
Globant
we are ready

Big Data Training 2015

Apache Hadoop

Fundamentals and Architecture

April 2015

Globant
we are ready

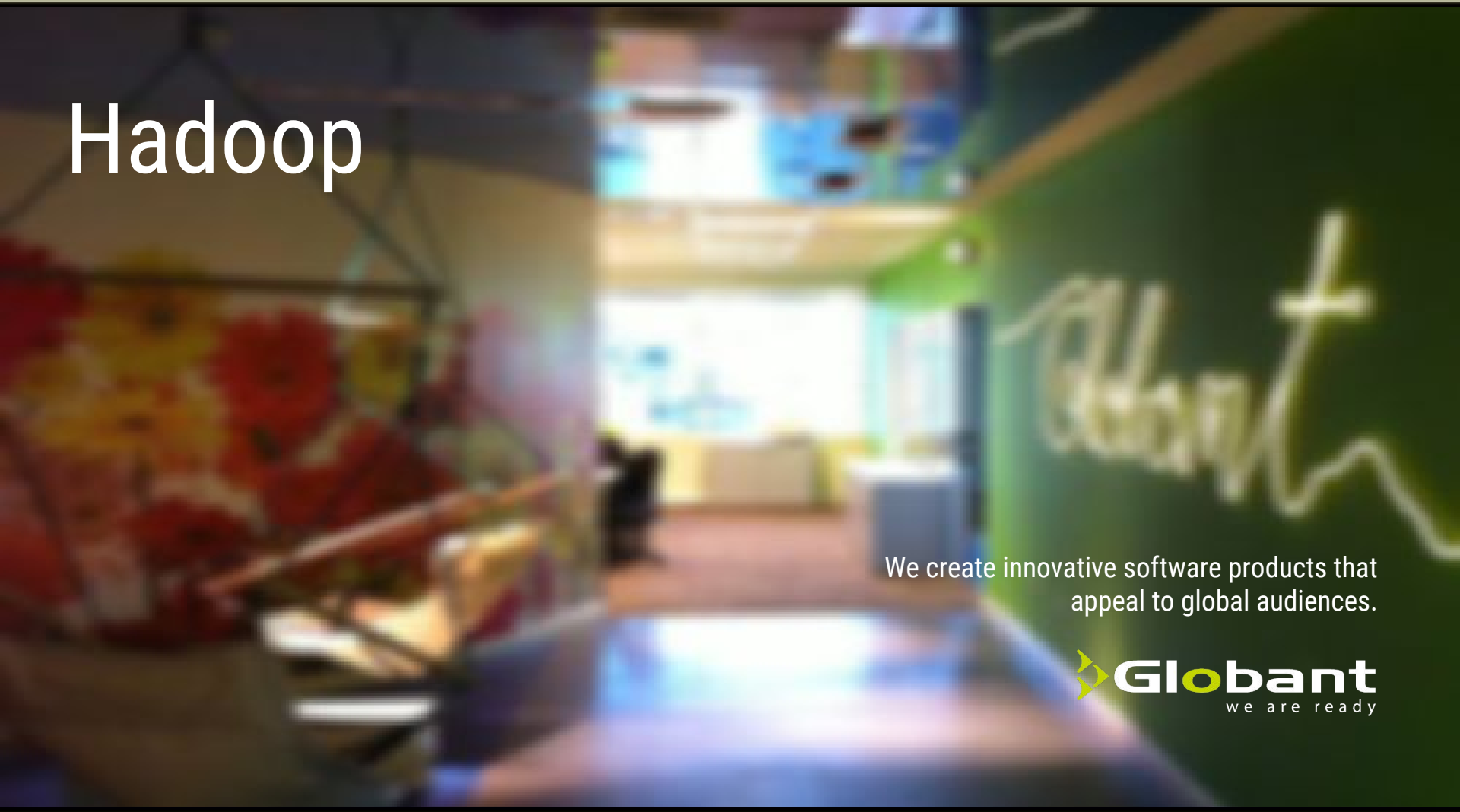
BIG DATA & HIGH PERFORMANCE

Gonzalo Zarza

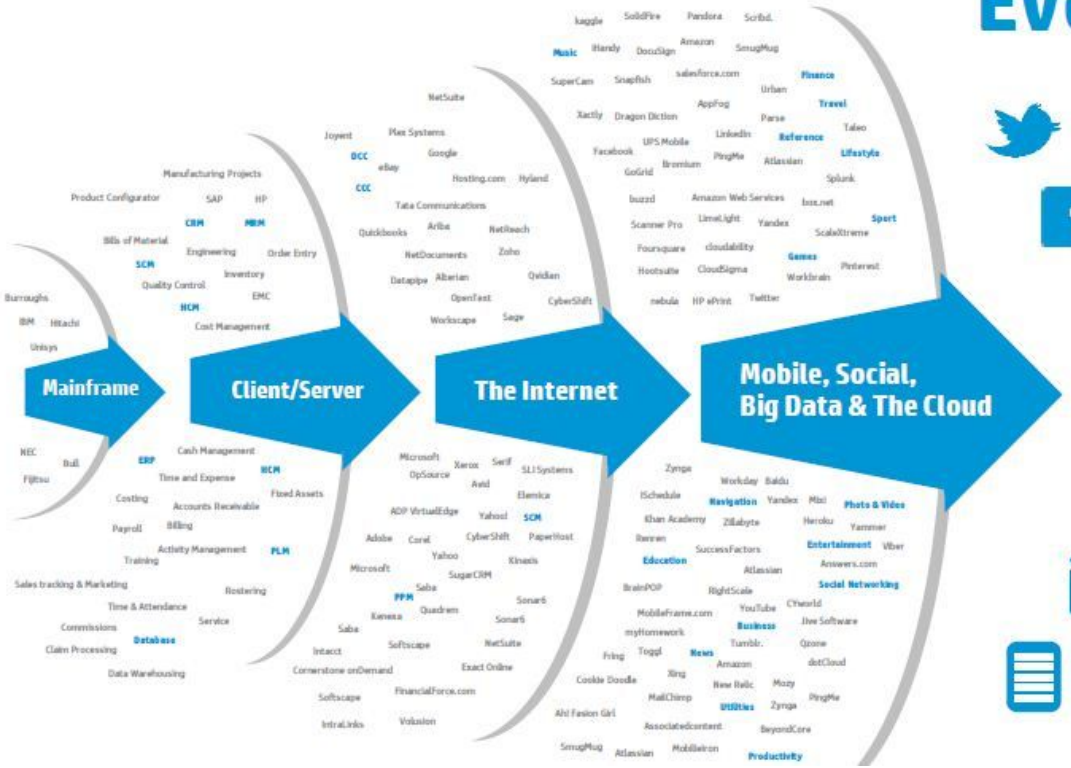
Big Data Architect
PhD in High-Performance Computing

Note: *this lecture has been prepared by Gonzalo Zarza, José Muguerza and Juan Gaviria*

Hadoop

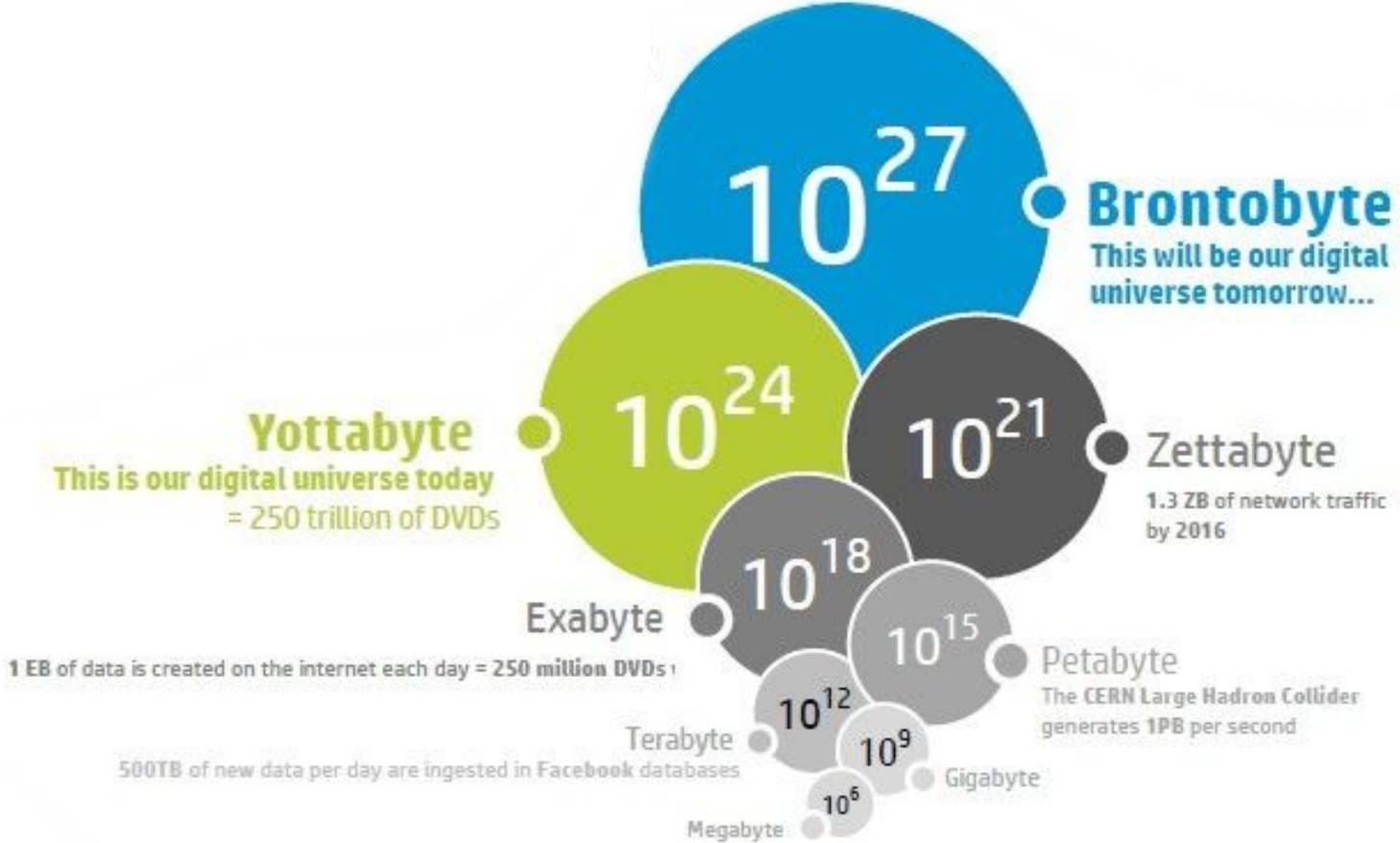

We create innovative software products that
appeal to global audiences.
Globant
we are ready

