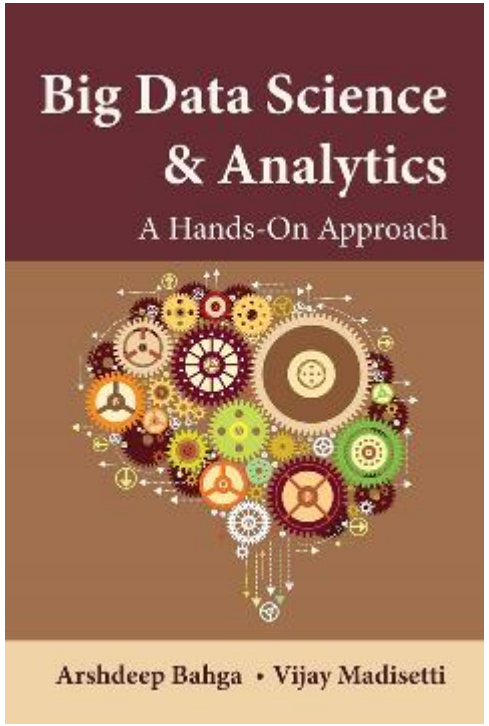
Batch Data Analysis

Prof. Gheith Abandah

Reference

- Chapter 7: **Batch Data Analysis**



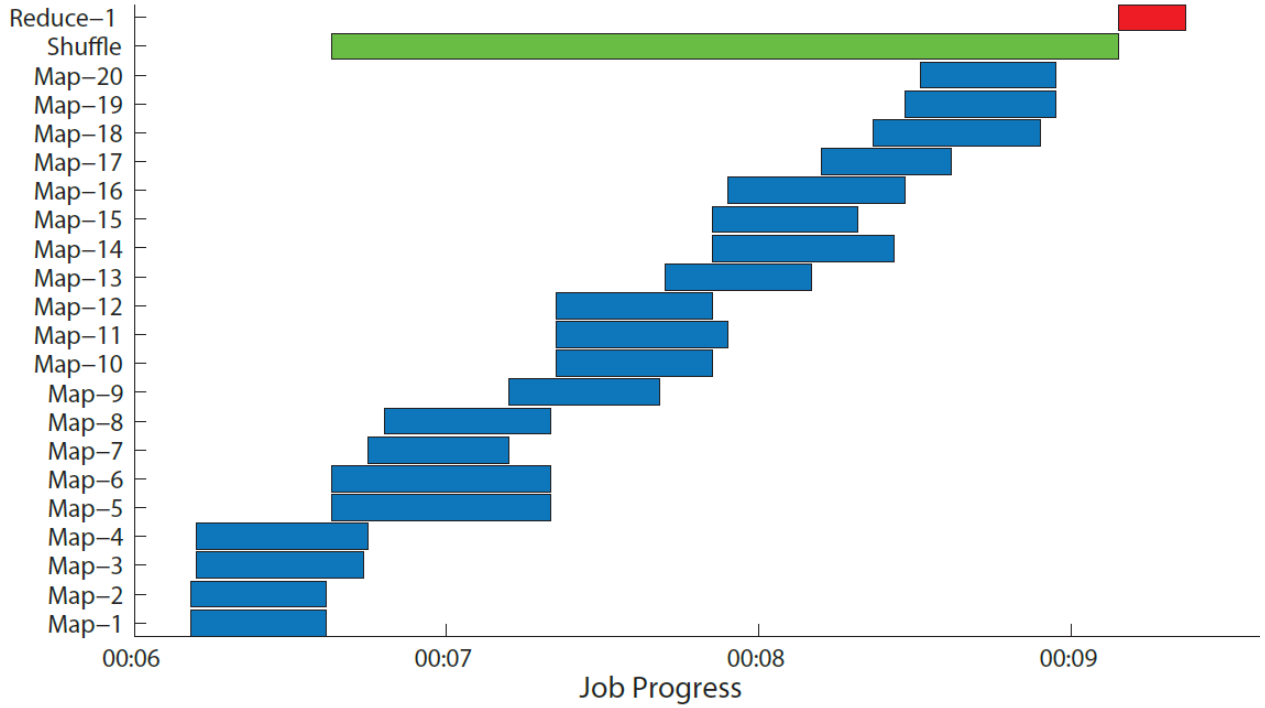
- Arshdeep Bahga and Vijay Madisetti, **Big Data Science and Analytics: A Hands-On Approach**, 2019.
  - Web site: http://www.hands-on-books-series.com/

