# Create your own Data pipeline with Spark and Azure HDInsight

Roberto Grandi
*@grandirob*

PASS

PASS SQL saturday

#566 | PARMA 2016

# Sponsors


Microsoft


ENGAGE
IT SERVICES


DATA SKILLS
UNDERSTANDING THE WORLD


MICROSYS


Elfo


CENTRO SISTEMI


STUDIOS


redgate


ApexSQL


SQLSENTRY

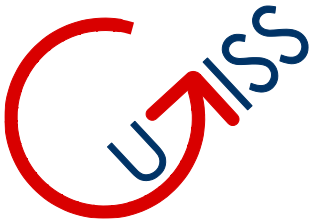

WindowServer.it


dotnet
MARCHE

# Organizers


ENGAGE IT SERVICES

getlatestversion.it




dotNET{podcast}

daniela malvisi COMMUNICATION



PASS SQL saturday

#sqlsatParma
#sqlsat566

# Speaker | @grandirob

- Backend Developer & Data Engineer
  @ 7Pixel (www.trovaprezzi.it, www.kirivo.it, www.drezzy.it )
- Data Engineer and DevOps Consultant
  @ ValueAmplify Consulting Group
- Past: Amazon.it, Medtronic.com, DaisyLabs
- Hobby: Data Science Lover

# Agenda

## Big Data Ecosystem (Story telling)

- Hadoop and Spark
- Azure HDInsight

## Build your own pipeline (Spark working samples)

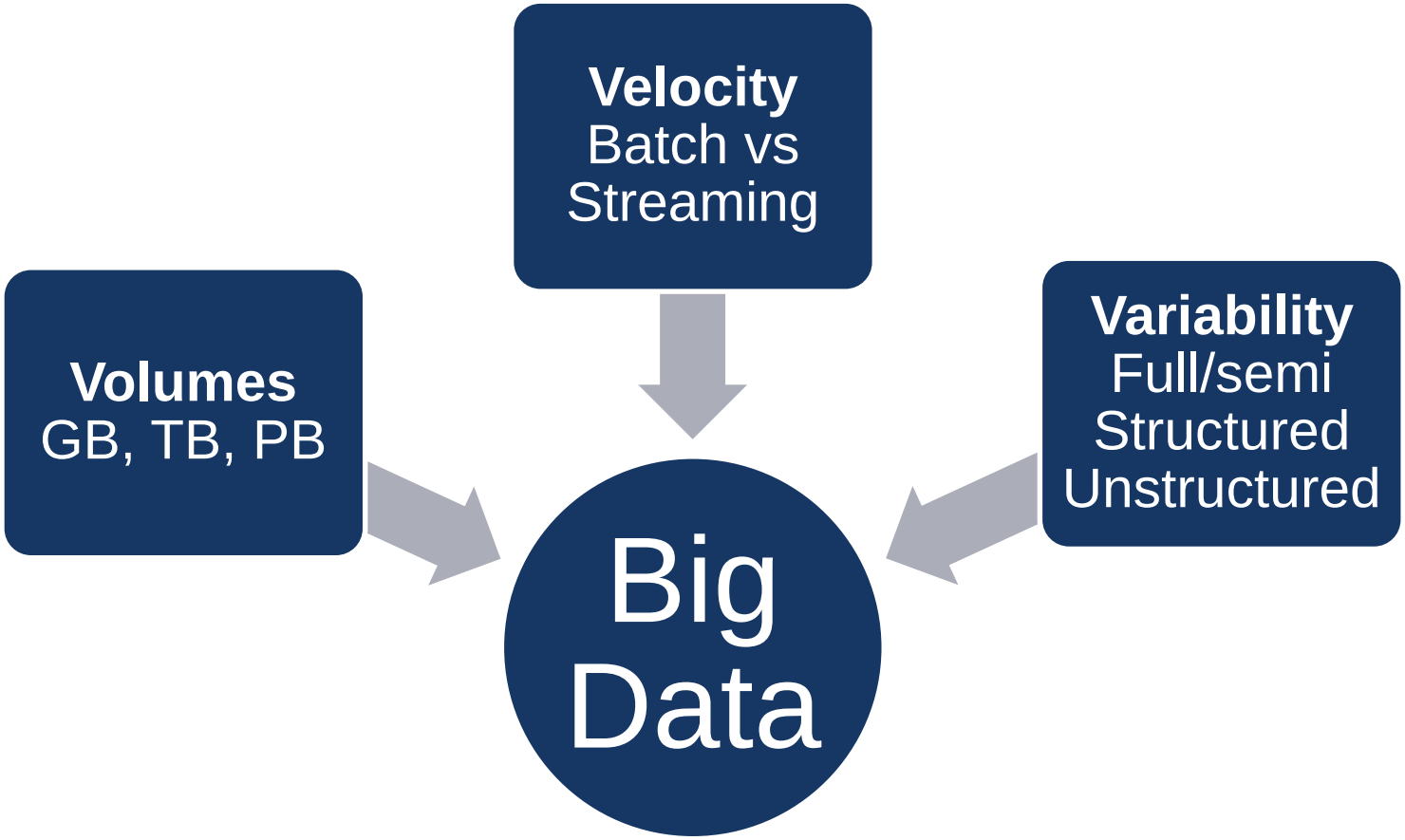
- Spark Local
- Interactive Offers Analysis (surprise ☺)
- Deploy Standalone App the cluster (big surprise)

