ANALYTICS BIG DATA SUMMIT

Apache Spark : Fast and Easy Data Processing

Sujee Maniyam

Elephant Scale LLC

sujee@elephantscale.com

http://elephantscale.com

Spark

- Fast & Expressive Cluster computing engine
- Compatible with Hadoop
- Came out of Berkeley AMP Lab
- Now Apache project
- Version 1.1 just released (Sep 2014)

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Comparison With Hadoop

Hadoop	Spark
Distributed Storage + Distributed Compute	Distributed Compute Only
MapReduce framework	Generalized computation
Usually data on disk (HDFS)	On disk / in memory
Not ideal for iterative work	Great at Iterative workloads (machine learning ..etc)
Batch process	- Upto 10x faster for data on disk - Upto 100x faster for data in memory
	Compact code Java, Python, Scala supported
	Shell for ad-hoc exploration

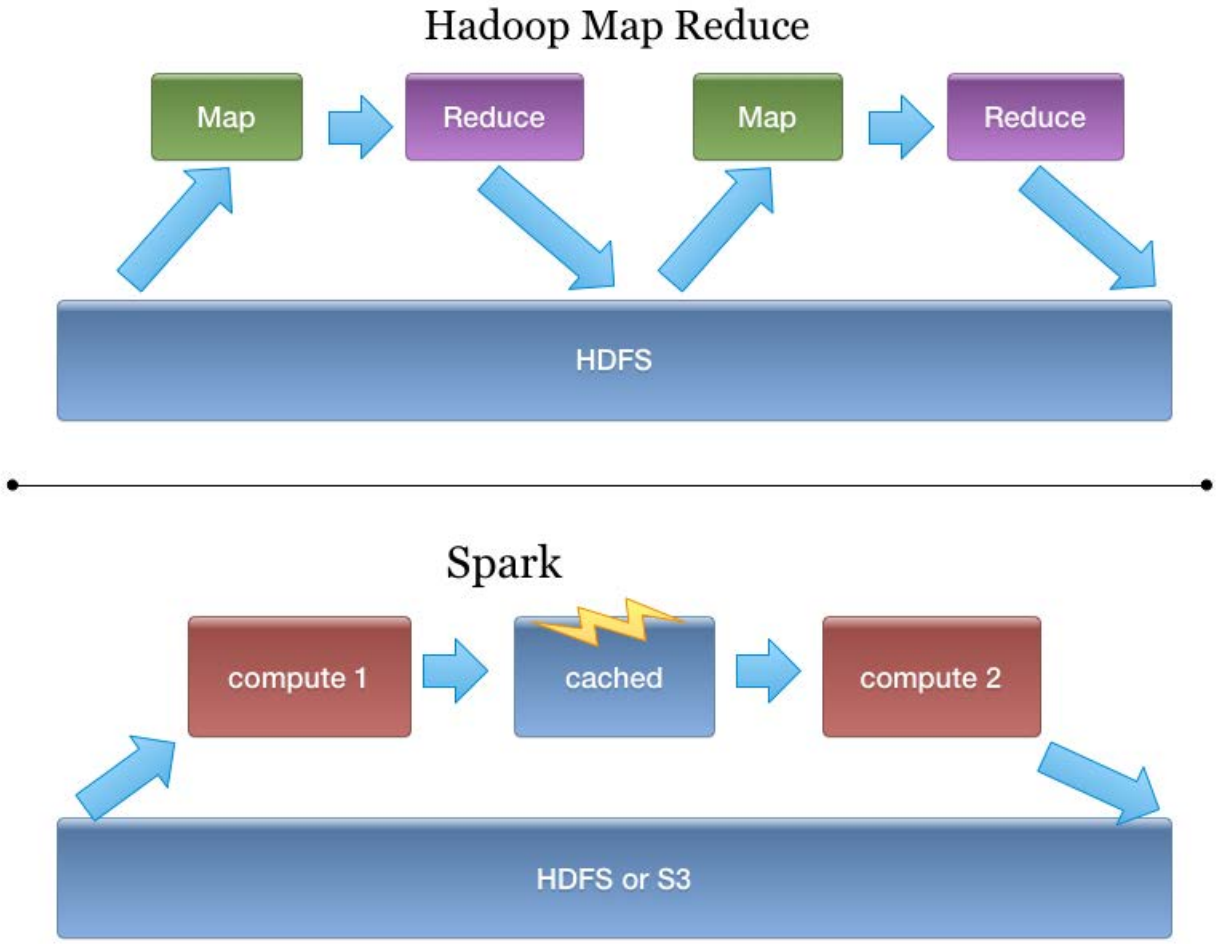
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Vs Hadoop