Outline

- Batch Analysis frameworks
  - Hadoop and MapReduce
  - Pig
  - Apache Oozie
  - Apache Spark
  - Apache Solr

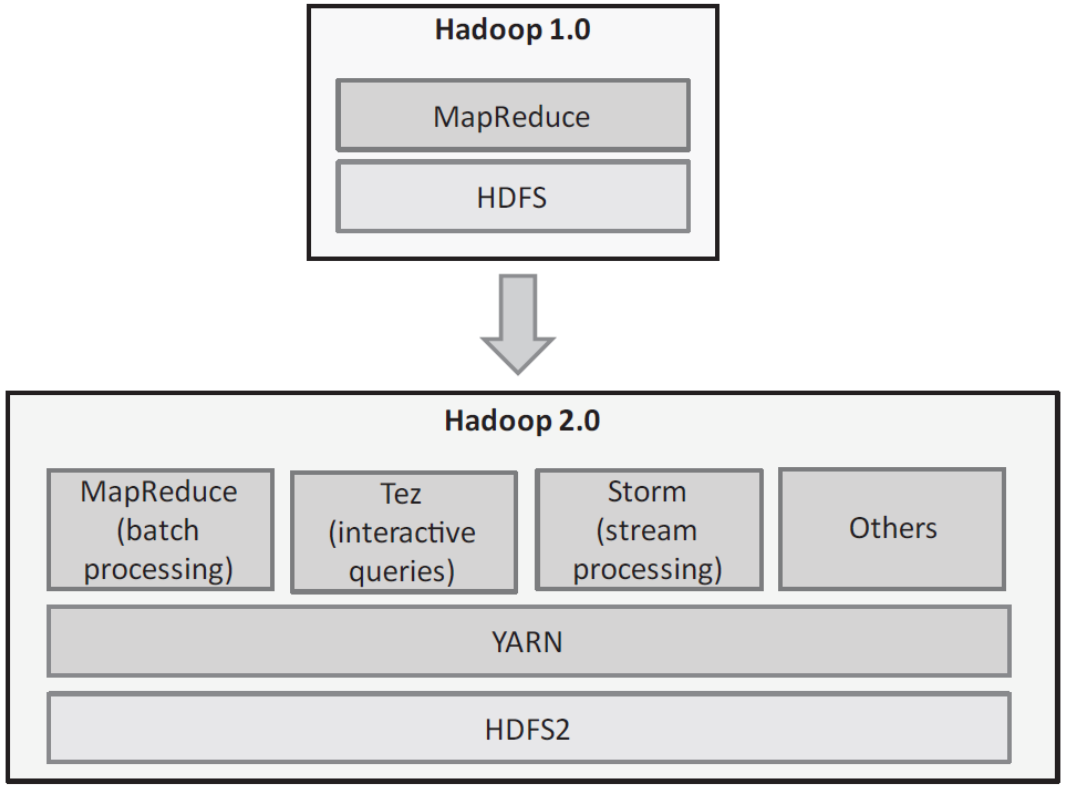
Hadoop and MapReduce

- **Apache Hadoop** is an open-source framework for **distributed batch processing** of **big data**.
- **MapReduce** is a parallel programming model suitable for analysis of big data.
- MapReduce algorithms allow **large-scale** computations to be **automatically parallelized** across a large cluster of servers.



