Università degli Studi di Milano - Bicocca

Dipartimento di Informatica Sistemistica e Comunicazione

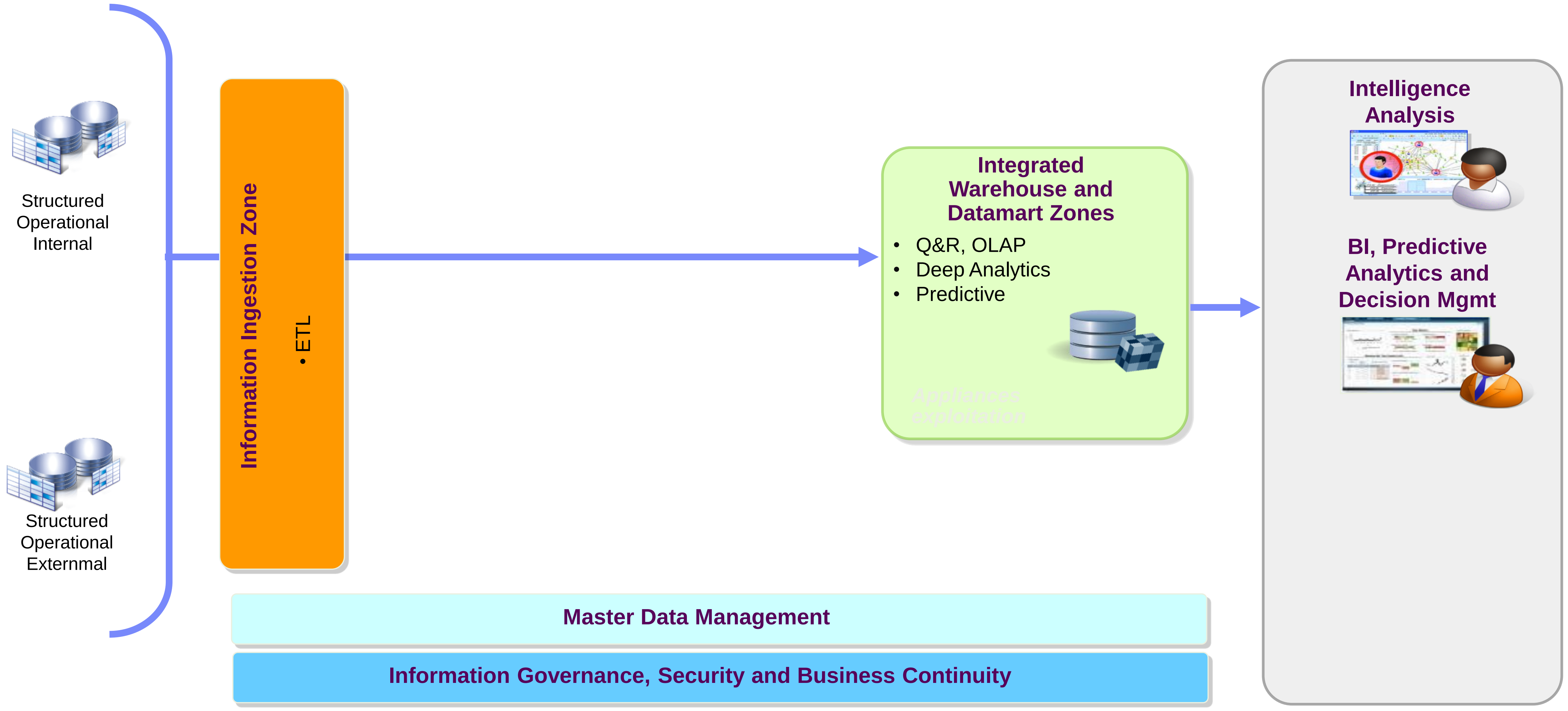
Infosphere BigInsights

Carlo Patrini – IBM Analytics (carlo.patrini@it.ibm.com

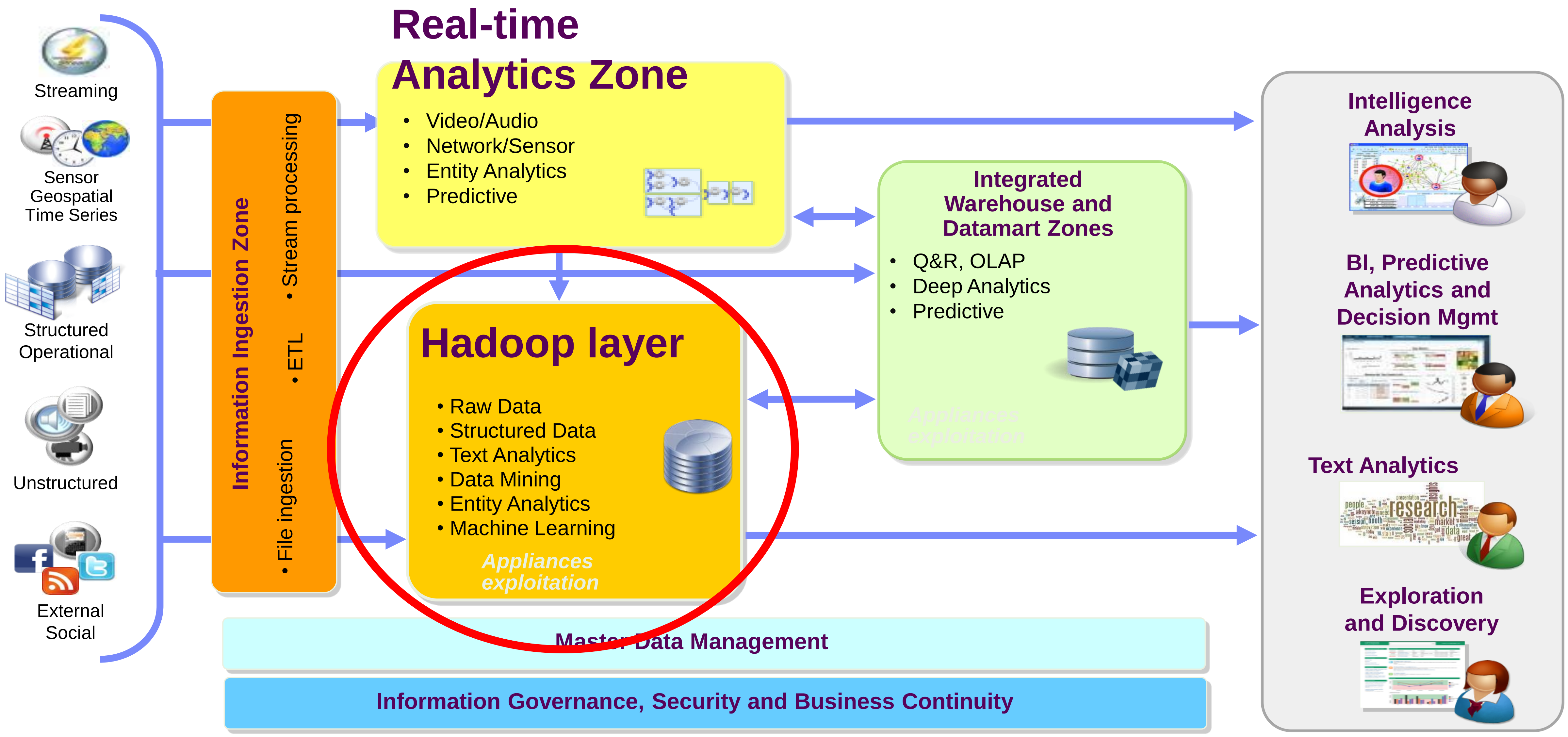
Francesco Airoldi – IBM Sales & Distribution (airfranz@it.ibm.com

Giugno 2015

La tradizionale filiera BI



La filiera delle IBM Big Data & Analytics



Biginsights basato su Hadoop ….. perchè ?

- CPU istruzioni al secondo – miglioramenti significativi

1990 44 Mips at 40 Mhz
2000 3.562 Mips at 1.2 Ghz
2010 147.600 Mips at 3.3 Ghz

- RAM Memory - miglioramenti significativi

1990 640 K
2000 64 Mb
2010 8-32 GB

- Disk capacity - miglioramenti significativi

1990 20 MB
2000 10 GB
2010 1 TB

- Disk latency (velocità di leggere e scrivere su disco) - miglioramenti poco significativi

Negli ultimi 7-10 anni non ci sono state enormi migliorie
correntemente la velocita è di circa 70 – 80 MB / sec

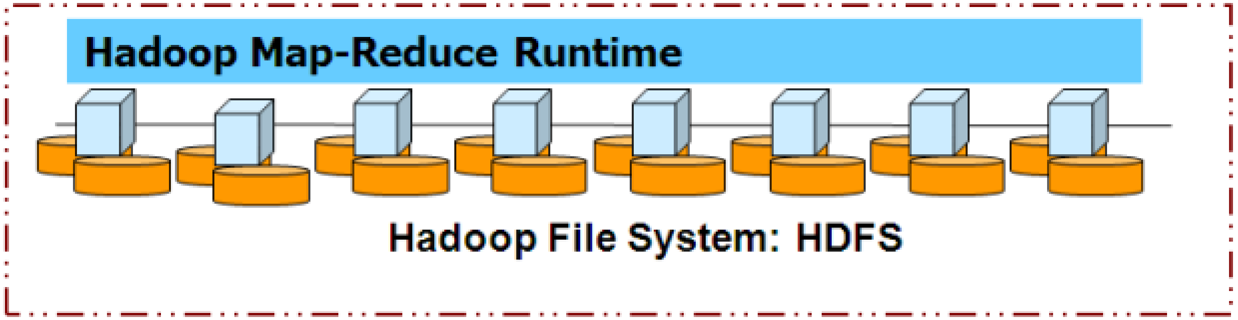
Quanto tempo ci vuole per scandire 1 TB ?

- 1 TB (at 80 MB / sec)

1 disk	3.4 hours
10 disks	20 min
100 disks	2 min
1000 disks	12 sec

- Per ovviare alla Disc Latency la risposta è la ..elaborazione parallela

- Hadoop : un nuovo modo per memorizzare ed elaborare i dati
  - Scritto in Java
  - Progettato per lavorare su hardware non specializzato
  - Gira in ambiente Linux
  - Scalabile, Flessibile,Robusto



What is Hadoop?

- **Apache Hadoop = free, open source framework** for data-intensive applications
  - Inspired by **Google technologies** (MapReduce, GFS)
  - **Yahoo has been** the largest contributor to the project (Doug Cutting),
  - Well-suited to batch-oriented, read-intensive applications
  - Originally built to address scalability problems of Nutch, an open source Web search technology

- Enables applications to work with thousands of nodes and petabytes of data in a highly parallel, cost effective manner
  - CPU + disks of commodity box = Hadoop “node”
  - Boxes can be combined into clusters
  - New nodes can be added as needed without changing
    - Data formats
    - How data is loaded
    - How jobs are written