## Conclusions and Ideas

# Big Data

- I sistemi/strumenti/SW immagazzinano sempre più informazioni
- Le applicazioni più interessanti, **Machine Learning e Real time Analytics**, non riescono a seguire all'incremento della mole di dati
- Non tutti si possono o vogliono permettere soluzioni costose $$$ (Oracle Exadata, SQL Parallel DW, etc..etc..)
- L'avvento del cloud → Scaling (auto)

# Le 3V dei Big Data



# Approccio Classico – Scale Up

Approccio RDBMS oriented (Oracle, SQL Server,…)

- Strumenti di ETL / Data Integration
- RDBMS e OLAP tutto fare
- Appliance costose

# Hadoop

- Nasce nel 2006 ispirato a due paper google
- Creato da Dough Nutch (lucene)
- Scritto in Java
- Spiazza il corrente modello di data management
- Approccio **Master-Slaves**

# Hadoop – Scale Out

## Hadoop sfrutta il parallelismo

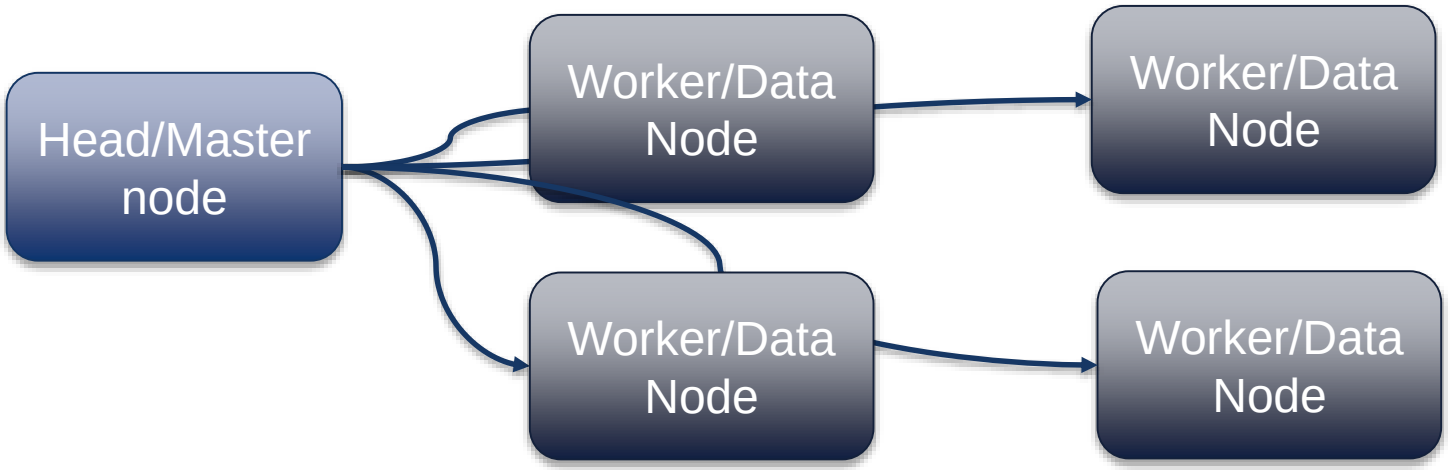
- Splitto I dati → sfrutto l'elaborazione in parallelo