- Spark is ‘easier’ than Hadoop
- ‘friendlier’ for data scientists
- Interactive shell (adhoc exploration)
- API supports multiple languages
  - Java, Scala, Python
- Great for small (Gigs) to medium (100s of Gigs) data

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Vs. Hadoop



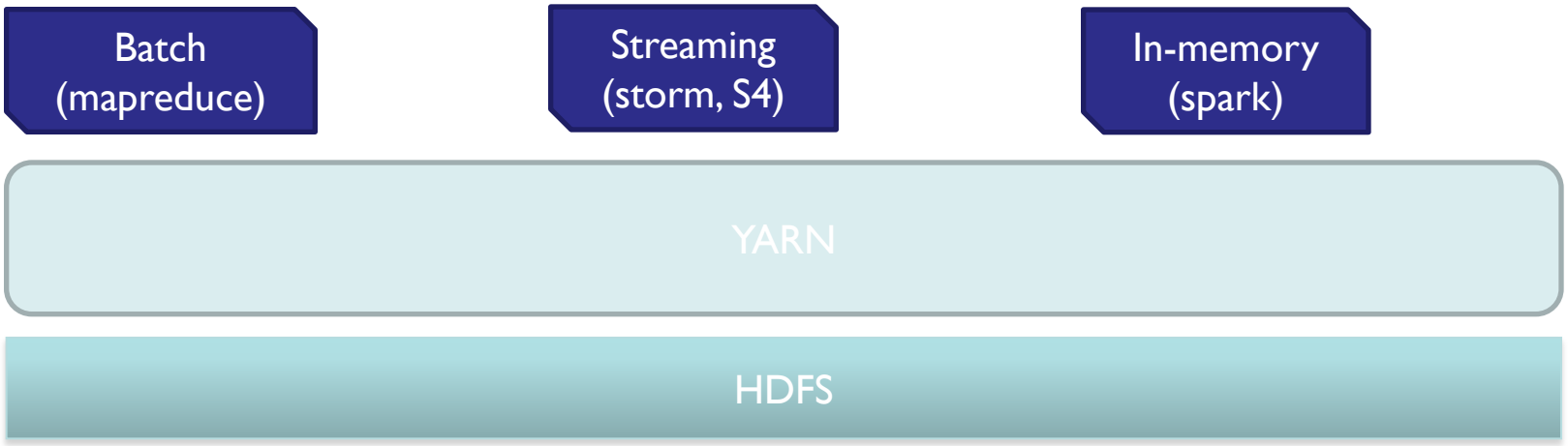
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Is Spark Replacing Hadoop?

- Right now, Spark runs on Hadoop / YARN
  - Complimentary
- Can be seen as generic MapReduce
- Spark is really great if data fits in memory (few hundred gigs),
- People are starting to use Spark as their only compute platform
- Future ???

