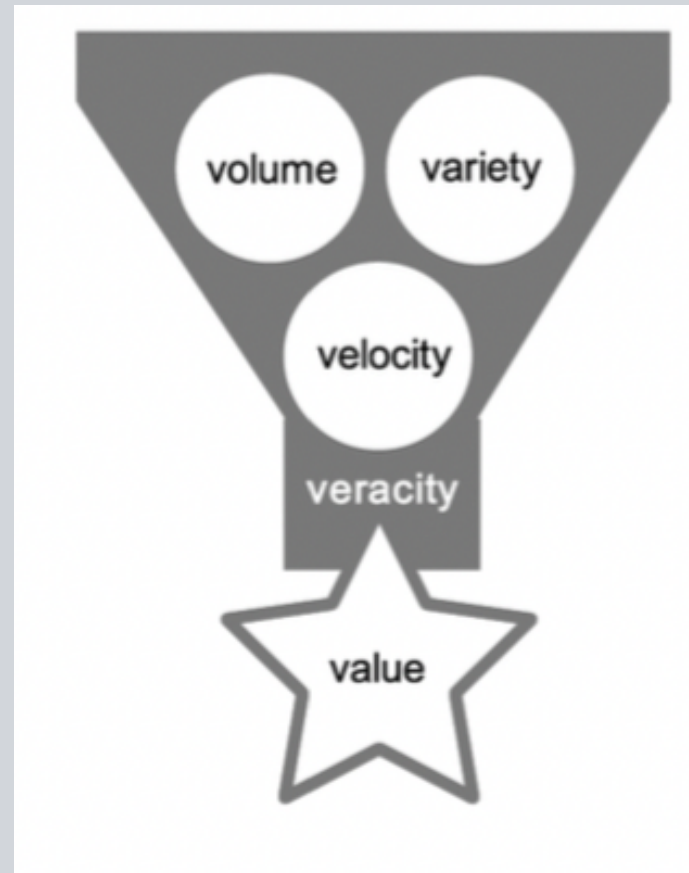
O que é Big Data?

Porque precisamos de novas técnicas ?

Big Data

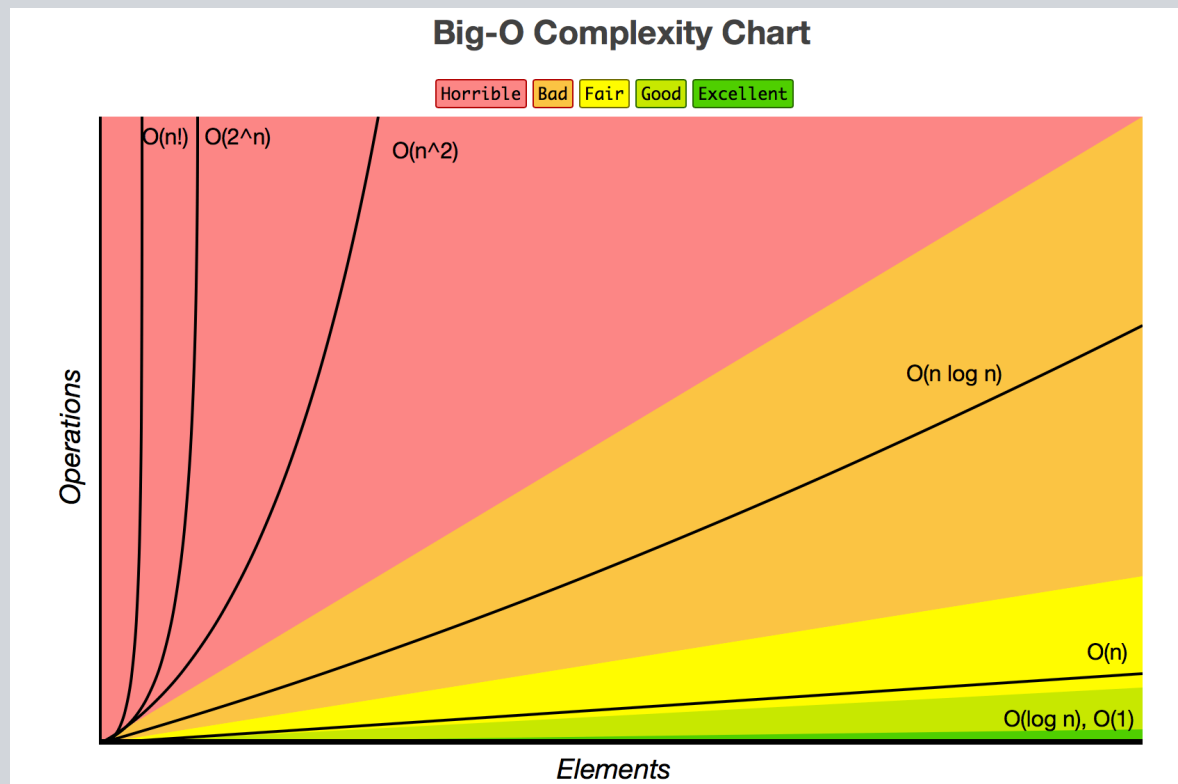
5 características do Big Data segundo Khan, Uddin e Gupta (2014).