- Uso il modo più semplice per memorizzare le informazioni → File Binary: testo, immagini, musica

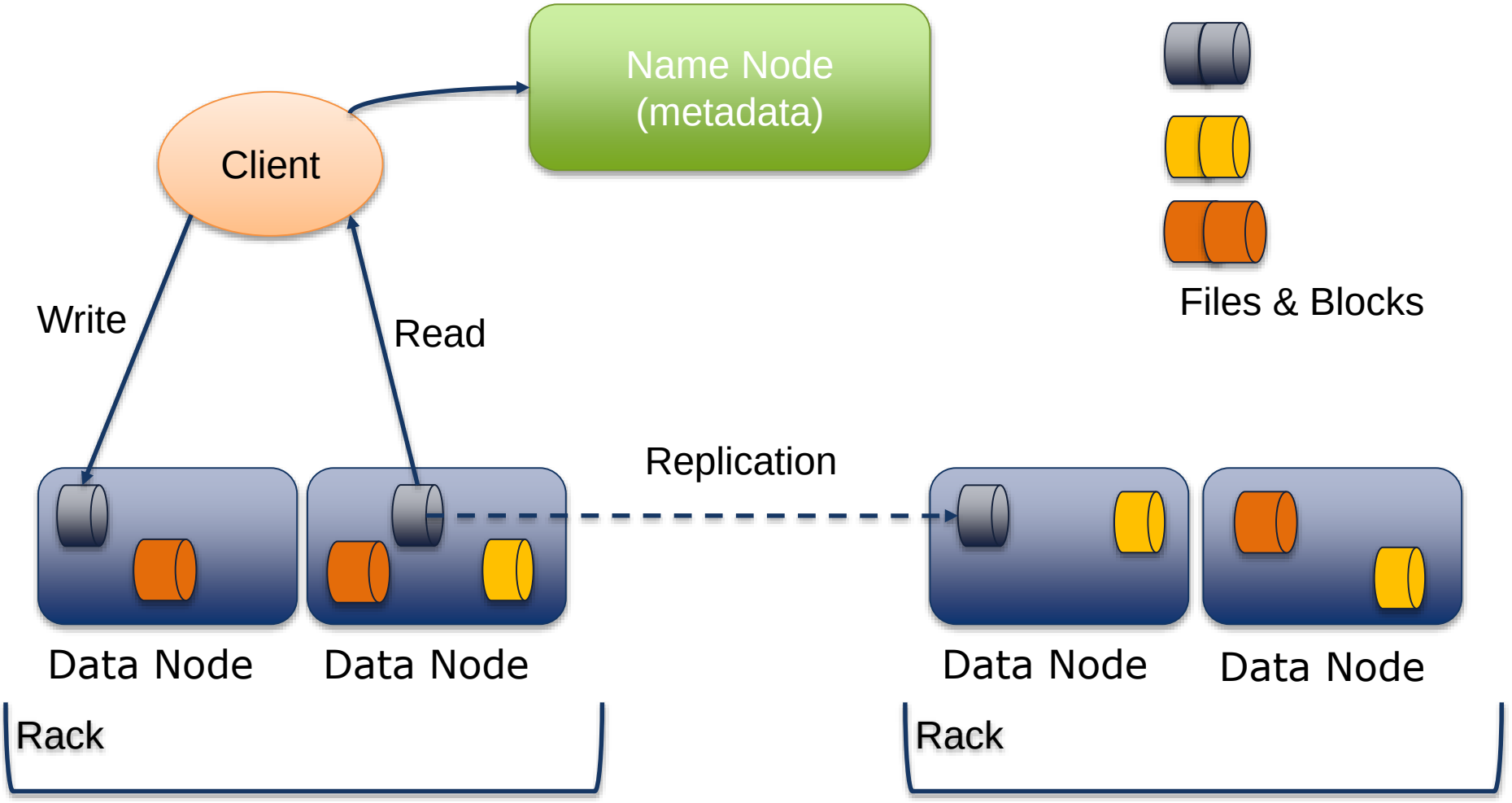
# Hadoop

Distributed File System, **HDFS,** → Persiste I dati

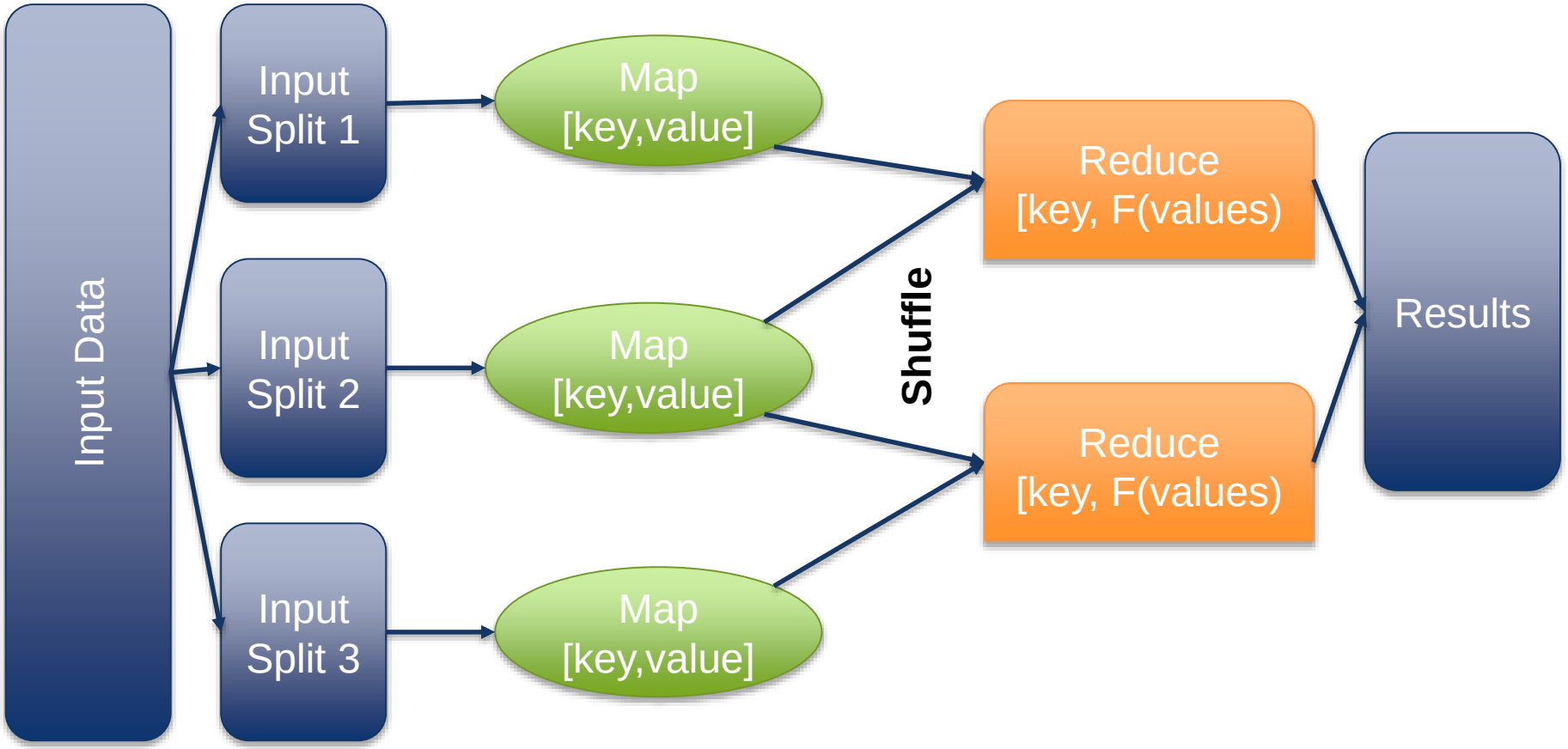
Data Processing, **Map Reduce,** → Parallel Computation



# HDFS


Name Node
(metadata)
Client
Files & Blocks
Write
Read
Replication
Data Node
Data Node
Data Node
Data Node
Rack
Rack

# Map Reduce


Input Data
Input Split 1
Input Split 2
Input Split 3
Map [key,value]
Map [key,value]
Map [key,value]
Shuffle
Reduce [key, F(values)
Reduce [key, F(values)
Results

