Big Data Analysis Platforms

SHYI-CHYI CHENG

Outline

- Review of Virtual Machine (虛擬機器回顧)
- Hadoop Platform (運算分析系統架構)
- MapReduce
- Introduction to Python (Python入門簡介)
- Python Spark Platform (Python Spark運算分析架構)
- Parallel Programming With Spark

Spark Ideas

- Expressive computing system, not limited to map-reduce model
- Facilitate system memory
  - avoid saving intermediate results to disk
  - cache data for repetitive queries (e.g. for machine learning)
- Compatible with Hadoop

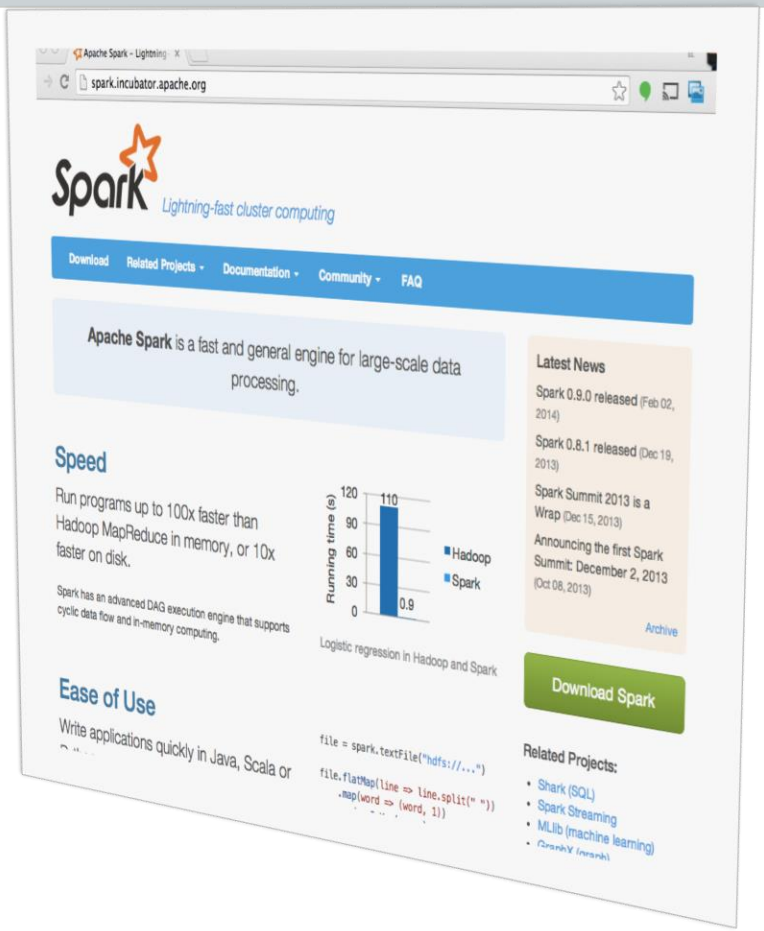
Apache Spark

- Originally developed in 2009 in UC Berkeley's AMP Lab
- Fully open sourced in 2010 – now a Top Level Project at the Apache Software Foundation

