



PROGRAMA DE  
PÓS-GRADUAÇÃO EM  
**ENGENHARIA DE SISTEMAS  
E PRODUTOS**

# Parallel and Distributed Computing: Hadoop

Dr. Francisco Borges  
franciscoborges@ifba.edu.br



# What is size?

# 72 horas

De vídeos por minuto no  
YouTube<sup>1</sup>

# 145 bilhões

emails enviados  
diariamente<sup>1</sup>



# 40TB

dados produzidos por um experimento  
do LHC/Cern em um segundo<sup>1</sup>

# 40 mil

Consultas no  
Google por  
segundo<sup>2</sup>

# 3.5 bi

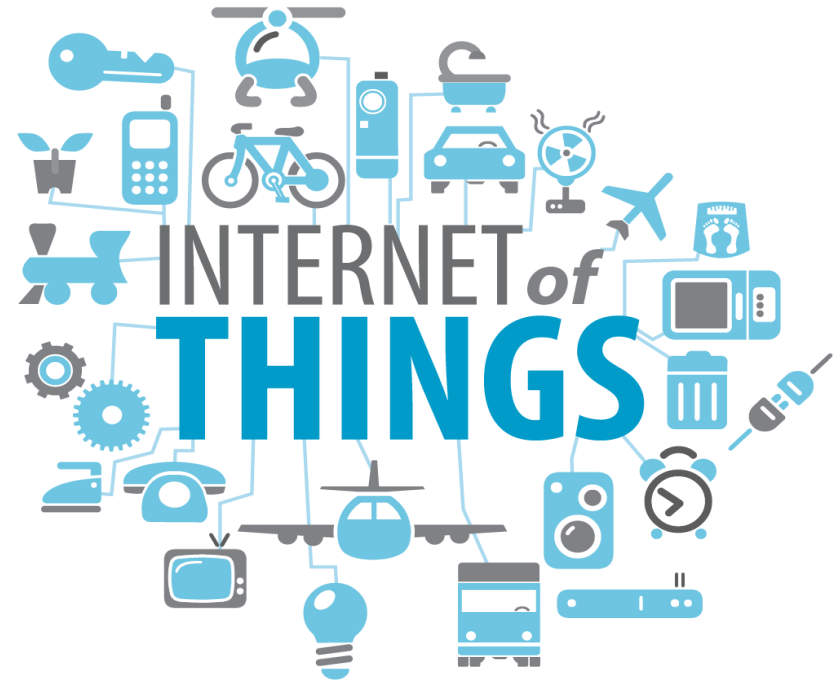
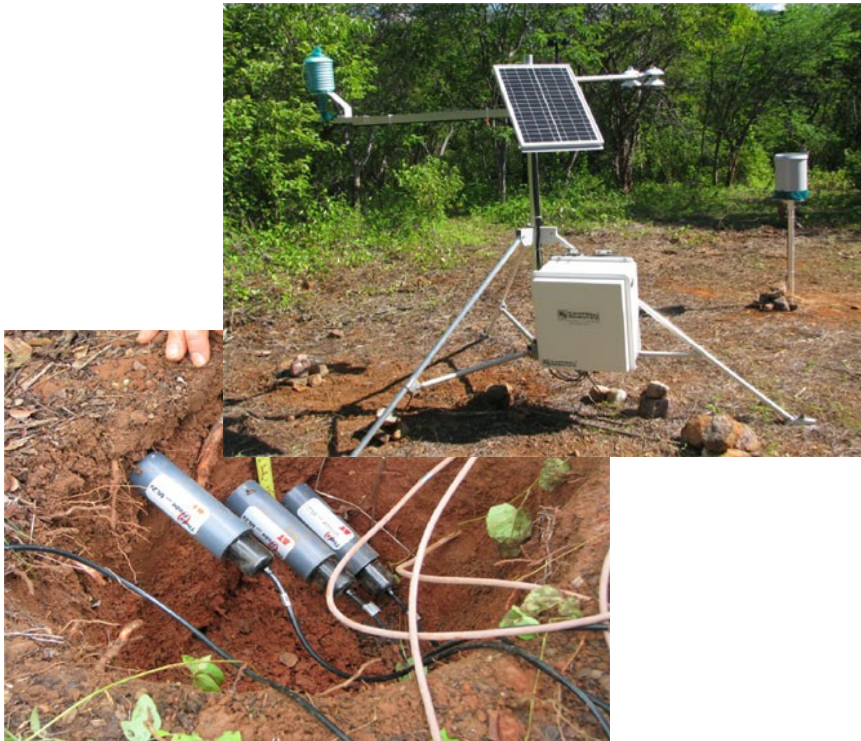
Consultas no  
Google por  
dia<sup>2</sup>

<sup>1</sup><http://marciaconner.com/blog/data-on-big-data/>, jul. 2012.

<sup>2</sup><http://www.internetlivestats.com/google-search-statistics/>, jun 2018



# What is source?





# What is the problem?



# 90%

produzidos nos  
dois últimos anos

# 4.4ZB

tamanho do  
universo digital

1ZB = 1024EB

1EB = 1024PB

1PB = 1024TB

1TB = 1024GB

# 1%

analisado para extrair  
alguma informação

# 33%

armazenados

<http://www.emc.com/leadership/digital-universe/2012iview/big-data-2020.htm>, dez.



# Which techniques?



# Techniques to Big Data



- Há muitas técnicas para lidar com big data, dentre elas:
  - A/B Testing
  - Association Rule learning
  - Classification
  - Cluster Analysis
  - Crowdsourcing
  - Data fusion and Data Integration
  - Data Mining
  - Ensemble Learning
  - Genetic Algorithmic
  - Machine Learning
  - Natural Language Processing
  - Neural Networks
  - Network analysis
  - Optimization
  - Pattern recognition
  - Predictive Modeling
  - Regression
  - Sentiment Analysis
  - Signal Processing
  - Spatial analysis
  - Statistics
  - Supervised Learning
  - Time Series Analysis
  - Unsupervised Learning
  - Visualization

etc...



# Tecnologies to Big Data

- Há muitas tecnologias para lidar com big data, dentre elas:
  - Big Table
  - Business Intelligence
  - Cassandra
  - Cloud Computing
  - Data mart
  - Data Warehouse
  - Distributed system
  - Dynamo
  - Extract, transform, and load (ETL)
  - Google File System
  - Hadoop
  - Hbase
  - MapReduce
  - Mashup
  - Metadata
  - Non-relational databases
  - R
  - Relational Databases
  - Semi-Structured data
  - SQL
  - Stream Processing
  - Structured Data
  - Unstructured Data
  - Visualization

etc...



# Content

- Background
  - What is parallel and distributed computing?
  - Clusters
  - Jobs
  - Process and threads
  - Bottlenecks
  - Distributed System Files
  - Framework