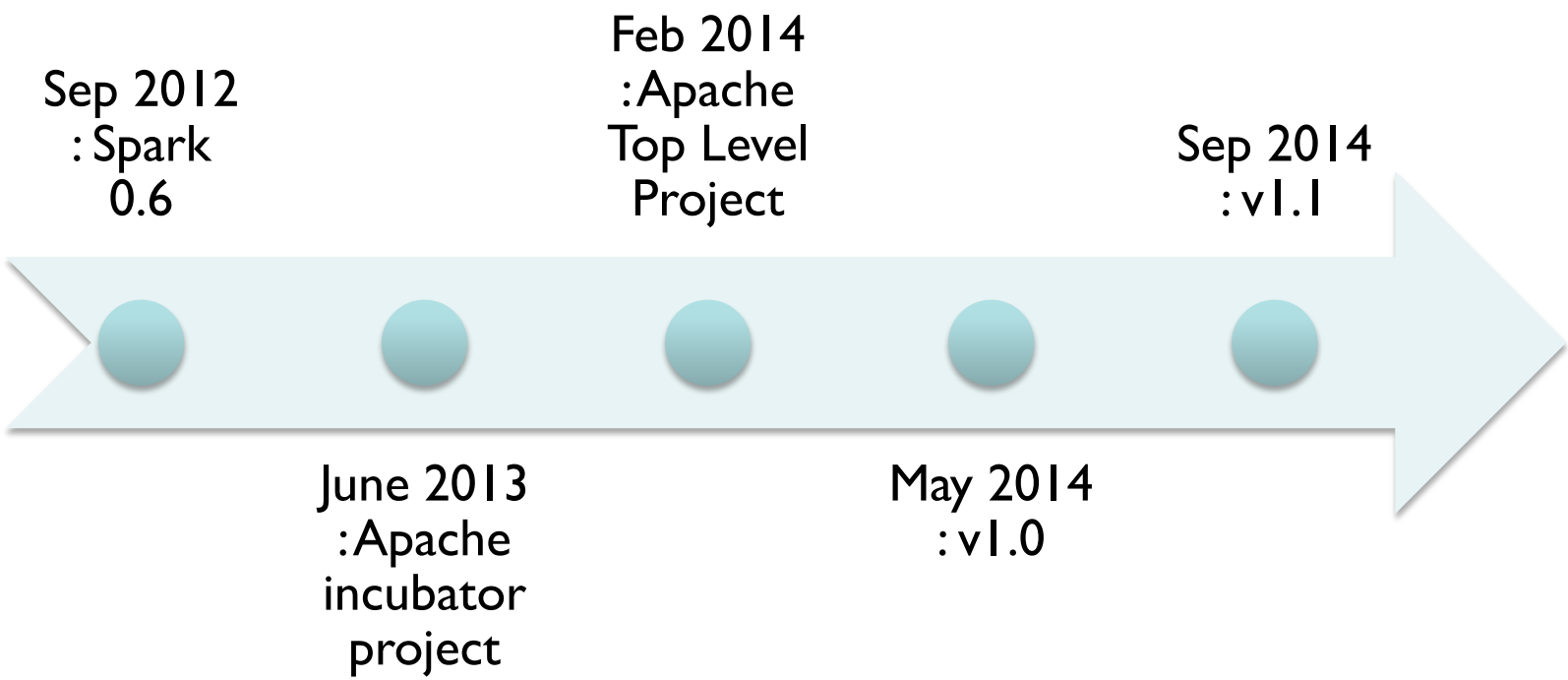
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Hadoop + Yarn : Universal OS for Cluster Computing



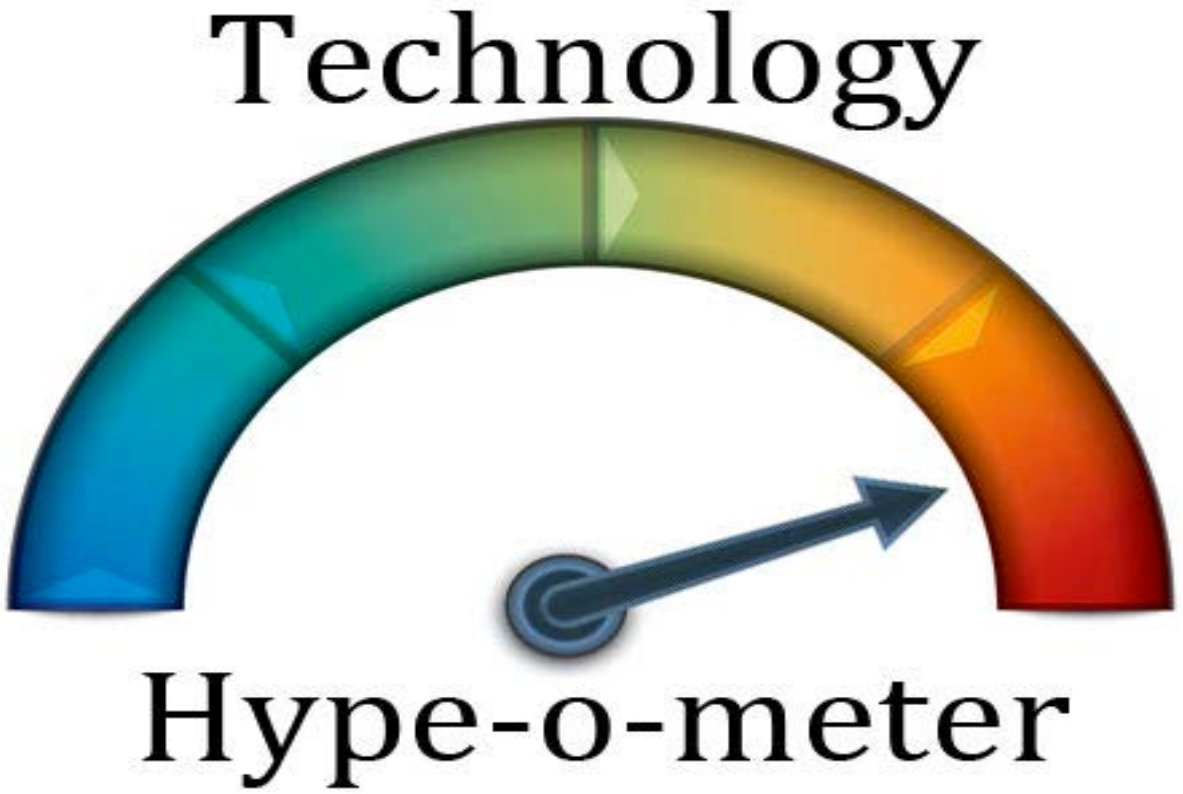
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Bit of History of Spark



