**HADOOP**

**What Is hadoop?**

Apache Hadoop project, led by Douglas Cutting, is an open source software framework that supports data-intensive distributed applications under a free license. It enables applications to work with thousands of nodes and petabytes of data. Hadoop was inspired by Google's MapReduce and Google File System (GFS) paper.

* Need to process 100TB data sets
* On 1 Node:

-Scanning @ 50MB/sec= 23 days

* On 1000 Node cluster:

-Scanning @ 50MB/sec= 33 mins

|  |  |
| --- | --- |
| Development status | Active |
| Written in | Java |
| Operating system | Cross-platform |
| Type | Distributed File System |
| License | Apache License 2.0 |
| Website | <http://hadoop.apache.org> |

**HDFS**

Primary storage system used by hadoop applications. Multiple replicas of data blocks distributed over compute nodes. Reliable, rapid computation

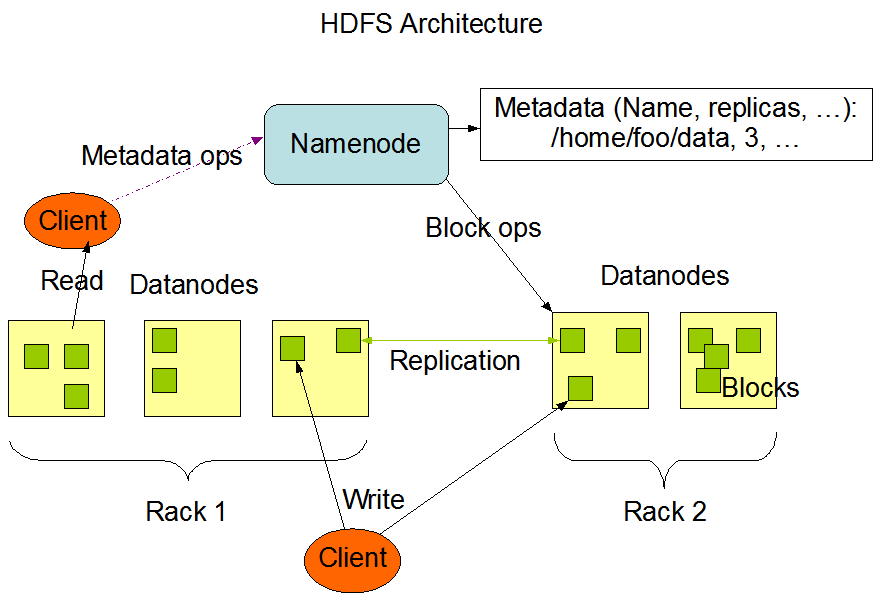
**Goals of HDFS**

* Very large distributed file system - 10k nodes, 100million files, 10-100 petabytes of data
* Hardware failure – detection and quick recovery of hardware failure
* Optimized for batch processing - Data locations exposed so that computations can move to where data resides– Provides very high aggregate bandwidth
* Portability Across Heterogeneous Hardware and Software Platforms

**HDFS ARCHITECTURE**

* NAMENODE – A MASTER SERVER
  + - * manages file systems namespace i.e., opening, closing, renaming files.
      * regulates access to files by clients
      * determines the mapping of blocks to the data nodes
* DATANODE – ONE OR MORE SLAVES
  + - * manage storage of blocks
      * performs block creation, deletion, replication upon instruction from the namenode
* SECONDARY NAMENODE –

The NameNode stores modifications to the file system as a log appended to a native file system file, edits. When a NameNode starts up, it reads HDFS state from an image file, fsimage, and then applies edits from the edits log file. The secondary NameNode merges the fsimage and the edits log files periodically and keeps edit log size within a limit.