# Hadoop - Limiti

Devo scrivere I job mapreduce in Java

I risultati intermedi devono essere per forza salvati su file o sorgenti dati esterne

Algoritmi iterative come ML sono penalizzati

# Spark

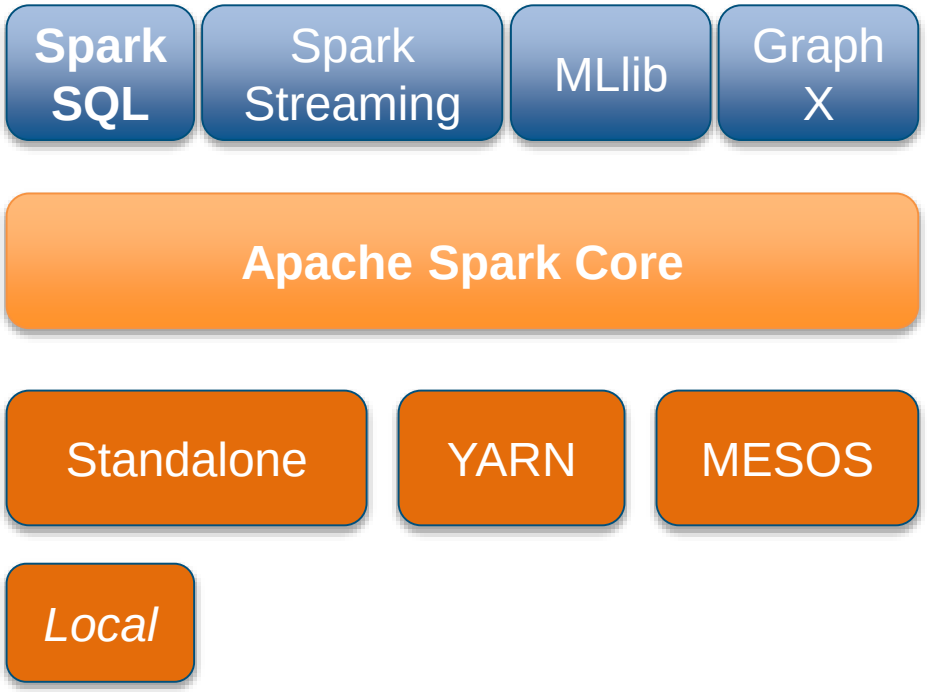
Distribued Computation Framework

Nasce per superare i limiti di MapReduce

*Scala based, functional inspired*

Api: Scala, Python, R, Java

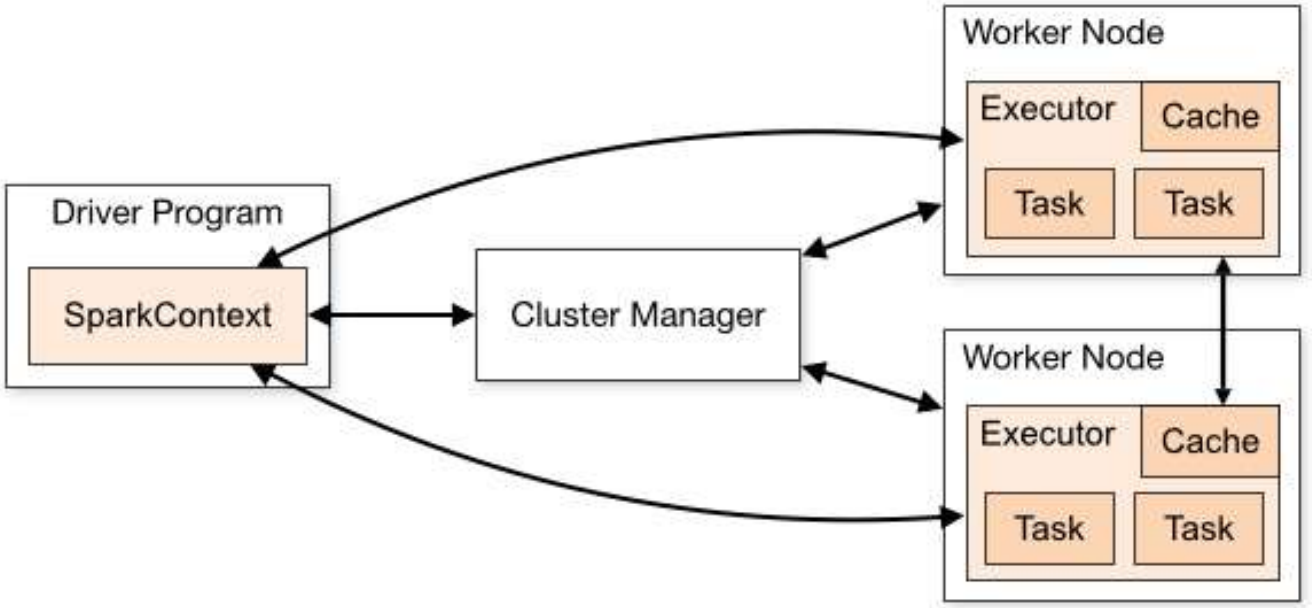
Sfrutta la memoria (del cluster, risultati intermedi)