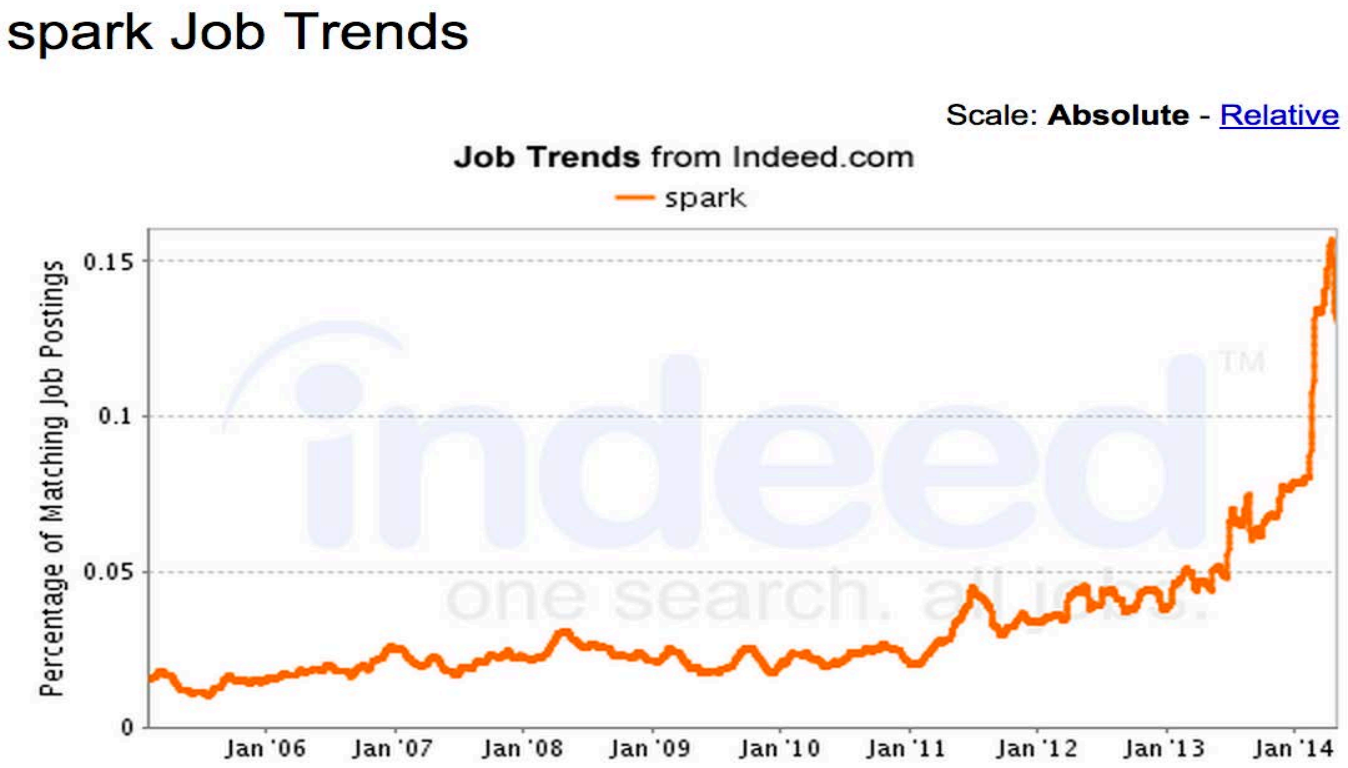
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Hypo-meter ☺



