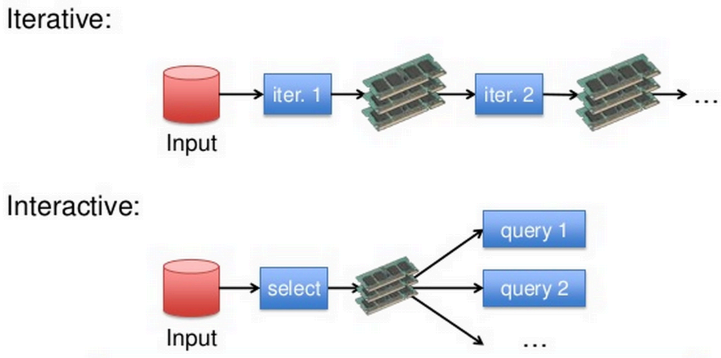
PySpark: next generation cluster computing engine
Wisely Chen

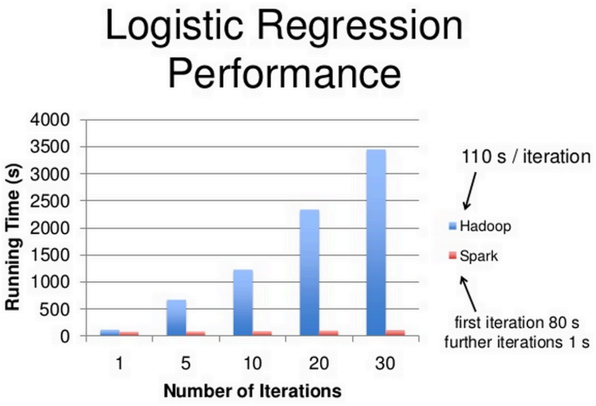
Abstract:

Apache Spark™ is a fast and general engine for large-scale data processing. It is an in-memory cluster computing framework, originally developed in UC Berkeley. Base on it's project page's evaluation, machine learning programming can run program 100x faster than Hadoop MapReduce. And Spark can run on Hadoop 2's YARN cluster manager, and can read any existing Hadoop data. Currently, it supports Scala, Java and Python for writing spark programs.

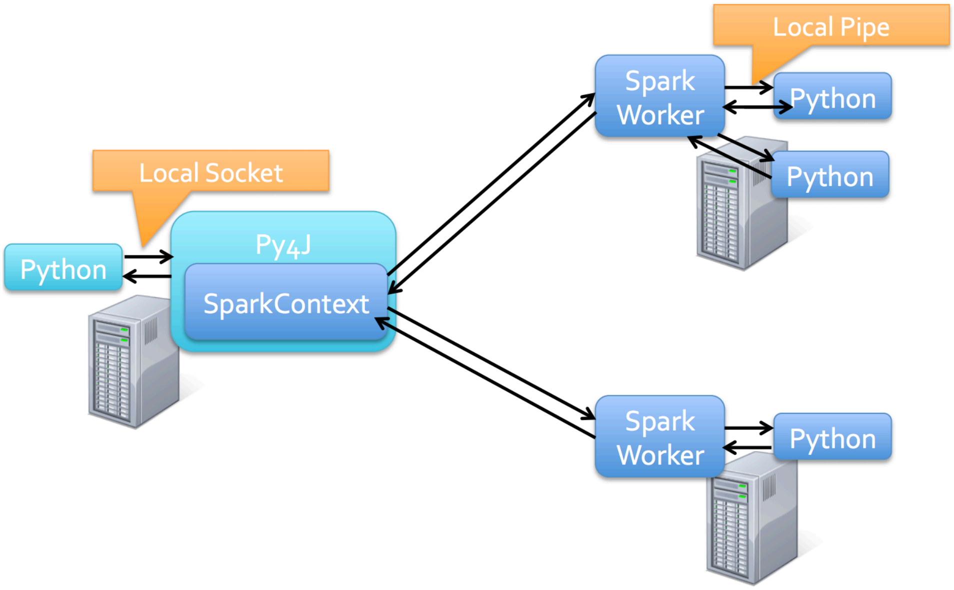
In this talk, I will introduce the general concept of Spark's infrastructure. In general hadoop mapreduce model, it's performance is limited by the replication/disk IO. Spark provide a new in-memory cluster computing framework, which is RDD (Resilient Distributed Datasets) which is a read-only, partitioned collection of records through which Spark achieves memory abstraction, fault tolerance and fast in-memory computation, followed by Job Scheduling and Memory Management.



Spark is 100x faster than original Map Reduce interactive programming based on this architecture, especially in machine learning algorithm.



I will show what is PySpark, a Python API which build on Spark framework. This API is built on top of the JAVA API using Py4j and it can interactive use through Python REPL.



PySpark will leverage the Spark's great computing power on machine learning program and coding like a common Python program. I will also demo how to use write a machine learning algorithm via PySPark. I will show how to write a series of PySpark machine learning program. The bellow is the simple word count example.

```
sc = SparkContext(...)
lines = sc.textFile(sys.argv[2], 1)
counts = lines.flatMap(lambda x: x.split(' ')).map(lambda x: (x, 1)) \
        .reduceByKey(lambda x, y: x + y)
for (word, count) in counts.collect():
        print "%s : %i" % (word, count)
```

We can find how easy the PySpark are.

At the end ead-to-head comparison between two programs doing same work - one written in Hadoop MapReduce and the other written using PySpark I will also conclude about the companies currently using Spark's use cases.