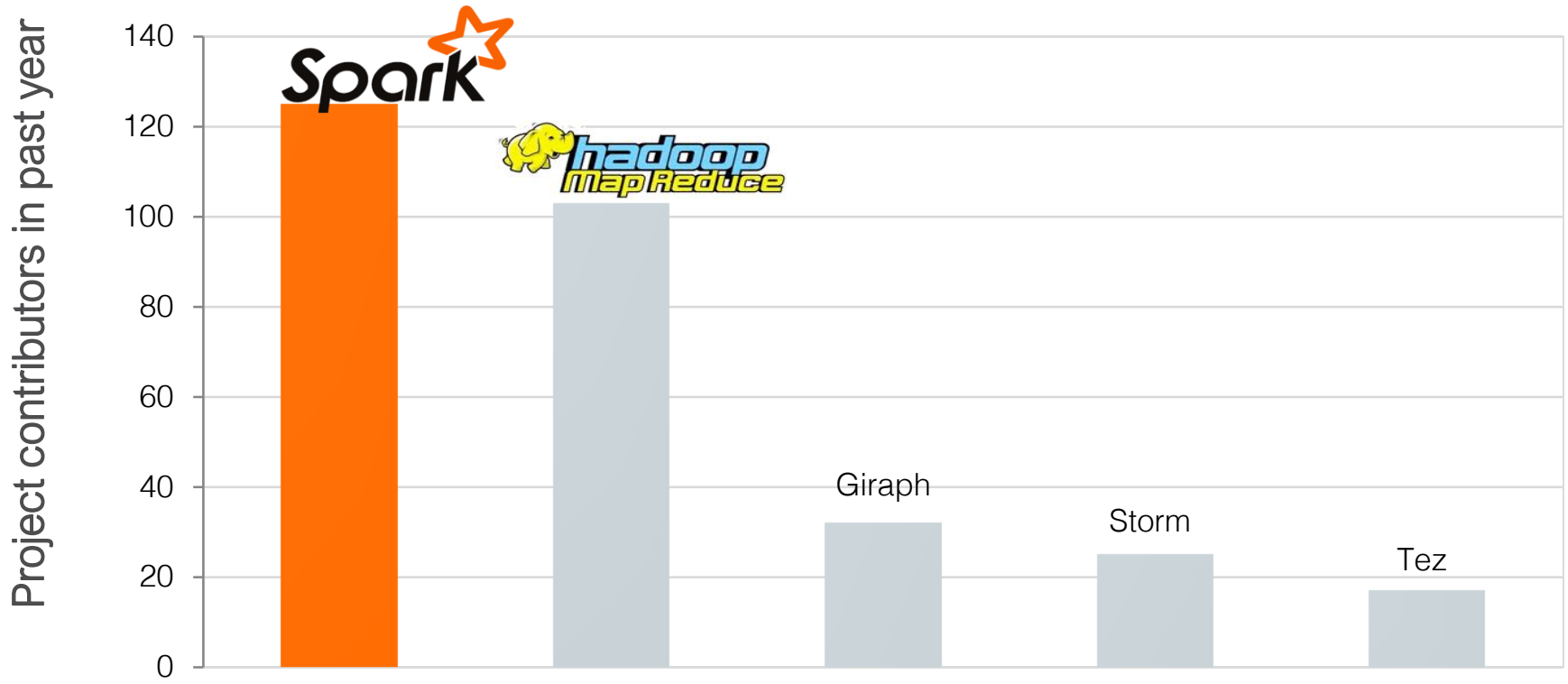
spark.apache.org

github.com/apache/spark