# Content

- Apache Hadoop Framework
- MapReduce
- Code example
- Homework



# What is parallel and distributed computing?



**Galaxy Formation**



**Planetary Movments**



**Climate Change**



**Rush Hour Traffic**



**Plate Tectonics**



**Weather**



**Auto Assembly**

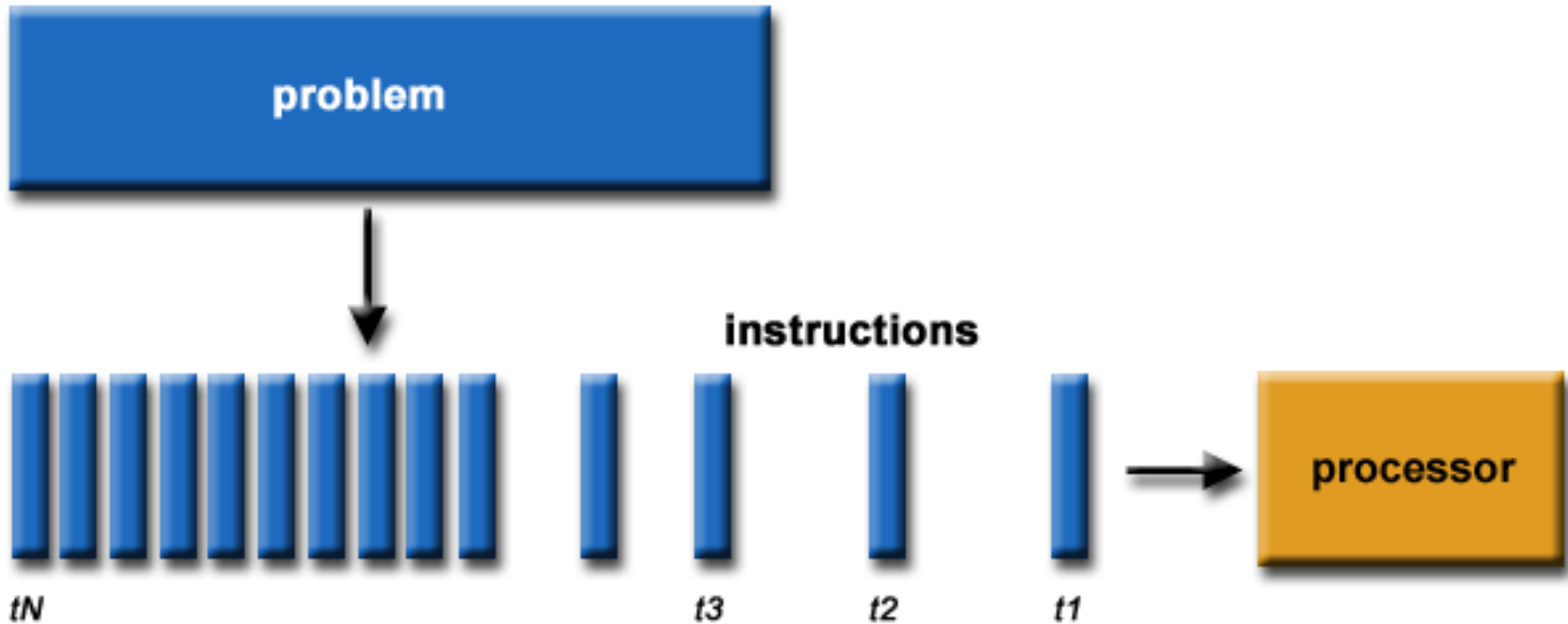


**Jet Construction**



**Drive-thru Lunch**

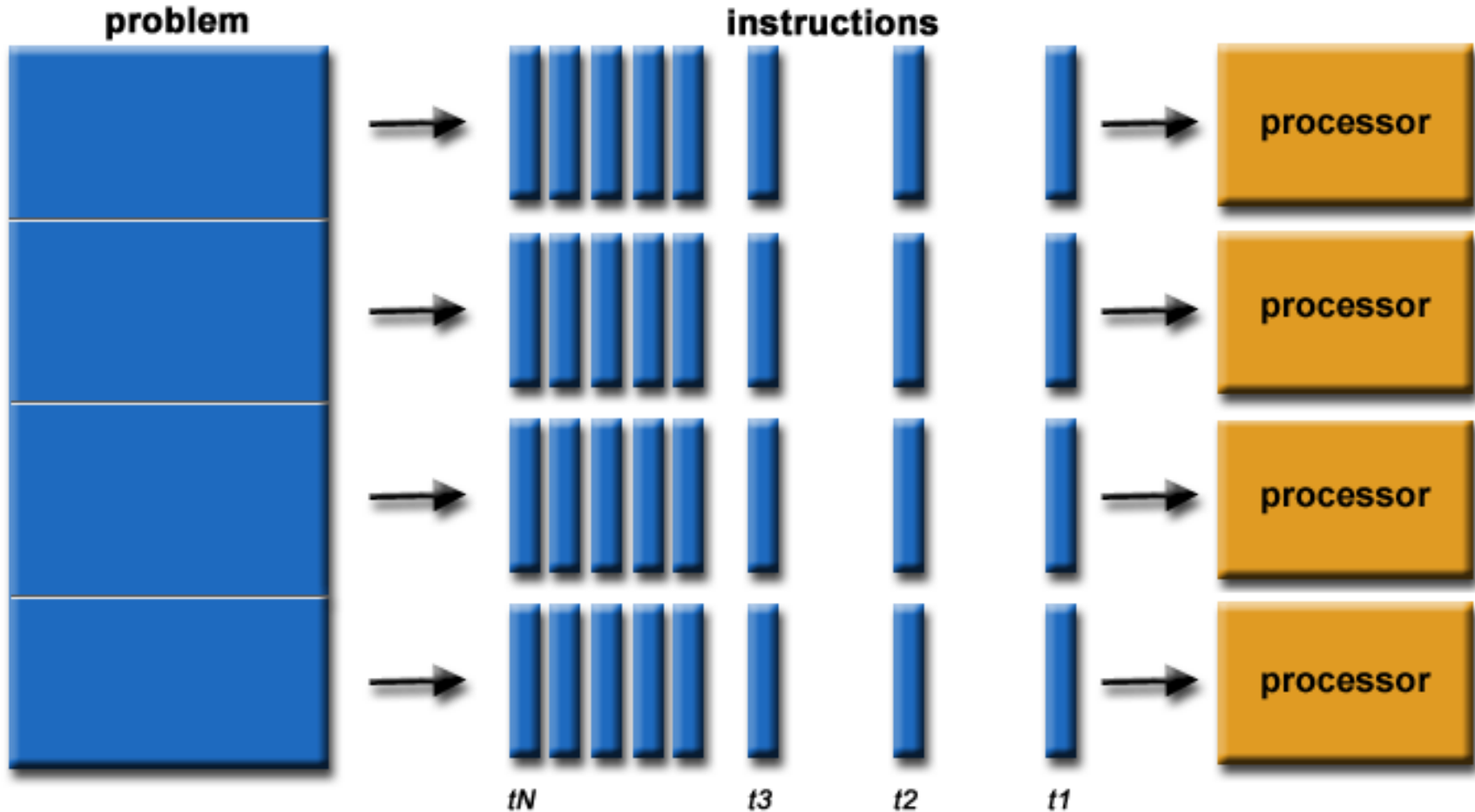
# Serial Computing



[https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)



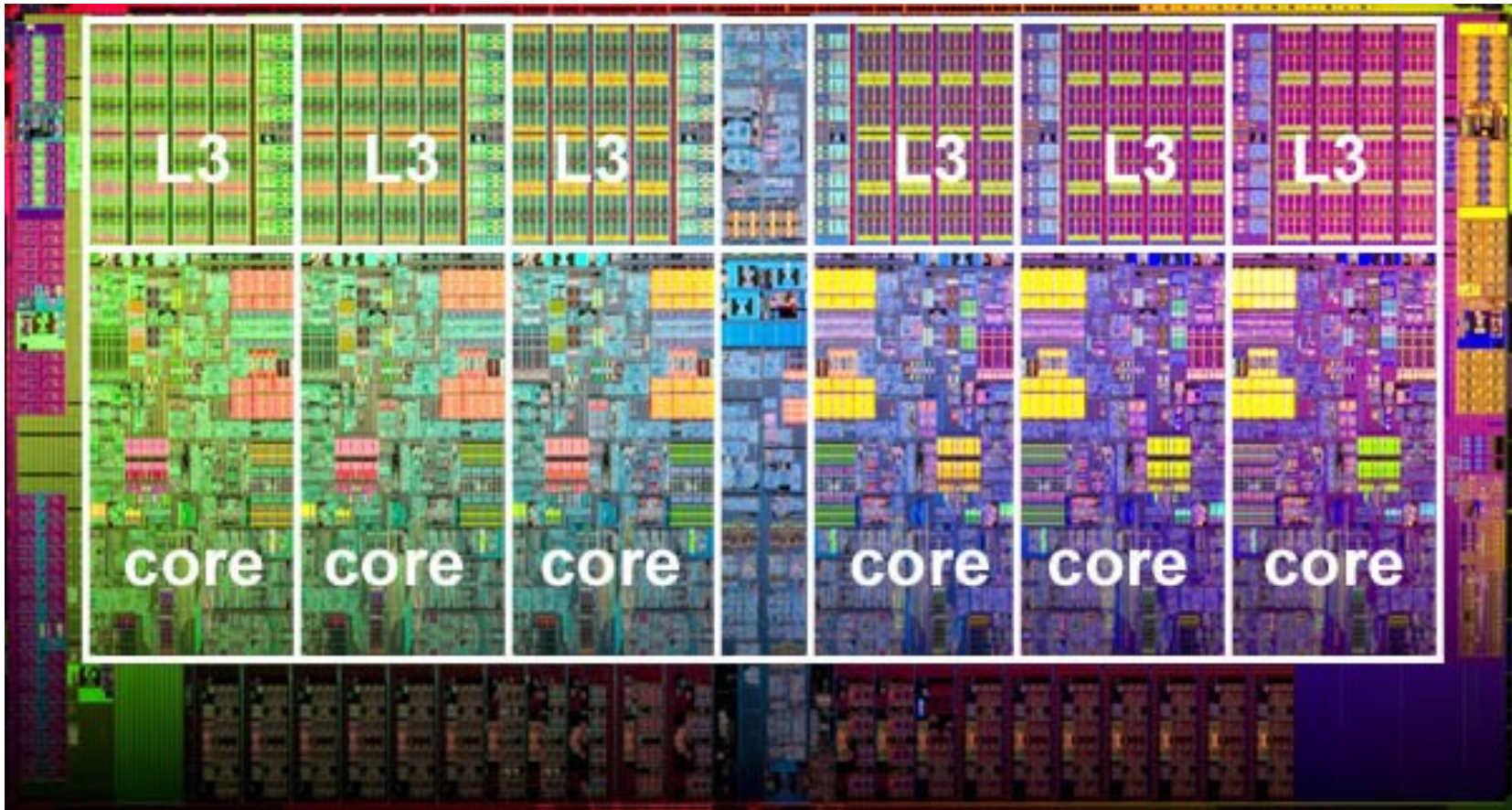
# Parallel Computing



[https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)