A new style of IT emerging

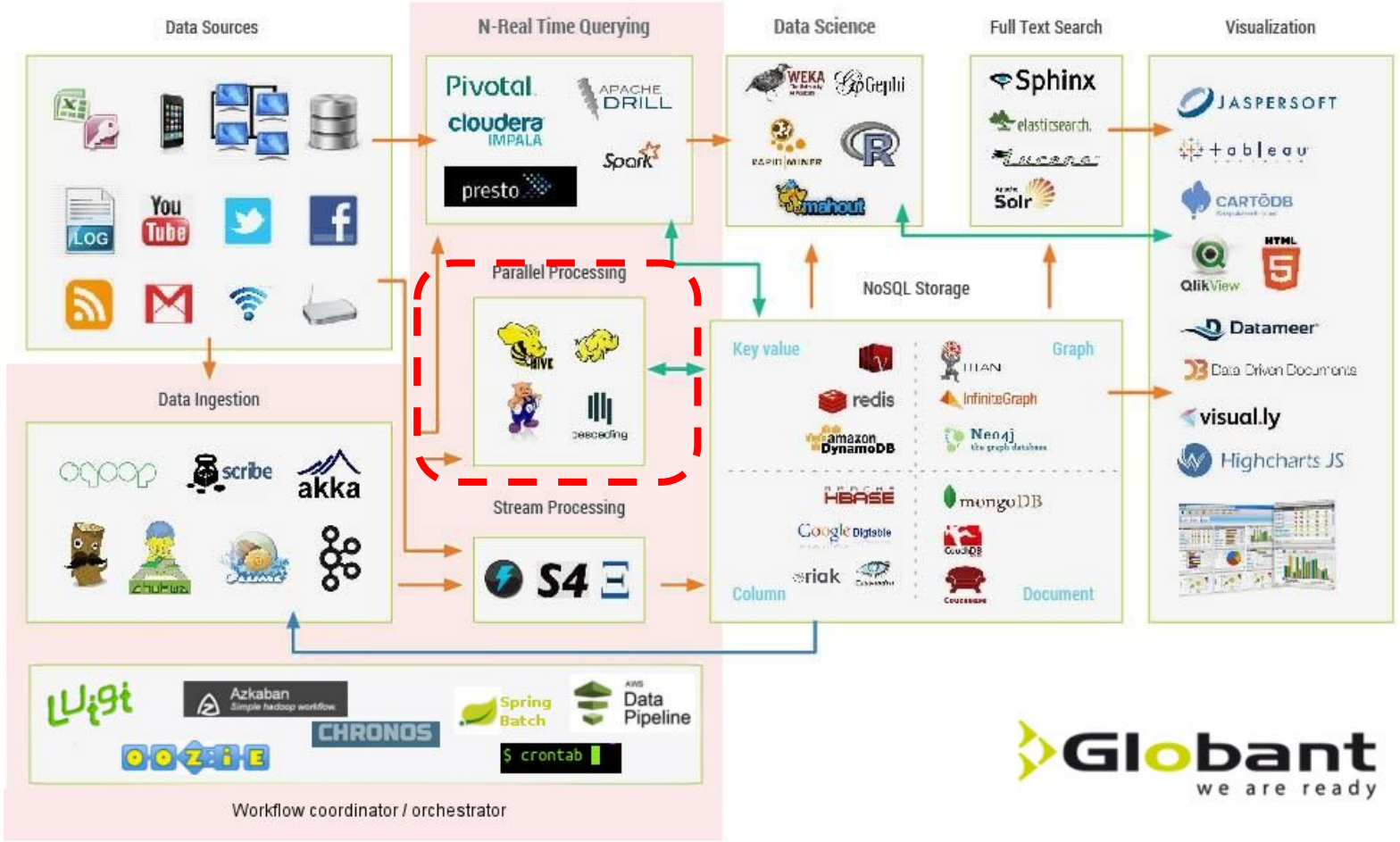


Every 60 seconds

- **98,000+** tweets
- **695,000** status updates
- **11million** instant messages
- **698,445** Google searches
- **168 million+** emails sent
- **1,820TB** of data created
- **217** new mobile web users


10^{27}
Brontobyte
This will be our digital universe tomorrow...
10^{24}
Yottabyte
This is our digital universe today
= 250 trillion of DVDs
10^{21}
Zettabyte
1.3 ZB of network traffic by 2016
10^{18}
Exabyte
1 EB of data is created on the internet each day = 250 million DVDs
10^{15}
Petabyte
The CERN Large Hadron Collider generates 1PB per second
10^{12}
Terabyte
500TB of new data per day are ingested in Facebook databases
10^{9}
Gigabyte
10^{6}
Megabyte