- Volume
- Variedade
- Velocidade
- Veracidade
- Valor



Big Data

Porque precisamos de novas técnicas ?
Complexidade computacional.



Problemas com Big Data

Exemplo:

- IoT na agricultura;
 - Cada maquina gera 25 registros de sensores por segundo;
 - Ligada 12 horas por dias envia 43.200 vezes 25 1.080.000
 - Mil Dispositivos gera mais de 1 bilhão de registros.

Problemas com Big Data

Exemplo:

- Para cada segundo é enviada a coordenada GPS, dados de coleta, insumos, velocidade, operação executava etc...

Problemas com Big Data

Exemplo:

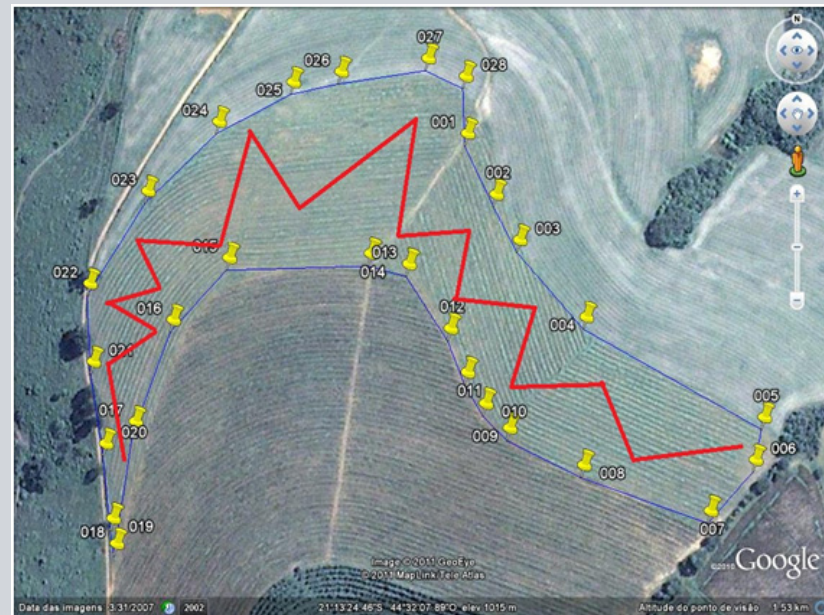
- Ao receber os dados o é necessário:
- Verificar em qual talhão



Problemas com Big Data

Exemplo:

- Ao receber os dados o é necessário:
- Verificar em qual talhão foi trabalhado.
- Quais atividades foram realizada.
- Consolidar os identificados dos registros enviados com os dados do banco.



Problemas com Big Data

Exemplo:

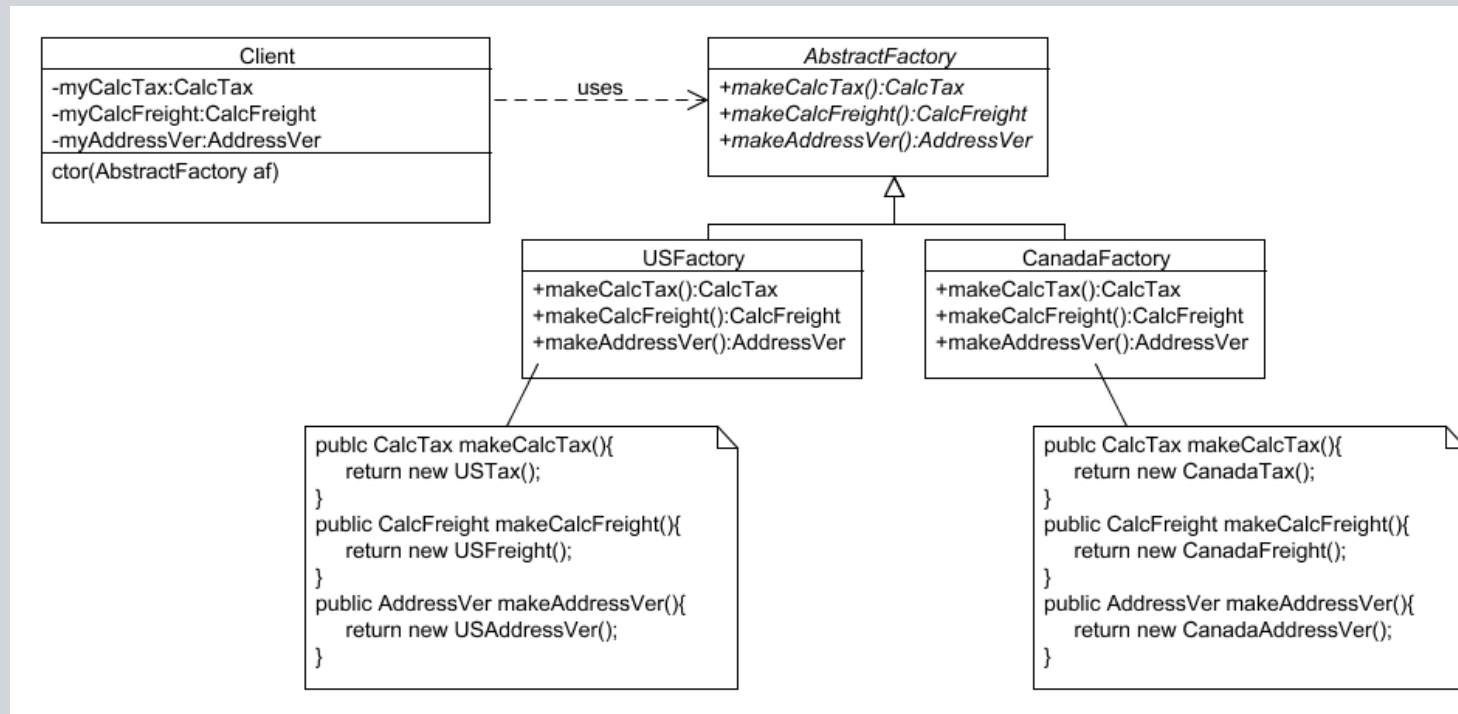
- Reduzir os dados para que consultas extração de dados sejam geradas rapidamente.
- Se em 1 minuto não houve mudança de atividade agrupa os 60 registros em apenas 1.

Como organizar o código para tratar todo esse processamento ?

Design Pattern:

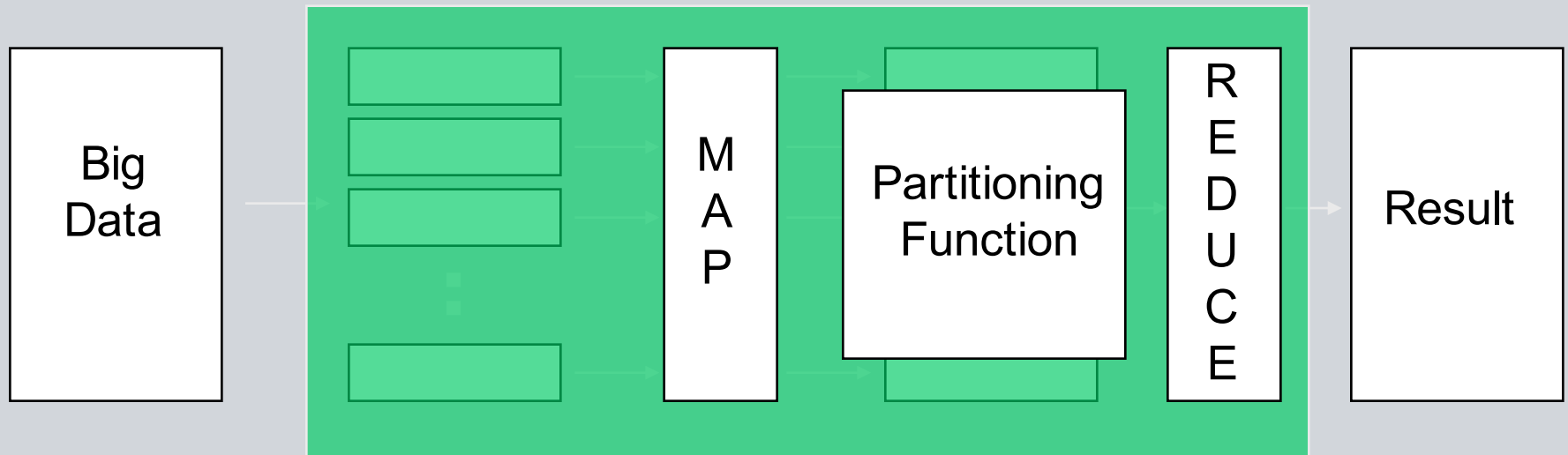
Flyweight

<https://github.com/iluwatar/java-design-patterns/tree/master/flyweight/src/main/java/com/iluwatar/flyweight>



Map Reduce

- Mapear os dados em estrutura de dados;
- Reduzir;