Hadoop YARN

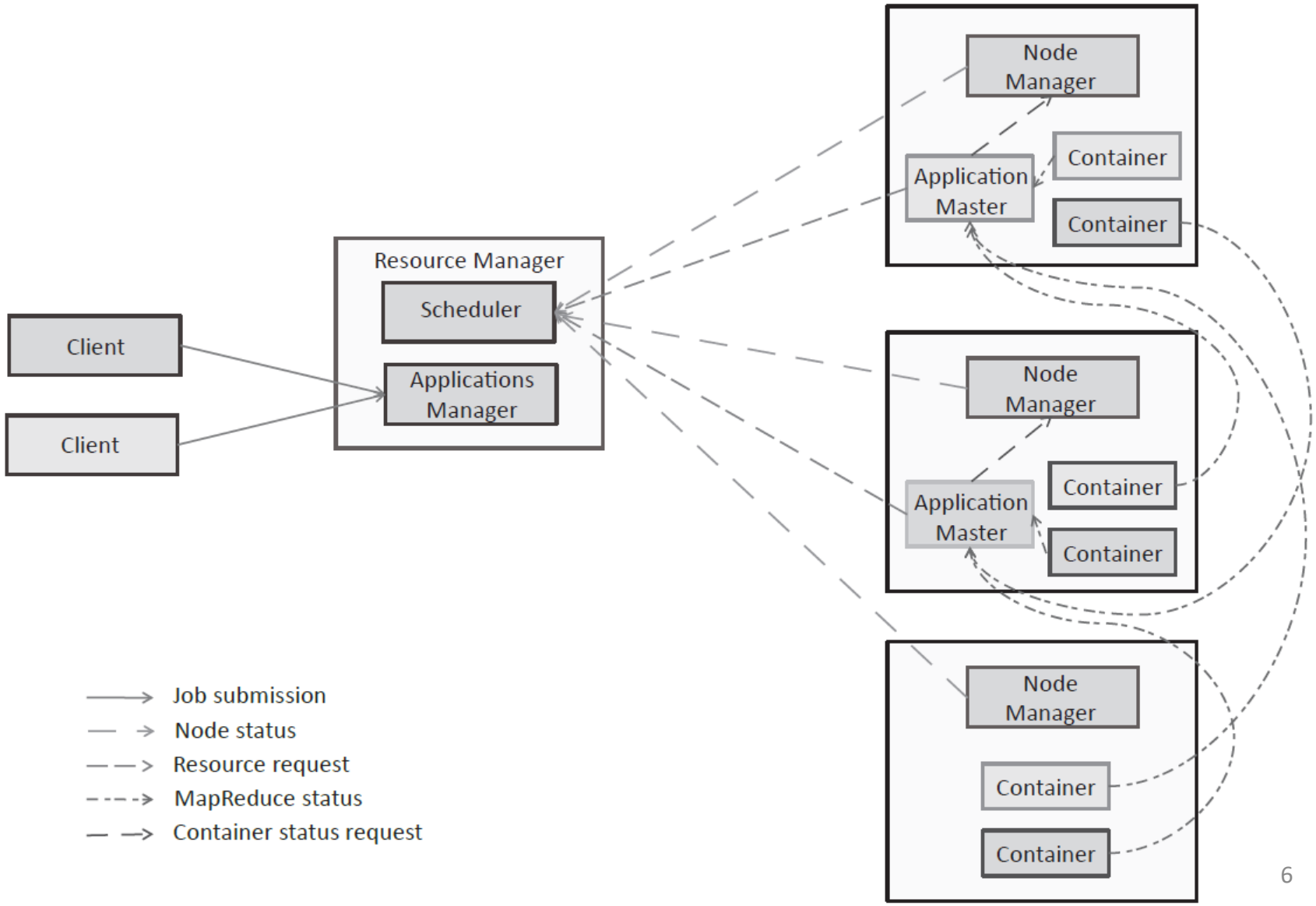
- **Hadoop Version 2**.
- The processing engine of Hadoop (MapReduce) has been separated from the **resource management component**.
- YARN is effectively an operating system for Hadoop that **supports** different processing engines on a Hadoop cluster:
  - **MapReduce** for batch processing,
  - **Apache Tez** for interactive queries
  - **Apache Storm** for stream processing



MapReduce job execution workflow

Applications Manager manages the running **Application Masters** in the cluster.

Application Master asks for resources from the **Resource Manager** and works with the **Node Managers** to execute and monitor the tasks.



Outline

- Batch Analysis frameworks
  - Hadoop and MapReduce
  - Pig
  - Apache Oozie
  - Apache Spark
  - Apache Solr

Pig

- Pig is a **high-level data processing language**.
- Developers write **data analysis scripts**.
- **Pig compiler** translates these scripts into MapReduce programs.