BIG DATA ECOSYSTEM



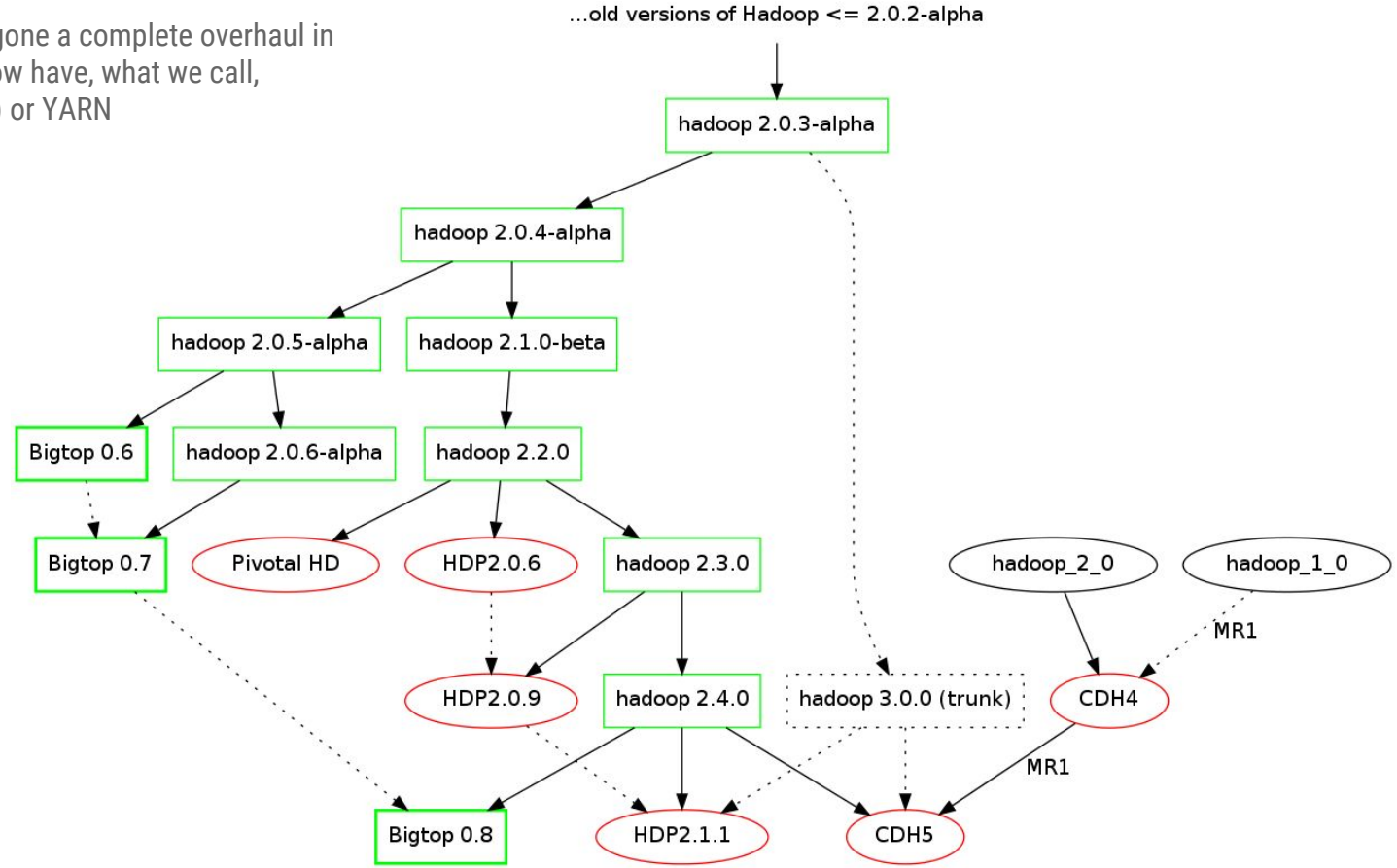
Globant
we are ready

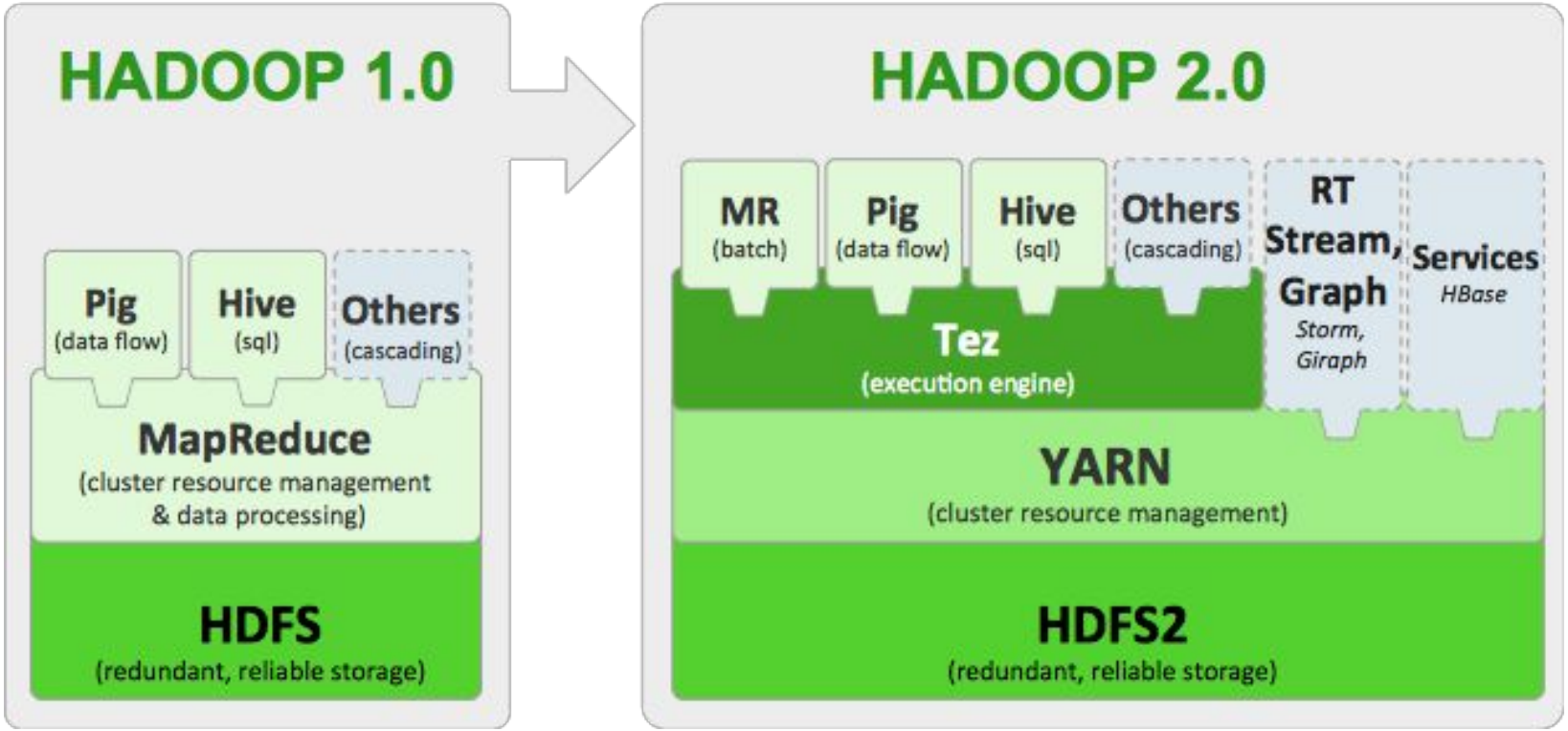
HADOOP

Description	**Apache Hadoop** es un framework de software que soporta aplicaciones distribuidas bajo una licencia libre.[1] Permite a las aplicaciones trabajar con miles de nodos y petabytes de datos. Hadoop se inspiró en los papers de Google para MapReduce (2004) y Google File System (GFS) (2003).
Last version	2.6.0 (Nov , 2014)
Maturity	Estable
Supported languages	Principalmente Java (también Python). APIs REST
Combined with other frameworks	Muchos, gran variedad de interacción….
Websites	https://hadoop.apache.org/ - Apache Hadoop YARN Book
Enterprise Support	Hortonworks, Cloudera, MapR, WanDisco, Intel, Altiscale
License	Apache License 2.0

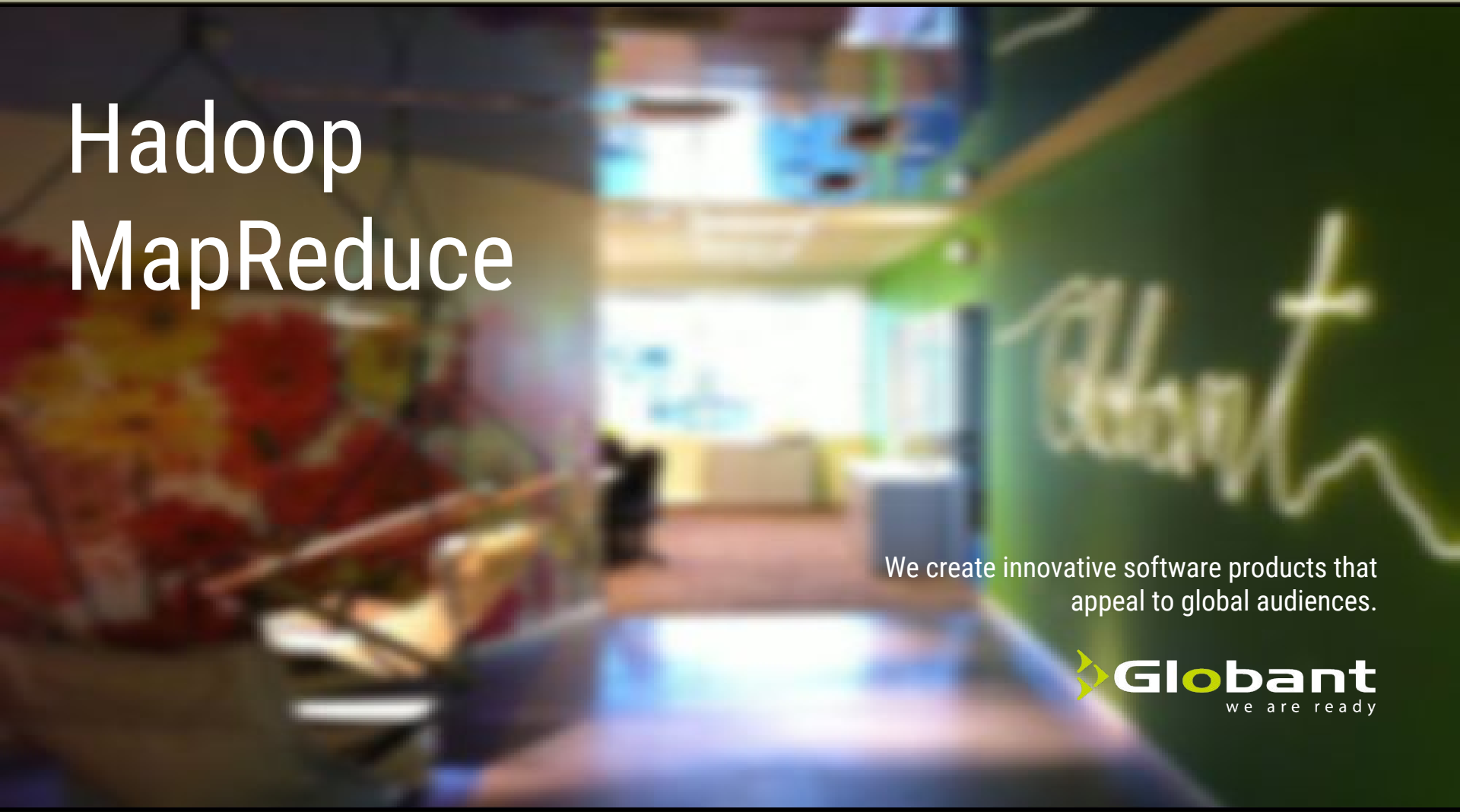

Map Reduce
Core
HDFS
Cloudera
Horton Works
Intel
Apache
Distributions
Greenplum
MapR
AWS EMR
hadoop
YARN
Flume
Mahout
Cascading
Related Projects
Pig
Sqoop
Hive
Oozie
HBase
Avro

MapReduce has undergone a complete overhaul in hadoop-0.23 and we now have, what we call, MapReduce 2.0 (MRv2) or YARN



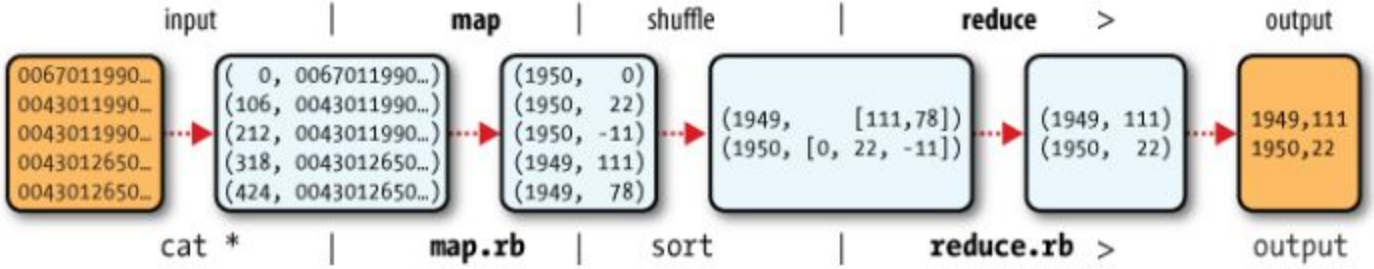

HADOOP 1.0
Pig
(data flow)
Hive
(sql)
Others
(cascading)
MapReduce
(cluster resource management
& data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MR
(batch)
Pig
(data flow)
Hive
(sql)
Others
(cascading)
RT
Stream,
Graph
Storm,
Giraph
Services
HBase
Tez
(execution engine)
YARN
(cluster resource management)
HDFS2
(redundant, reliable storage)