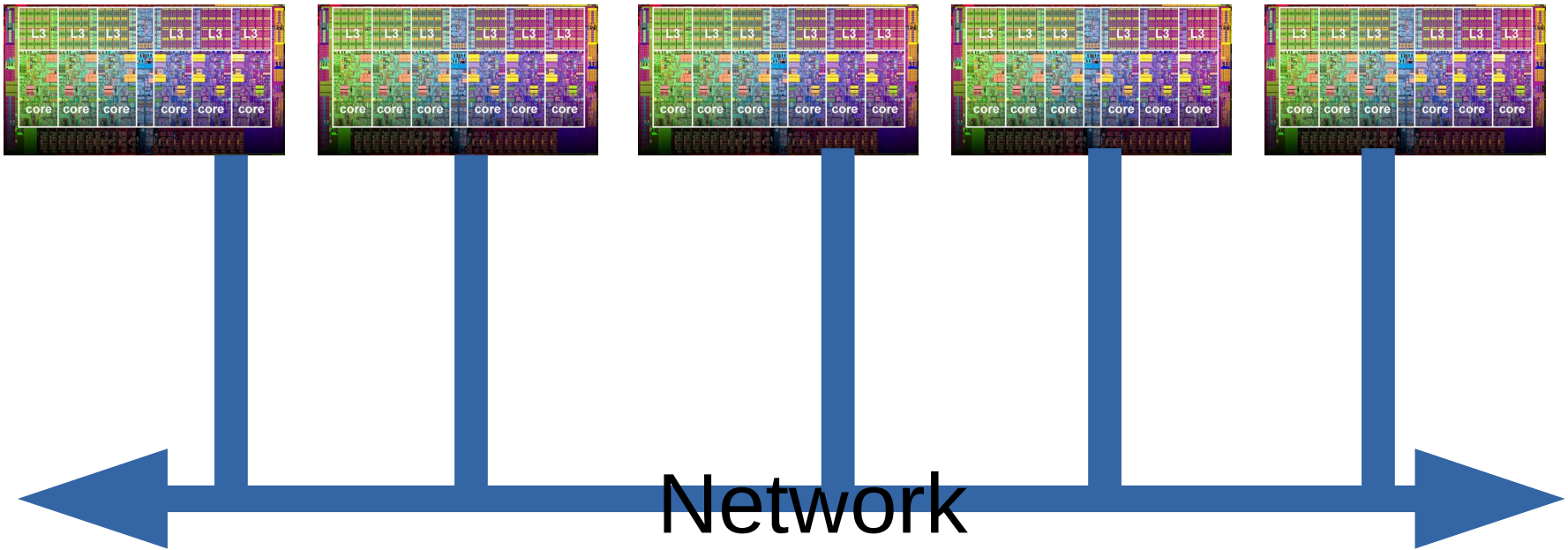
# Parallel Architecture



Intel Xeon processor with 6 cores and 6 L3 cache units

[https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)

# Distributed Architecture



[\\_1https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)



# Cluster

# Cluster



-  **compute node**
-  **infiniband switch**
-  **management hardware**

[https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)



# Job



# Job

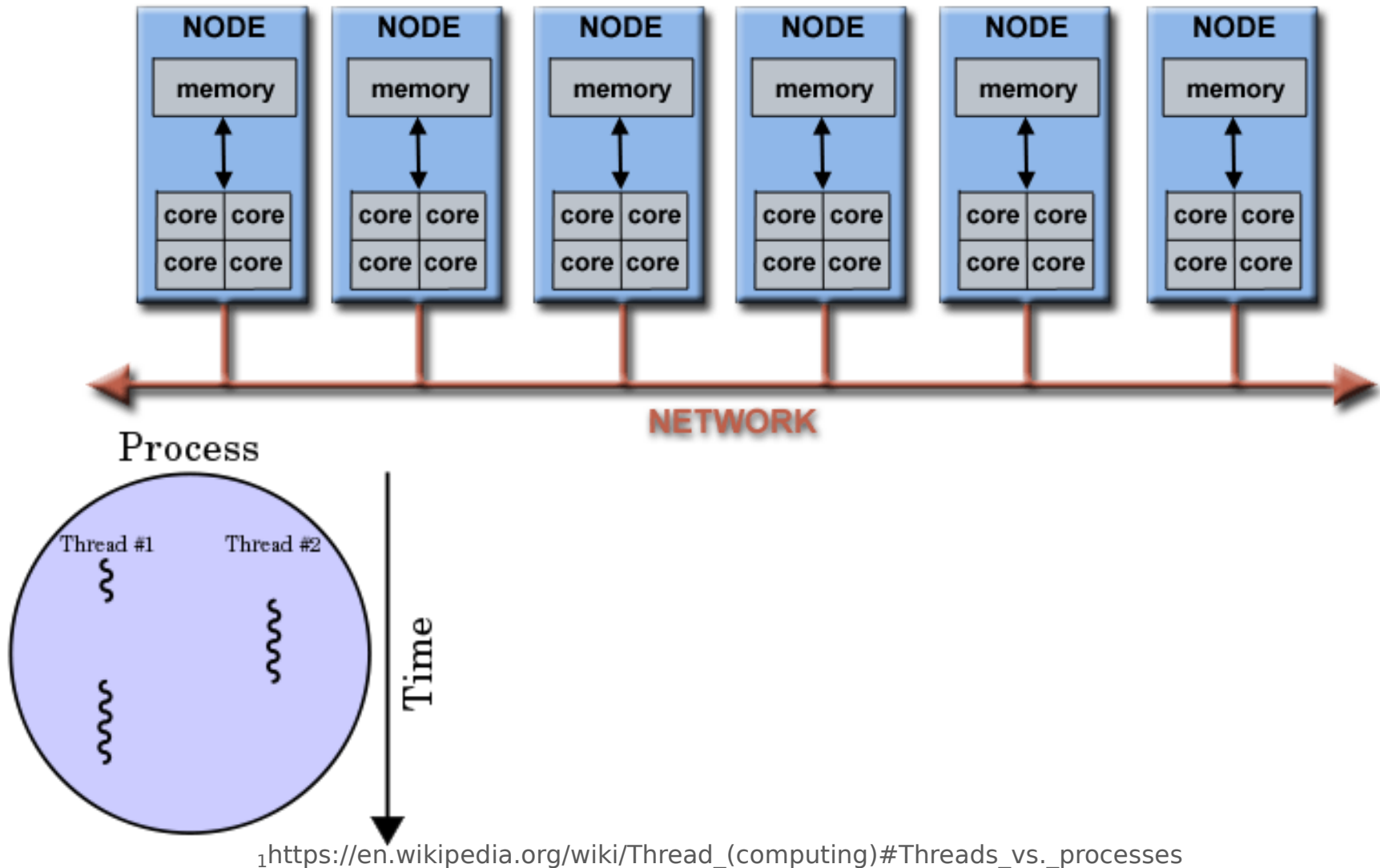
- “A job represents a task to be performed on a node in the cluster ... A job may have specific resource requirements but in general should be agnostic to which node in the cluster it runs on as long as its resource requirements are met.”<sup>1</sup>
- Oracle Grid Engine, previously known as Sun Grid Engine (SGE)
- Scripts

<sup>1</sup><http://star.mit.edu/cluster/docs/latest/guides/sge.html>



# Process and Threads

# Process and threads



[https://en.wikipedia.org/wiki/Thread\\_\(computing\)#Threads\\_vs.\\_processes](https://en.wikipedia.org/wiki/Thread_(computing)#Threads_vs._processes)