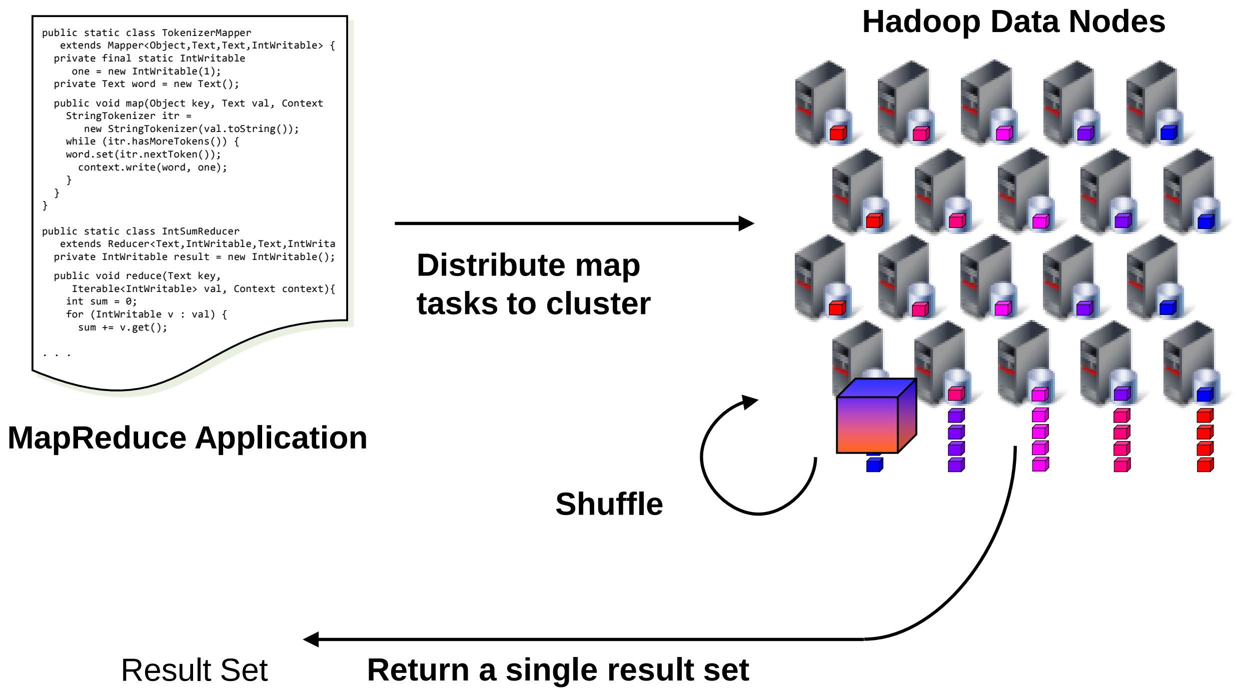
Two Key Aspects of Hadoop

- **MapReduce framework**
  - MapReduce is a software framework introduced by Google to support distributed computing on large data sets of clusters of computers.
  - How Hadoop understands and assigns work to the nodes (machines)

- **Hadoop Distributed File System = HDFS**
  - Where Hadoop stores data
  - A file system that spans all the nodes in a Hadoop cluster
  - It links together the file systems on many local nodes to make them into one big file system

Hadoop ed il paradigma Map Reduce

- I dati sono memorizzati su un sistema distribuito di server
- Le funzioni elaborative vengono inviate dove ci sono I dati
- Ogni server elabora I dati di propria competenza e condivide i risultati
- Il sistema può scalare raggiungendo migliaia di nodi e PB di dati



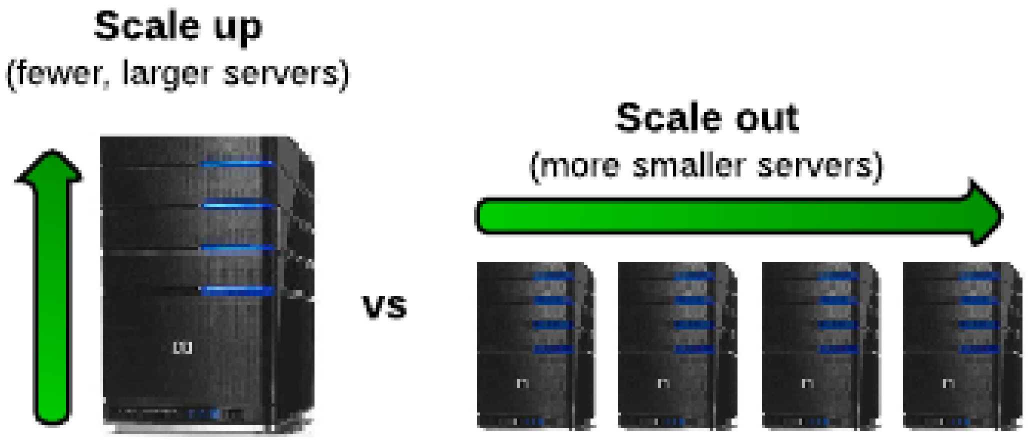
1. Map Phase
   (spezza il job in piccole parti)
2. Shuffle
   (riordina I risultati parziali per le elaborazione finale)
3. Reduce Phase
   (rielabora il tutto per ottenere un singolo risultato)

Scale-out vs scale-up

- ❑ **Just reading 100 terabytes is slow**
  - –Standard computer (100 MBPS) ~11 days
  - –Across 10Gbit link (high end storage) 1 day
  - –1000 standard computers 15 minutes!
- ❑ **Seek times for random disk access is a problem**
  - –1 TB data set with 10^{10} 100-byte records

    Updates to 1% would require 1 month

    Reading and rewriting the whole data set would take 1 day[*]
- ❑ **One node is not enough!**
- ❑ **Need to scale out not up!**



Scaling out

- **Bad news: nodes fail, especially if you have many**
  - Mean time between failures for 1 node = 3 years, 1000 nodes = 1 day
  - Super-fancy hardware still fails and commodity machines give better performance per dollar

- **Bad news II: distributed programming is hard**
  - Communication, synchronization, and deadlocks
  - Recovering from machine failure
  - Debugging
  - Optimization

- **Bad news III: repeat for every problem**

An alternative solution

In 2003/4 Google publishes seminal whitepapers on a new programming paradigm to handle data at Internet Scale.

Original assumptions:

- scale-out architecture
- build a super computer on commodity hardware (commodity != low-end - not tied to expensive, proprietary offerings from a single vendor)
- hides system-level details from the developers (the datacenter is the computer)
- move processing to the data (because cluster have limited bandwith)
- process data sequentially (seeks are expensive)
- expect failure
- shared-nothing architecture (processing tasks have no dependency on one other)

Google: 3 significant papers

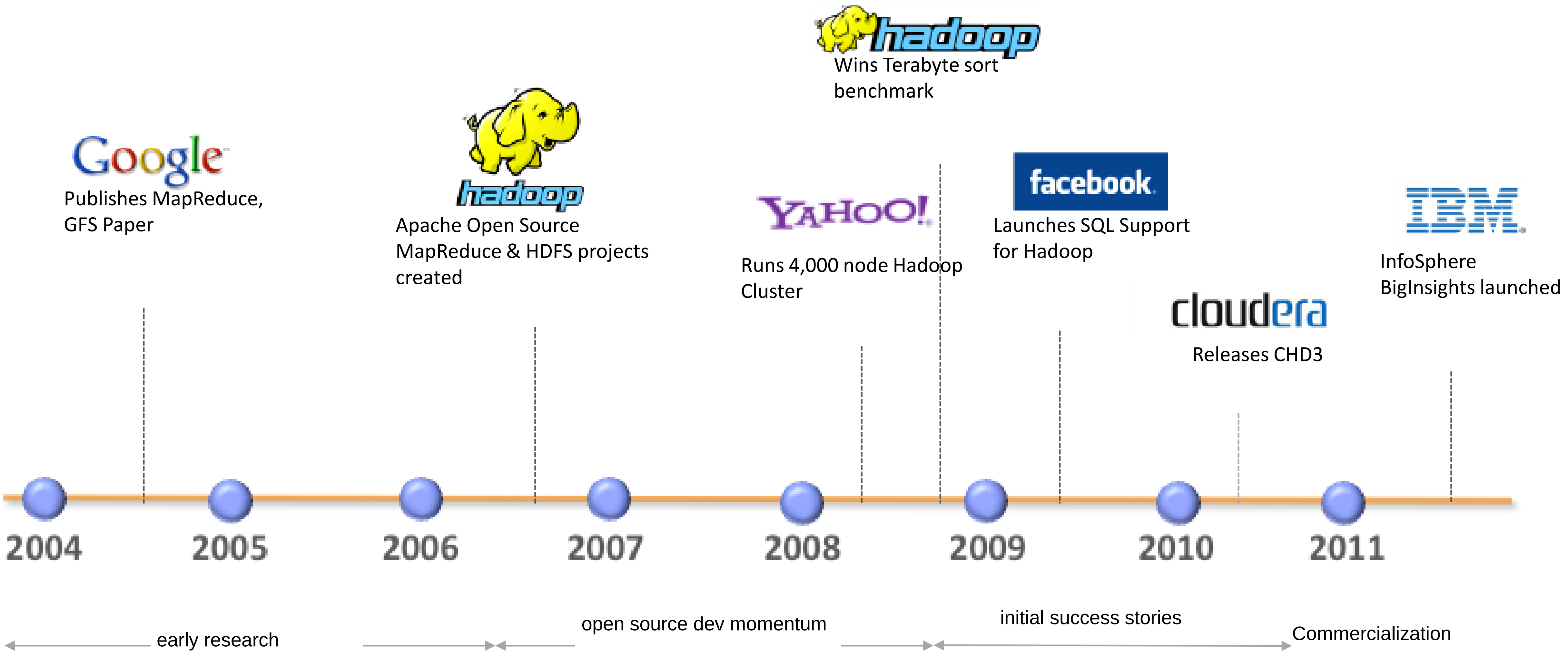
The Google File System
Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Lake George, NY, October, 2003.

MapReduce: Simplified Data Processing on Large Clusters
Jeffrey Dean and Sanjay Ghemawat
San Francisco, CA, December, 2004.

"A simple and powerful interface that enables automatic parallelization and distribution of large-scale computations, combined with an implementation of this interface that achieves high performance on large clusters of commodity PCs"

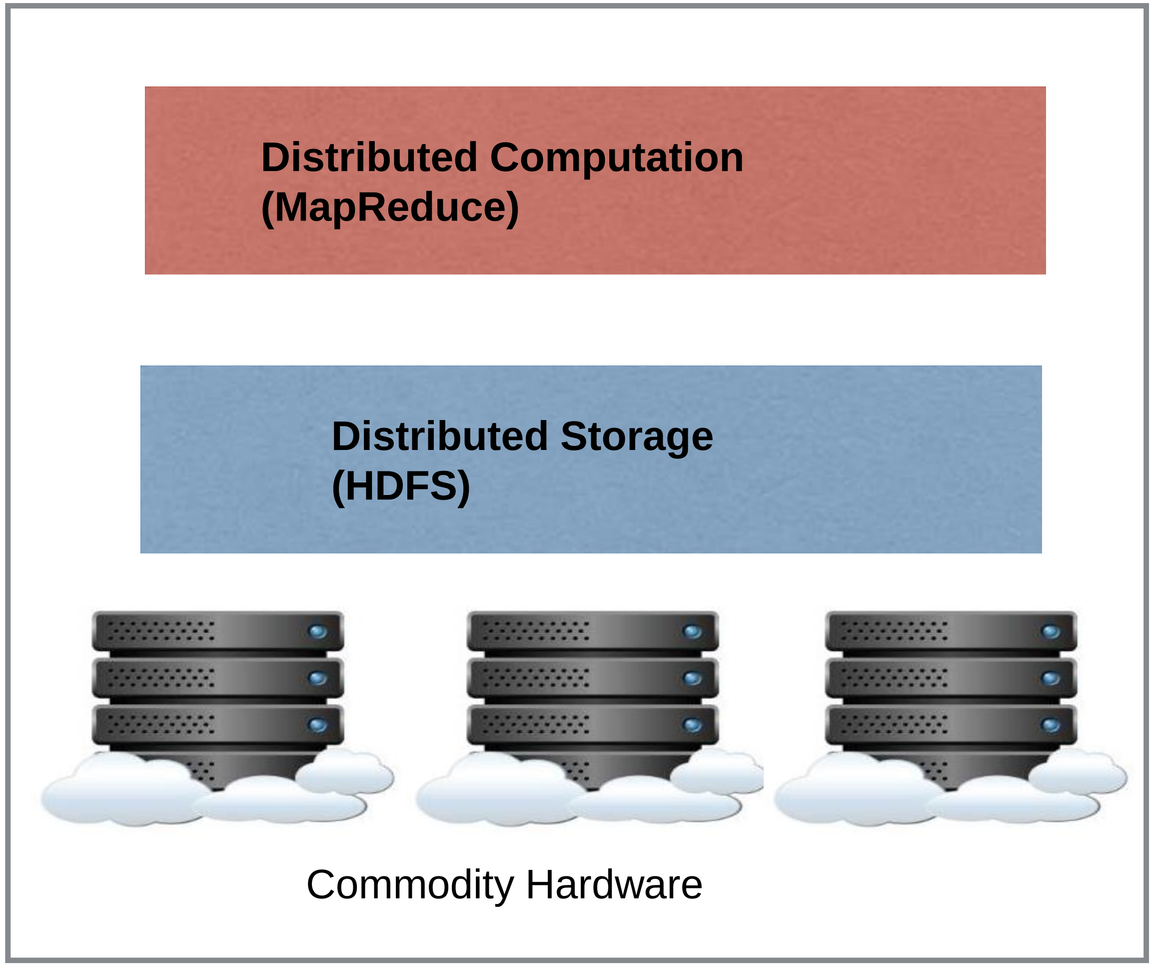
Bigtable: A Distributed Storage System for Structured Data
Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber
Seattle, WA, November, 2006.