Hadoop MapReduce



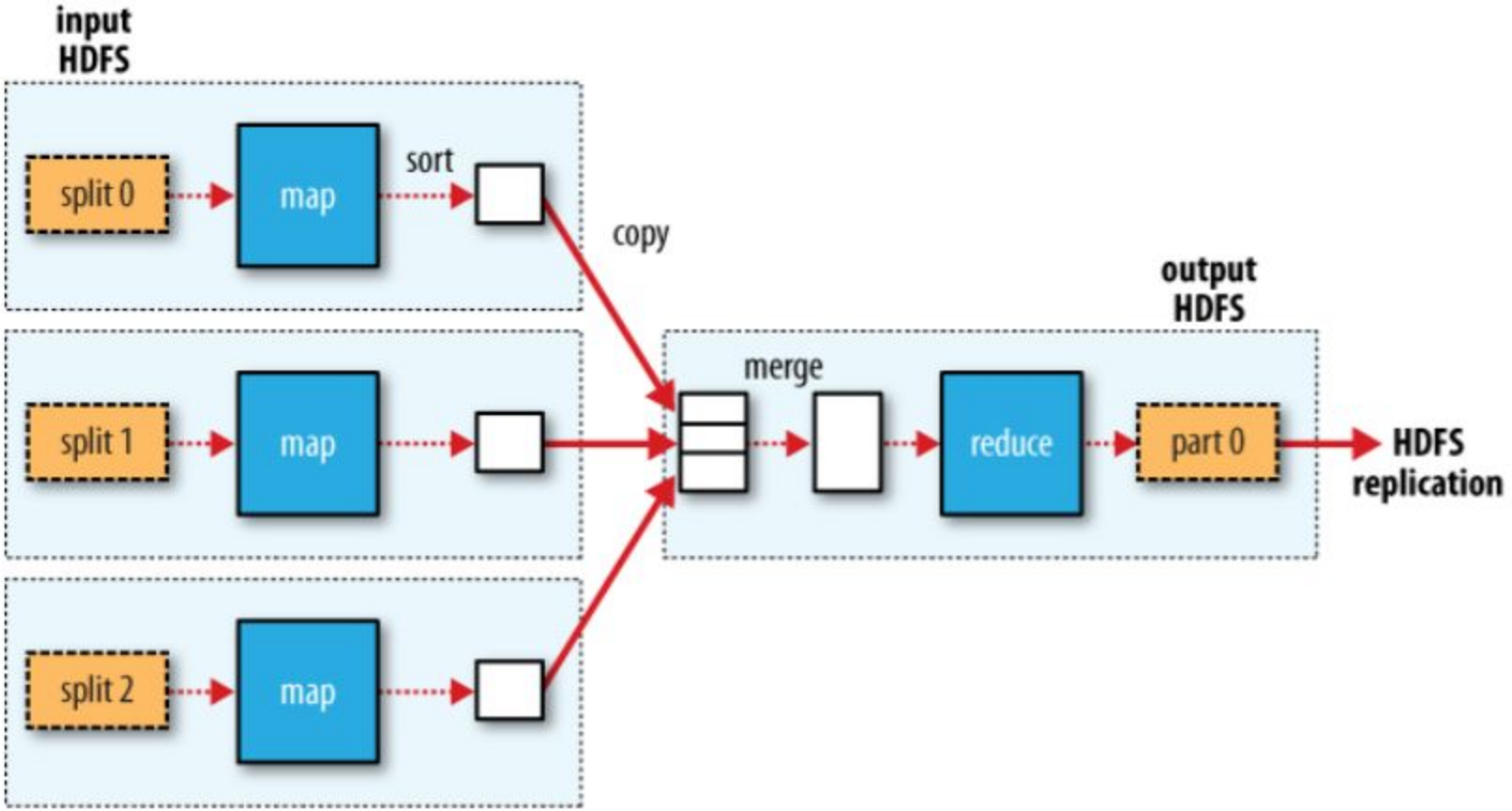
MapReduce is the heart of Hadoop. It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster.

MapReduce works by breaking the processing into **two phases**: the **map** phase and the **reduce** phase. Each phase has **key-value pairs** as input and output, the types of which may be chosen by the programmer. The programmer also specifies two functions: the map function and the reduce function.

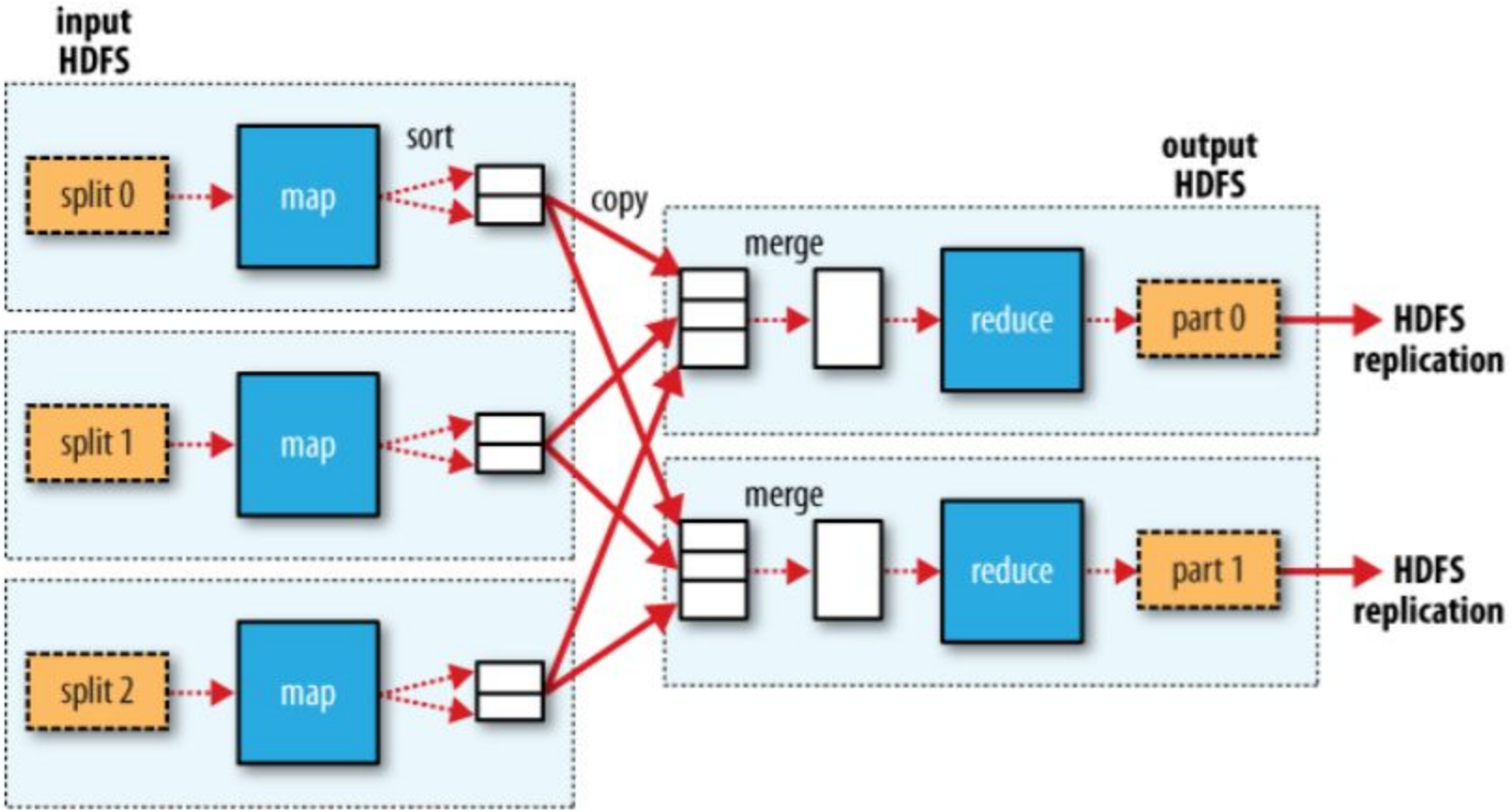


Note: *A MapReduce job is a unit of work that the client wants to be performed; it consists of the input data, the MapReduce program, and configuration information. Hadoop runs the job by dividing it into tasks, of which there are two types: map tasks and reduce tasks.*

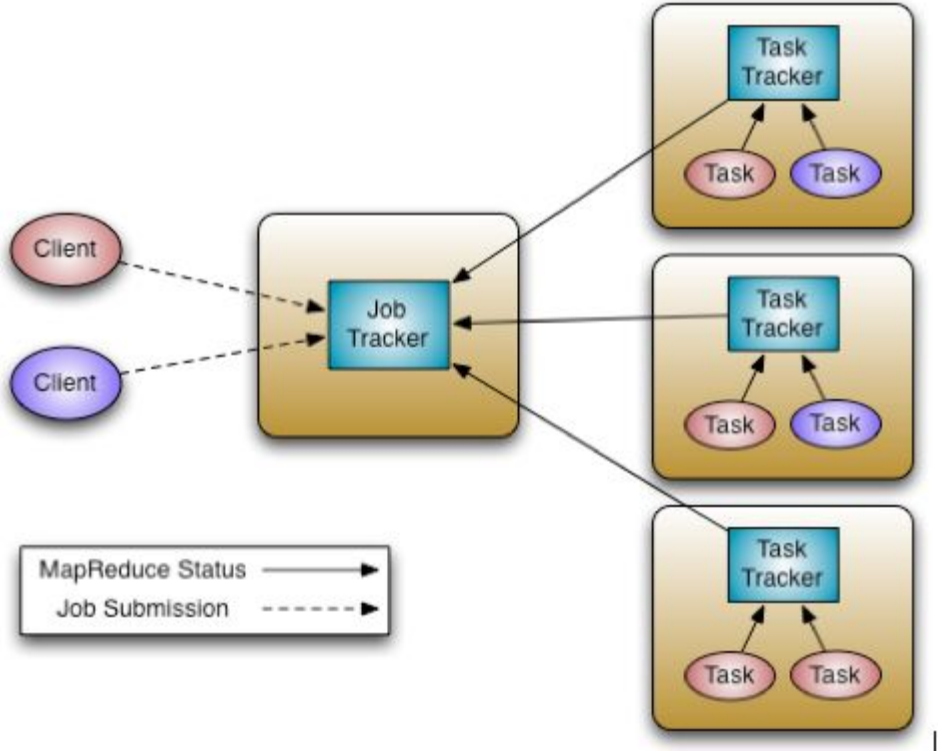
Hadoop divides the input to a MapReduce job into fixed-size pieces called input splits, or just splits. Hadoop creates one map task for each split, which runs the user- defined map function for each record in the split.

