

EPL646 – Advanced Topics in Databases

Hadoop

<http://www.cs.ucy.ac.cy/~dzeina/courses/epl646/labs/lab.html>



Εισαγωγή - Hadoop

- **MapReduce** model is patented and is proprietary to Google
- Hadoop is an open-source implementation of **MapReduce**
- The implementation was based mainly on two articles published by the initiators of MapReduce and described model
- Created by the Apache Foundation with financial support mainly from the Yahoo!



Εισαγωγή - Hadoop

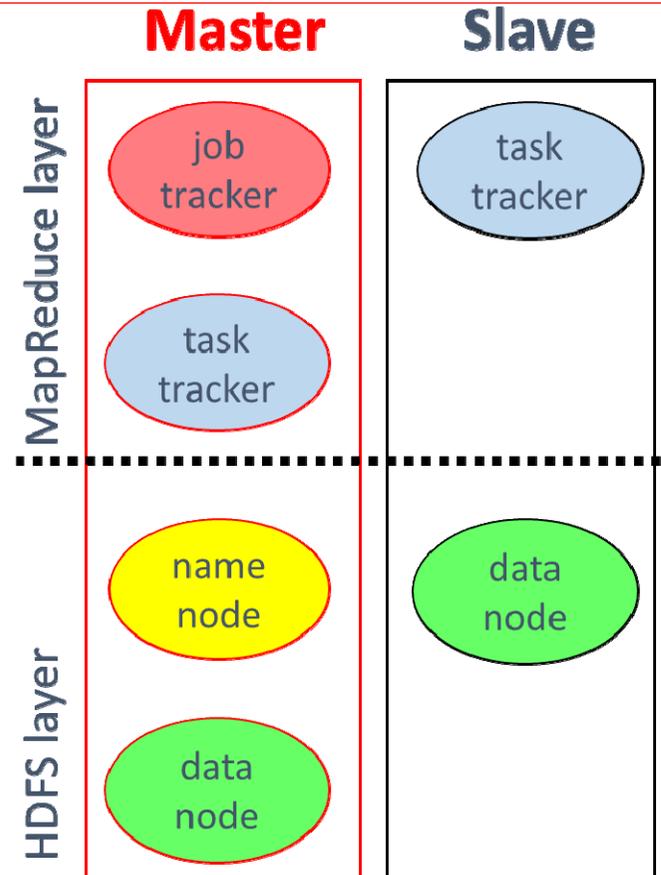
- The Hadoop allows editing and analyzing vast amounts of data to **Petabyte** scale (1 PB = 1024 TB).
- The purpose of the analysis is the extraction of useful information and trends
 - In addition to Yahoo!, it is used by other organizations such as Facebook, AOL, Netflix, Amazon, Apple, eBay, etc.
 - Facebook said in June of 2012 that it holds the world's largest Hadoop cluster with total data volume ~ 100 PB and increases by ~ 0.5 PB day.

HDFS

- HDFS: the Distibuted Hadoop File System
 - Inspired by the Google File System (GFS) which of course is patented
 - Designed for storing large amounts of data
 - An HDFS cluster size is usually 64MB. The NTFS (Microsoft) and ext4 (Linux) filesystems usually have a cluster size of just 4 KB.
- HDFS provides reliability by copying data to more than 1 nodes
 - When a node is not available, data can be retrieved from other nodes (because there are duplicates)
 - This beneficial characteristic is called replication
- The master node of HDFS has the **name node** role
 - The name node maintains the index that describes **where** each file or file chunk is located (i.e. in what node)
 - All remaining nodes have the **data node** role (i.e. store data in the cluster)

Hadoop Architecture

- The **MapReduce** mechanism of Hadoop follows the master-slave model
 - The master is responsible for the distribution of tasks
 - The slaves perform the tasks and return the result to the master
 - The master node in Hadoop has the **job tracker** role (i.e. the supervision and distribution of tasks)
 - The slave nodes have the **task tracker** role (task execution assigned from the job tracker)



The master node may perform and the roles of slave nodes (task tracker & data node) especially when we have a small cluster with limited resources

That's how we will design our own cluster

Example – WordCount

- Typical example of how MapReduce works
 - Counts how many times each word is displayed in one or more files
 - Processed through Hadoop and thus the processing is distributed
- The map function takes as input <key, value> pairs where:
 - the **key** is the offset of a row in the file (the location of the line in the file)
 - the **value** is a whole line of one of the files
 - We don't care about the location of the line within the file and thus we ignore the key.
- The map function gives output to the reduce function <key, value> pairs where:
 - the **key** is every one of the words
 - the **value** is always the number 1