A brief history of Hadoop



So what exactly is Hadoop?

A framework for running applications (aka jobs) on large clusters built on **commodity hardware** capable of processing **petabytes of data**



- it implements a computational paradigm named **Map/Reduce**, where the application is divided into self contained units of work, each of which may be executed or re-executed on any node in the cluster
- it provides a distributed file system (**HDFS**) that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster.
- node **failures are automatically handled** by the framework.

Some figures from production sytems

The largest known production environments:

- By 2011: Facebook, which consisted of 4,000 nodes (including 8- and 16-core CPUs) and supported 21PB of storage
- By (October) 2013: Yahoo!, spanning more than 35,000 nodes

Facebook Datawarehousing Hadoop cluster (2011 data):

- 12 TB of compressed data added per day
- 800 TB of compressed data scanned per day
- 25,000 map-reduce jobs per day
- 65 millions files in HDFS
- 30,000 simultaneous clients to the HDFS NameNode

Nowadays Facebook Datawarehouse is 300 PB and add about 600 TB of compressed data per day

Hadoop is not for all types of work

Not good for

- Not to process transactions (random access)
- Not good when work cannot be parallelized
- Not good for low latency data access
- Not good for processing lots of small files
- Not good for intensive calculations with little data

Good for

- Data intensive, bulk-processing oriented, typically "embarrassingly parallel"
- Examples
  - Index building at Google and Yahoo!
  - Article clustering
  - Statistical machine translation
  - Spam detection
  - Ad optimization
  - Natural Language Processing
  - Image analysis
  - OCR
  - IBM's Watson

Hadoop and the Apache ecosystem

HDFS Storage

- Distributed
- Reliable
- Commodity gear

+

MapReduce

- Parallel Programming
- Fault Tolerant

+

New emerging workloads

core

Hadoop term is also used for a family of related projects that fall under the umbrella of infrastructure for distributed computing and large-scale data processing

+

Oozie
Workflow

Zookeeper
Service coordination

HIVE

HBASE

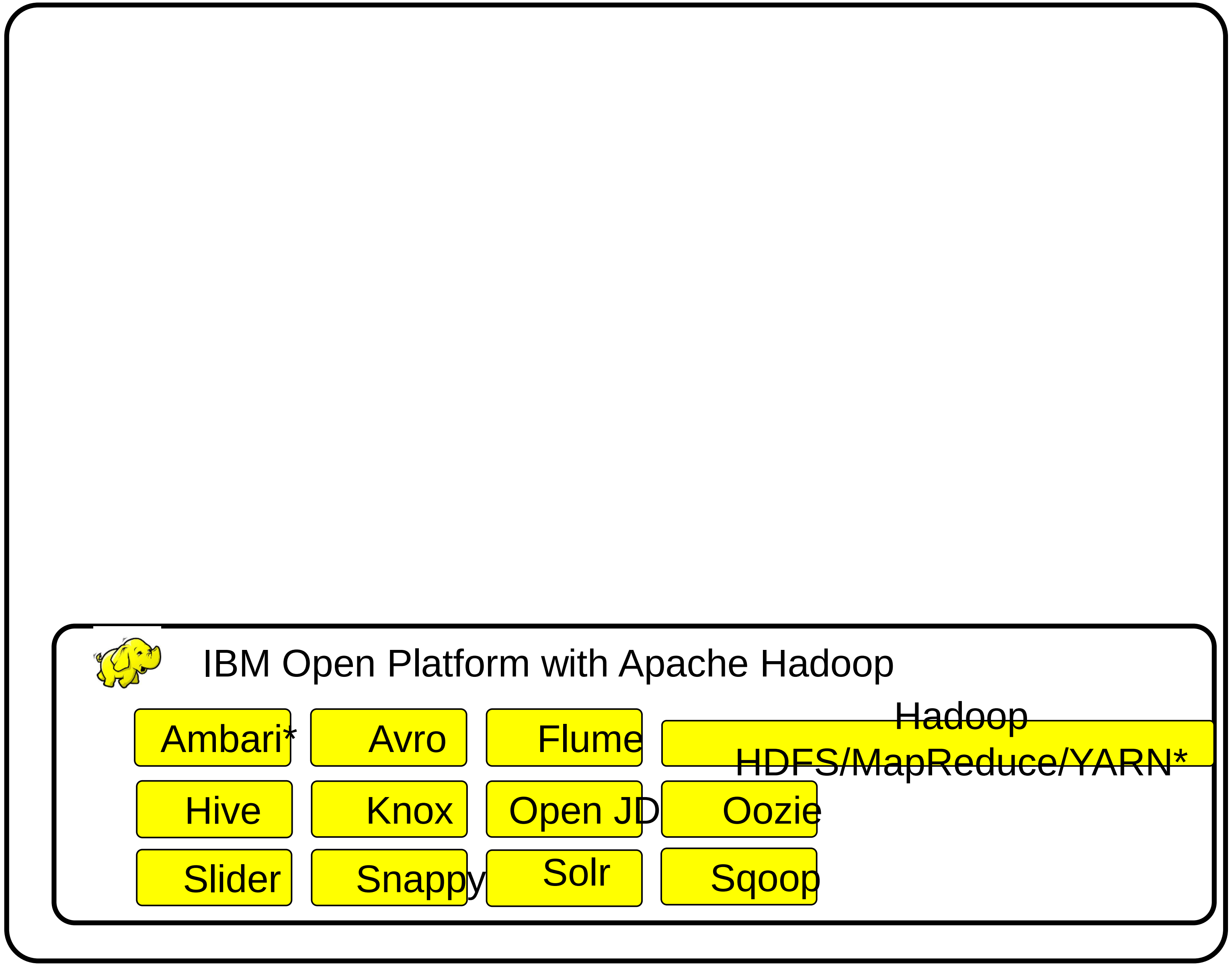
Sqoop
Integration

Flume, Chukwa, Scribe
Data collection

+ . . .

related

HADOOP

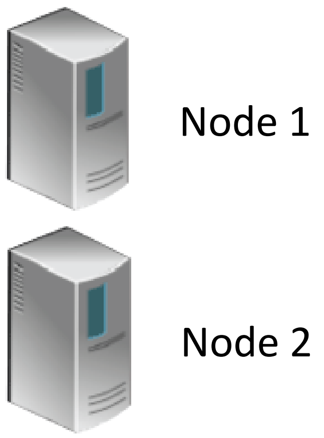

IBM Open Platform with Apache Hadoop
Ambari*
Avro
Flume
Hadoop HDFS/MapReduce/YARN*
Hive
Knox
Open JD
Oozie
Slider
Snappy
Solr
Sqoop

HDFS - Terminology review

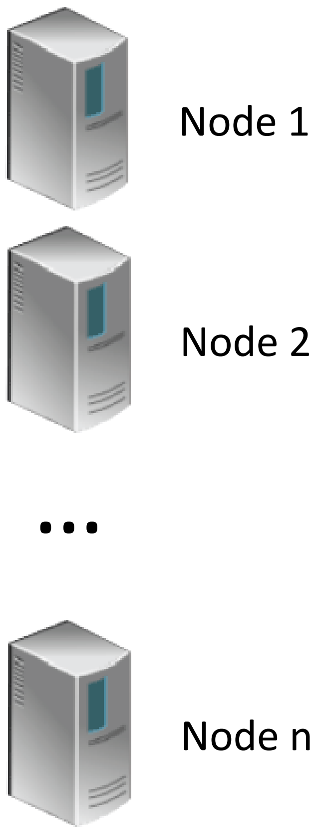
Node 1

6

HDFS - Terminology review

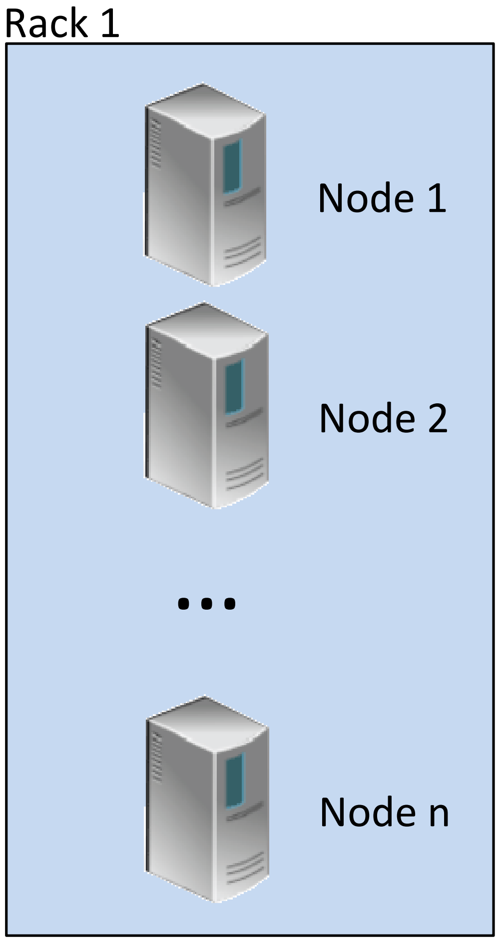


HDFS - Terminology review



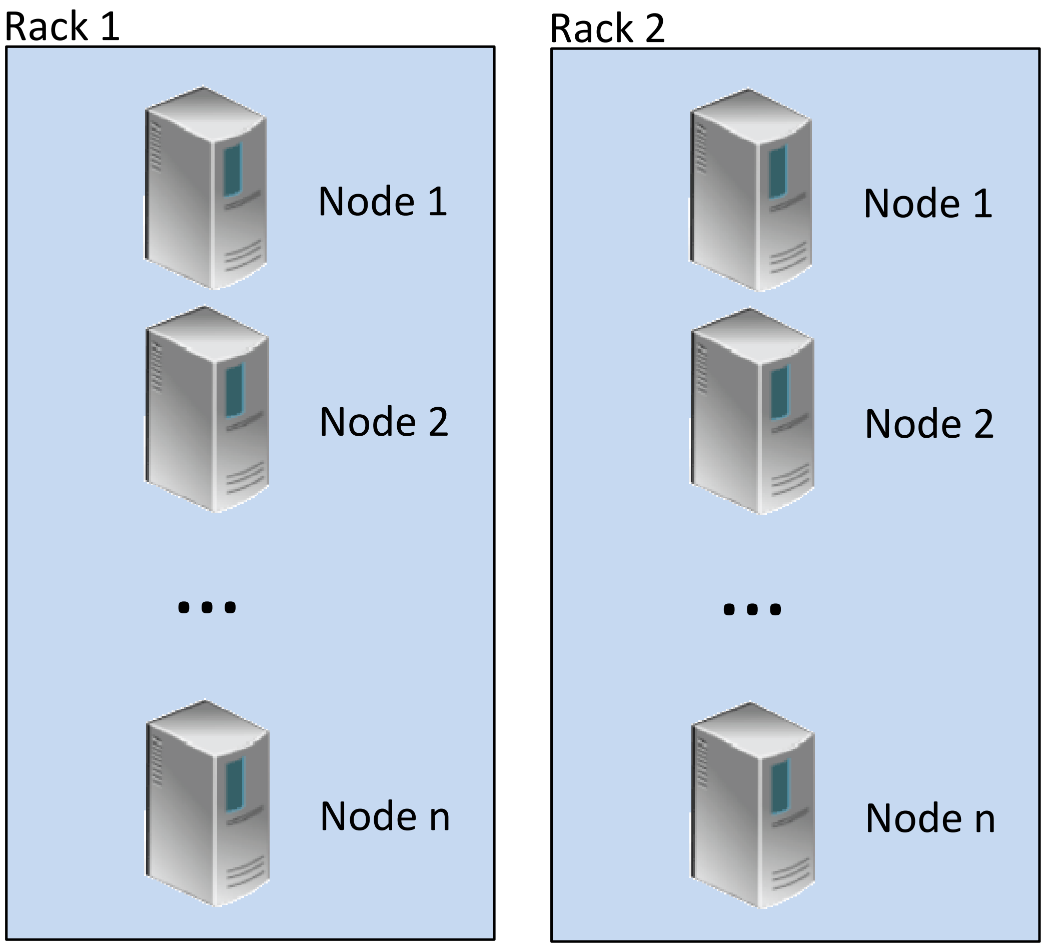
HDFS - Terminology review

Università degli Studi di Milano - Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione

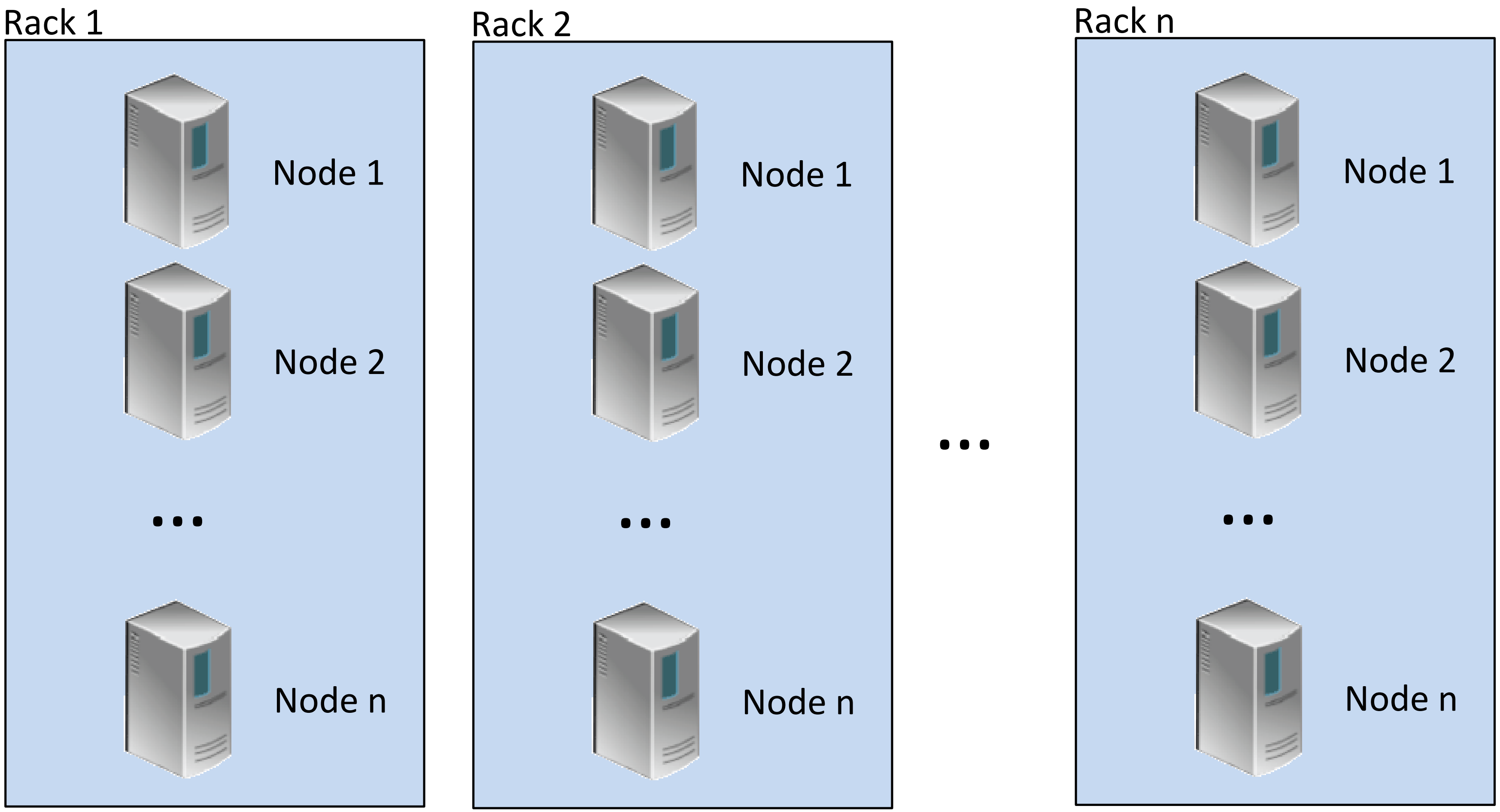


9

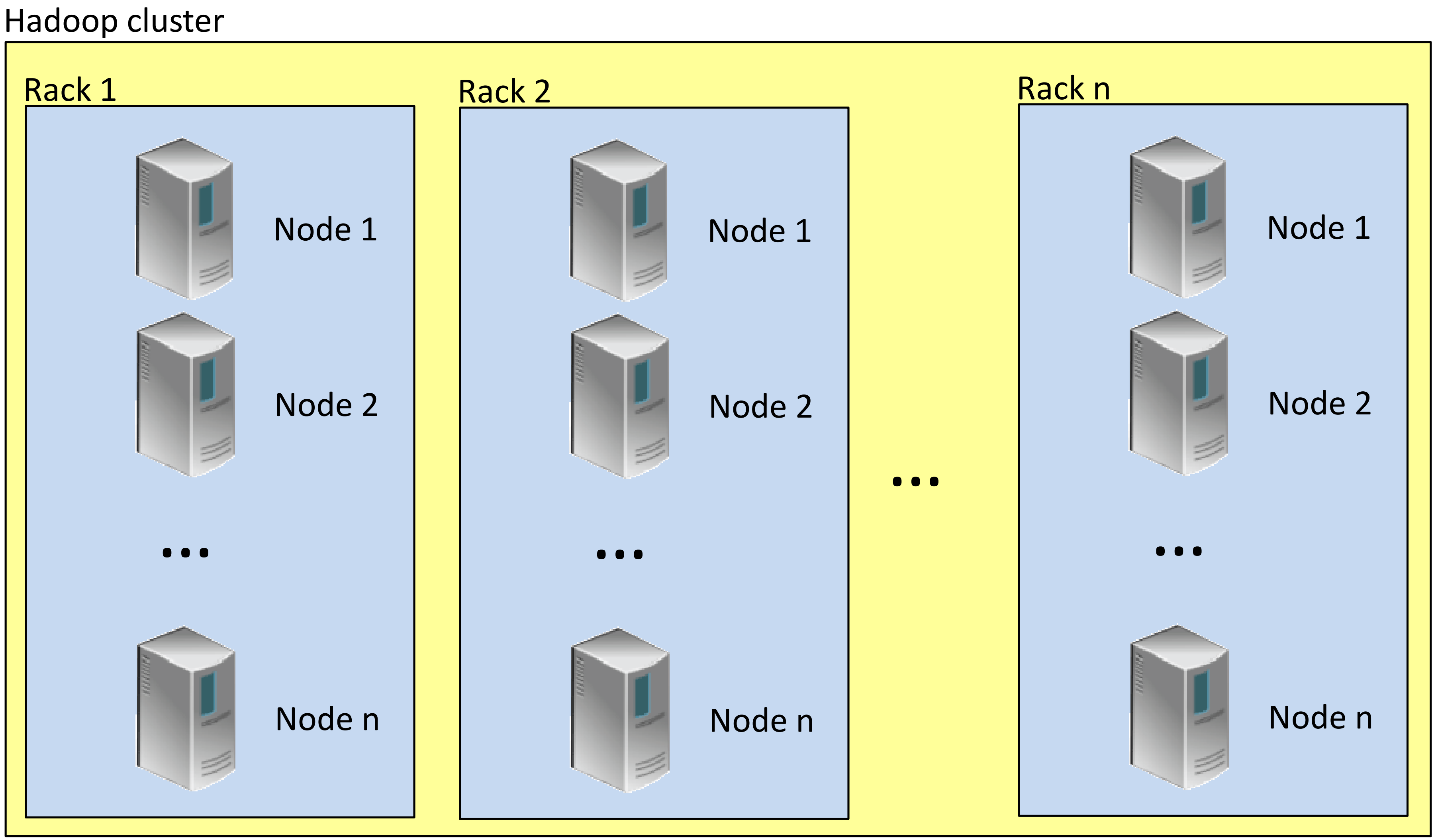
HDFS - Terminology review



HDFS - Terminology review

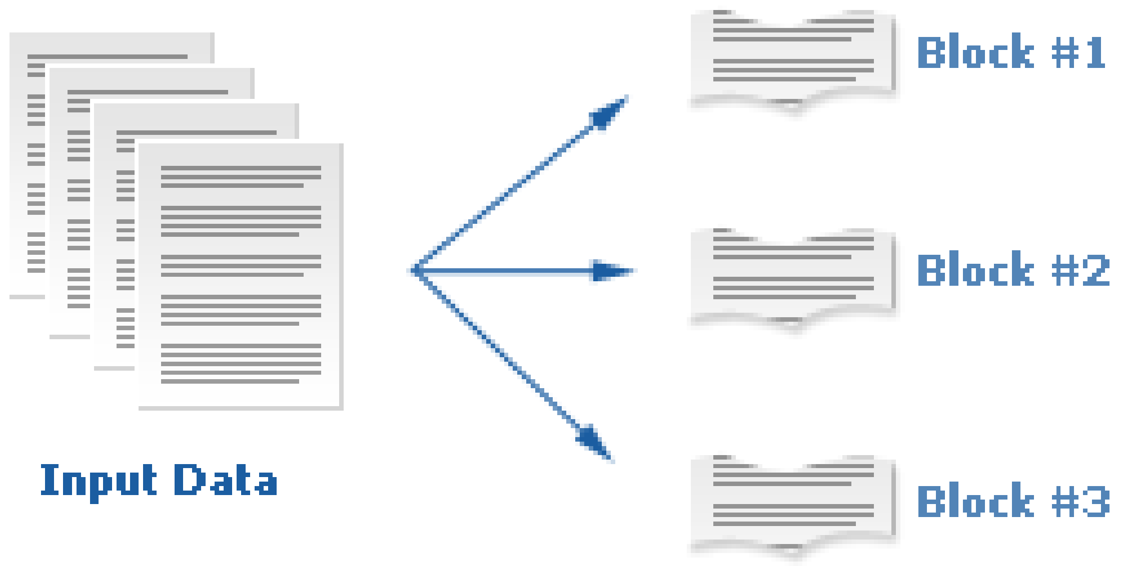


HDFS - Terminology review



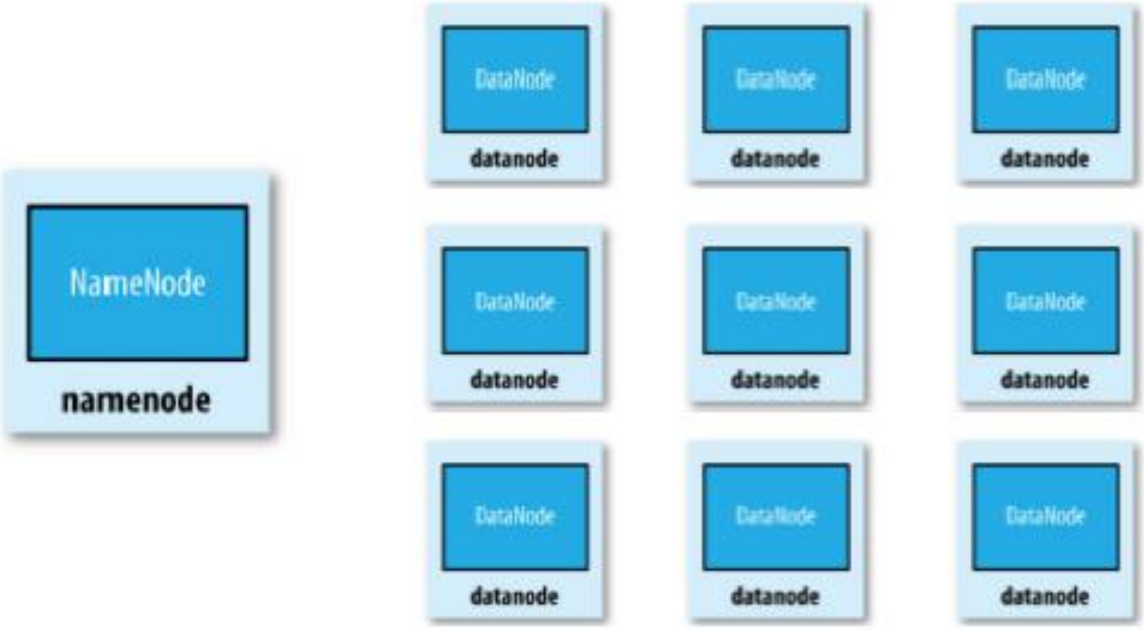
HDFS

- Hadoop file system that runs on top of existing file system
- Designed to handle very large files with streaming data access patterns
- Uses <u>blocks</u> to store a file or parts of a file
- Can create, delete, copy, but NOT update