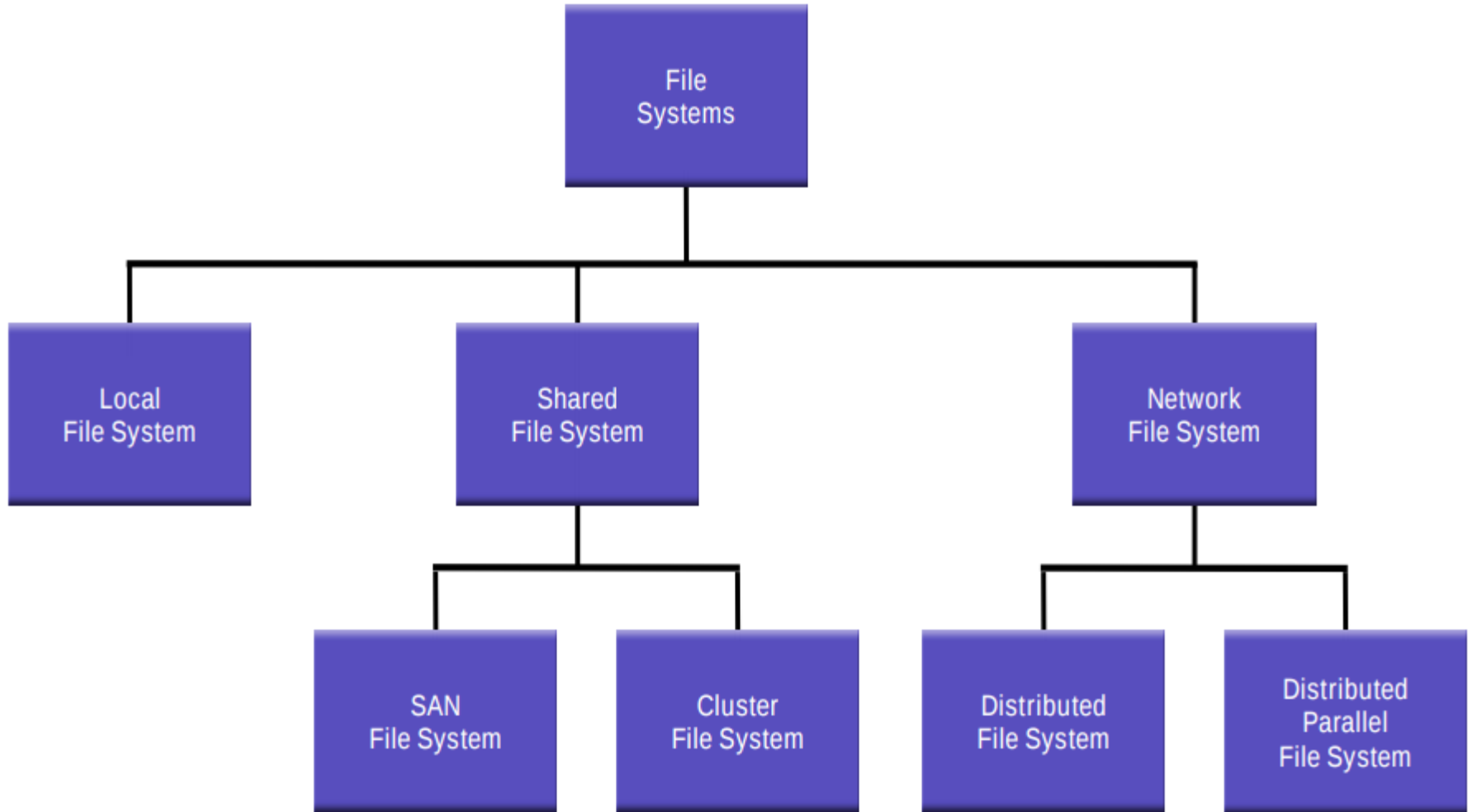
# Bottleneck



# Distributed System Files



# A File System Taxonomy



<sup>1</sup><https://pdfs.semanticscholar.org/presentation/cf6c/2598c4e6981f1e18cac03e0670da4ea4b41e.pdf>

# DSF



- A Distributed File System ( DFS ) enables programs to store and access remote files exactly as they do local ones, allowing users to access files from any computer on a network.
- GFS vs HDFS<sup>1</sup>

<sup>1</sup><http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.4159&rep=rep1&type=pdf>



# Framework

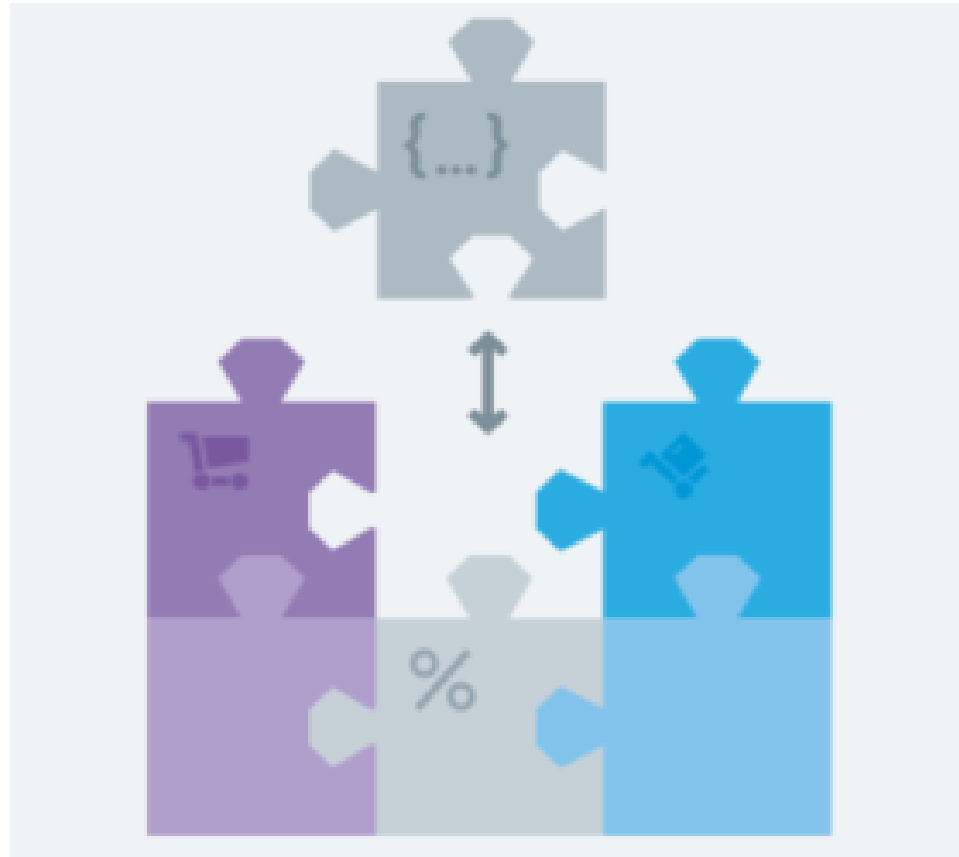


# Definition

- A software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software<sup>1</sup>

<sup>1</sup>[https://en.wikipedia.org/wiki/Software\\_framework](https://en.wikipedia.org/wiki/Software_framework)