# Spark – Come funziona

## Applicazione Spark

- Scala
- Python
- Java
- R




Info: http://spark.apache.org/docs/latest/cluster-overview.html

# Use Case – Cross Team working

"Mi analizzi questi web log"?

- I'm an Analyst: *".. Se ci stà su excel no problem"*
- I'm a BI Dev: *"..con SSIS, un dtsx ed un RDBMS faccio tutto"*
- I'm a Dev: *".. Riscrivo da capo, hai specifiche chiare?"*

# Use Case - Sfide

*"aspetta.....i log sono su azure..."*

*".. Mica vorrai consumare banda.."*

*".... e sono compressi"*

*"Vedi un pò tu quello che c'è dentro.."*

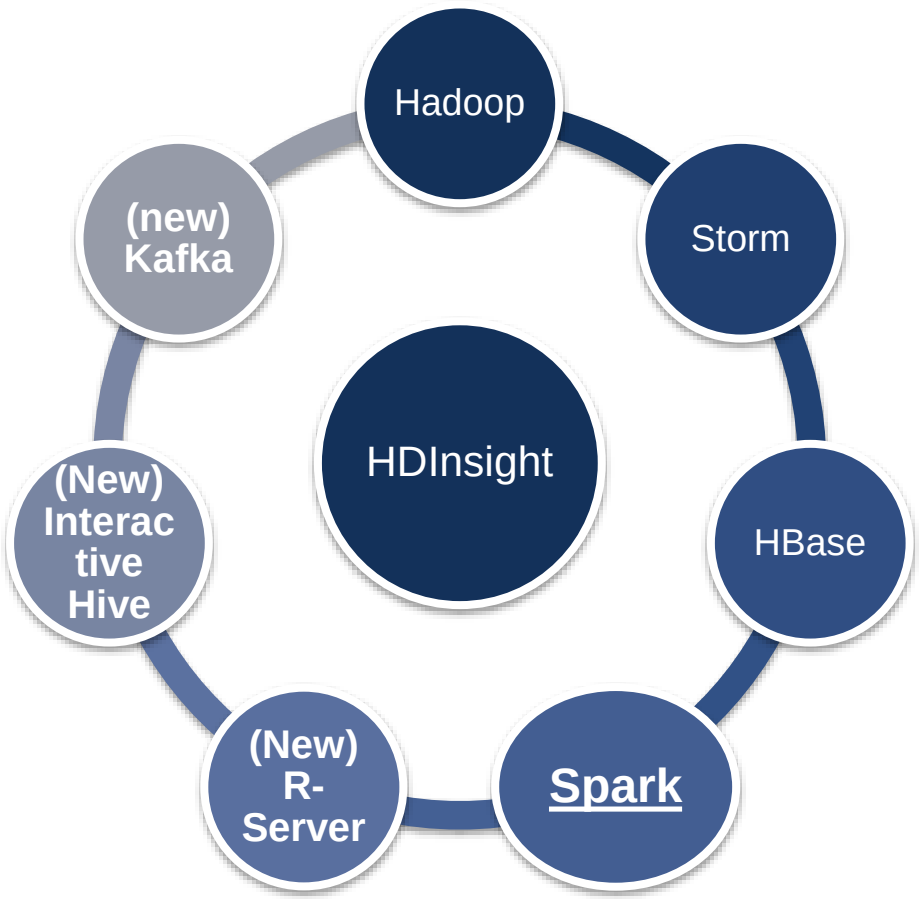
# HDInsight

E' la la soluzione Big Data **As a Service** di Microsoft

Basata sulla distribuzione Hadoop **HortonWorks**, contiene:

- Hadoop Yarn (Hortonworks)
- Spark (usa => Yarn)
- R
- Kafka
- Hive

# HDInsight Clusters and Evolutions



# DEMO - HDInsight

- Cluster Provisioning
- Azure portal, PowerShell, Azure CLI

# HDInsight – Punti di forza

**Scalability**: pochi limiti, se non la spesa

**Services**: infrastruttura gestita dal provider cloud

**Pricing**: Nessun costo upfront HW e SW

**Prototyping**: Tempo abbastanza rapido di sviluppo

# Spark

## RDD: Resilient Distributed Dataset

- Una collection di “oggetti”
- Partizionata e Distribuita
- Su Disco o in **Memoria**
- Processabile in Parallelo e Fault tolerant

# RDD

RDD (classe astratta): non può essere istanziata, può essere solo generata

File di testo

Database (SQL e NoSQL)

Streaming Sources

# Spark Context

```
val conf = new SparkConf().setMaster("local[*]")
val sc = new SparkContext(conf);

val linesRDD = sc.textFile("wasb:///example/data/file.csv")

val count = linesRDD.count()
```

# RDD Operations

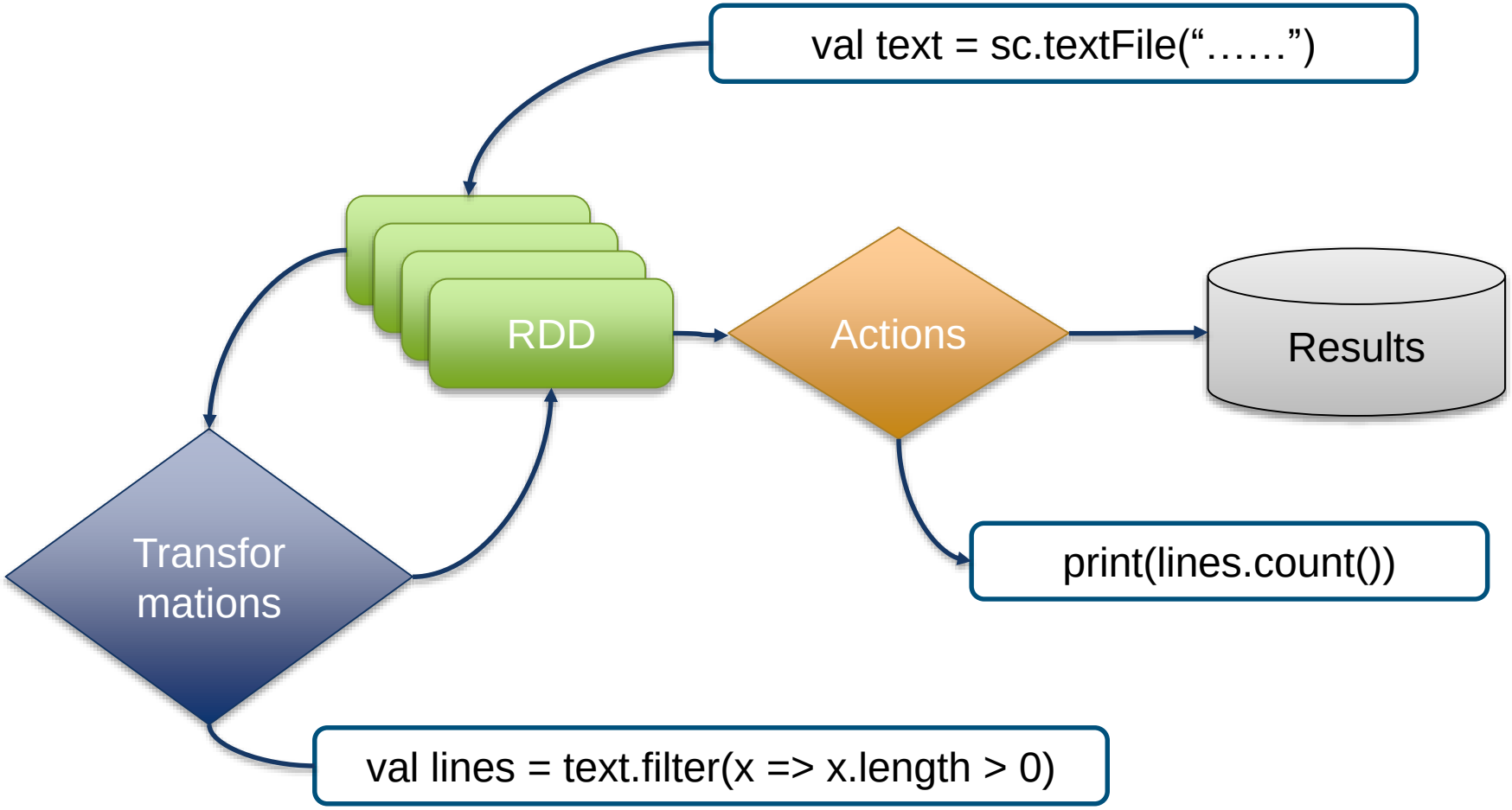
## Transformations

- Transformano un RDD in un altro RDD: **map, flatMap, filter**
- Lazy

## Actions

- Eseguono un operazione su un RDD: count, take, collect
- Triggerano le trasformazioni

# Working with RDD



# DEMO –Spark Local

## Spark-Shell

# Spark – I Due Cappelli

## Data Scientist

- Focus Modelli
- Esplorazione
- Analista (con steroidi)
- R/Python

## Data Engineer

- Focus Processo
- Gestione informazioni ed infrastruttura
- SW Developer
- Scala/Python

# Spark SQL

- Modulo dedicato ai dati strutturati
- Basato su Dataframe e Dataset
- Con Spark 2.0 accediamo tramite SparkSession

```
val spark = new SparkSession.builder().getOrCreate();

val csv = spark.read.csv("wasb:///example/data/file.csv")