**How hadoop will do this ?**

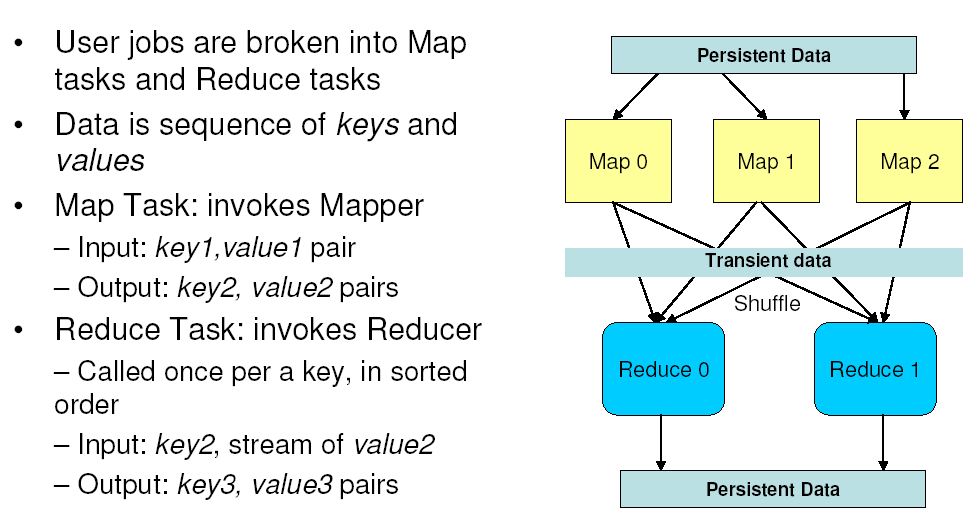
**What is MAPREDUCE?**

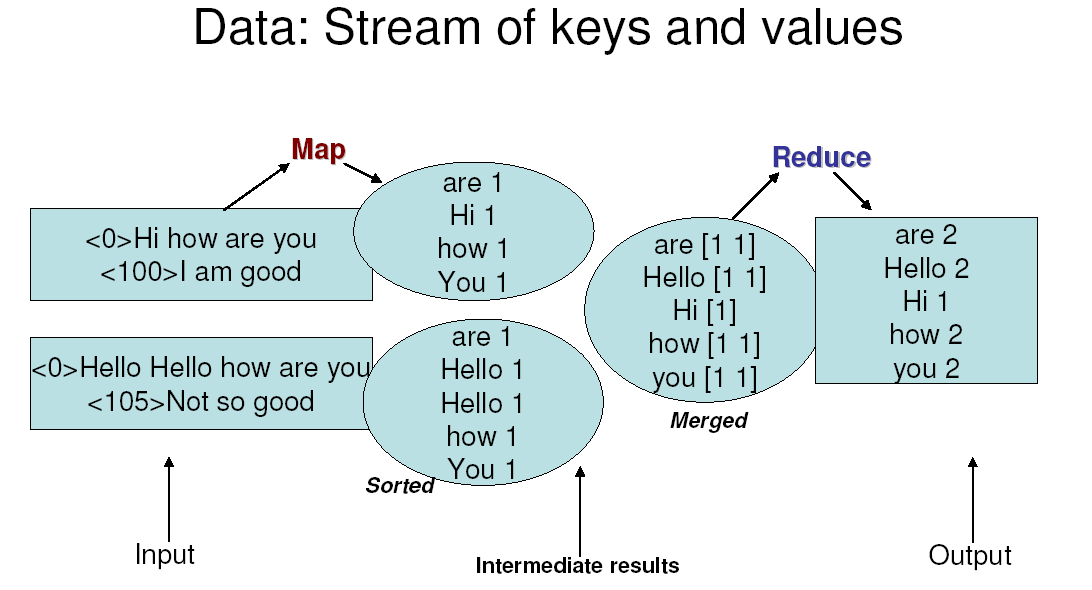
Programming model used by Google. A combination of the Map and Reduce models with an associated implementation. Used for processing and generating large data sets. highly scalable and can be used across many computers.

**How MAPREDUCE works?**

Map returns information. Reduces accepts information. Reduce applies a user defined function to reduce the amount of data.