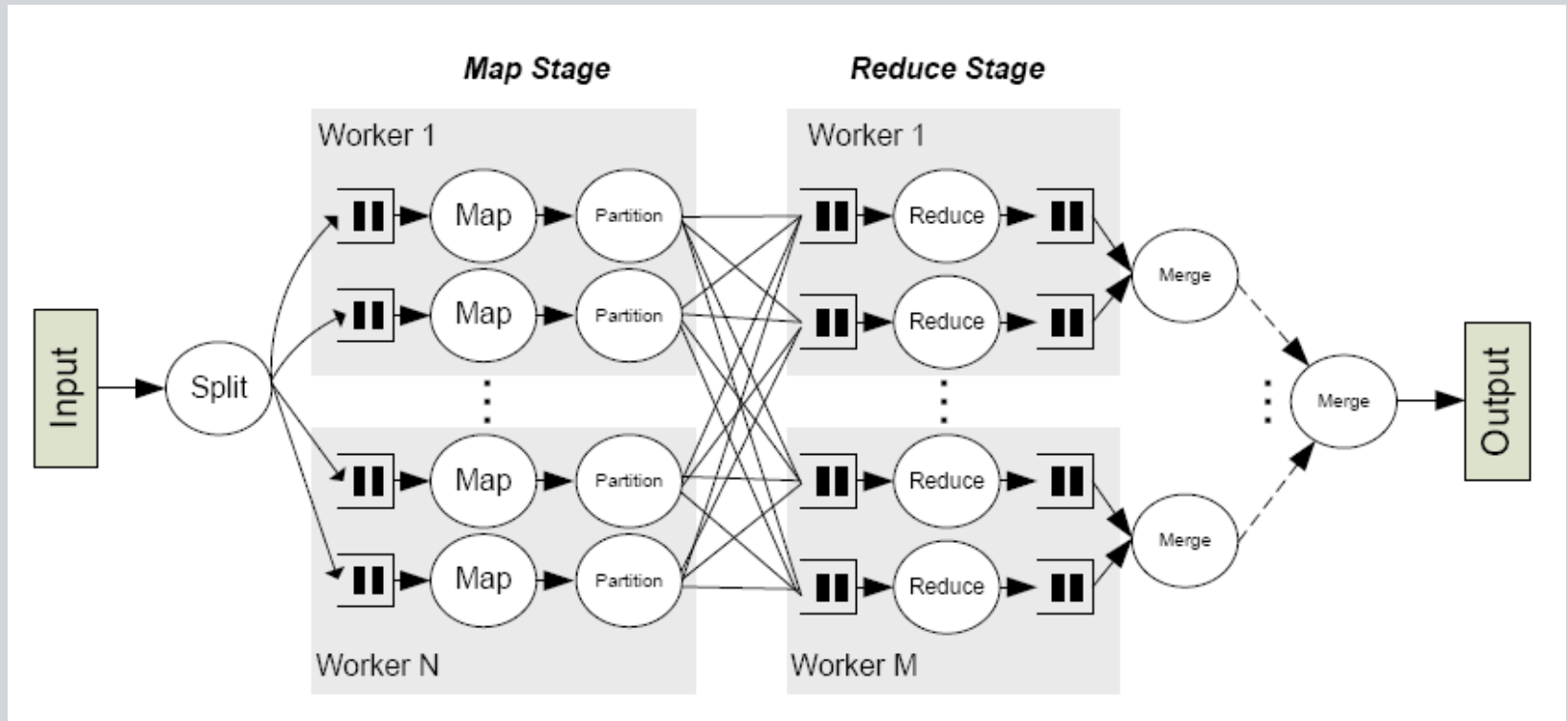


Ferramenta Hadoop

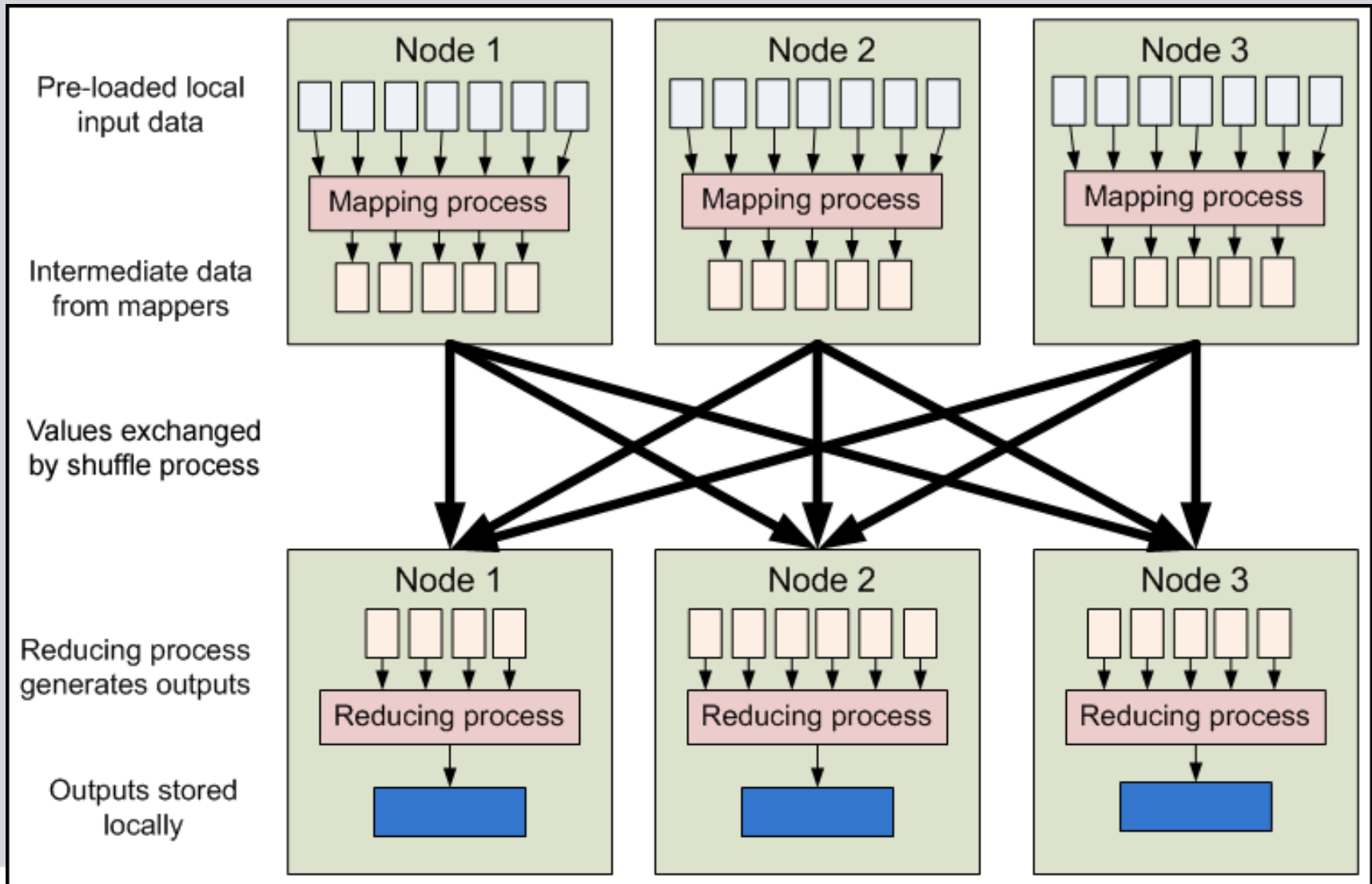
Google desenvolveu o Hadoop;

Framework para realizar o processamento Big Data em ambientes distribuídos.

Ferramenta Hadoop



Ferramenta Hadoop



Ferramenta Hadoop

Como usar ?

Instalar o Hadoop:

<http://hadoop.apache.org/releases.html>

Apache Hadoop Releases

Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-256.