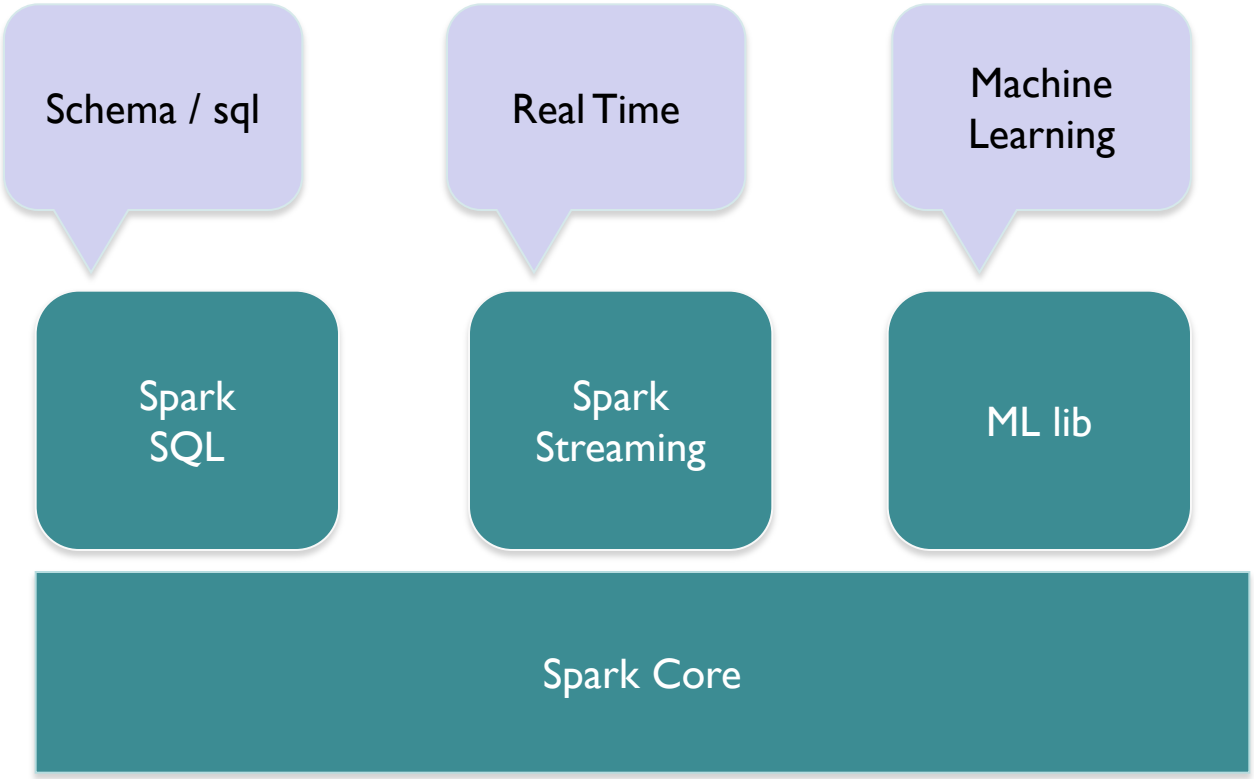
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Job Trends



2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Eco-System



2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Core

- Distributed compute engine
- Sort / shuffle algorithms
- In case of node failures -> re-computes missing pieces