# Definition





# Apache Hadoop Framework



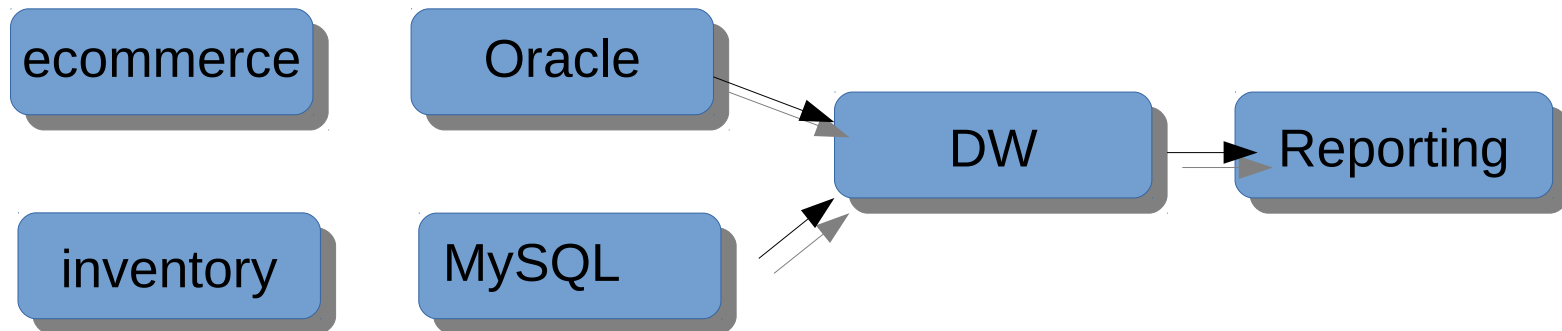
# Apache Hadoop Framework



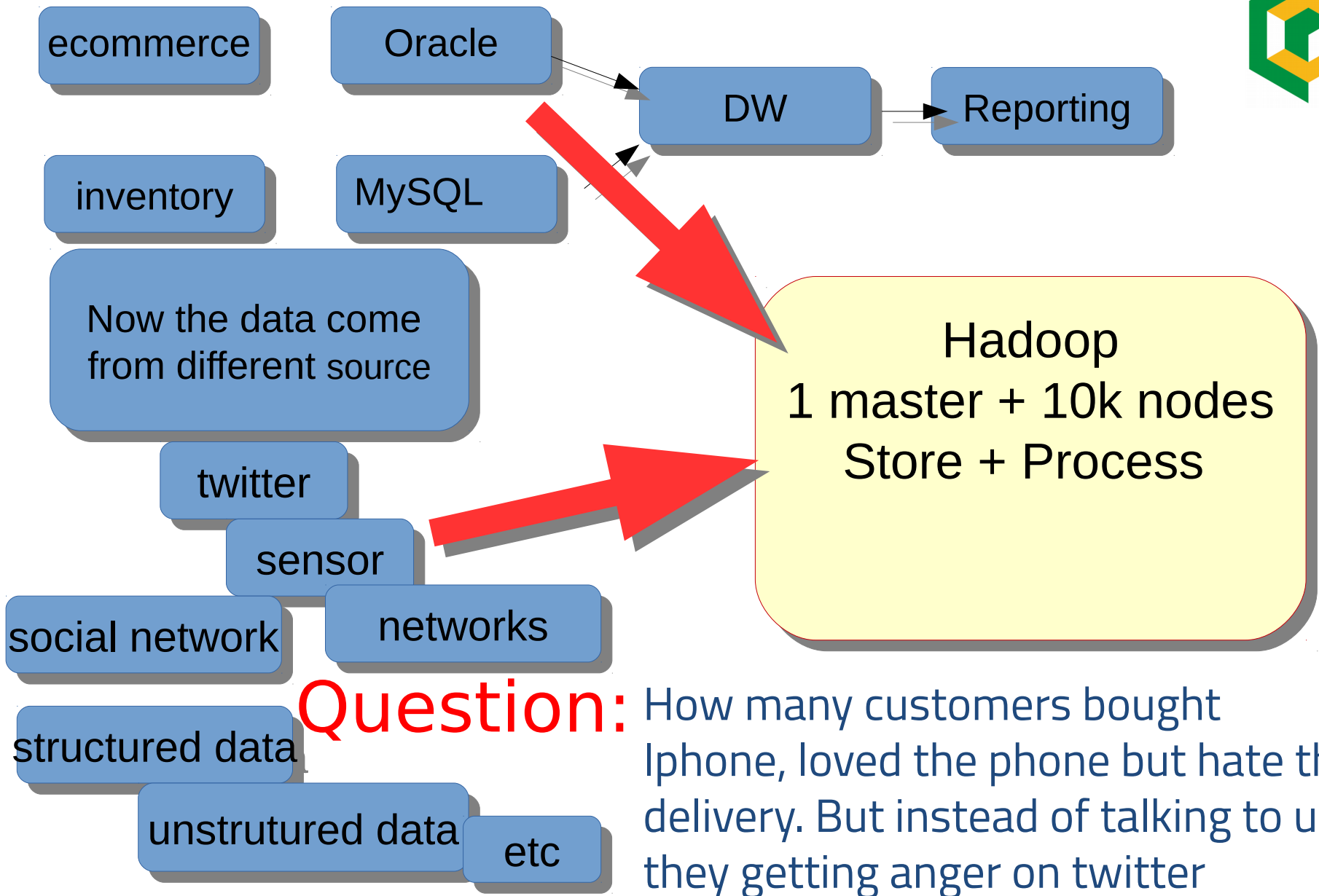
- What is Hadoop?
- Where does Hadoop really sit in an enterprise?

**Story<sup>1</sup>:** I'm Francisco Borges. I'm CEO of this commerce site called as Mamazon... And you are my thecnical guys!

<sup>1</sup><https://intellipaat.com/tutorial/hadoop-tutorial>



**Question:** How many Iphones should be order next month?

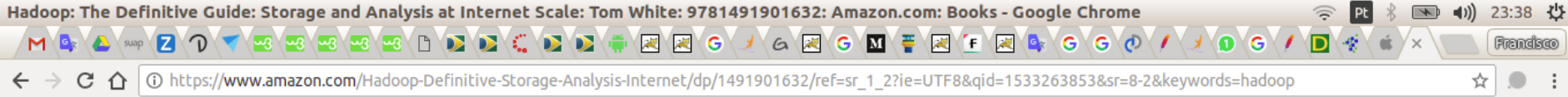


ecommerce

Oracle

85% → screen guard

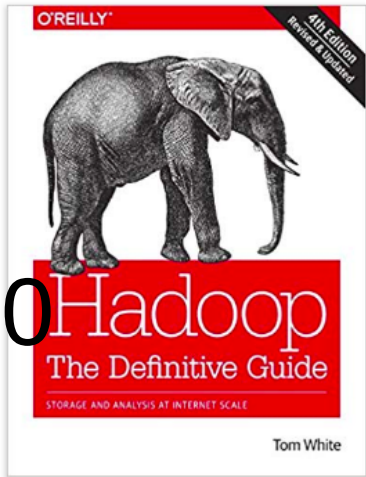
81% → backcover



#1 Best Seller In Parallel Computer Programming

Look inside ↓

Related Text

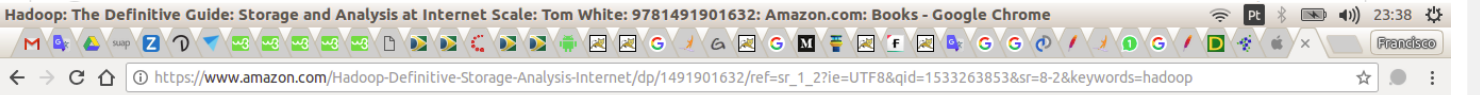


ISBN-13: 978-1491901632  
ISBN-10: 1491901632  
Why is ISBN important? ↓

Have one to sell?

Share

Kindle \$8.09 - \$19.70  
Paperback **\$13.09 - \$19.28**  
Other Sellers See all 2 versions



Customers who bought this item also bought

Page 1 of 15

- Learning Spark: Lightning-Fast Big Data Analysis  
Holden Karau  
★★★★☆ 68  
Paperback  
\$19.79 ✓prime
- Programming Hive: Data Warehouse and Query Language for Hadoop  
Edward Capriolo  
★★★★☆ 15  
Paperback  
\$37.21 ✓prime
- Spark: The Definitive Guide: Big Data Processing Made Simple  
Bill Chambers  
★★★★☆ 12  
Paperback  
\$50.56 ✓prime
- Data Analytics with Hadoop: An Introduction for Data Scientists  
Benjamin Bengfort  
★★★★☆ 4  
Paperback  
\$15.00 ✓prime
- Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale  
Neha Narkhede  
★★★★☆ 9  
Paperback  
\$25.18 ✓prime
- Hadoop Application Architectures: Designing Real-World Big Data...  
Mark Grover  
★★★★☆ 11  
Paperback  
\$40.74 ✓prime
- Advanced Analytics with Spark: Patterns for Learning from Data at...  
Sandy Ryza  
★★★★☆ 10  
Paperback  
\$30.47 ✓prime