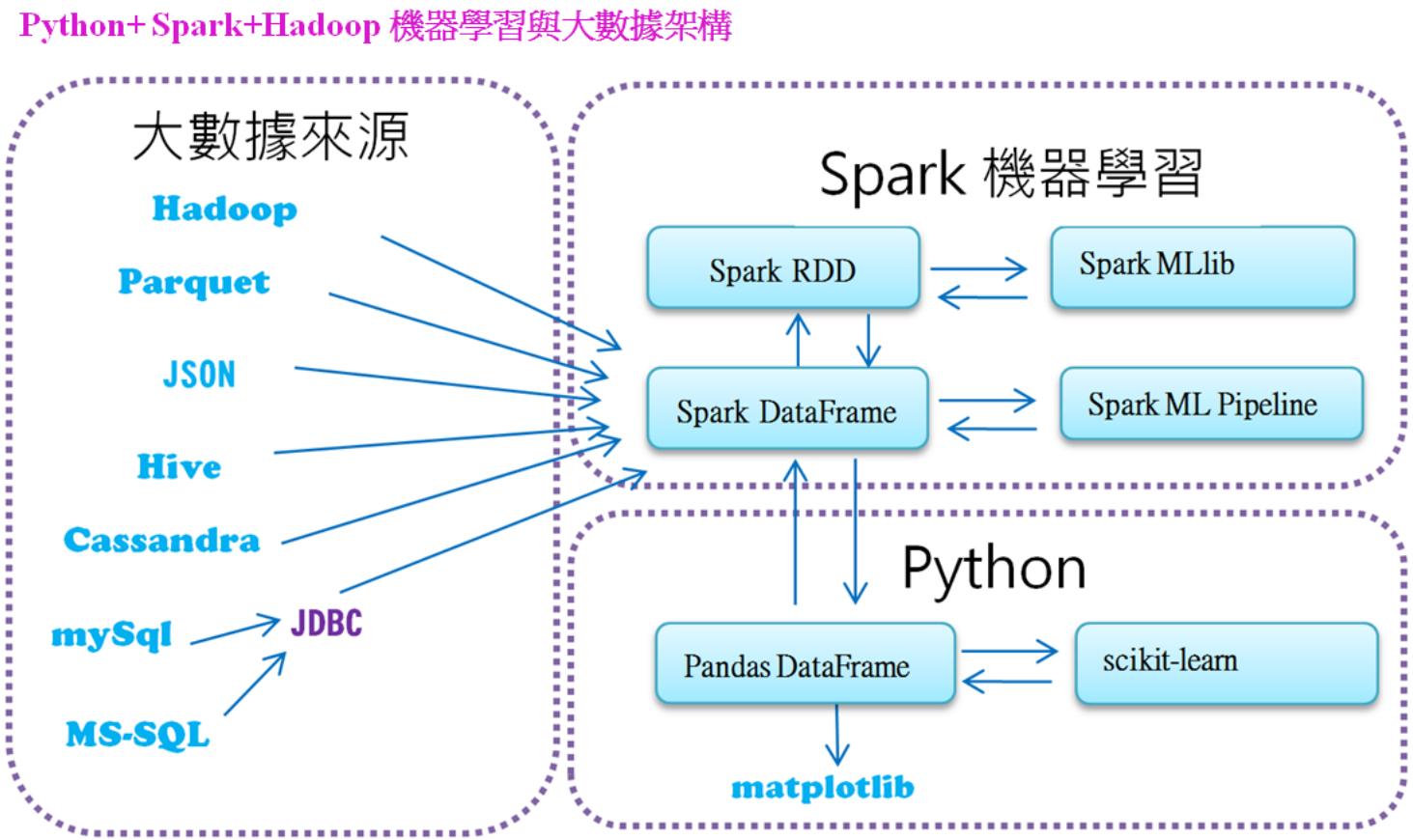
user@spark.apache.org