HDFS: organization

- Files are divided into chunks (blocks)
- Chunks are replicated at different compute nodes (usually 3+)
- Nodes holding copies of one chunk are located on different racks
- Chunk size and the degree of replication can be decided by the user
- A special node (the **NameNode**) stores, for each file, the positions of its chuncks



NameNode

- **Entire metadata is kept in RAM**
  - Ensure enough RAM in NameNode
  - If run out of RAM, NameNode will crash
- **NameNode mainly consists of:**
  - fsimage: Contains the metadata <u>on disk</u> (not exact copy of what is in RAM, but a checkpoint copy)
  - edit logs: Records all write operations, synchronizes with metadata in RAM after each write
- **In case of 'power failure' on NameNode**
  - Can recover using fsimage + edit logs
- **Need to format NameNode to use it:**
  - hadoop namenode -format

DataNode

Università degli Studi di Milano - Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione

- Many per Hadoop cluster
- Manages blocks with data and serves them to clients
- Periodically reports to NameNode the list of blocks it stores
- Use inexpensive commodity hardware for this node

MapReduce flow

1) Some number of Map tasks each are given one or more chunks of data.

2) These Map tasks turn the chunk into a sequence of key-value pairs. The way key-value pairs are produced is determined by the code written by the user for the Map function.

3) The key-value pairs from each Map task are collected by a master controller and sorted and grouped by key (Shuffle and sort).

4) The keys are divided among all the Reduce tasks, so all key-value pairs with the same key wind up at the same Reduce task.

5) The Reduce tasks work on one key at a time, and combine all the values associated with that key in some way. The way values are combined is determined by the code written by the user for the Reduce function.

6) Output key-value pairs from each reducer are written persistently back onto the distributed file system

7) The output ends up in r files, where r is the number of reducers. The r files often serve as input to yet another MapReduce job

Example: word count

Problem: counting the number of occurrences for each word in a collection of documents.

Input: a repository of documents, each document is an element

Map: reads a document and emits a sequence of key-value pairs where keys are words of the documents and values are equal to 1:

(w1, 1), (w2, 1), . . . , (wn, 1)

Grouping: groups by key and generates pairs of the form

(w1, [1, 1, . . . , 1]) , . . . , (wn, [1, 1, . . . , 1])

Reduce: adds up all the values and emits:

(w1, k) , . . . , (wn, l)

Output: (w,m) pairs, where w is a word that appears at least once among all the input documents and m is the total number of occurrences of w among all those documents.

Simple data flow example

input

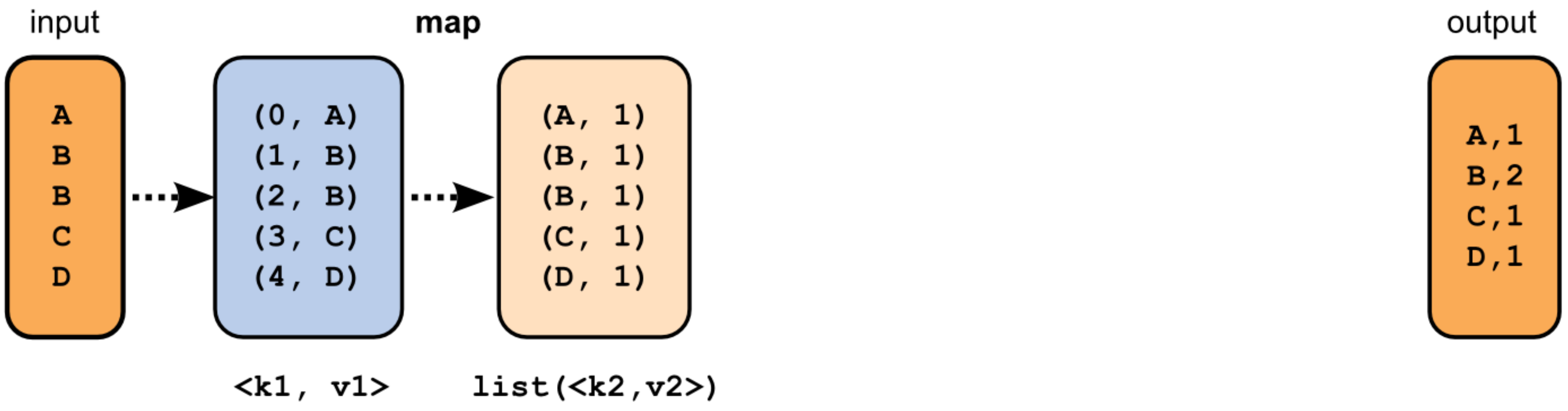
```
A
B
B
C
D
```

output

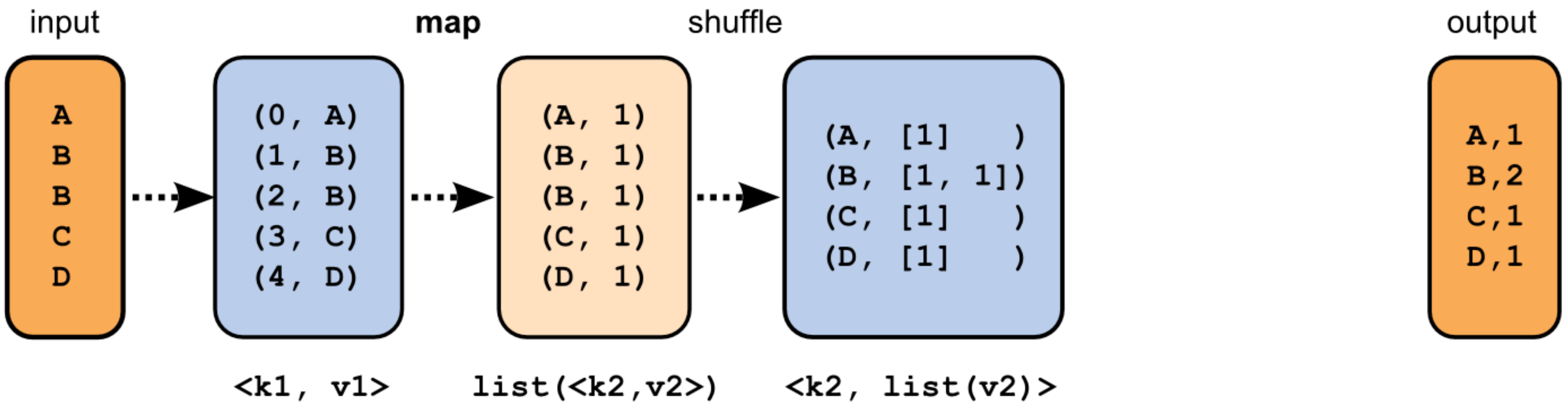
```
A,1
B,2
C,1
D,1
```

Simple data flow example

Simple data flow example

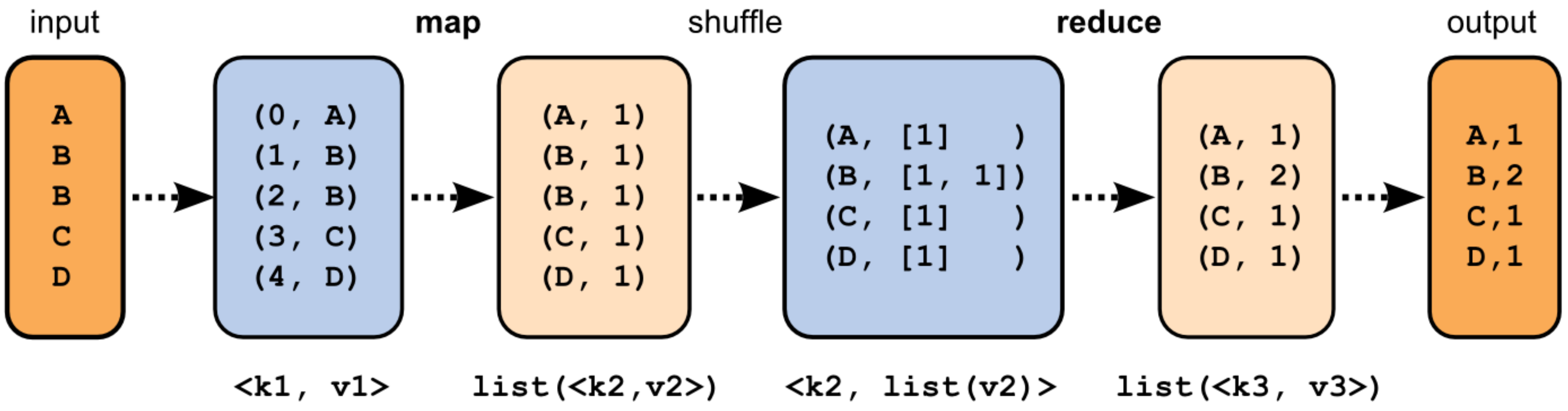


Simple data flow example



53

Simple data flow example



BIGINSIGHTS

IBM InfoSphere BigInsights offre una serie completa di funzionalità di analytics avanzate che consentono alle aziende di analizzare volumi elevati di dati strutturati e non strutturati nel loro formato nativo.
Il software unisce la tecnologia Apache Hadoop open source con le innovazioni IBM, quali analytics dei testi avanzata, IBM BigSheets e Big SQL per la consultazione dei dati e una serie di funzioni amministrative, di sicurezza e prestazioni.
Il risultato è una soluzione economica e facile da utilizzare per l'analytics di big data complessi

InfoSphere BigInsights is 100% standard, open source Hadoop

Avoid proprietary lock-in. BigInsights is based on Open Data Platform and includes the rich tools that Hadoop users expect. Where IBM does provide value-added features, they are carefully implemented so that customers have a choice whether to use IBM enhancements or standard Hadoop functionality.

100% Open Source Hadoop, but Enterprise Grade

IBM BigInsights for Apache Hadoop

SQL on Hadoop
Big SQL – optimized ANSI compliant SQL

Data Visualization
BigSheets spreadsheet interface

Predictive Modeling
Big R, Machine Learning

Text Analytics
Advanced text processing with AQL, Text extraction web interface

Storage Integration
GPFS - POSIX Distributed Filesystem

Enterprise Manageability
Adaptive MapReduce, Multi-tenant scheduling

Application Tooling
Toolkits and accelerators

Search & Exploration
Watson Explorer

Real-time Analytics
InfoSphere Streams

Data Governance and Security
DataClick, LDAP, Secure cluster

IBM Open Platform with Apache Hadoop

Ambari*
Avro
Flume
Hadoop HDFS/MapReduce/YARN*
Hive
Knox
Open JD
Oozie
Slider
Snappy
Solr
Sqoop

VESTAS ha ottimizzato una analisi predittiva su 2.5 Peta

http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=AB&infotype=PM&appname=SWGE_IM_IP_USEN&htmlfid=IMC14702USEN&attachment=IMC14702USEN.PDF#loaded

Open Data Platform Initiative

Why is IBM involved?