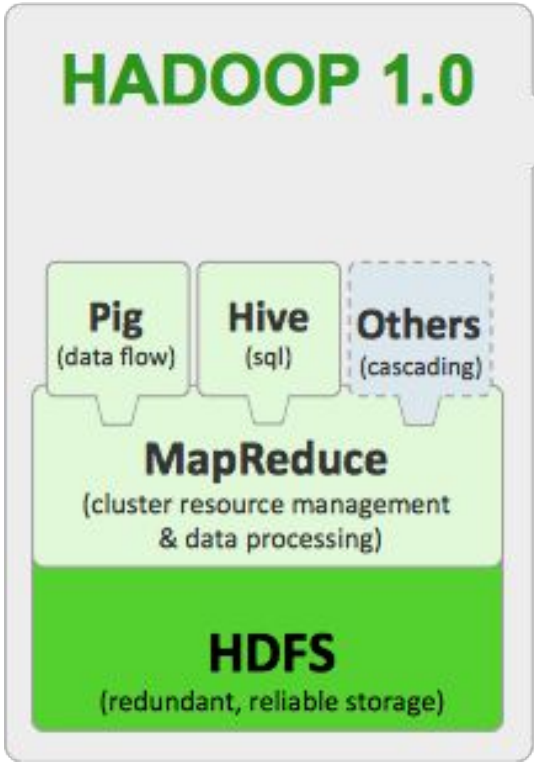


Source: *Hadoop, the definitive guide (3rd edition)*

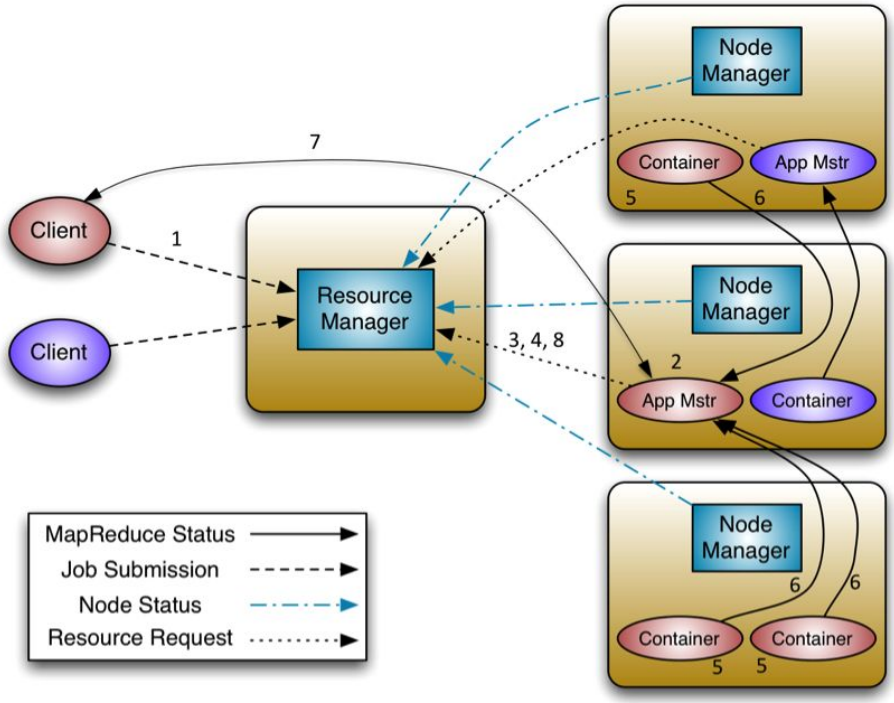


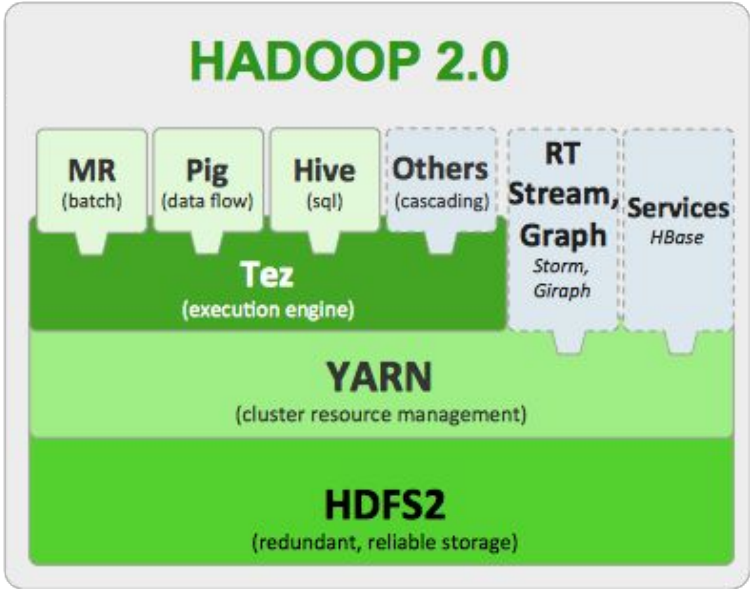
Source: *Hadoop, the definitive guide (3rd edition)*

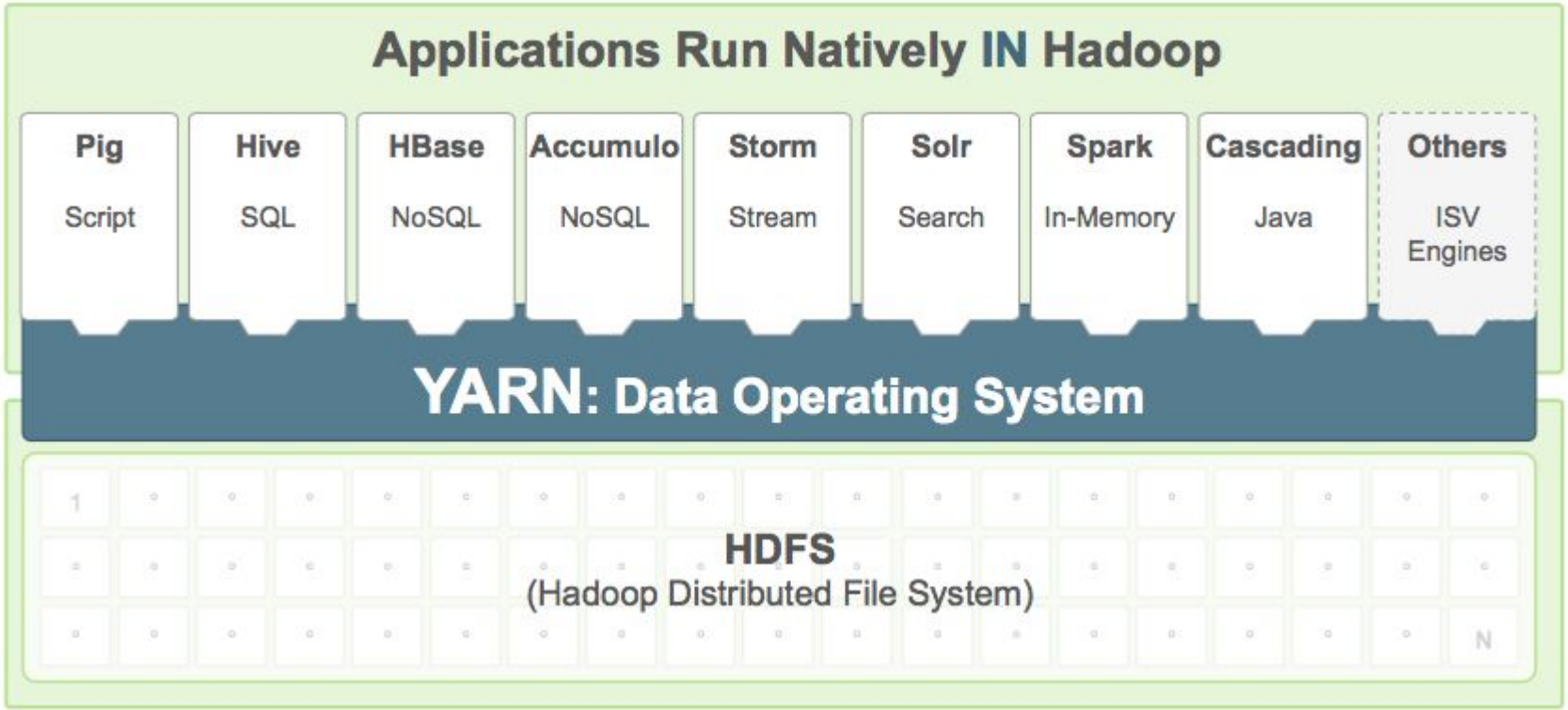

Client
Client
Job Tracker
Task Tracker
Task
Task
Task Tracker
Task
Task
Task Tracker
Task
Task
MapReduce Status
Job Submission


