1 Εγκατάσταση Spark 2.4.4 σε single node (Pseudo Distributed)

Επικοινωνία:
kbitsak@cslab.ece.ntua.gr Κωνσταντίνος Μπιτσάκος, nchalv@cslab.ece.ntua.gr Νικόλαος Χαλβαντζής

Στο κείμενο αυτό περιγράφεται η εγκατάσταση του Spark framework σε single node (pseudo-distributed) locally. Περιγράφεται επίσης πως μπορούμε να εκτελέσουμε ένα job στο Spark cluster που έχουμε δημιουργήσει. Ο οδηγός αυτός είναι βασισμένος στις οδηγίες που δίνονται στην επίσημη σελίδα του Spark (https://spark.apache.org/docs/2.4.4/index.html) και μπορεί να χρησιμοποιηθεί για οποιοδήποτε Ubuntu-based Linux σύστημα.

Η εγκατάσταση περιλαμβάνει τα εξής συστήματα:

- HDFS: Κατανεμημένο filesystem, από το οποίο διαβάζουν και γράφουν τα Spark jobs.
- Spark: Open source framework γενικού σκοπού για επεξεργασία δεδομένων, που περιλαμβάνει υλοποίηση του προγραμματιστικού μοντέλου MapReduce.

Οι υπολογιστές ή διεργασίες (κόμβοι) ενός Spark cluster μπορεί να έχουν έναν ή παραπάνω από τους παρακάτω ρόλους:

HDFS DataNode:	Οι DataNodes περιέχουν κομμάτια (blocks) από τα αρχεία του HDFS. Αναλαμβάνουν να «σερβίρουν» δεδομένα σε κλήσεις εξωτερικών πελατών.
HDFS NameNode:	Πρόκειται για τον κεντρικό κόμβο του HDFS. Ο NameNode περιέχει την πληροφορία με την αντιστοίχιση των blocks των αρχείων με τους αντίστοιχους DataNodes (δηλαδή σε ποιον DataNode βρίσκεται κάθε block).
Spark Master node:	Ο master είναι ο κεντρικός κόμβος του Spark που μέσω του ενσωματωμένου cluster manager ελέγχει τους διαθέσιμους πόρους και με βάση αυτούς δρομολογεί και διαχειρίζεται τις κατανεμημένες εφαρμογές που τρέχουν στο cluster.
Spark Worker node:	Ο worker είναι μια διεργασία του Spark, η οποία τρέχει σε κάθε κόμβο του cluster και διαχειρίζεται τις προς εκτέλεση διεργασίες

	των κατανεμημένων εφαρμογών στον κόμβο αυτό.

1.1

Προαιρετικό: ενημέρωση του αρχείου hosts του μηχανήματός μας.

Με αυτό το βήμα ενημερώνουμε το σύστημά μας για το localhost και τον master, μιας και το hadoop όπως και το spark χρειάζονται να κάνουν ssh μεταξύ των κόμβων.

Το αρχείο hosts υπάρχει σε κάθε υπολογιστή, και στην ουσία αποτελεί μια τοπική βάση DNS την οποία συμβουλεύεται πρώτη κάθε φορά που προσπαθεί να κάνει resolve ένα όνομα σε διεύθυνση IP.

```
/etc/hosts
```

Ανοίγουμε το αρχείο hosts σαν root χρήστης (ή με χρήση sudo) και συμπληρώνουμε στο τέλος:

```
127.0.0.1       localhost
127.0.0.1       master
```

1.2 Ρύθμιση για passwordless ssh

Το Spark αλλά και το HDFS για να λειτουργήσουν θα πρέπει οι υπολογιστές που αποτελούν το Spark ή το HDFS cluster να μπορούν να συνδέονται μεταξύ τους με την χρήση ssh χωρίς κωδικό. Αυτό επιτυγχάνεται με την χρήση ζεύγους ιδιωτικού/δημόσιου κλειδιού[1]. Η βασική ιδέα της τεχνικής αυτής είναι ότι ο κάθε χρήστης μπορεί να έχει ένα ιδιωτικό κλειδί (αρχείο) το οποίο έρχεται ζεύγος με ένα δημόσιο κλειδί. Ο χρήστης τοποθετεί το δημόσιο κλειδί στους υπολογιστές τους οποίους θέλει να έχει πρόσβαση, και με την χρήση του ιδιωτικού μπορεί να συνδέεται σε αυτούς χωρίς κωδικό. Η πιστοποίηση του χρήστη γίνεται καθώς μόνο ο χρήστης έχει στην κατοχή του το ιδιωτικό κλειδί. Στο μηχάνημα master τρέχουμε

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

Η εντολή δημιουργεί ένα ιδιωτικό κλειδί id_rsa και ένα δημόσιο κλειδί id_rsa.pub. Στη συνέχεια τρέχουμε

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

η παραπάνω εντολή βάζει το δημόσιο κλειδί id_rsa.pub στον κατάλογο με τα αποδεκτά δημόσια κλειδιά του χρήστη user του master μηχανήματος.