Pig Script Example

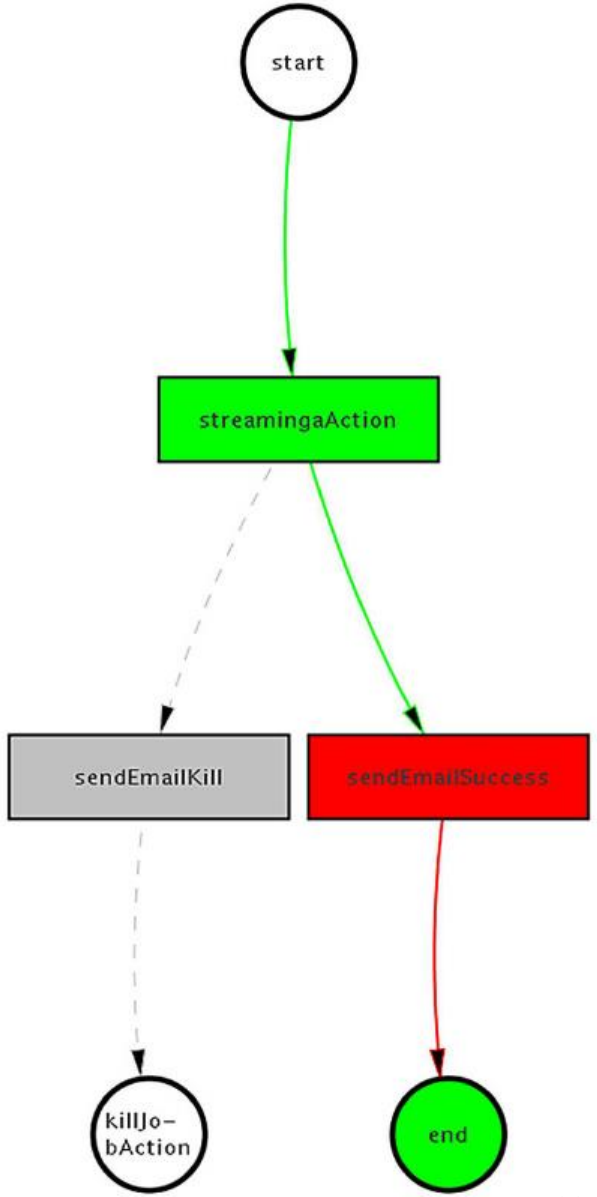
```
# LOAD example
data = LOAD ‘data.txt’ as (text:chararray);
# FOREACH example
monthTemp = FOREACH data GENERATE SUBSTRING(text, 10,12) as month,
(double)SUBSTRING(text, 38,45) as temp;
DUMP monthTemp;
(01,22.9)
:
(12,5.6)
# FILTER example
low = FILTER monthTemp by temp<20.0;
DUMP low;
(01,10.4)
:
(12,4.8)
```

Outline

- Batch Analysis frameworks
  - Hadoop and MapReduce
  - Pig
  - Apache Oozie
  - Apache Spark
  - Apache Solr

Apache Oozie

- For batch analysis applications that require **more than one MapReduce job** to be chained.
- Oozie is **workflow scheduler system** that allows managing Hadoop jobs.
- Oozie creates workflow which is a **collection of jobs** arranged as Direct Acyclic Graphs (**DAG**).
- An action is executed only when the preceding action is completed.
- Uses an XML-based Process Definition Language called **Hadoop Process Definition Language** (hPDL).