HADOOP 1.0
Pig
(data flow)
Hive
(sql)
Others
(cascading)
MapReduce
(cluster resource management
& data processing)
HDFS
(redundant, reliable storage)

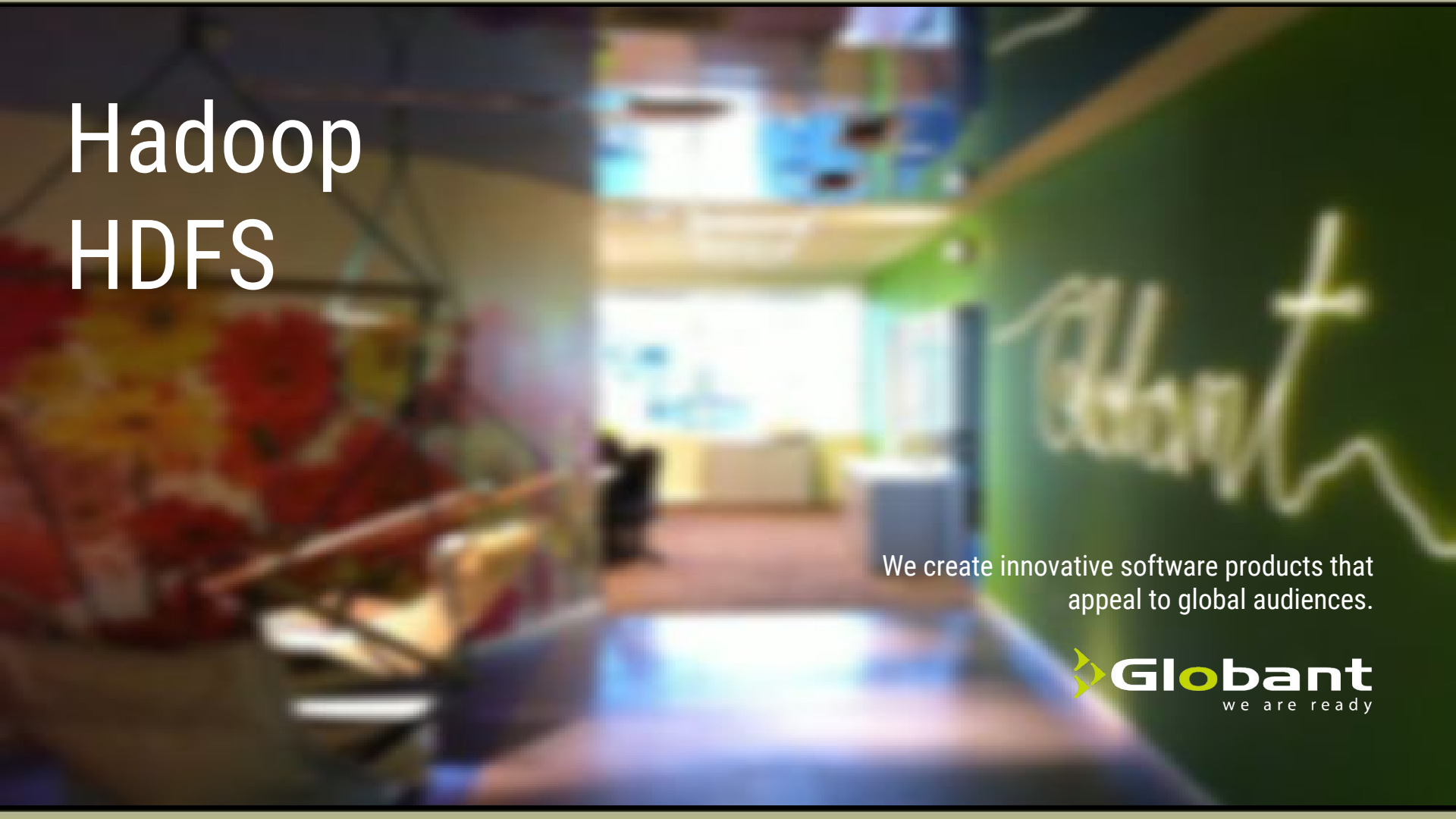
ARCHITECTURE MAPREDUCE 2.0

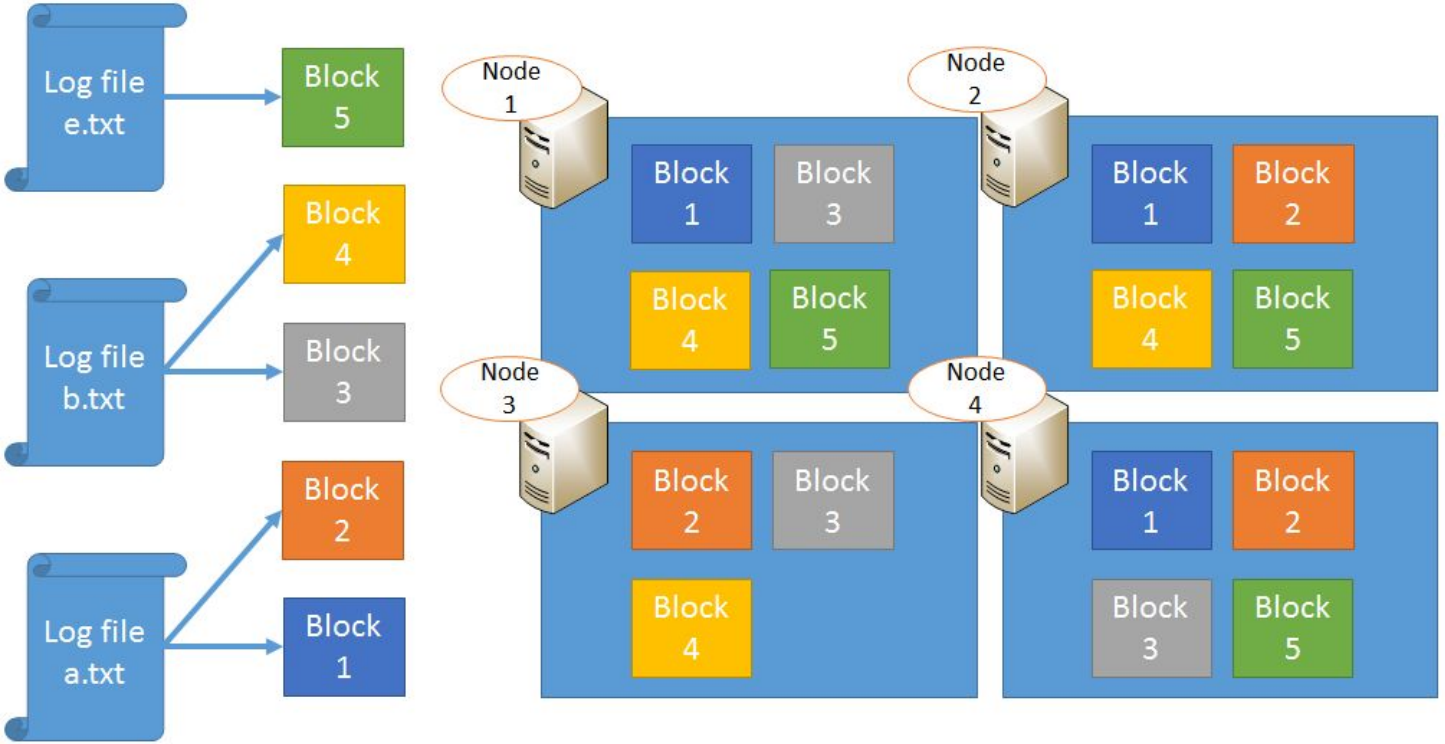


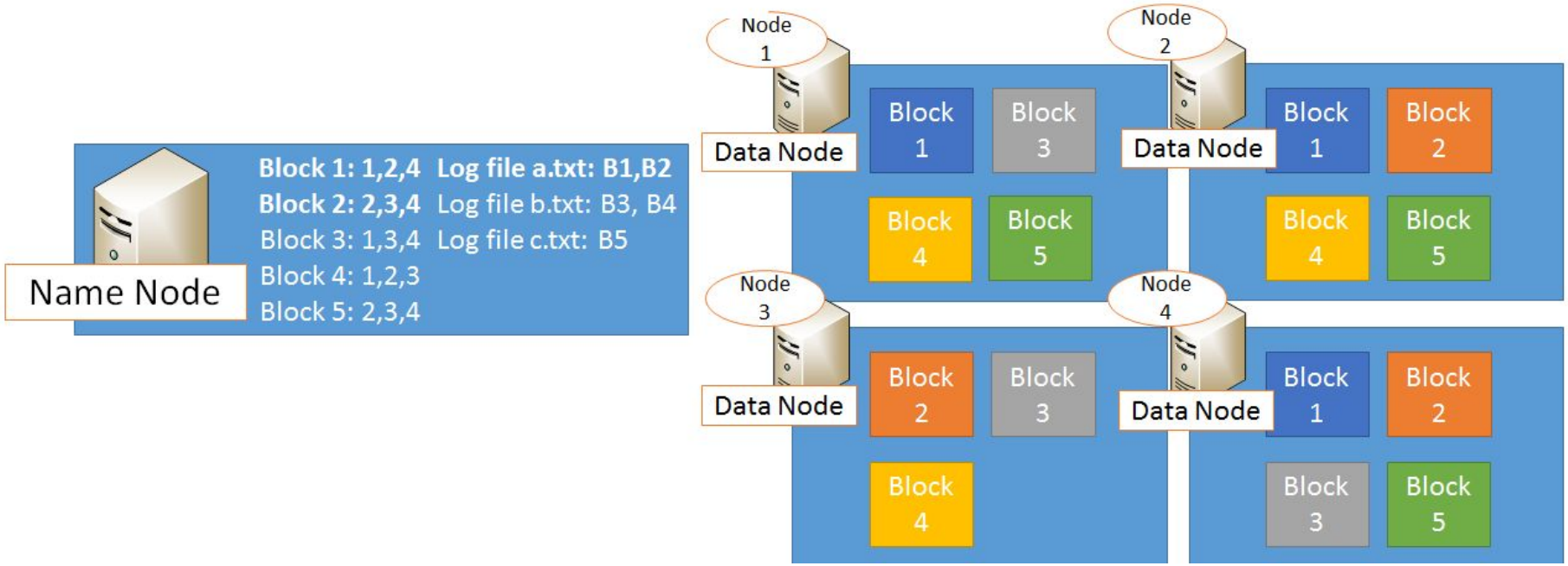



Applications Run Natively IN Hadoop
Pig
Script
Hive
SQL
HBase
NoSQL
Accumulo
NoSQL
Storm
Stream
Solr
Search
Spark
In-Memory
Cascading
Java
Others
ISV Engines
YARN: Data Operating System
HDFS
(Hadoop Distributed File System)