Version	Release Date	Tarball	PGP	SHA-256
3.0.0-alpha1	03 September, 2016	source	signature	checksum file
2.7.3	25 August, 2016	binary	signature	checksum file
		source	signature	227785DC 6E3E6FF8..
2.6.4	11 February, 2016	binary	signature	D489DF38 08244B90..
		source	signature	F755D961 18216335..
2.5.2	19 Nov, 2014	binary	signature	C58F08D2 E0813035..
		source	signature	139E8272 09C5637E..
		binary	signature	09DB4650 A3825208..

To verify Hadoop releases using GPG:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the signature file `hadoop-X.Y.Z-src.tar.gz.asc` from [Apache](#).
3. Download the [Hadoop KEYS](#) file.
4. `gpg --import KEYS`
5. `gpg --verify hadoop-X.Y.Z-src.tar.gz.asc`

To perform a quick check using SHA-256:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the checksum `hadoop-X.Y.Z-src.tar.gz.md5` from [Apache](#).
3. `shasum -a 256 hadoop-X.Y.Z-src.tar.gz`

All previous releases of Hadoop are available from the [Apache release archive](#) site.

Many third parties distribute products that include Apache Hadoop and related tools. Some of these are listed on the [Distributions wiki page](#).

Home » Dyn About Projects People Get Involved Download Support Apache

THE APACHE SOFTWARE FOUNDATION

We suggest the following mirror site for your download:

<http://apache.mirrors.tds.net/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

Other mirror sites are suggested below. Please use the backup mirrors only to download PGP and MD5 signatures to verify your downloads or if no other mirrors are working.

HTTP¶

<http://apache.claz.org/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://apache.cs.utah.edu/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://apache.mesi.com.ar/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://apache.mirrors.hoobly.com/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://apache.mirrors.lucidnetworks.net/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

Ferramenta Hadoop

Como usar ?

Instalar o Hadoop:

<http://hadoop.apache.org/releases.html>

```
wget http://apache.mirrors.tds.net/hadoop/common/hadoop-2.7.4/hadoop-2.7.4.tar.gz
```

```
tar -xzvf hadoop-2.7.4.tar.gz
```

```
sudo mv hadoop-2.7.4 /usr/local/hadoop
```

```
readlink -f /usr/bin/java | sed "s:bin/java::"
```

```
sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

```
#export JAVA_HOME=${JAVA_HOME}
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre/
```


Ferramenta Hadoop

Como usar ?

Instalar o Hadoop:

<http://hadoop.apache.org/releases.html>

```
/usr/local/hadoop/bin/hadoop
```

```
mkdir ~/input
```

```
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/  
mapreduce/hadoop-mapreduce-examples-2.7.3.jar grep ~/input ~/grep_example  
'principal[.]*'
```

Ferramenta Hadoop

Projeto:

<https://goo.gl/TNnQMv>

Ferramenta Hadoop

Projeto:

<https://goo.gl/TNnQMv>

Atividade:

Mudar o código para contar apenas uma coleção específica de palavras.

Ferramenta Hadoop

Projeto:

ht

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();

    Job job = new Job(conf, "wordcount");
    job.setJarByClass(WordCount.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    job.setMapperClass(Map.class);
    job.setReducerClass(Reduce.class);

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.waitForCompletion(true);
}
```

Ferramenta Hadoop

Projeto:

```
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)
        throws IOException, InterruptedException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            context.write(word, one);
        }
    }
}
```

Ferramenta Hadoop

Projeto:

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

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

Create Cluster - Quick Options [Go to advanced options](#)

General Configuration

Cluster name

Logging

S3 folder

Launch mode Cluster Step execution

Software configuration

Release

- Applications**
- Core Hadoop: Hadoop 2.7.3 with Ganglia 3.7.2, Hive 2.3.0, Hue 4.0.1, Mahout 0.13.0, Pig 0.17.0, and Tez 0.8.4
 - HBase: HBase 1.3.1 with Ganglia 3.7.2, Hadoop 2.7.3, Hive 2.3.0, Hue 4.0.1, Phoenix 4.11.0, and ZooKeeper 3.4.10
 - Presto: Presto 0.170 with Hadoop 2.7.3 HDFS and Hive 2.3.0 Metastore
 - Spark: Spark 2.2.0 on Hadoop 2.7.3 YARN with Ganglia 3.7.2 and Zeppelin 0.7.2

Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type

Number of instances (1 master and 2 core nodes)

Contato

E-mail: klebermagno@gmail.com



UNIVERSIDADE FEDERAL
DE SANTA CATARINA