**Map Reduce data flow?**

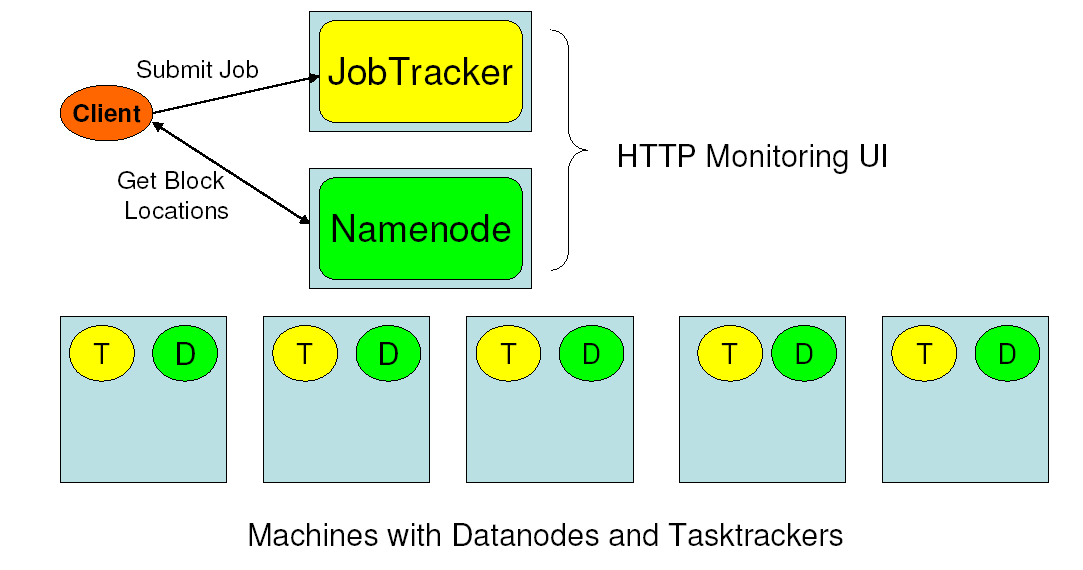


**Word Count using Map-reduce?**

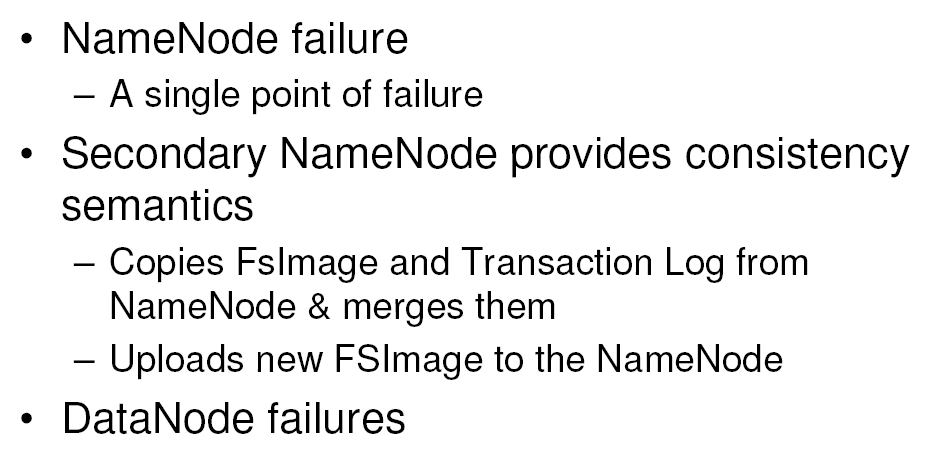
**MapReduce Architecture?**

* MapReduce Master JobTracker –
  + - * + Accepts MR jobs submitted by user
        + Asssign Maps and Reduces tasks to TaskTrackers
        + Monitor tasks and TaskTracker status, re-executes failed tasks
* MapReduce Slaves TaskTrackers –
  + - * Runs Maps and Reduce tasks upon instruction from the JobTracker
      * Manage storage and transmission of intermediate data