Παράδειγμα – WordCount

- The reduce function takes as input a **< key, <list of values>>** pair where:
 - **key** is one of the words
 - **value** is a list that contains the number 1 as many times as many occurrences of the key we have in our files
- The reduce function gives as output a **<key, value>** pair where:
 - **key** the word it took as input
 - **value** the sum of the ones in the list of values

Example– WordCount

Line 1 the government said yesterday it was on track to cover the financing needs for the next three months but for the next three years there was no choice to cover



Input to map: key = 0, value = the government said yesterday [...]

Output: <the,1>, <government,1>, <said,1>, <yesterday,1>, <it,1>, <was,1>, <on,1> <choice,1> , <to,1> <cover,1>



Input to reduce:
key = the, value = <1,1,1,1>
key = government, value = <1>

Output: <the, 4>, <government, 1>, <said, 1> ... <to, 2>, <cover, 2>

Πρακτική – WordCount

- Θα χρησιμοποιήσουμε τον virtual machine που έχει ήδη εγκατεστημένο το Hadoop, Java, Eclipse και ό,τι άλλα χρειάζεται για να τρέξουμε το παράδειγμα word count.
- Χρήσιμες πληροφορίες:
- Hadoop API:
 - <https://hadoop.apache.org/docs/r2.7.4/>
- Namenode:
 - <http://hadoopmaster:50070>
- Resource Manager:
 - <http://hadoopmaster:8088>

ResourceManager

The screenshot displays the Hadoop ResourceManager web interface. The browser address bar shows 'localhost:8088/cluster'. The page title is 'All Applications'. The Hadoop logo is visible in the top left. The interface is divided into several sections:

- Cluster Metrics:** A table showing various cluster statistics.

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0B	0B	0B	0	0	0	0	0	0	1	0
- Scheduler Metrics:** A table showing scheduler configuration.

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>
- Applications Table:** A table with columns for ID, User, Name, Application Type, Queue, StartTime, FinishTime, State, FinalStatus, Progress, Tracking UI, and Blacklisted Nodes. The table is currently empty, displaying 'Showing 0 to 0 of 0 entries'.
- Navigation:** A sidebar on the left contains links for 'Cluster', 'About Nodes', 'Node Labels', 'Applications', and 'Scheduler'. The 'Applications' link is expanded, showing a list of application states: NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, and KILLED. A 'Tools' link is also present at the bottom of the sidebar.

HDFS name node

Overview 'localhost:54310' (active)

Started:	Tue Nov 07 21:45:24 EET 2017
Version:	2.7.4, rcd915e1e8d9d0131462a0b7301586c175728a282
Compiled:	2017-08-01T00:29Z by kshvachk from branch-2.7.4
Cluster ID:	CID-7277add8-0342-4b6e-9e5e-4683ba25a357
Block Pool ID:	BP-1727959122-127.0.1.1-1506114028539

Summary

Security is off.

Safe mode is ON. The reported blocks 2 has reached the threshold 0.9990 of total blocks 2. The number of live datanodes 1 has reached the minimum number 0. In safe mode extension. Safe mode will be turned off automatically in 6 seconds.

5 files and directories, 2 blocks = 7 total filesystem object(s).

Heap Memory used 58.74 MB of 129 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 37.52 MB of 38.38 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	18.58 GB
DFS Used:	160 KB (0%)
Non DFS Used:	9.1 GB
DFS Remaining:	8.51 GB (45.81%)
Block Pool Used:	160 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)

Πρακτική – WordCount

1. Ξεκινήστε το Hadoop ☺ (start-all.sh).
 1. Βεβαιωθείτε ότι ξεκίνησε με την εντολή jps
2. Συμπληρώστε τον κώδικα του WordCount μέσα στην Eclipse.
3. Τρέξτε τον κώδικα :
 1. input(ο φάκελος HDFS όπου βρίσκονται τα αρχεία μας)
 2. output (ο φάκελος στο HDFS όπου θέλουμε να αποθηκευθούν τα αποτελέσματα του MapReduce).
4. Αντιγράψτε τα επεξεργασμένα αρχεία από το HDFS στο local file system (δηλ. στα Ubuntu).

Questions?

<http://www.cs.ucy.ac.cy/~dzeina/courses/epl646/labs/lab.html>