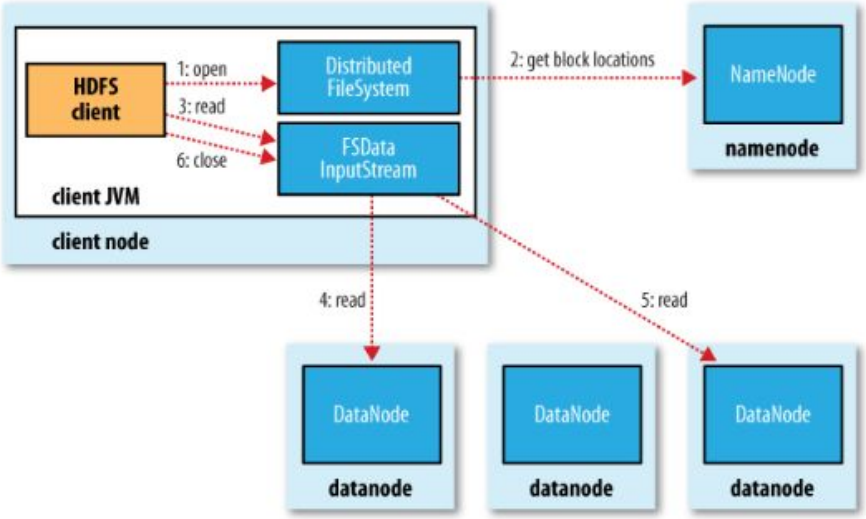
Hadoop HDFS


We create innovative software products that
appeal to global audiences.
Globant
we are ready


Log file e.txt
Block 5
Log file b.txt
Block 4
Block 3
Log file a.txt
Block 2
Block 1
Node 1
Block 1
Block 3
Block 4
Block 5
Node 2
Block 1
Block 2
Block 4
Block 5
Node 3
Block 2
Block 3
Block 4
Node 4
Block 1
Block 2
Block 3
Block 5

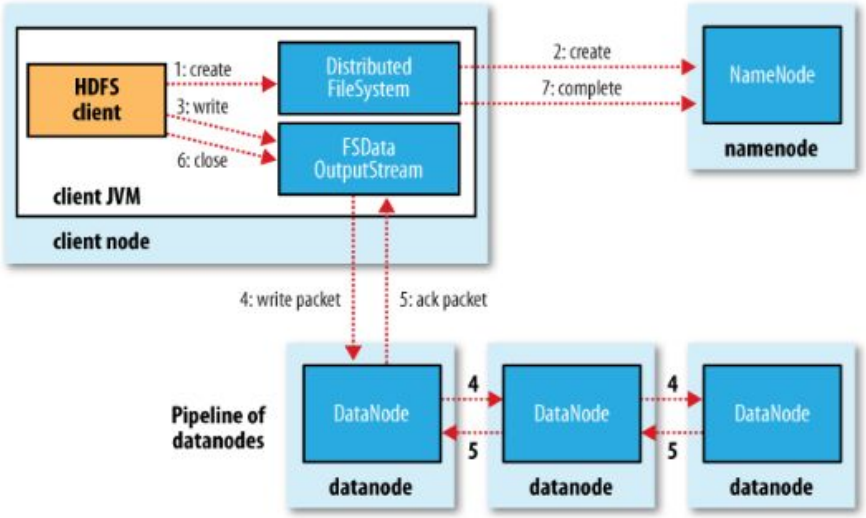

Name Node
Block 1: 1,2,4
Block 2: 2,3,4
Block 3: 1,3,4
Block 4: 1,2,3
Block 5: 2,3,4
Log file a.txt: B1,B2
Log file b.txt: B3, B4
Log file c.txt: B5
Node 1
Data Node
Block 1
Block 3
Block 4
Block 5
Node 2
Data Node
Block 1
Block 2
Block 4
Block 5
Node 3
Data Node
Block 2
Block 3
Block 4
Node 4
Data Node
Block 1
Block 2
Block 3
Block 5

HDFS: DATA FLOWS

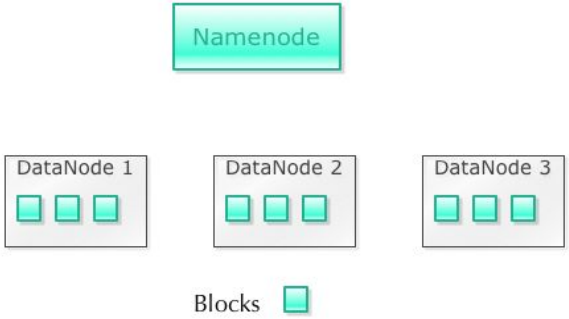


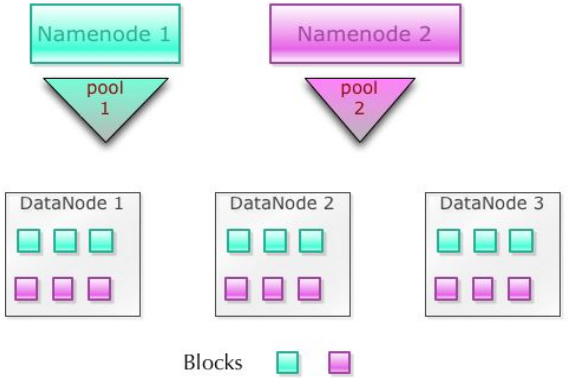
Read Flow

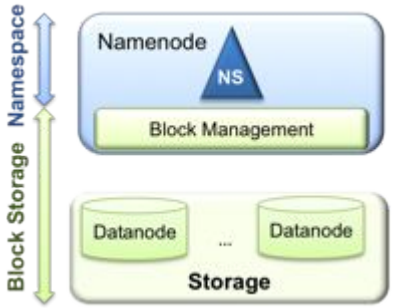
Write Flow

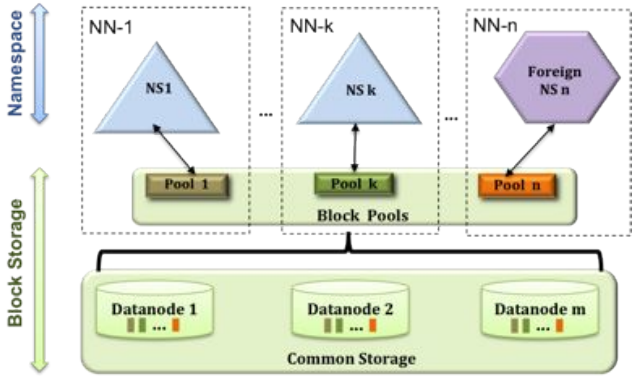


Source: *Hadoop, the definitive guide (3rd edition)*

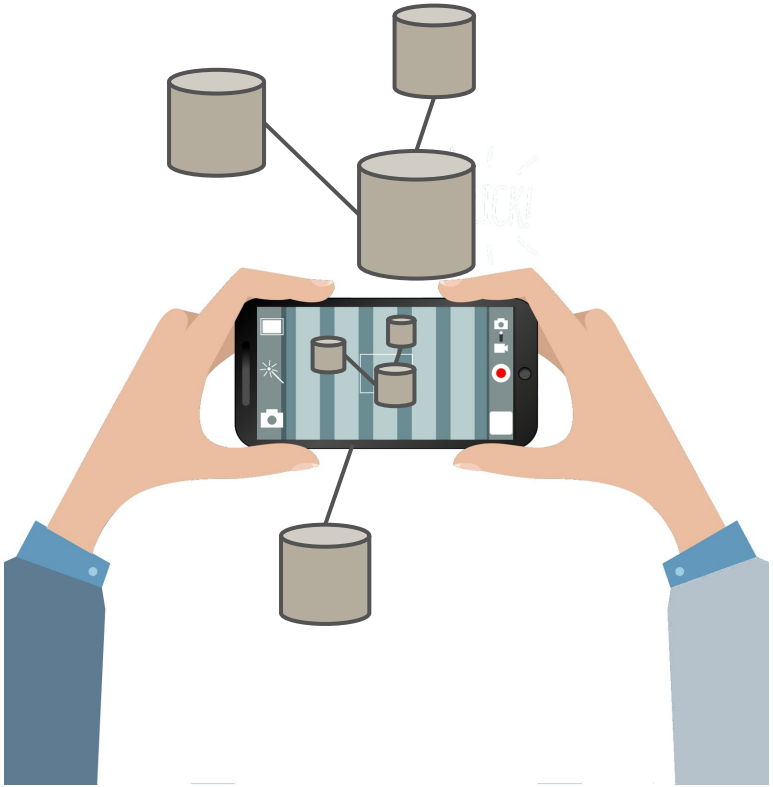

Namenode
DataNode 1
DataNode 2
DataNode 3
Blocks


Namenode 1
Namenode 2
pool 1
pool 2
DataNode 1
DataNode 2
DataNode 3
Blocks


Namespace
Block Storage
Namenode
NS
Block Management
Datanode
...
Datanode
Storage


Namespace
Block Storage
NN-1
NN-k
NN-n
NS1
NS k
Foreign NS n
Pool 1
Pool k
Pool n
Block Pools
Datanode 1
Datanode 2
Datanode m
Common Storage

*HDFS Snapshots are read-only **point-in-time copies** of the file system.*



HADOOP IS NOT GOOD FOR...

