Spark

WE'LL COVER THE FOLLOWING

Apache Spark is one of the most recent and the quickest developing projects in the big data space. Its design is intended to offer solution to similar problems as Hadoop. It additionally provides similar advantages, including Distributed computing, higher level interfaces, and Fault Tolerance. Spark separates itself in different significant ways:

- Spark offers various performance improvements and makes extraordinary use of the memory. The documentation of Spark claims programs keep running up to 100X quicker than Map Reduce with respect to memory and 10X faster with respect to disk.
- Spark is designed with a rich set of interfaces and APIs making it simple to write applications of distributed data processing. It is possible for you to write spark jobs in Scala, R, Java, and Python. You can additionally utilize SQL to the created query tables in Hive, while making use of Spark as the initial execution engine.
- The initial concept of Spark's Resilient distributed datasets is utilized by all of the other tools present in the Spark environment, so any further developments made to the core is translated into advancements for the remaining tools.
- The APIs of Spark support functional programming concepts, making the code concise and expressive much more in this way, when it is compared to Map Reduce code.

Example code

```
text_file = spark.textFile("hdfs://...")

text_file.flatMap(lambda line: line.split())
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a+b)
```

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it interactively from the Scala, Python and R shells.

http://spark.apache.org/