val json = spark.read.json("wasb:///example/data/file.json")
```

# DEMO

## Spark SQL

- Data science e Jupyter Notebook
- MarketPlace Offers Analysis

# Spark SQL e Parquet

SparkSQL offre supporto native per il formato Parquet ( parquet.apache.org )

- Columnar Storage: permette di leggere solo le colonne necessarie

- RDDs/Dataframe possono essere persistiti in formato Parquet mantenendo lo schema

- Best Practices: utilizzare parquet come storage intermedio per formati più lenti

# DEMO

## Spark Coding

Costruire applicazioni Spark per HDInsight… ☺

# Spark – What's new

- Grande cambiamenti da 1.6.x a 2.0.x
- Spark SQL e Dataframes + Dataset
- Performances First!

# Spark - Vantaggi

- **Data pipeline più veloce**: è un solo batch job senza salvataggi intermedi
- Persistenza in memoria
- Api per diversi linguaggi
- Figure diverse di lavorare sulla stessa piattaforma

# HDInsight – Da migliorare

- **Evoluzioni**: Spark VS **R-Server** VS **Kafka** VS Hive RealTime
- **Migrazioni**: Jupyter ma non Zeppelin, R-server
- **Configurazioni**: templates vs configurazione
- HDInsight: complessità speculativa, mi serve realmente? Alternative:
  - Self Provisioning
  - Mesos e DC/OS , Azure Container Service

# Conclusioni

Non ci sono più confine e limiti!

Linguaggi

Strumenti

Architettura/Infrastruttura

# Q&A

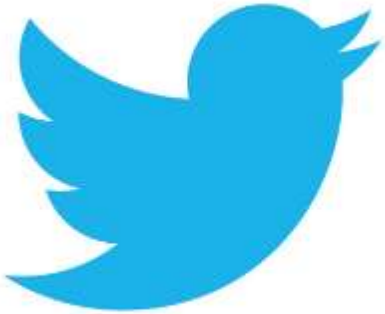
Domande…?

….alla prossima puntata ☺

(spark streaming e ML)

# Risorse

- Apache Spark Documentation
- Azure HDInsight
  - IntelliJ Plugin
  - Eclipse Plugin (new!)
- Azure QuickStart Templates
  - Spark 2.0 on SLES
  - HDInsight on Linux VMs



**#sqlsatParma**
**#sqlsat566**

roberto.grandi@gmail.com
@grandirob

# THANKS!