Strong history of leadership in open source & standards

Supports our commitment to open source currency in all future releases

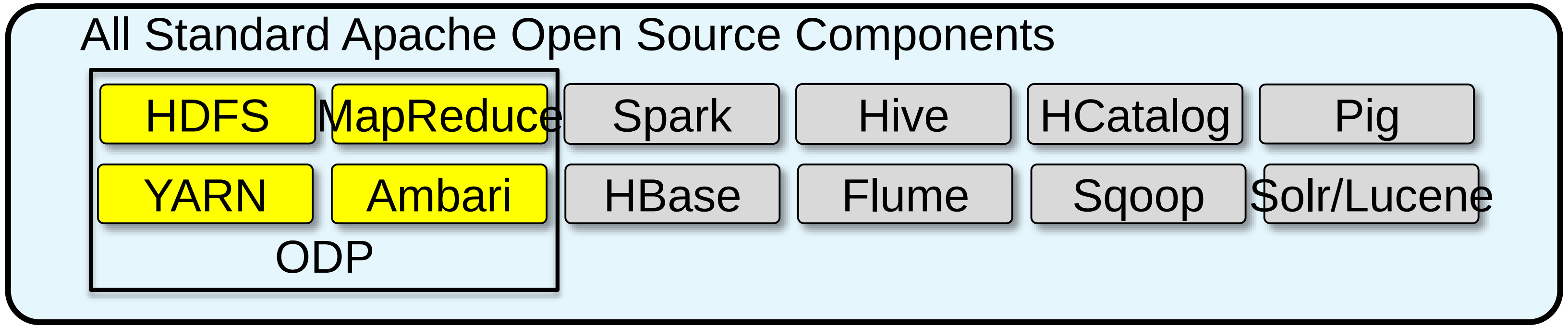
Accelerates our innovation within Hadoop & surrounding applications

Open Data Platform (ODP) vs. Apache Software Foundation (ASF)

ODP supports the ASF mission

ASF provides a governance model around individual projects without looking at ecosystem

ODP aims to provide a vendor-led consistent packaging model for core Apache components as an ecosystem

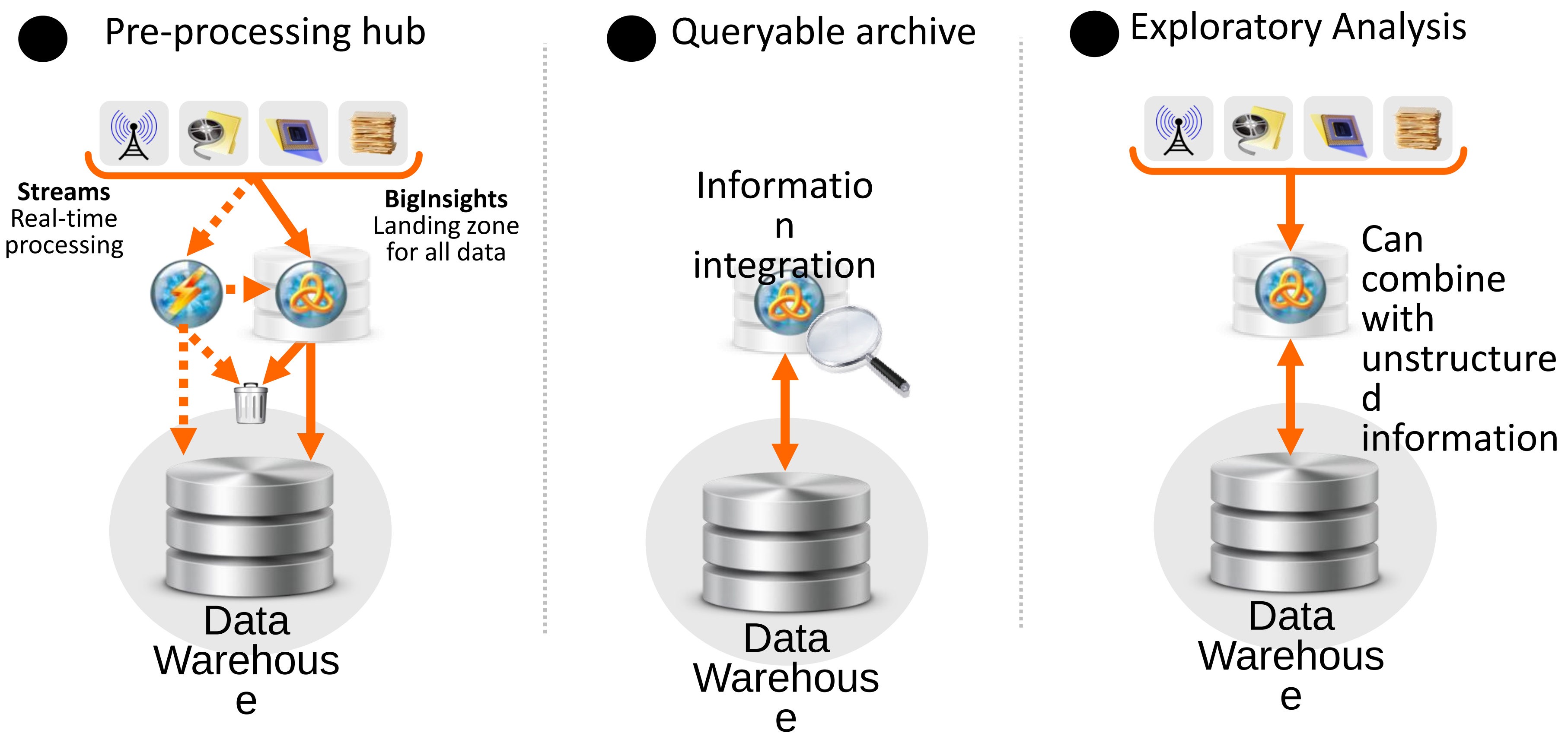


Big SQL – Lightning fast, ANSI compliant, native Hadoop formats

While some Hadoop vendors are building their own SQL-on-Hadoop implementations from scratch, Big SQL is ANSI compliant, lightning fast and runs on native Hadoop file formats. Brought to you by the inventors of SQL, Big SQL provides federated access so customers can query IBM and third party data sources with the same feature rich SQL.

SQL Access for Hadoop: Why?

- Data warehouse modernization is a leading Hadoop use case



- Limited availability of skills in MapReduce, Pig, etc.
- SQL opens the data to a much wider audience
  - Familiar, widely known syntax
  - Common catalog for identifying data and structure

IBM InfoSphere BigInsights – Big SQL

IBM's SQL for Hadoop

- Makes Hadoop data accessible to a wider audience
- Familiar, widely known syntax
- Leverage native Hadoop data sources

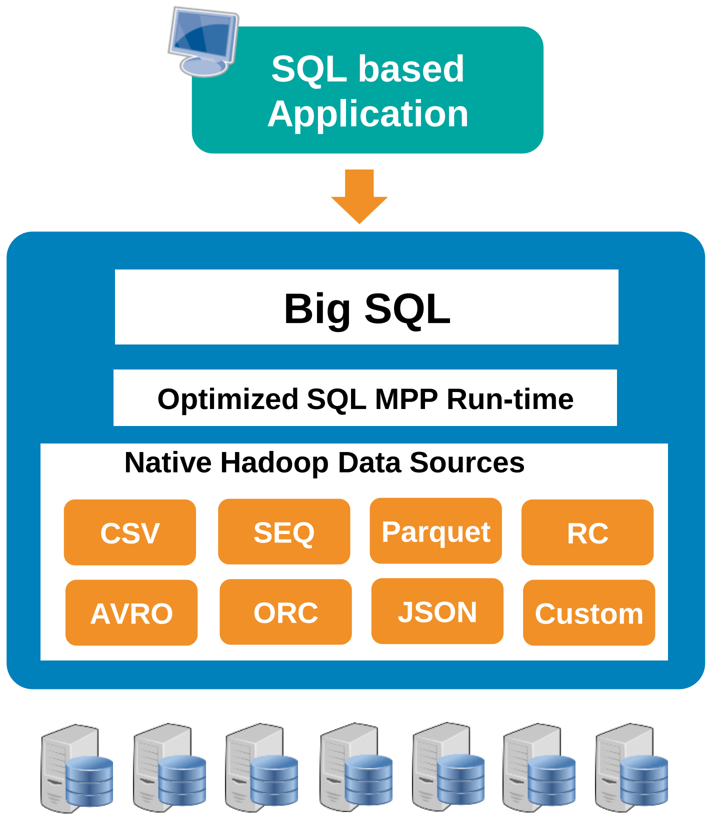
Complements the Data Warehouse

- Exploratory analytics
- Sandbox, Data Lake

Included in BigInsights

Use familiar SQL tools

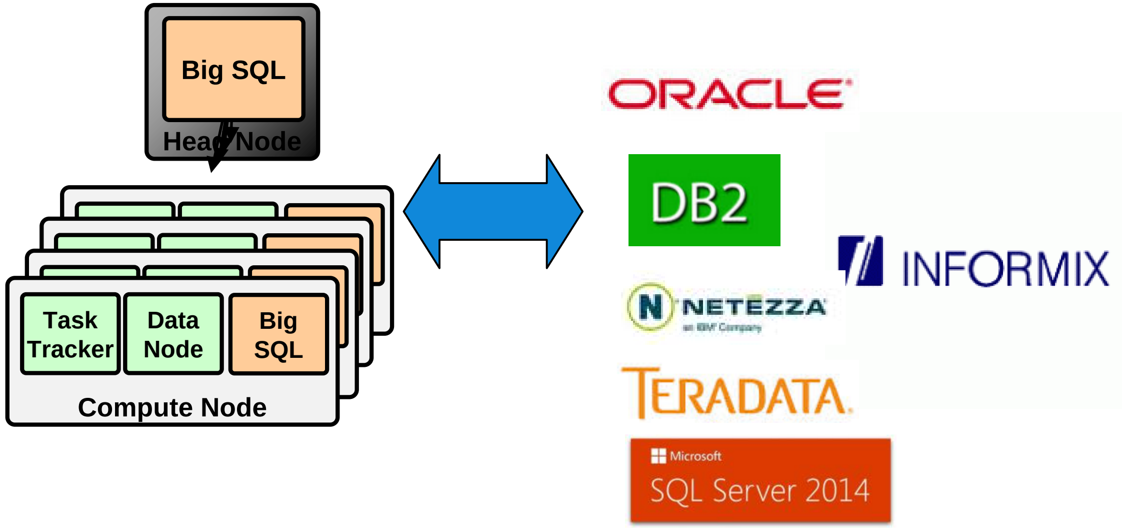
- Cognos, SPSS, Tableau, MicroStrategy



A word about . . . query federation

- Data rarely lives in isolation

- Big SQL transparently queries heterogeneous systems
  - Join Hadoop to RDBMSs
  - Query optimizer understands capabilities of external system
    - *Including available statistics*
  - As much work as possible is pushed to each system to process



45

About the Hadoop-DS Benchmark

- Created by IBM
- The Big Data Decision Support Benchmark (*Hadoop-DS*) is inspired by, and is highly compliant with TPC-DS
  - Fully complies with the TPC-DS schema requirement
  - Uses all 99 queries
  - Meets the multi-user requirement
  - Has been audited by a TPC-DS auditor but as a non-TPC benchmark
- Select deviations from TPC-DS due to Hadoop limitations:
  - No data maintenance operations, referential integrity enforcement, or ACID property validation as these are not feasible with HDFS
  - Additional statistics used
  - Metric adjustments
  - No price/performance measures included
  - **Not an official TPC benchmark result**