Customers who viewed this item also viewed

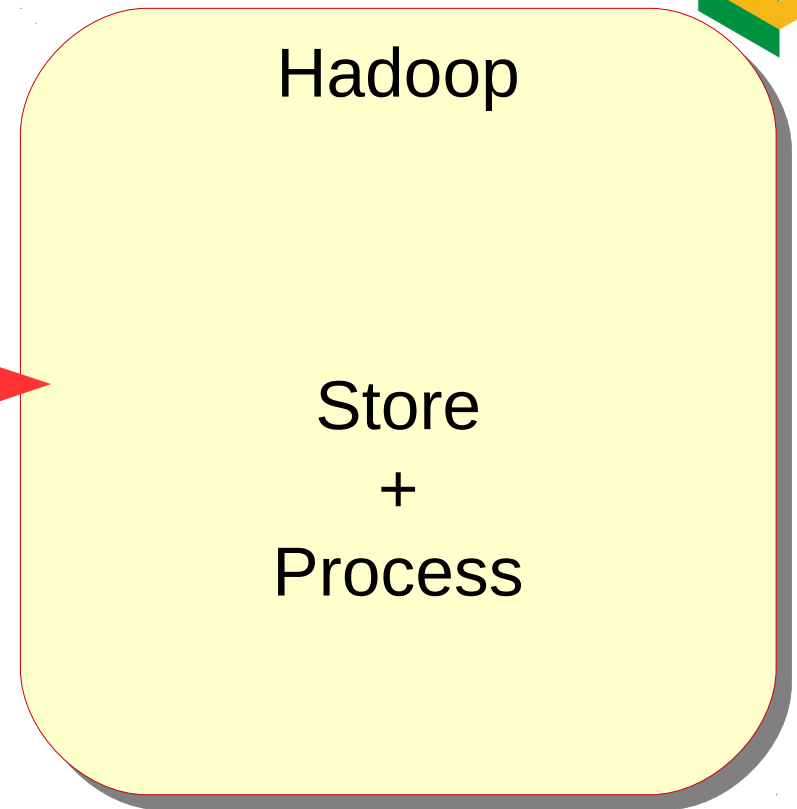
Page 1 of 9

- Learning Spark: Lightning-Fast Big Data Analysis  
Holden Karau  
★★★★☆ 68
- Data Analytics with Hadoop: An Introduction for Data Scientists  
Benjamin Bengfort
- Hadoop: The Definitive Guide  
Tom White  
★★★★☆ 82
- Programming Hive: Data Warehouse and Query Language for Hadoop  
Edward Capriolo
- Advanced Analytics with Spark: Patterns for Learning from Data at...  
Sandy Ryza
- Spark: The Definitive Guide: Big Data Processing Made Simple  
Bill Chambers
- High Performance Spark: Best Practices for Scaling and Optimizing Apache...  
Holden Karau

ENGINE.



**Big Data**



Okay, Hadoop is all about store and process BUT:

- 1 – How does store really work?
- 2 – How does process really work?

Hadoop is giant system that says: give me big data, I will store and process it for you.

**Story:** Hi I am Borges Consultancy Services and we have presence in 150 countries. Yesterday, our Facebook page got 1 Billion likes.



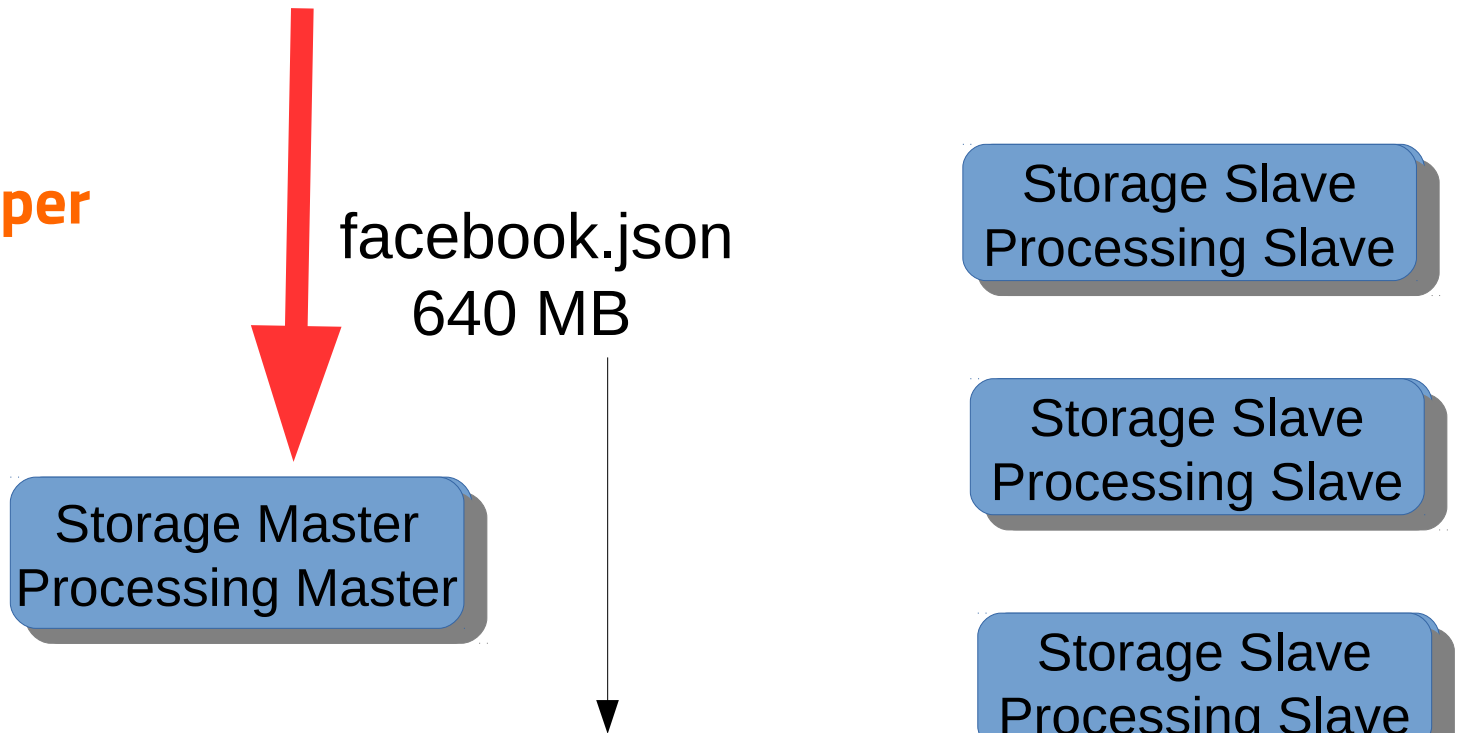
Our CEO Mr. Novais is really excited and wants a huge dashboard where we see whole globe and on hover of each country, we see how many like we got from that country...!

He calls you who is Project Manager and ask you to get this dashboard made in 4 days time...

You has two members in this team:

- Mr. Admin
- Mr. Developer

**Admin  
Developer**



**When he writes the file on Storage Master the file is broken into blocks of:**

Admin says you: boss setup is done. Go ahead and inject data and write program

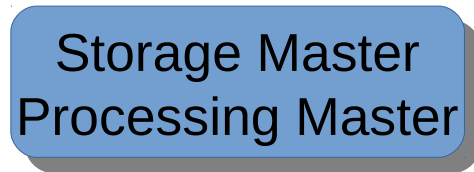
**128 MB so 640 MB is divided by 128 MB means 5 blocks And each block is replicate 3 times.**



**Developer**



facebook.json  
640 MB



**128 MB each block: 5**  
**3 replicator factor**

**Storage Master has just metadata:**  
**facebook.json → blocks: 1, 2, 3, 4, 5**  
**Block 1: slave → 1, 2, 3**  
**Block 2: slave → 2, 3, 4**

...



**Storage Master stores only metadata**





**Developer**

facebook.json  
640 MB

Storage Master  
Processing Master

Storage Slave  
Processing Slave

Storage Slave  
Processing Slave

Storage Slave  
Processing Slave

Storage Slave  
Processing Slave

● groupBy.java

**Processing Master copy  
this program in all nodes.**

**Slave 1 → blocks: 1, 2 (local to you)**

**Slave 2 → block: 3 (local to you)**

**Slave 3 → block: 4 (local to you)**

**Slave 4 → block: 5 (local to you)**

→ **Output**

# Concept:

Hadoop concept is simple:

Data cut in pieces traded across slaves

Write a program spread it across  
slaves

Process these pieces

Take output



## Processes Name



Hadoop 1.x → you can only write program in MapReduce

Hadoop 2.x → you can write program in non-MapReduce

Processing  
(MapReduce)  
(JobTracker &  
TaskTracker)

Storage  
(HDFS)  
(Namenode &  
Datanode)