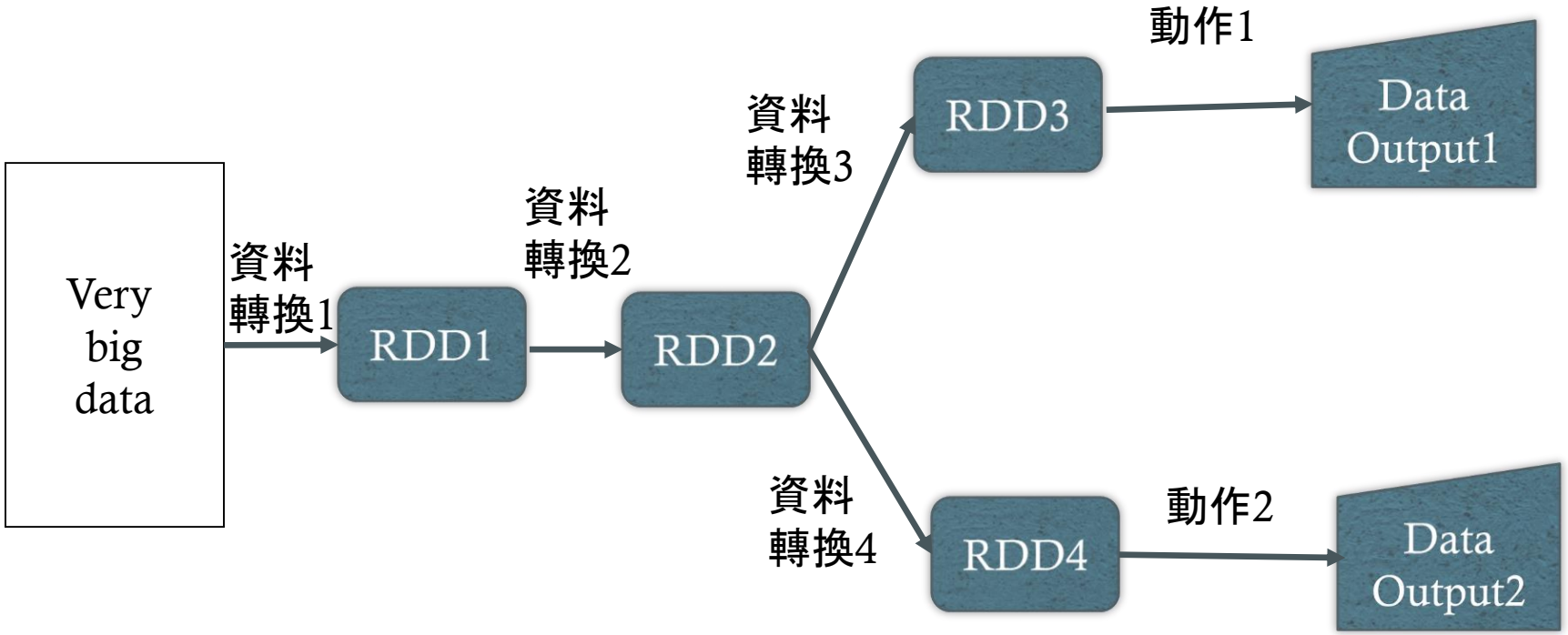
Spark is the Most Active Open Source Project in Big Data