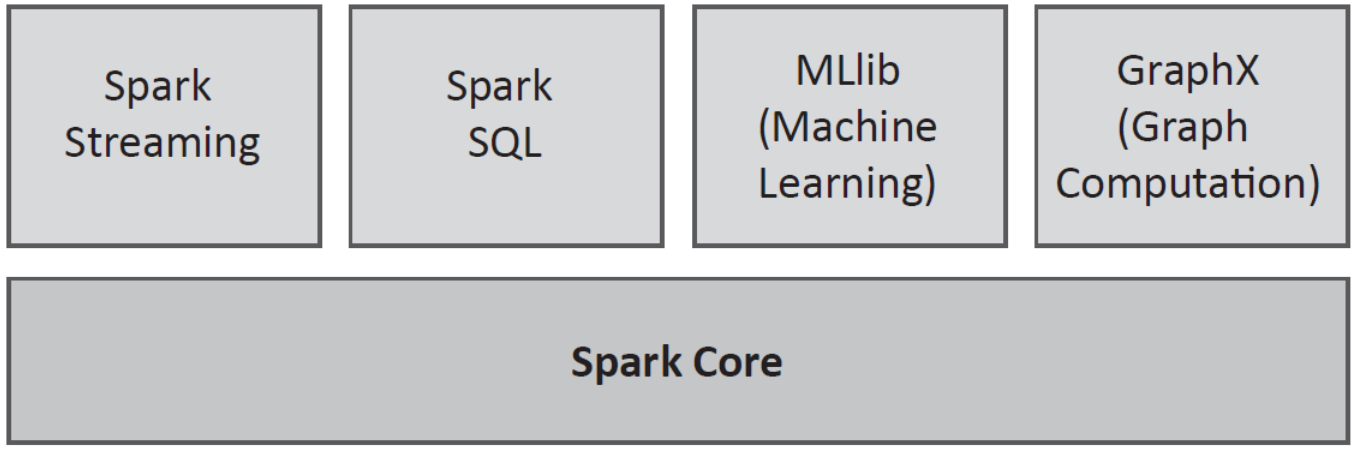


Outline

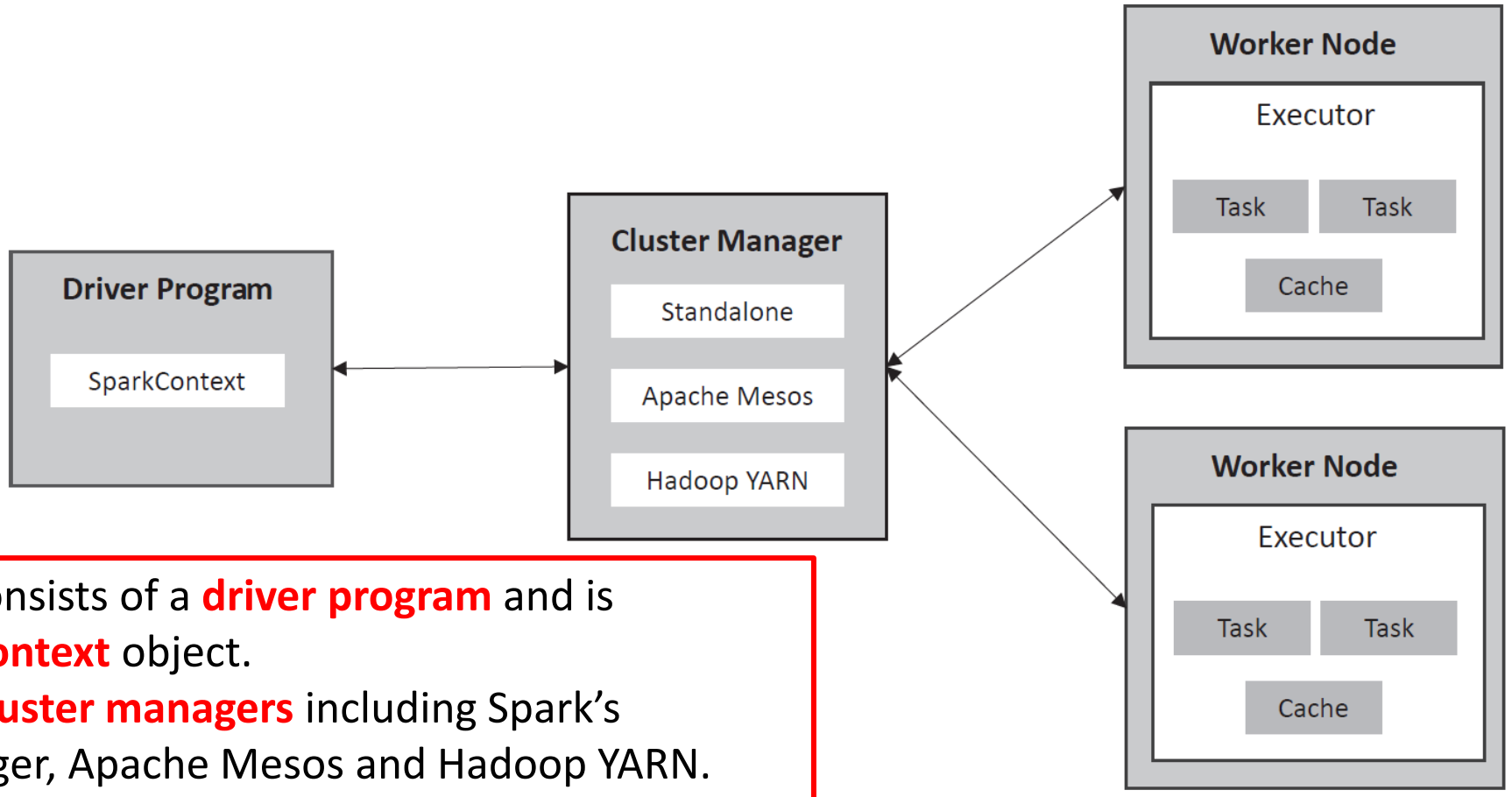
- Batch Analysis frameworks
  - Hadoop and MapReduce
  - Pig
  - Apache Oozie
  - Apache Spark
  - Apache Solr