**HDFS + MR Cluster**



**HDFS : Handling Failures**

****

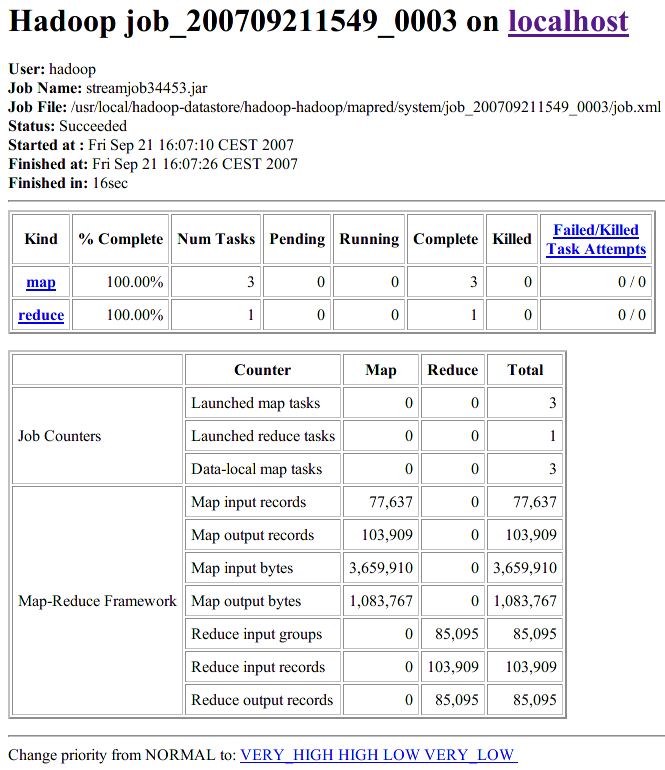
**Hadoop Web Apps for monitoring**

The Job Tracker can be found at:  http://localhost:50030

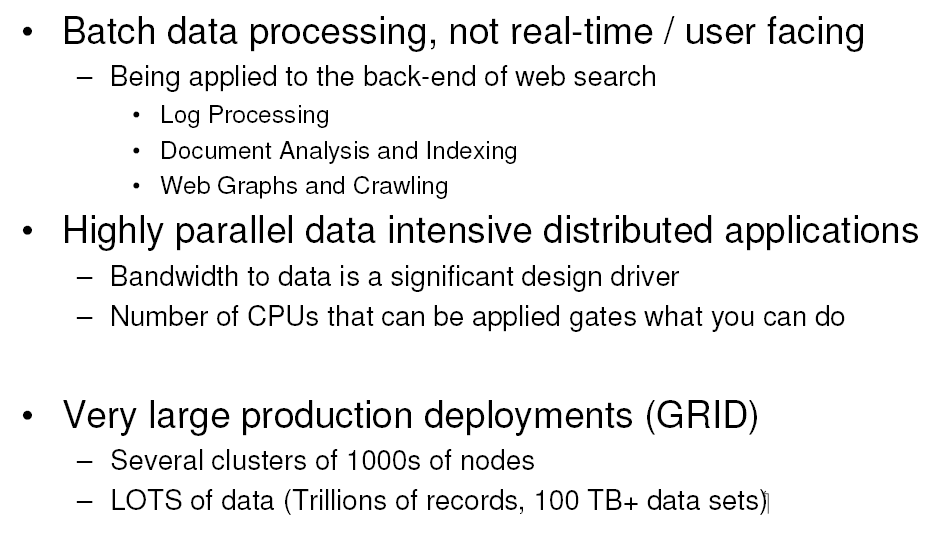
The Task Tracker can be found at: http://localhost:50060