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Streaming

- Process data streams in **real time**
- Stock ticks / click streams …etc

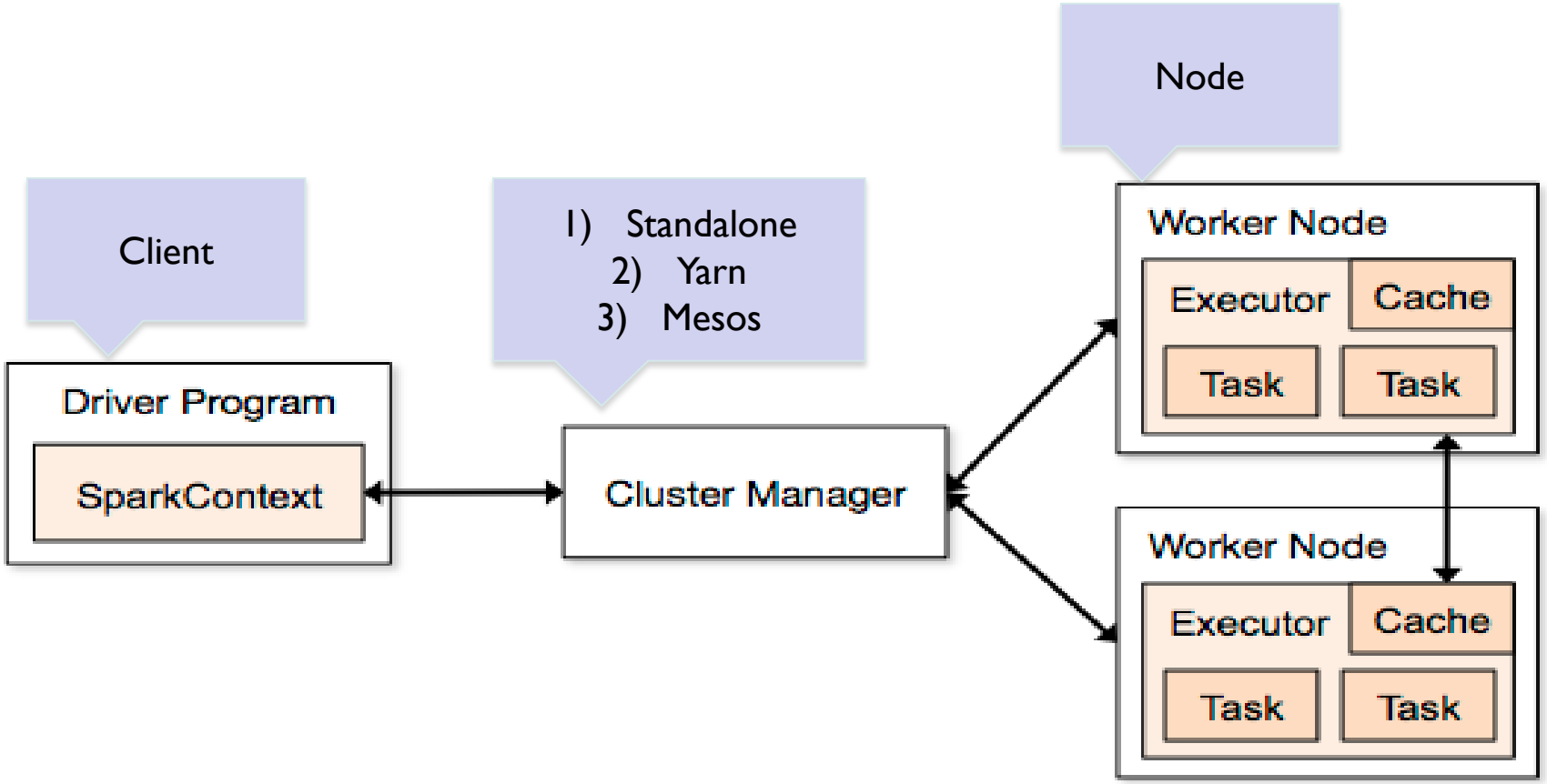
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Machine Learning (ML Lib)

- Out of the box ML capabilities !
- Lots of common algorithms are supported
- Classification / Regressions
  - Linear models (linear R, logistic regression, SVM)
  - Decision trees
- Collaborative filtering (recommendations)
- K-Means clustering
- …
- More to come

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Architecture



2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Architecture

- Multiple ‘applications’ can run at the same time
- Driver (or ‘main’) launches an application
- Each application gets its own ‘executor’
  - Isolated (runs in different JVMs)
  - Also means data can not be shared across applications
- Cluster Managers:
  - multiple cluster managers are supported
  - 1) Standalone : simple to setup
  - 2) YARN : on top of Hadoop
  - 3) Mesos : General cluster manager (AMP lab)

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Spark Data Model : RDD

- Resilient Distributed Dataset (RDD)
- Can live in
  - Memory (best case scenario)
  - Or on disk (FS, HDFS, S3 …etc)
- Each RDD is split into multiple partitions
- Partitions may live on different nodes
- Partitions can be computed in parallel on different nodes

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

RDD : Loading

- Use Spark context to load RDDs from disk / external storage

```
val sc = new SparkContext(...)

val f = sc.textFile("/data/input1.txt")  // single file
sc.textFile("/data/")  // load all files under dir
sc.textFile("/data/*.log")  // wild card matching
```