Python Spark 大數據分析架構

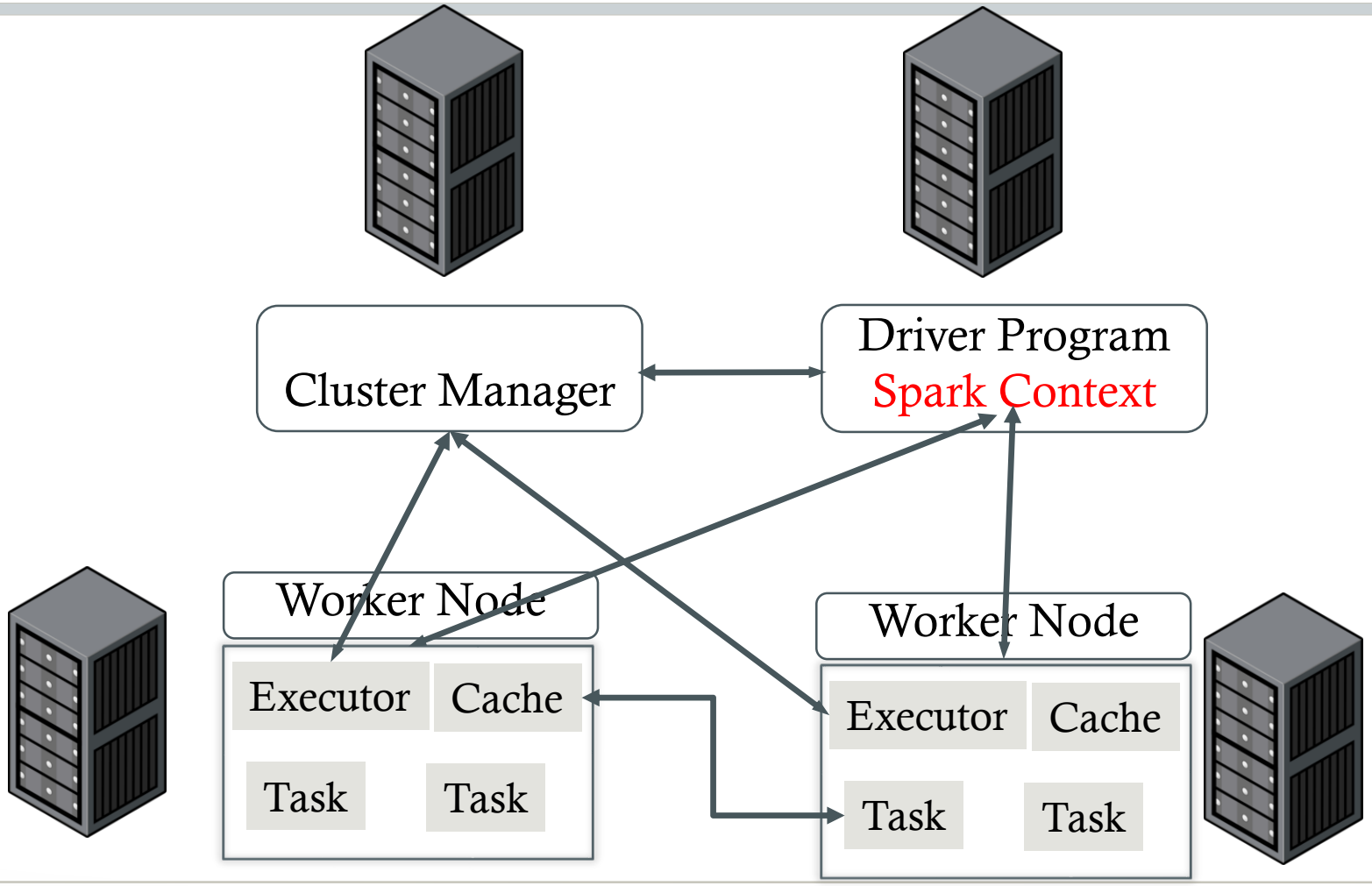


Python Spark RDD



RDD: Resilient Distributed Dataset, 是一種分散式記憶體架構

Python Spark架構



Python Spark架構

- Driver program: spark 程式碼，定義Spark Context開發Spark應用程式
- Spark Context透過Cluster Manager在整個群組電腦執行
- 每個Work Node包含
  - Executor負責執行Spark程式

Cluster Manager執行模式

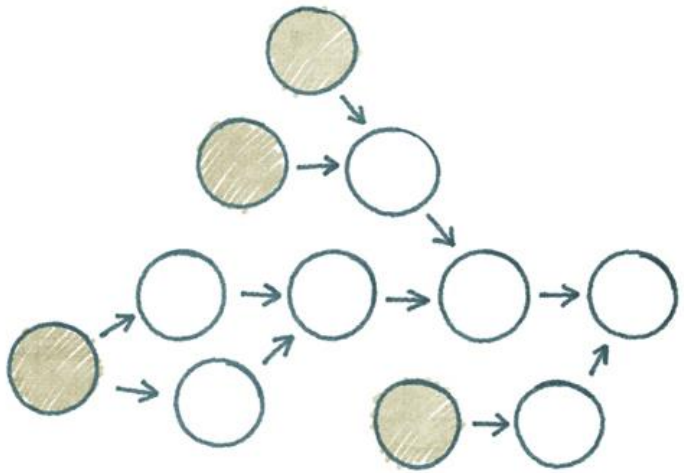
- 本機執行(Local Machine)
  - 只要在程式中載入Spark套件即可。
- Spark Standalone cluster:不需要架設Hadoop環境
- Hadoop YARN (Yet Another Resource Manager): Spark在Yarn Hadoop 環境中執行
- 在雲端執行:例如Amazon的AWS EC2平台

Supported Languages

- Java
- Scala
- Python
- Hive

Directed Acylic Graph (DAG)