IBM Big SQL – Runs 100% of the TPC-DS queries

Competitive environments require significant effort

IBM

Query 01	Query 34	Query 67
Query 02	Query 35	Query 68
Query 03	Query 36	Query 69
Query 04	Query 37	Query 70
Query 05	Query 38	Query 71
Query 06	Query 39	Query 72
Query 07	Query 40	Query 73
Query 08	Query 41	Query 74
Query 09	Query 42	Query 75
Query 10	Query 43	Query 76
Query 11	Query 44	Query 77
Query 12	Query 45	Query 78
Query 13	Query 46	Query 79
Query 14	Query 47	Query 80
Query 15	Query 48	Query 81
Query 16	Query 49	Query 82
Query 17	Query 50	Query 83
Query 18	Query 51	Query 84
Query 19	Query 52	Query 85
Query 20	Query 53	Query 86
Query 21	Query 54	Query 87
Query 22	Query 55	Query 88
Query 23	Query 56	Query 89
Query 24	Query 57	Query 90
Query 25	Query 58	Query 91
Query 26	Query 59	Query 92
Query 27	Query 60	Query 93
Query 28	Query 61	Query 94
Query 29	Query 62	Query 95
Query 30	Query 63	Query 96
Query 31	Query 64	Query 97
Query 32	Query 65	Query 98
Query 33	Query 66	Query 99

cloudera IMPALA

Query 01	Query 34	Query 67
Query 02	Query 35	Query 68
Query 03	Query 36	Query 69
Query 04	Query 37	Query 70
Query 05	Query 38	Query 71
Query 06	Query 39	Query 72
Query 07	Query 40	Query 73
Query 08	Query 41	Query 74
Query 09	Query 42	Query 75
Query 10	Query 43	Query 76
Query 11	Query 44	Query 77
Query 12	Query 45	Query 78
Query 13	Query 46	Query 79
Query 14	Query 47	Query 80
Query 15	Query 48	Query 81
Query 16	Query 49	Query 82
Query 17	Query 50	Query 83
Query 18	Query 51	Query 84
Query 19	Query 52	Query 85
Query 20	Query 53	Query 86
Query 21	Query 54	Query 87
Query 22	Query 55	Query 88
Query 23	Query 56	Query 89
Query 24	Query 57	Query 90
Query 25	Query 58	Query 91
Query 26	Query 59	Query 92
Query 27	Query 60	Query 93
Query 28	Query 61	Query 94
Query 29	Query 62	Query 95
Query 30	Query 63	Query 96
Query 31	Query 64	Query 97
Query 32	Query 65	Query 98
Query 33	Query 66	Query 99

HIVE

Query 01	Query 34	Query 67
Query 02	Query 35	Query 68
Query 03	Query 36	Query 69
Query 04	Query 37	Query 70
Query 05	Query 38	Query 71
Query 06	Query 39	Query 72
Query 07	Query 40	Query 73
Query 08	Query 41	Query 74
Query 09	Query 42	Query 75
Query 10	Query 43	Query 76
Query 11	Query 44	Query 77
Query 12	Query 45	Query 78
Query 13	Query 46	Query 79
Query 14	Query 47	Query 80
Query 15	Query 48	Query 81
Query 16	Query 49	Query 82
Query 17	Query 50	Query 83
Query 18	Query 51	Query 84
Query 19	Query 52	Query 85
Query 20	Query 53	Query 86
Query 21	Query 54	Query 87
Query 22	Query 55	Query 88
Query 23	Query 56	Query 89
Query 24	Query 57	Query 90
Query 25	Query 58	Query 91
Query 26	Query 59	Query 92
Query 27	Query 60	Query 93
Query 28	Query 61	Query 94
Query 29	Query 62	Query 95
Query 30	Query 63	Query 96
Query 31	Query 64	Query 97
Query 32	Query 65	Query 98
Query 33	Query 66	Query 99

Key points

- With competing solutions, many queries needed to be re-written, some significantly
- Owing to various restrictions, some queries could not be re-written or failed at run-time
- Re-writing queries in a benchmark scenario where results are known is one thing – doing this against real databases in production is another

Works without modification

Minor modification

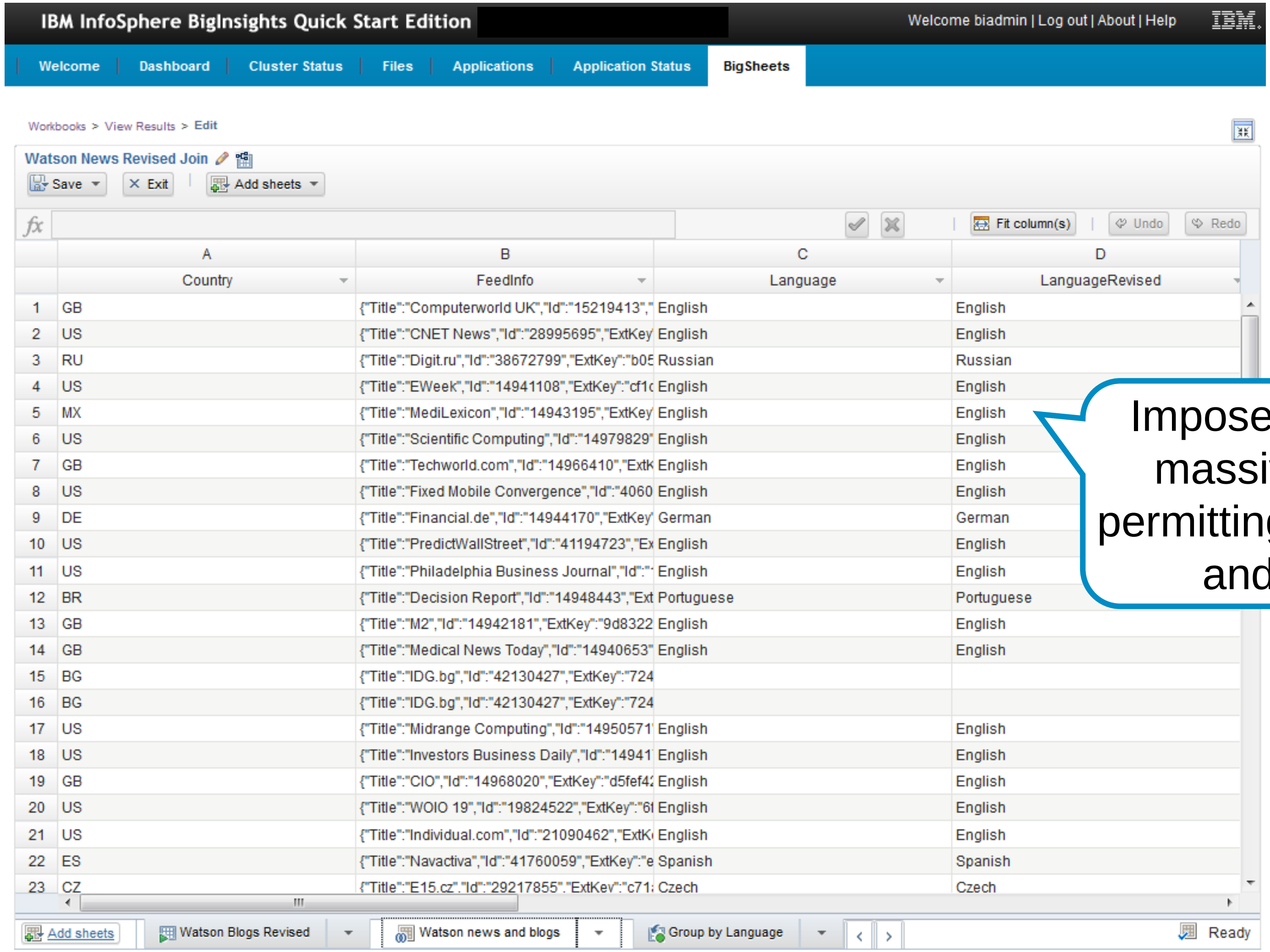
Extensive modification

Not working

BigSheets – Spreadsheet-like data access for business users

Shockingly, there are those who prefer not to code in Java or write Pig scripts to get data from Hadoop. BigSheets provides an easy-to-use spreadsheet interface allowing business users to extract, manipulate and visualize data from a variety of Hadoop and non-Hadoop data sources.

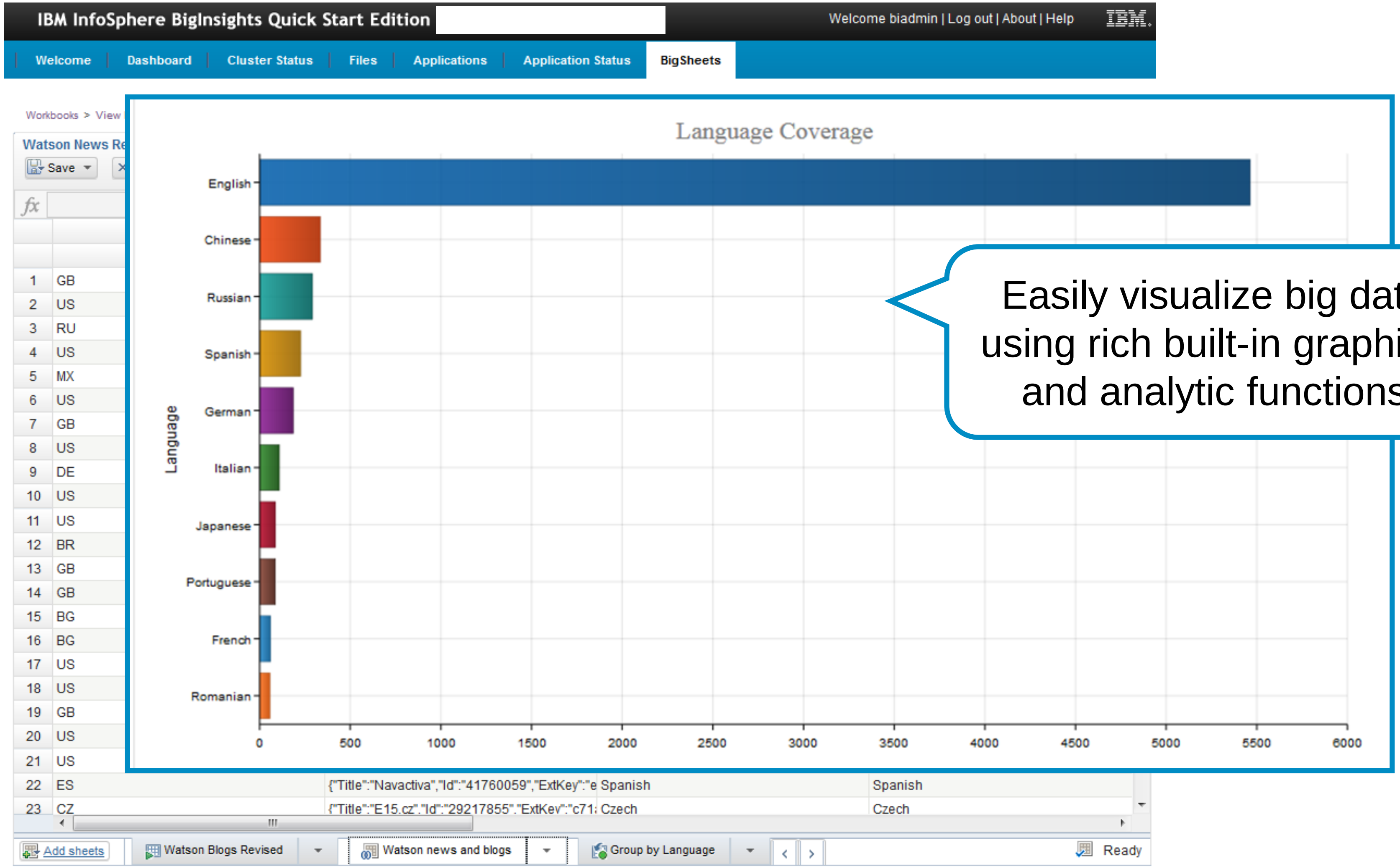
Spreadsheet style analysis tool for business users



Impose structure on massive datasets permitting manipulation and analysis

Università degli Studi di Milano - Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione

Spreadsheet style analysis tool for business users



Easily visualize big data using rich built-in graphing and analytic functions

Big Text – Simplify text analytics and natural language processing

Building applications able to extract meaning from text is hard. Rather than build your own solution, stand on the shoulders of the inventors of Watson, the reigning Jeopardy champ. Build your own advanced analytic applications, parse jargon-laden text in multiple languages, and do what your competitors cannot.

BigInsights and Text Analytics

- Distills structured info from unstructured text
  - Sentiment analysis
  - Consumer behavior
  - Illegal or suspicious activities
  - …
- Parses text and detects meaning with annotators
- Understands the context in which the text is analyzed
- Features pre-built extractors for names, addresses, phone numbers, etc.

Unstructured text (document, email, etc)

Football World Cup 2010, one team distinguished themselves well, losing to the eventual champions 1-0 in the Final. Early in the second half, Netherlands' striker, Arjen Robben, had a breakaway, but the keeper for Spain, Iker Casillas made the save. Winger Andres Iniesta scored for Spain for the win.