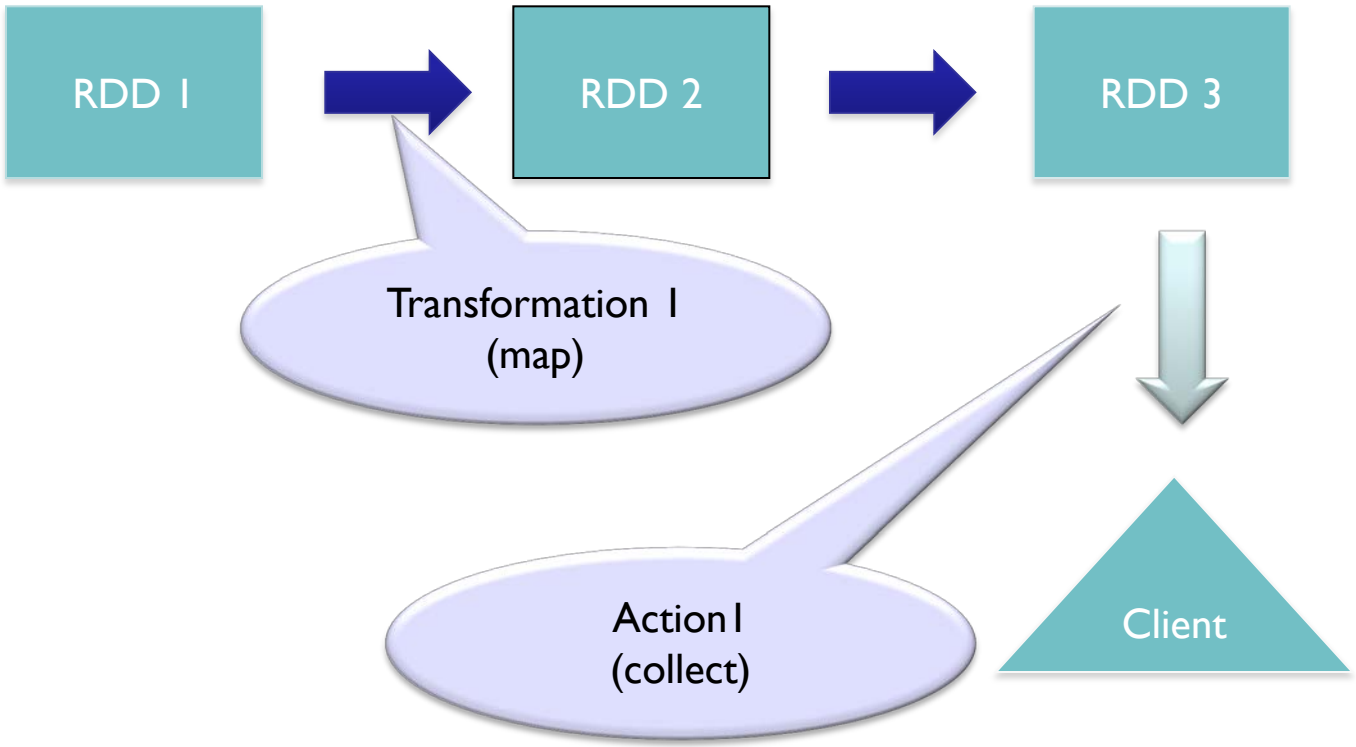
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

RDD Operations

- Two kinds of operations on RDDs
  - 1) Transformations
    - Create a new RDD from existing ones (e.g. Map)
  - 2) Actions
    - E.g. Returns the results to clients (e.g. Reduce)
- Transformations are **lazy**.. Actions force transformations

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Transformations / Actions



2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

RDD Transformations

Transformation	Description	Example
filter	Filters through each record (aka grep)	f.filter(line => line.contains("ERROR"))
union	Merges two RDDs	rdd1.union(rdd2)
…see docs …		