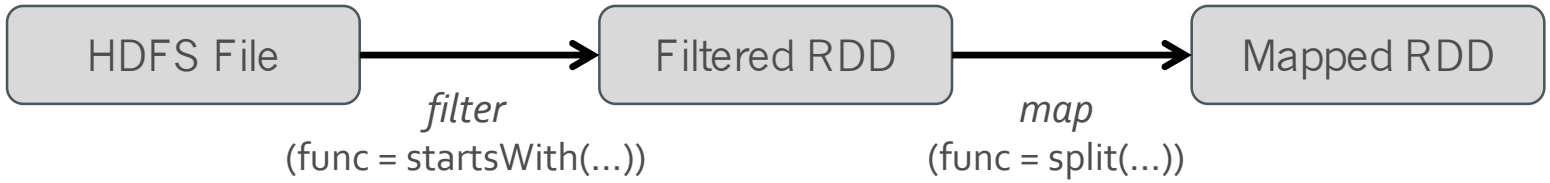
- Directed
  - Only in a single direction
- Acyclic
  - No looping
- Why does this matter?
  - This supports fault-tolerance



RDD Fault Recovery

RDDs track *lineage* information that can be used to efficiently re-compute lost data

```
msgs = textFile.filter(lambda s: s.startsWith("ERROR"))
               .map(lambda s: s.split("\t")[2])
```



RDD Operations

- Transformations
  - Creation of a new dataset from an existing
    - map, filter, distinct, union, sample, groupByKey, join, etc…
- Actions
  - Return a value after running a computation
    - collect, count, first, takeSample, foreach, etc…

Check the documentation for a complete list

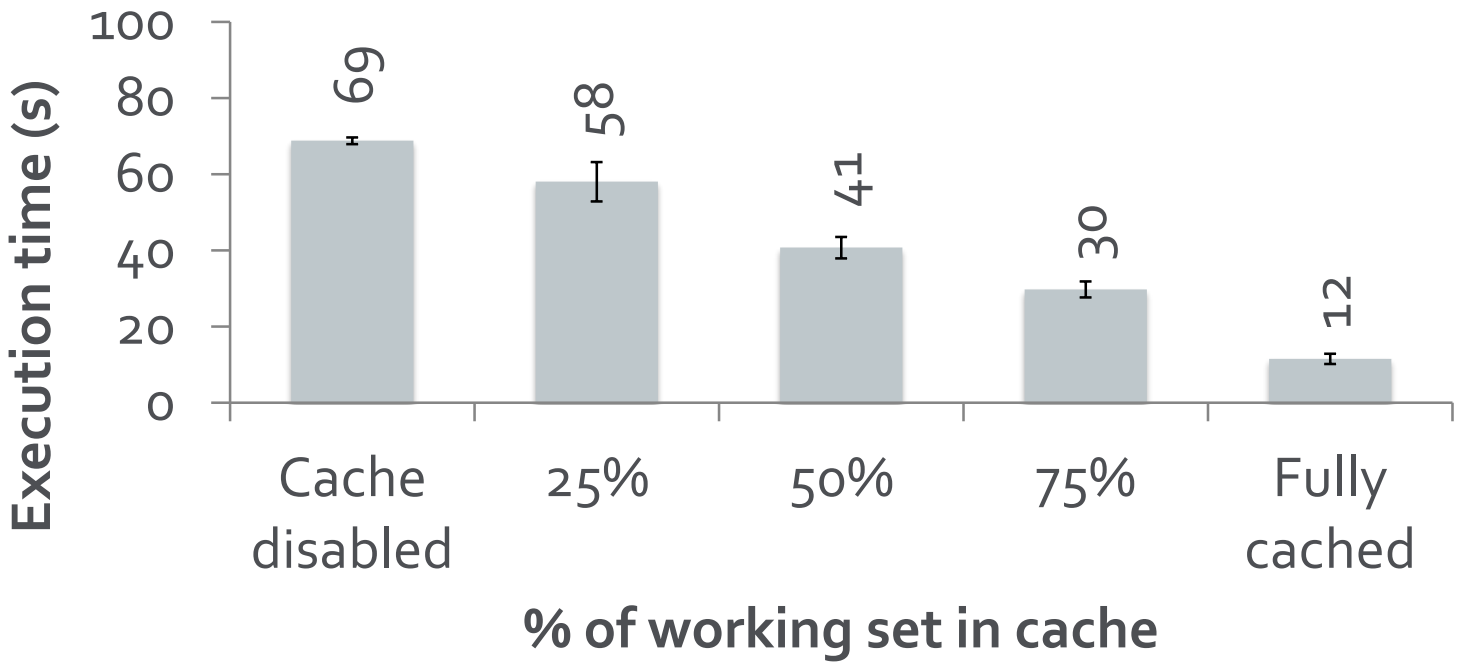
http://spark.apache.org/docs/latest/scala-programming-guide.html#rdd-operations