MapReduce

Spark

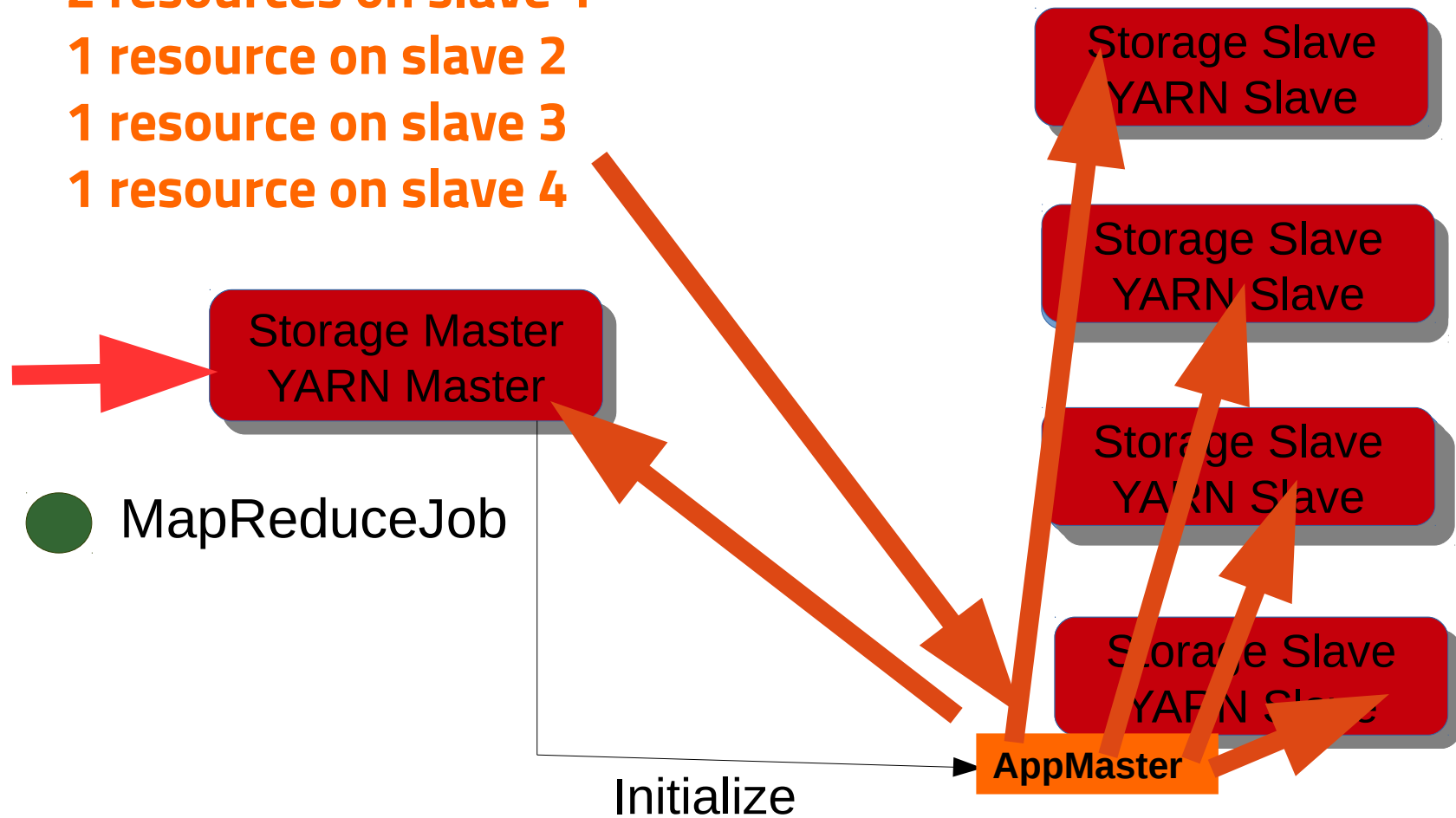
Others

Resource Manager  
YARN  
NodeManager

Storage  
(HDFS)  
(Namenode &  
Datanode)



2 resources on slave 1  
1 resource on slave 2  
1 resource on slave 3  
1 resource on slave 4





# MapReduce



# MapReduce ...

- is basically a software **framework** or **programming model**, which enable users to write programs so that data can be processed parallelly across multiple systems in a cluster.
- MapReduce consists of two parts: Map and Reduce.

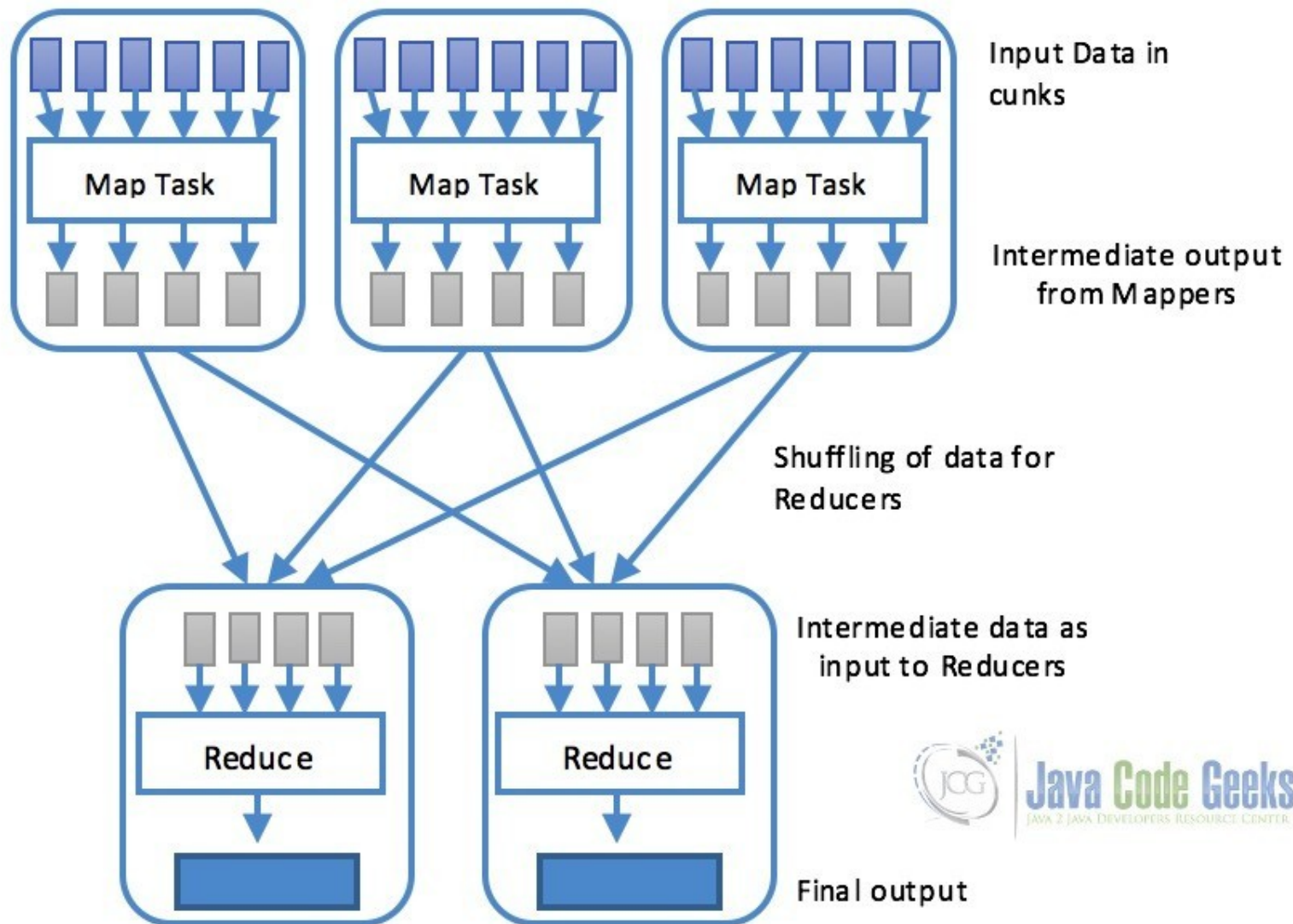
1<https://examples.javacodegeeks.com/enterprise-java/apache-hadoop/hadoop-hello-world-example/>



# Map and Reduce ...

- **Map:** Map task is performed using a `map()` function that basically performs **filtering and sorting**. This part is responsible for processing one or more chunks of data and producing the output results which are generally referred as intermediate results.
- **Reduce:** Reduce task is performed by `reduce()` function and performs a summary operation. It is responsible for consolidating the results produced by each of the Map task.

1<https://examples.javacodegeeks.com/enterprise-java/apache-hadoop/hadoop-hello-world-example/>





# Step by step MapReduce Flow<sup>1</sup>



- **Step 1:** One block is processed by one mapper at a time. In the mapper, a developer can specify his own business logic as per the requirements. In this manner, Map runs on all the nodes of the cluster and process the data blocks in parallel.
- **Step 2:** Output of Mapper also known as intermediate output is written to the local disk. An output of mapper is not stored on HDFS as this is temporary data and writing on HDFS will create unnecessary many copies.