Classification and Insight

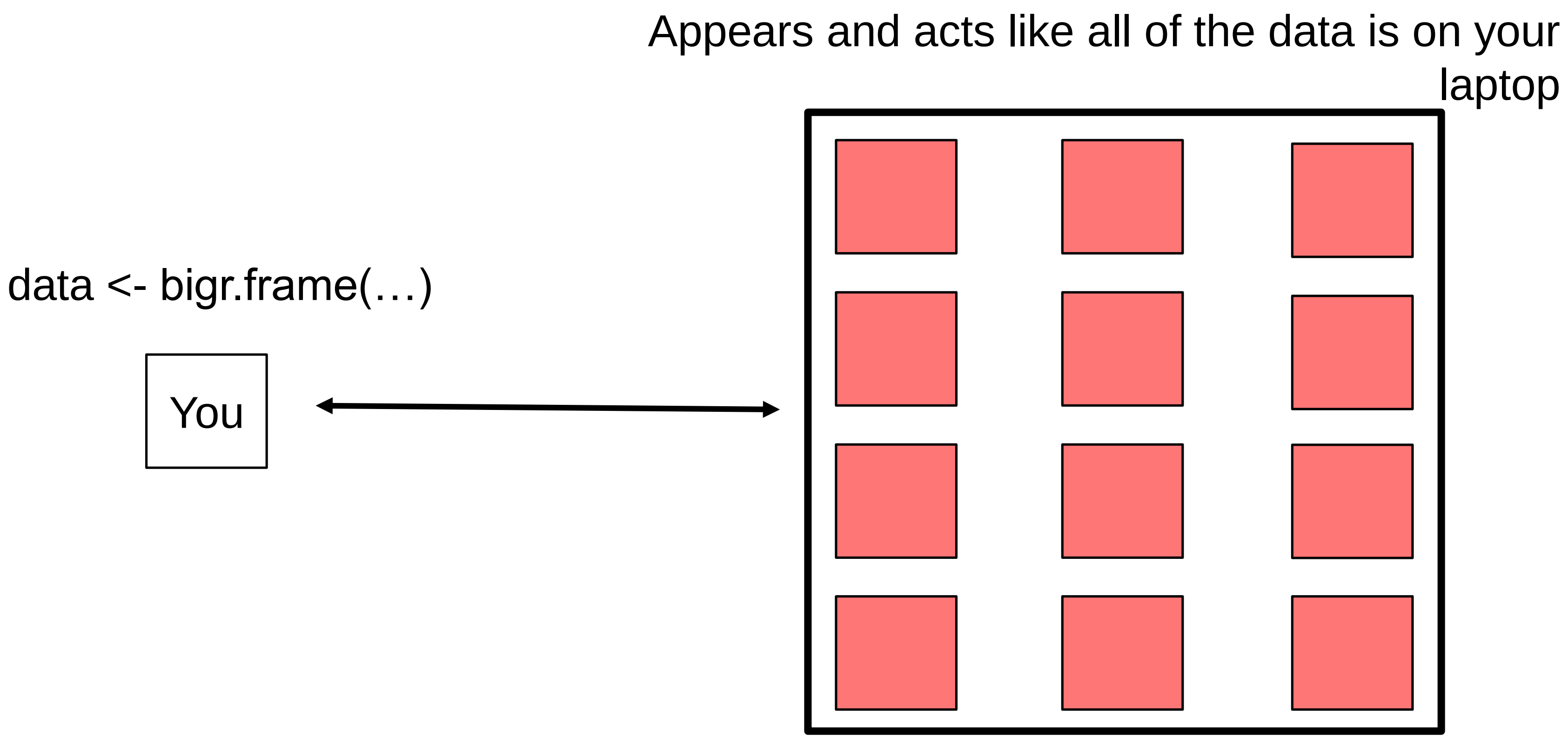
World Cup 2010 Highlights

Name	Position	Country
Arjen Robben	Striker	Netherlands
Iker Casillas	Keeper	Spain
Andres Iniesta	Winger	Spain

Big R – Deep R language integration on Hadoop

It seems that everyone has a solution for R on Hadoop. What's unique about Big R is that it implements standard R-language functions for parallel processing, supports routines available from the Comprehensive R Archive Network (CRAN), it uses standard R developer tools, and supports advanced machine analytic functions in R.

Big R Data Structures: Proxy to entire dataset



In-Hadoop Analytics – Deploy the analytics to the data

The whole point of Hadoop is to move the compute to the data, but IBM takes this to the next level with native analytic functions accessible from R, SQL and other languages. Provide seamless access to advanced in Hadoop statistical and machine learning functions avoiding the cost and complexity of software development and additional tooling.

Link utili

Try it for free!

Non-production, no-limit version of IBM InfoSphere BigInsights

http://IBM.co/QuickStart

For more info

A few essential links

IBM Analytics
http://www.ibm.com/analytics/

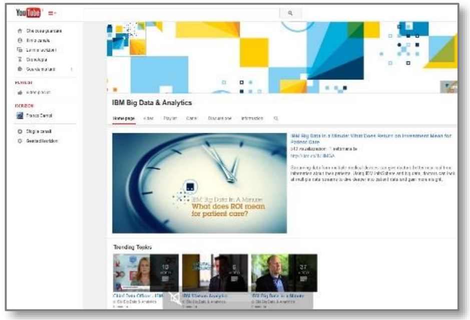
IBM Big Data
http://www-01.ibm.com/software/data/bigdata/

IBM Watson
http://www.ibm.com/smarterplanet/us/en/ibmwatson/

For more info

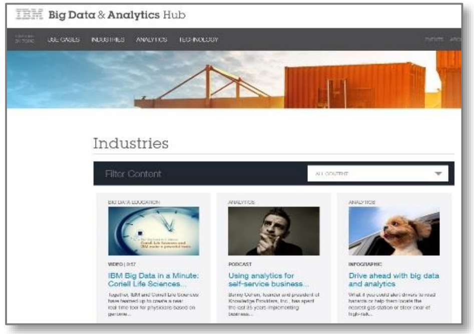
Other relevant links

Big Data university
http://bigdatauniversity.com/



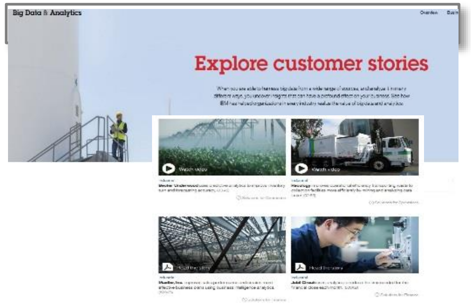
IBM Big Data Youtube Channel

https://www.youtube.com/user/ibmbigdata



IBM Big Data Hub

http://www.ibmbigdatahub.com



IBM Big Data case studies

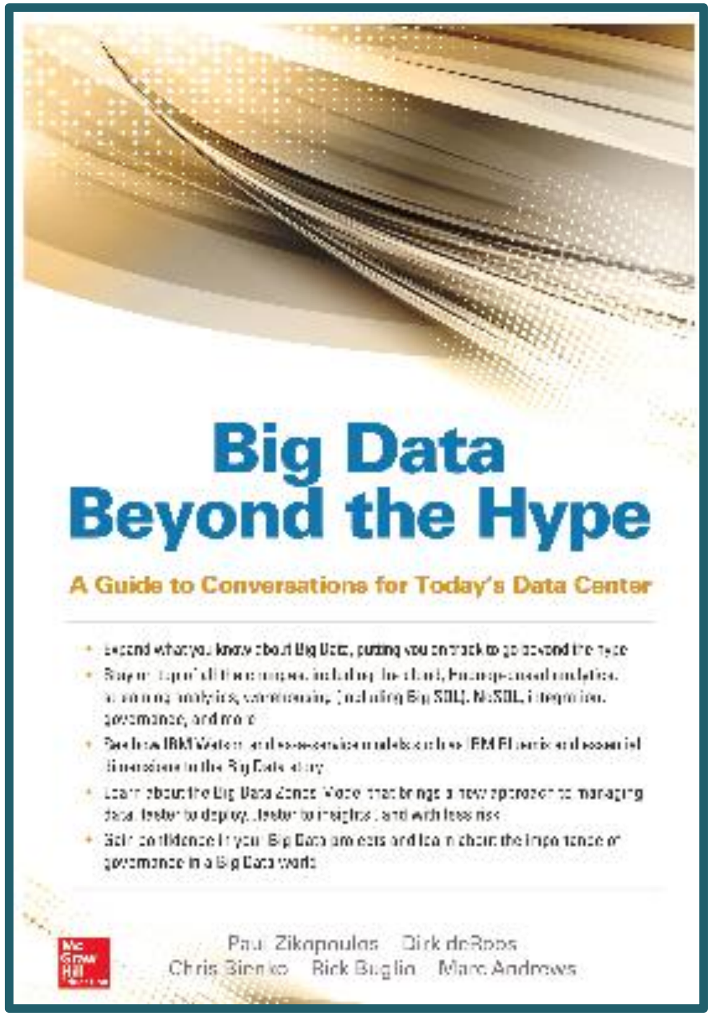
http://www.ibm.com/big-data/us/en/big-data-and-analytics/case-studies.html#filter2=customerinsight

IBM Analytics Zone

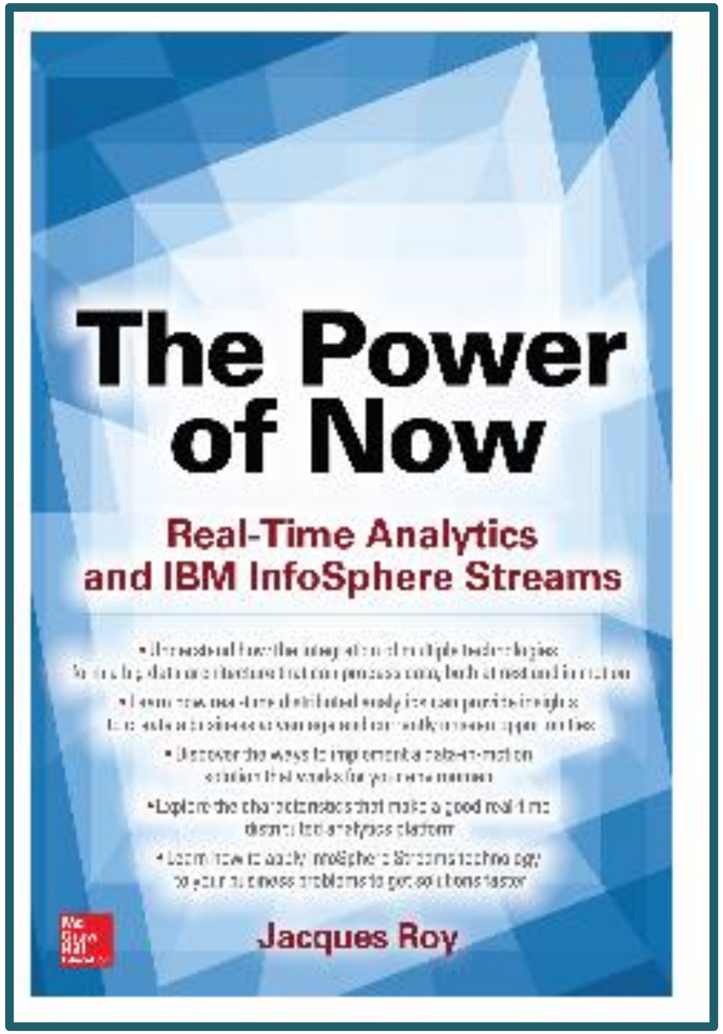
https://www.analyticszone.com

Due buoni libri …free…sui Big Data

Università degli Studi di Milano - Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione



https://www.ibm.com/developerworks/community/blogs/SusanVisser/entry/big_data_beyond_the_hype_a_guide_to_conversations_for_today_s_data_center?lang=en



https://www.ibm.com/developerworks/community/blogs/SusanVisser/entry/the_power_of_now_real_time_analytics_and_ibminfosphere_streams?lang=en http://bigdatauniversity.com/

Università degli Studi di Milano - Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione

Uno degli ultimi COURSES…su SPARK

Mi fermo qui
GRAZIE
a tutti voi