Apache Spark

- Open-source cluster computing framework for **data analytics**.
- Supports **in-memory cluster computing** (faster than Hadoop).
- Allows **real-time**, **batch and interactive queries** and provides APIs for Scala, Java and **Python** languages.
- Spark **tools** for data analysis:



Components of a Spark cluster



- Each Spark application consists of a **driver program** and is coordinated by a **SparkContext** object.
- Spark supports various **cluster managers** including Spark's standalone cluster manager, Apache Mesos and Hadoop YARN.
- The cluster manager allocates resources for applications on the **worker nodes**.

Spark Operations

- Spark provides a data abstraction called **resilient distributed dataset** (RDD) which is a collection of elements partitioned across the nodes in a Spark cluster.
- Spark RDDs support two types of **operations**:
  - **Transformations** are used to create a new dataset from an existing one.
  - **Actions** return a value to the driver program after running a computation on the dataset.

Example Transformations

```
from pyspark import SparkContext

sc = SparkContext(appName="TransformationApp")

lines = sc.textFile("file:///root/spark/README.md")

# map transformation example
lineLengths = lines.map(lambda s: len(s))
lineLengths.take(5)
[14, 0, 78, 72, 73]

# filter transformation example
filteredLines = lines.filter(lambda line: line.find('Spark')>0)
filteredLines.take(3)
[u'# Apache Spark', u'rich set of tools including Spark SQL',
 u'and Spark Streaming for stream processing.']
```

Example Actions

```
# reduce transformation example
lineLengths = lines.map(lambda s: len(s))
totalLength = lineLengths.reduce(lambda a, b: a + b)
3526

# count transformation example
lines.count()
98
```

Apache Spark Python program for computing word count

```
from operator import add
from pyspark import SparkContext

sc = SparkContext(appName="WordCountApp")
lines = sc.textFile("file.txt")

counts = lines.flatMap(lambda x: x.split(' ')).map(
   lambda x: (x, 1)).reduceByKey(add)

output = counts.collect()

for (word, count) in output:
    print("%s: %i" % (word, count))
```

Summary

- Batch Analysis frameworks
  - Hadoop and MapReduce
  - Pig
  - Apache Oozie
  - Apache Spark
  - Apache Solr

Apache Solr

- Scalable and open-source framework for **searching** data, which is built on Apache Lucene the open-source library for **indexing** and search.
- To enable searching of documents, Solr creates an **index** of the documents.
- Solr provides a REST-like **web service** that can be used for indexing and querying.

Outline

- Batch Analysis frameworks
  - Hadoop and MapReduce
  - Pig
  - Apache Oozie
  - Apache Spark
  - Apache Solr