1. Low-latency Data Access

Applications that require real-time query, and low-latency access to data in tens of milliseconds will not work well with Hadoop.
Hadoop is not a substitute for a database. Database index records that will gains low-latency and fast response.
But if you really want to replace the database for real time needs, try HBase, which is a column-oriented database for random and real time read/write.

2. Structured Data

Hadoop is not fit for structured data with strong relationship. Hadoop works well for semi-structured and unstructured data. It stores data in files, doesn't index them like RDBMS. Therefore, each ad hoc query for Hadoop is processed by MapReduce job which will bring the latency cost.

3. When data isn't that big

How big the data is big enough for Hadoop? The answer is TB or PB. When your analytics data is only tens of GB, Hadoop is heavy. Don't follow the fashion and use Hadoop, just follow your requirements.

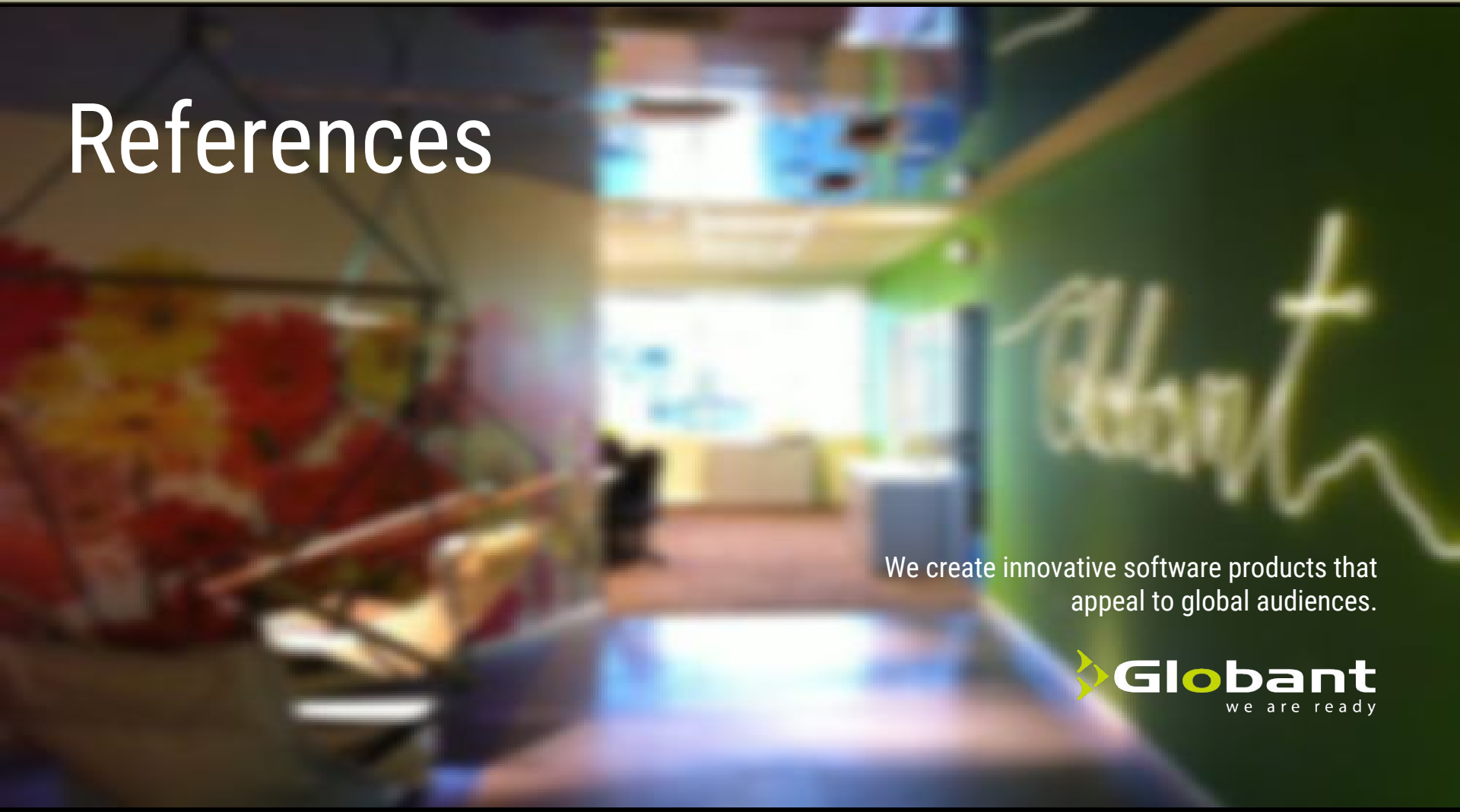
4. Too many small files

When there are too many small files, the NameNode will hit its memory limit where the block map and the metadata are hosted. And to handle the NameNode bottleneck, Hadoop introduces HDFS Federation.

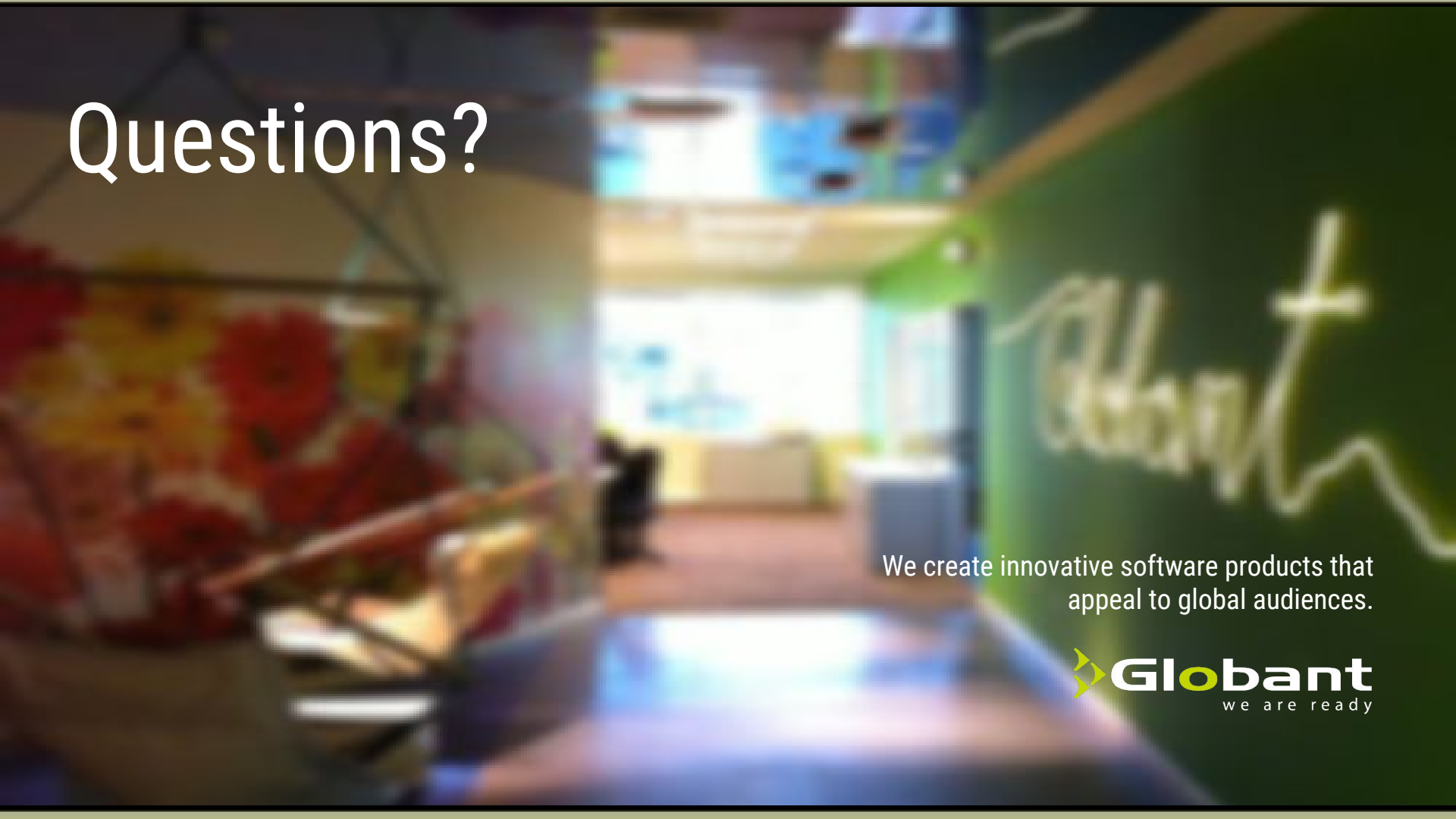
5. MapReduce may not the best choice

MapReduce is a simple programming model in parallel. But for MapReduce parallelism, you need to make sure each MR job and the data where the job runs on is independent from all the others. Every MR shouldn't have dependencies.
But if you want to do some data sharing during MR, you can do like this:
- Iteration: run multiple MR jobs, with the output of one being the input of the next MR.
- Shared state information. But don't share information in memory, since each MR job is run on single JVM.

References


We create innovative software products that
appeal to global audiences.
Globant
we are ready

- **Hadoop: The Definitive Guide, 3rd Edition.** O'Reilly Media / Yahoo Press - Online @ Globant's Big Data Training Site
- **Guía de Instalación de Hadoop**. Globant Team - Online @ Globant's Big Data Training Site
- *To be continued...*


Questions?
We create innovative software products that appeal to global audiences.
Globant
we are ready

*Thank **You**!*

Globant
we are ready