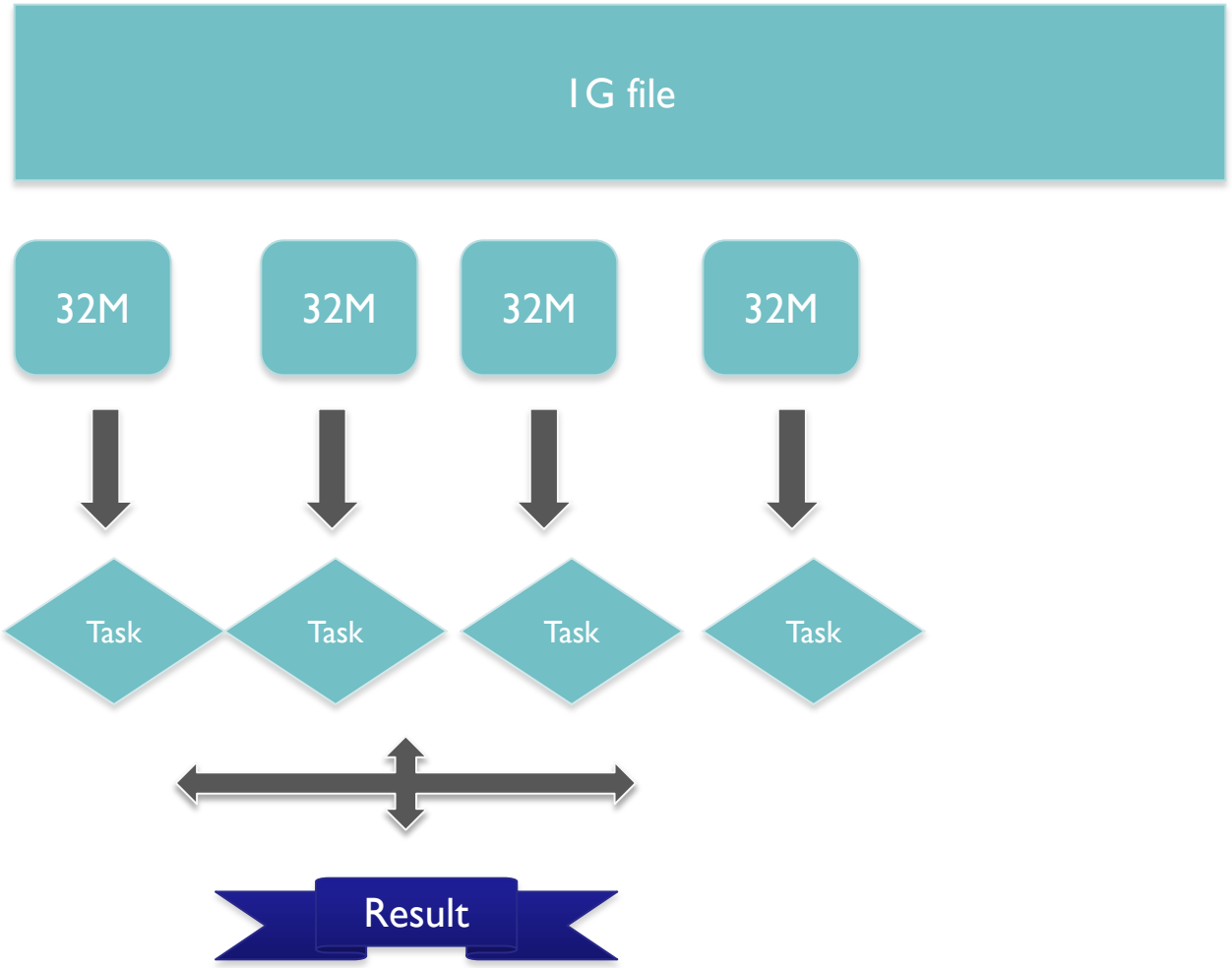
2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

RDD Actions

Action	Description	Example
count()	Counts all records in an rdd	f.count()
first()	Extract the first record	f.first ()
take(n)	Take first N lines	f.take(10)
collect()	Gathers all records for RDD. All data has to fit in memory of ONE machine (don't use for big data sets)	f.collect()
…. See documentation ..		

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Partitions Explained



2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

RDD : Saving

- saveAsTextFile () and saveAsSequenceFile()

f.saveAsTextFile("/output/directory") // a directory

- Output usually is a directory
  - RDDs will be saved as multiple files in the dir
  - Each partition → one output file

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

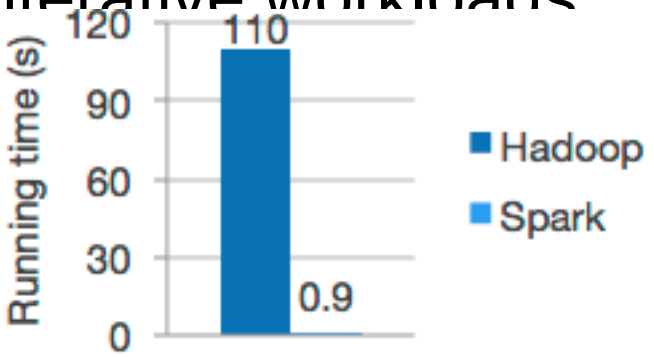
Caching of RDDs

- RDDs can be loaded from disk and computed
  - Hadoop mapreduce model
- Also RDDs can be cached in memory
- Subsequent operations are much faster

f.persist() // on disk or memory

f.cache() // memory only

- In memory RDDs are great for iterative workloads
  - Machine learning algorithms



2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Demo Time !

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Demo : Spark-shell

- Invoke spark shell
- Load a data set
- Do basic operations (count / filter)

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Demo: RDD Caching

- From Spark shell
- Load an RDD
- Demonstrate the difference between cached and non-cached

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Demo : Map Reduce

- Quick Word count

```
val input = sc.textFile("…")
    val counts = input.flatMap(
                    line => line.split(" ")).
                    map(word => (word, 1)).
                    reduceByKey(_+_)
```

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Thanks !

Sujee Maniyam

sujee@elephantscale.com

http://elephantscale.com

Expert consulting & training in Big Data

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.

Credits

- http://spark.apache.org/
- http://www.strategictechplanning.com

2015 SNIA Analytics and Big Data Summit. © Elephant Scale LLC. All Rights Reserved.