RDD Persistence / Caching

- Variety of storage levels
  - memory_only (default), memory_and_disk, etc…
- API Calls
  - persist(StorageLevel)
  - cache() – shorthand for persist(StorageLevel.MEMORY_ONLY)
- Considerations
  - Read from disk vs. recompute (memory_and_disk)
  - Total memory storage size (memory_only_ser)
  - Replicate to second node for faster fault recovery (memory_only_2)
    - Think about this option if supporting a web application

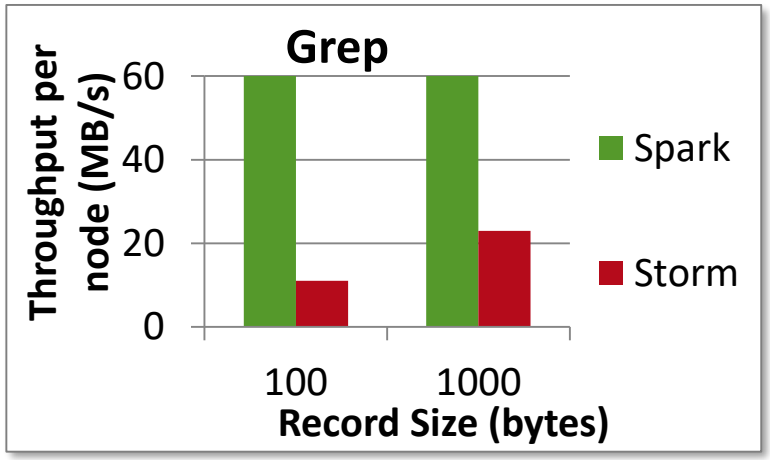
http://spark.apache.org/docs/latest/scala-programming-guide.html#rdd-persistence

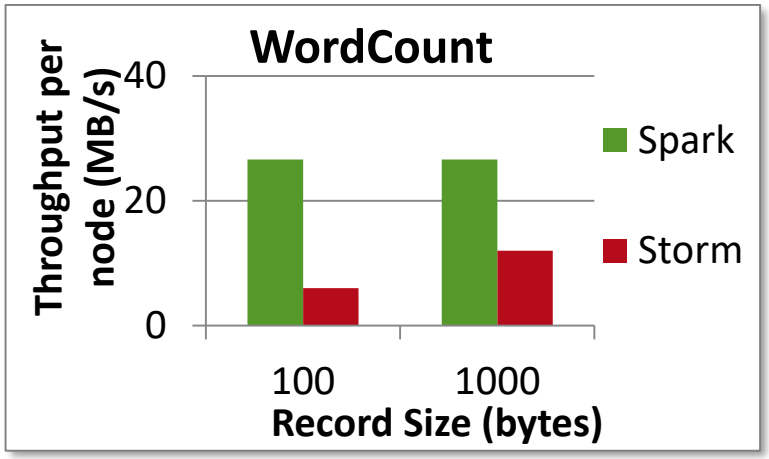
Cache Scaling Matters



Comparison to Storm

- Higher throughput than Storm
  - Spark Streaming: **670k** records/sec/node
  - Storm: **115k** records/sec/node
  - Commercial systems: **100-500k** records/sec/node