<sup>1</sup><https://data-flair.training/blogs/hadoop-mapreduce-flow/>

# Step by step MapReduce Flow<sup>1</sup>



- **Step 3:** Output of mapper is shuffled to reducer node (which is a normal slave node but reduce phase will run here hence called as reducer node). The shuffling/copying is a physical movement of data which is done over the network.
- **Step 4:** Once all the mappers are finished and their output is shuffled on reducer nodes then this intermediate output is merged & sorted. Which is then provided as input to reduce phase.

<sup>1</sup><https://data-flair.training/blogs/hadoop-mapreduce-flow/>

# Step by step MapReduce Flow<sup>1</sup>



- **Step 5:** Reduce is the second phase of processing where the user can specify his own custom business logic as per the requirements. An input to a reducer is provided from all the mappers. An output of reducer is the final output, which is written on HDFS.

<sup>1</sup><https://data-flair.training/blogs/hadoop-mapreduce-flow/>

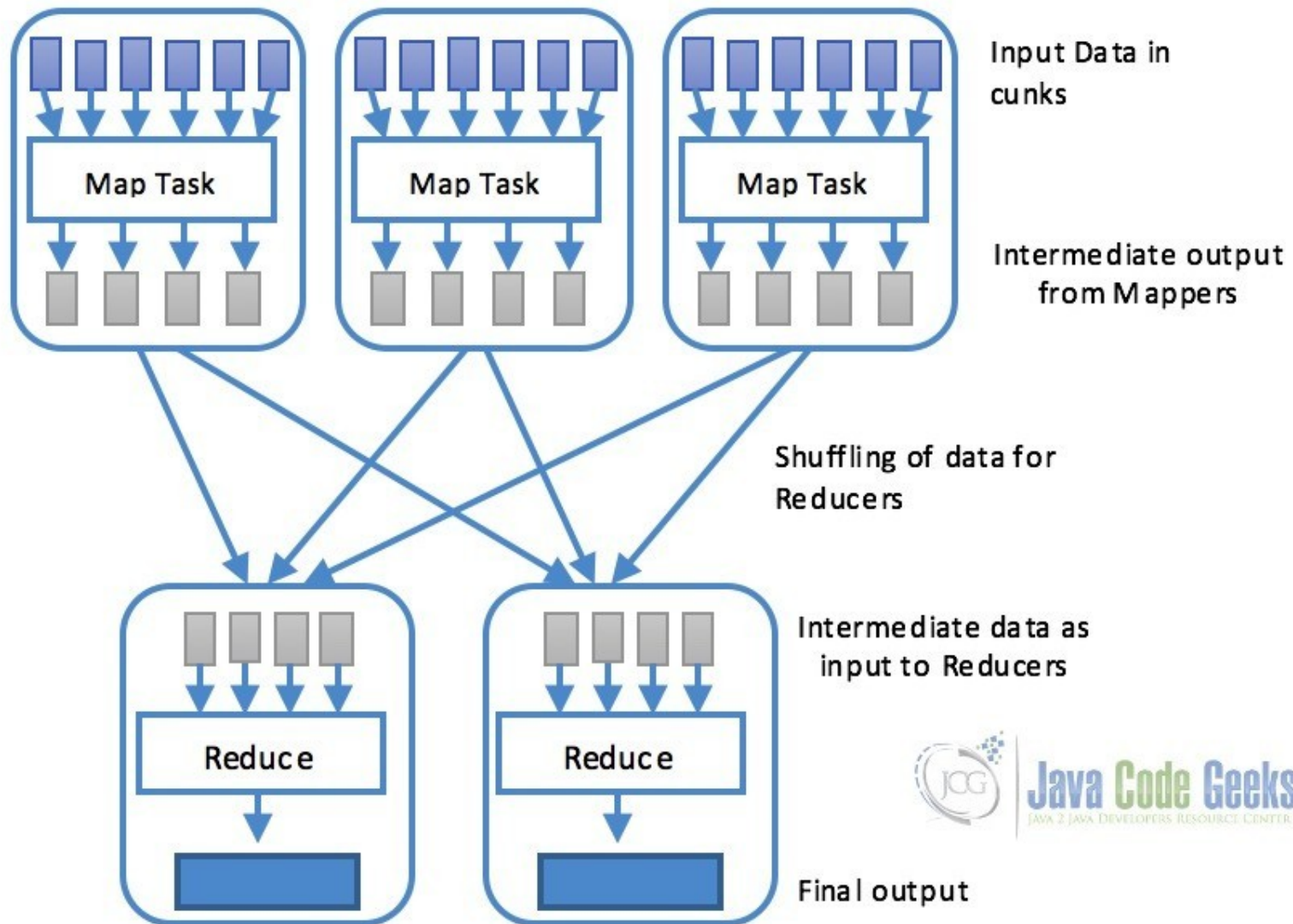
# Hadoop Word-Count Example



- Hello World” program of the Hadoop and MapReduce.
- Consists of MapReduce job that counts the number of occurrences of each word in a file.
- The Map task maps the data in the file and counts each word in data chunk provided to the map function.
- The outcome of the this task is passed to reduce which combines the data and output the final result on the disk.

1<https://examples.javacodegeeks.com/enterprise-java/apache-hadoop/hadoop-hello-world-example/>

# Hadoop Word-Count Example



# MapReduce: Inputs and Outputs<sup>1</sup>



- The MapReduce framework operates exclusively on  $\langle \text{key}, \text{value} \rangle$  pairs.
- the framework views the input to the job as a set of  $\langle \text{key}, \text{value} \rangle$  pairs
- and produces a set of  $\langle \text{key}, \text{value} \rangle$  pairs as the output of the job, conceivably of different types.

<sup>1</sup><https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.htm>

# MapReduce: Inputs and Outputs<sup>1</sup>



- the key and value classes have to be serializable by the framework and hence need to implement the **Writable interface**.
- Additionally, the key classes have to implement the **WritableComparable interface** to facilitate sorting by the framework.

<sup>1</sup><https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.htm>

# MapReduce: Inputs and Outputs<sup>1</sup>



- Input and Output types of a MapReduce job:

(input)  $\langle k1, v1 \rangle \rightarrow$  **map**  $\rightarrow \langle k2, v2 \rangle \rightarrow$  **combine**  $\rightarrow$   
 $\langle k2, v2 \rangle \rightarrow$  **reduce**  $\rightarrow \langle k3, v3 \rangle$  (output)

<sup>1</sup><https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.htm>



# Hadoop Word-Count Example



- Source code:

[https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Source\\_Code](https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Source_Code)



# Homework



# Homework

- Install Hadoop Single Node Setup
- Implements Letters-Count. Instead of count words you must count total number of letters for each word of a text . Example:

**input:** parallel and distributed computing using hadoop by Dr Francisco Borges and Dr Renato Novais

# Homework



- **output:**

Number letters	Quantity of words
8	1
3	2
11	1
9	2
5	1
6	4
2	3



# Homework

## Requirements

- You can use whatever input file but file must be at least 1000 words. Use this dummy text generation: <http://www.blindtextgenerator.com>
- You have to execute your code on Hadoop Single Node and get the result.

## Suggestion

- Use Eclipse for coding and execution.



# Must be read

- <https://data-flair.training/blogs/hadoop-tutorial/>
- <https://hadoop.apache.org/docs/stable/index.html>



# References

- <http://marciaconner.com/blog/data-on-big-data>
- <http://www.internetlivestats.com/google-search-statistics>
- <http://www.emc.com/leadership/digital-universe/2012iview/big-data-2020.htm>
- <https://intellipaat.com/tutorial/hadoop-tutorial>
- <https://examples.javacodegeeks.com/enterprise-java/apache-hadoop/hadoop-hello-world-example/>



# References

- <https://data-flair.training/blogs/hadoop-tutorial/>
- <https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>
- <https://data-flair.training/blogs/hadoop-mapreduce-flow/>
- [https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)
- <http://star.mit.edu/cluster/docs/latest/guides/sge.html>





# References

- [https://en.wikipedia.org/wiki/Thread\\_\(computing\)#Threads\\_vs.\\_processes](https://en.wikipedia.org/wiki/Thread_(computing)#Threads_vs._processes)
- <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.4159&rep=rep1&type=pdf>
- <https://pdfs.semanticscholar.org/presentation/cf6c/2598c4e6981f1e18cac03e0670da4ea4b41e.pdf>



PROGRAMA DE  
PÓS-GRADUAÇÃO EM  
**ENGENHARIA DE SISTEMAS  
E PRODUTOS**

# Parallel and Distributed Computing: Hadoop

Dr. Francisco Borges  
franciscoborges@ifba.edu.br