**INTRO**

<http://analyticstraining.com/2015/all-about-hadoop-the-what-how-and-why/>

Apache Hadoop (more commonly in Hadoop): an open-source framework mainly used to process and store big data. Need to have additional software such as Apache HBase, Apache Spark ... to make it work more efficient

**Hadoop requirements:**

* JRE 1.6 and higher
* SSH between nodes, responsible for startup and shutdown scripts

**Hadoop basic feature:**

* Open-source
* Is a software framework (more than a program)
* Distributed framework (HDFS): data will be divided and store in different computers -> allow to work with millions of MBs simultaneously.
* Hadoop Common: libraries and utilities used by other Hadoop modules.
* Hadoop YARN(Yet Another Resource Negotiator): a resource management for processes running on Hadoop.
* Hadoop MapReduce: Utilize to process data. 2 different processes: Map and reduce. Map: Master node breaks large file into chunks (usually 64Mb, can be change), worker nodes can operate on different chunks at the same time. Then, reduce is that the master node take the result and combine data based on key to produce output.
* Locational settings: to solve the problem: nodes have location info, when their network switch identity -> error locating, redundancy.
* Hadoop clustering: master node in charge of several worker nodes. Job of master nodes is NameNodes, TaskTrackers, DataNodes, JobTrackers while that of worker nodes are only DataNodes and TaskTracker.

**Why?**

Every firm has different issues with data, depend on that to choose appropriate big data solution -> not always recommend Hadoop. However, the following are the reasons to pick Hadoop rather than RDBMS:

* Massive storage space: split data in to blocks and store it many hardware
* Low cost: use common hardware.
* Computing power: more nodes = more power.
* Scalability: more node = bigger system
* Flexibility: can store both structured and unstructured data.
* Failure tolerance: no single point of failure. One goes down, jobs reassign to other. Multiple copy of all data are stored automatically

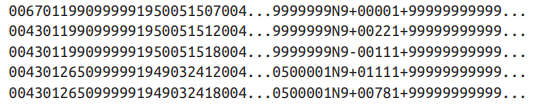
**Challenges of Hadoop:**

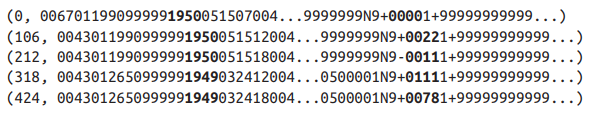
* MapReduce not good for all problems: multiple file is created in MapReduce -> inefficient for advanced analytic computing
* Professional workers: require sufficient Java skills to be productive with MapReduce.
* Data security.

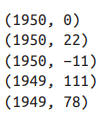
**BASIC**

1. **Overview:**

* A cluster have many machine
* Each machine in has 2 parts: **MapReduce** (called as a **TaskTracker**) and **HDFS** (called as a **DataNode**)
* **JobTracker**: single, receive jobs from users and divide it to **TaskTrackers**. **TaskTrackers** run assigned task and then return the result. If a **TaskTracker** fail, the **JobTracker** forward the task of that **TaskTracker** to other one.
* **Name node**: single. Keep info on data location. Replicate the data.

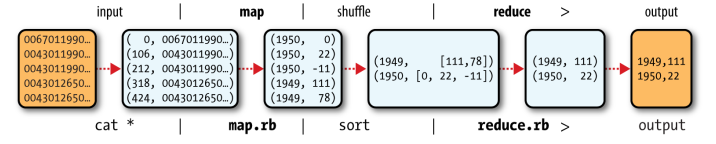
1. **MapReduce:**
2. **How Map and Reduce** **program works**: has 2 main phases: Map and Reduce. Each phase has key-value pair as input and output, can be chosen by user. Example: Raw data about weather

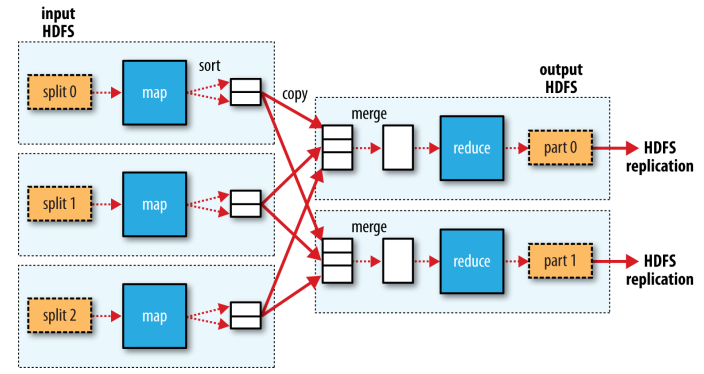
Input into map function as key-value pairs:

Keys are the line offsets, which we don’t have to worry in map function. The map function get the year and the air temperature( the bold text), emitting them out as output.

Before sending to the reduce function, the output is sorted by key.

In the reduce function, the maximum temperature is picked.

Map and Reduce data flow:

1. **MapReduce in Hadoop:**
   * MapReduce is a job which consists of the input data, the MapReduce program and configuration info.
   * Hadoop runs it by dividing it into tasks, use YARN (explain later) to assign task for nodes and re-assign task in case the node fail.
   * The input is divided into fixed-size pieces namely splits, 128MB by default and changeable. Hadoop create one map task for each split.
   * The number of reduce tasks is varied from 0 to many:
     + In case of many reduce task, the output of each map will be partitioned into N parts for N reduce tasks.
2. The Hadoop Distributed Filesystem (HDFS)
3. Blocks

Files in HDFS are divided into chunks, usually 128 MB and can be changed.

Why?

* + File can be larger than single disk storage.
  + Simplify storage management.
  + Provide fault tolerance and availability (thanks to replications)

1. NameNodes and DataNodes

HDFS cluster operates in a master-worker pattern: a NameNode (master) and a handful of DataNodes (workers)

* + NameNode stores filesystem tree, metadata persistently on disk in the form of 2 files: the namespace image and the edit log. Besides, it also stores the block locations temporarily.
  + Datanodes store and retrieve blocks when they are told to and report back periodically the lists of blocks they are storing.

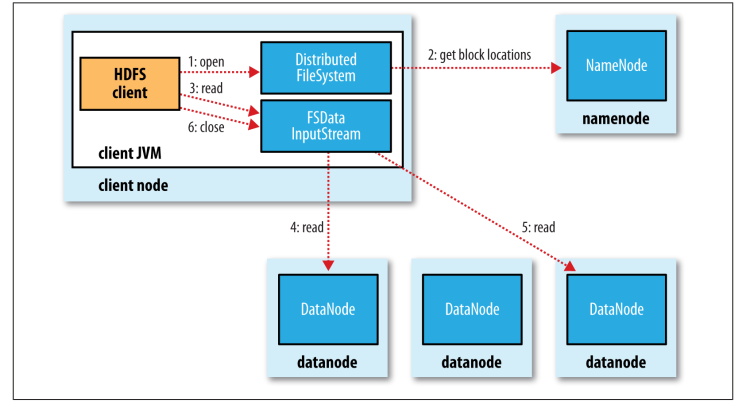
1. Hadoop high availability (HA)

Only one NameNode would cause SPOF. HA fixes it with a secondary NameNode. Hadoop 2 has HDFS HA, in which a pair of

NameNodes works in an active-standby configuration. If the first fails the second will take place and vice verse.

* + Share the edit note for synchronization when the first fails and second comes up.
  + Report from DataNodes send to both NameNodes

1. File Reading:



Note: if the InputStream cannot connect to the DataNode, it will try the next closest one and report the failure.

1. File writing: 