Interactive Shell

- Iterative Development
  - Cache those RDDs
  - Open the shell and ask questions
    - We have all wished we could do this with MapReduc
  - Compile / save your code for scheduled jobs later
- Scala – spark-shell
- Python – pyspark


```
mbo@mbo-ubuntu-vbox:~/mbo/spark$ MASTER=spark://localhost:7077 ./spark-shell
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 0.9.0-SNAPSHOT
      /_/

Using Scala version 2.9.3 (Java HotSpot(TM) Client VM, Java 1.6.0_45)
Initializing interpreter...
Creating SparkContext...
Spark context available as sc.
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Existing Jobs

- Java MapReduce
  - Port them over if you need better performance
    - Be sure to share the results and learning's
- Pig Scripts
  - Port them over
  - Try SPORK!
- Hive Queries….

Spark SQL

- Shark is officially dead, long-live Spark SQL
- Hive-compatible (HiveQL, UDFs, metadata)
  - Works in existing Hive warehouses without changing queries or data!
- Augments Hive
  - In-memory tables and columnar memory store
- Fast execution engine
  - Uses Spark as the underlying execution engine
  - Low-latency, interactive queries
  - Scale-out and tolerates worker failures

Word Count

- Java MapReduce (~15 lines of code)
- Java Spark (~ 7 lines of code)
- Scala and Python (4 lines of code)
  - interactive shell: skip line 1 and replace the last line with counts.collect()
- Java8 (4 lines of code)

```
SparkContext sc = new SparkContext(master, appName, [sparkHome], [jars]);
```

```
JavaRDD<String> file = sc.textFile("hdfs://...");
```

```
JavaRDD<String> counts = file.flatMap(line -> Arrays.asList(line.split(" ")))
        .mapToPair(w -> new Tuple2<String, Integer>(w, 1))
        .reduceByKey((x, y) -> x + y);
```

```
counts.saveAsTextFile("hdfs://...");
```

Network Word Count – Streaming

```
// Create the context with a 1 second batch size
val ssc = new StreamingContext(args(0), "NetworkWordCount", Seconds(1),
System.getenv("SPARK_HOME"), StreamingContext.jarOfClass(this.getClass))
```

```
// Create a NetworkInputDStream on target host:port and count the
// words in input stream of \n delimited text (eg. generated by 'nc')
val lines = ssc.socketTextStream("localhost", 9999, StorageLevel.MEMORY_ONLY_SER)
```

```
val words = lines.flatMap(_.split(" "))
```

```
val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)
```

```
wordCounts.print()
```

```
ssc.start()
```

Configuration

http://spark.apache.org/docs/latest/

Most Important

- Application Configuration
  http://spark.apache.org/docs/latest/configuration.html
- Standalone Cluster Configuration
  http://spark.apache.org/docs/latest/spark-standalone.html
- Tuning Guide
  http://spark.apache.org/docs/latest/tuning.html

Resources

- Pig on Spark
  - http://apache-spark-user-list.1001560.n3.nabble.com/Pig-on-Spark-td2367.html
  - https://github.com/aniket486/pig
  - https://github.com/twitter/pig/tree/spork
  - http://docs.sigmoidanalytics.com/index.php/Setting_up_spork_with_spark_0.8.1
  - https://github.com/sigmoidanalytics/pig/tree/spork-hadoopasm-fix
- Latest on Spark
  - http://databricks.com/categories/spark/
  - http://www.spark-stack.org/

Homework4: 安裝Python Spark Standalone Cluster

- 安裝步驟
  - Scala下載及安裝
    http://www.scala-lang.org/files/archive/
  - Spark下載及安裝
    http://spark.apache.org/downloads.html
  - 安裝Ananconda
    - Anaconda套件方便我們安裝Python環境
- Writing a report to describe your work

Any Questions?