The NameNode / Filesystem / log browser can be found at:  http://localhost:50070

The SecondaryNameNode can be fund at: http://localhost:50090



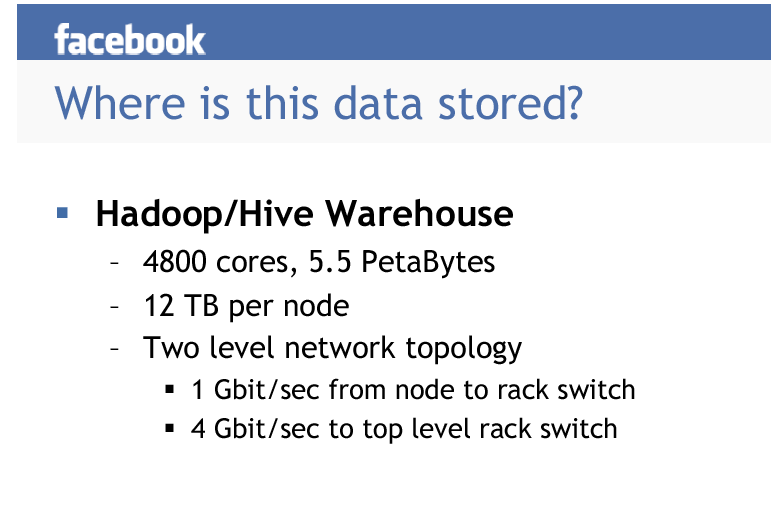
**Applications**

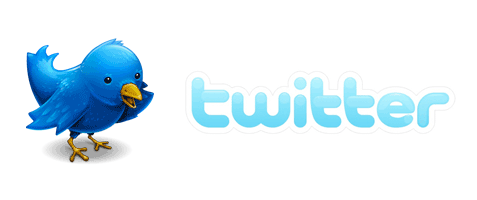
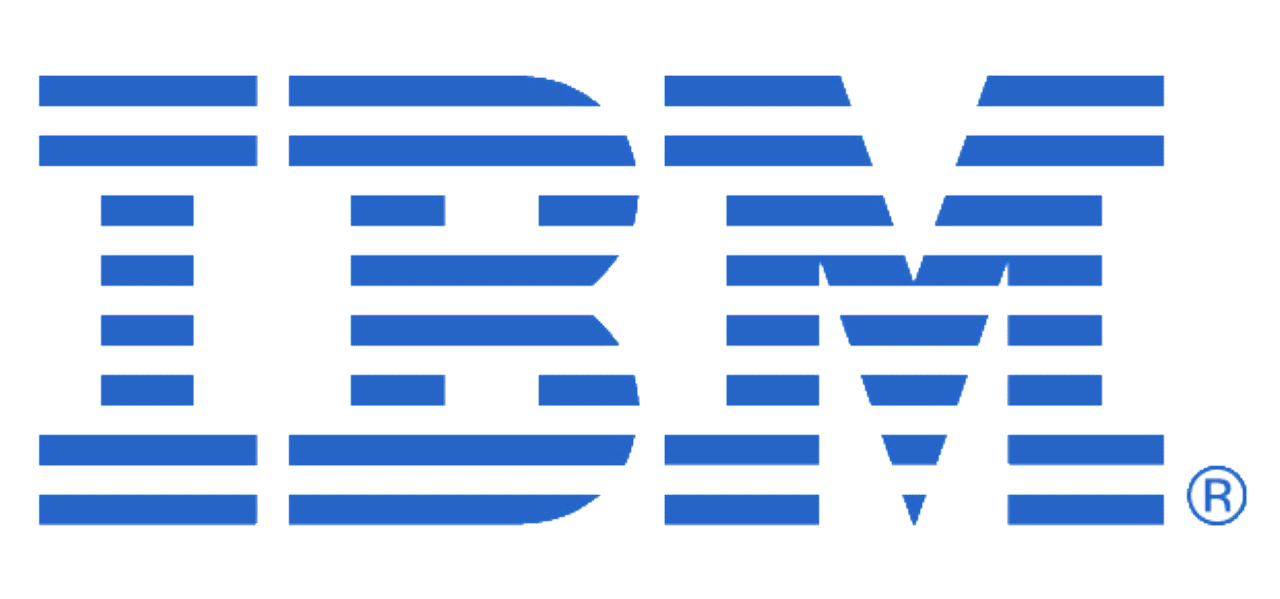


Hadoop at Yahoo



* Yahoo has ~20,000 machines running Hadoop
* largest clusters are currently 2000 nodes
* Several PetaBytes of user data (compressed, replica less)
* It runs hundreds of thousands of jobs every month





**COMPILED AND EDITED BY**

**SRIPARNA CHAKRABORTY**

**REFERENCES**

**“Pro Hadoop” by Jasson Venner**

**Web references**

**Apache hadoop site -http://hadoop.apache.org/**

**Hadoop wiki -http://wiki.apache.org/hadoop/**

**WITH SPECIAL THANKS TO PROF. SUNIRMAL KHATUA OF THE COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, UNIVERSITY OF CALCUTTA, WITHOUT WHOM THIS PRESENTATION WOULD NOT HAVE BEEN POSSIBLE**