---

[1] http://en.wikipedia.org/wiki/Public-key_cryptography

1.3 Εγκατάσταση Java 8

Για να εγκαταστήσουμε την OpenJDK Java 8, πρέπει πρώτα να ενημερώσουμε τα apt repositories των πακέτων.

```
sudo apt-get update
sudo apt-get install openjdk-8-jdk
```

Μόλις ενημερώσουμε τα repositories κι εγκαταστήσουμε την java (πατάμε yes σε ότι ζητηθεί), ελέγχουμε αν η εγκατάσταση ολοκληρώθηκε με επιτυχία.

```
kostas@kostas-HP-ProBook-450-G2:~$ java –version
openjdk version "1.8.0_222"
OpenJDK Runtime Environment (build 1.8.0_222-8u222-b10-
1ubuntu1~16.04.1-b10)
OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
```

Η παραπάνω διαδικασία πρέπει να ακολουθηθεί και για το slave μηχάνημα.

1.4 Εγκατάσταση HDFS

Τα παρακάτω βήματα πρέπει να γίνουν σε όλους τους υπολογιστές του cluster.

Κατεβάζουμε το installation package από ένα repository και το αποσυμπιέζουμε στο home folder μας.

```
cd ~
wget http://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-
2.7.7.tar.gz
tar -xzf hadoop-2.7.7.tar.gz
```

Κάνουμε edit το ~/.bashrc ώστε να κάνουμε export τις παρακάτω μεταβλητές περιβάλλοντος. Προσθέτουμε τις παρακάτω γραμμές στο τέλος του bashrc.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_INSTALL=/home/kostas/workspace/hadoop/hadoop-2.7.7
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export HADOOP_CONF_DIR=$HADOOP_INSTALL/etc/hadoop
```

Το configuration του Hadoop βρίσκεται στο path: $HADOOP_INSTALL/etc/hadoop. Στον κατάλογο αυτό θα ρυθμίσουμε τα εξής αρχεία:

- core-site.xml
- hadoop-env.sh
- hadoop-env.sh

Περισσότερες λεπτομέρειες για τα αρχεία παραμετροποίησης μπορούν να βρεθούν στη διεύθυνση: http://hadoop.apache.org/docs/r2.7.7/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html

Κάνουμε edit τα αρχεία(vim <file-name>) και προσθέτουμε τα εξής configuration properties:

core-site.xml

```
…
<!-- Put site-specific property overrides in this file. -->
<configuration>
      <property>
            <name>fs.defaultFS</name>
            <value>hdfs://localhost:9000</value>
      </property>
</configuration>
```

hdfs-site.xml

```
…
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

hadoop-env.sh: τροποποιούμε το JAVA_HOME

```
…
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 (ή όποιο άλλο
path, έχουμε εγκατεστημένη τη Java)
…
```

1.5 Εγκατάσταση Spark

Τα παρακάτω βήματα πρέπει να γίνουν σε όλους τους υπολογιστές του cluster.

Κατεβάζουμε το installation package από ένα repository και το αποσυμπιέζουμε στο home folder μας.

```
cd ~
wget https://archive.apache.org/dist/spark/spark-2.4.4/spark-2.4.4-
bin-hadoop2.7.tgz
tar -xzf spark-2.4.4-bin-hadoop2.7.tgz
```

Κάνουμε edit το ~/.bashrc ώστε να κάνουμε export τις παρακάτω μεταβλητές περιβάλλοντος. Προσθέτουμε τις παρακάτω γραμμές στο τέλος του bashrc.

```
export SPARK_HOME=/home/kostas/workspace/spark/spark-2.4.4-bin-
hadoop2.7
export PATH=$PATH:$SPARK_HOME/bin
alias start-all.sh='$SPARK_HOME/sbin/start-all.sh'
alias stop-all.sh='$SPARK_HOME/sbin/stop-all.sh'
```

Το configuration του Spark βρίσκεται στο path: $SPARK_HOME/conf. Στον κατάλογο αυτό υπάρχουν κάποια αρχεία με την κατάληξη '.template'. Εμείς θα δημιουργήσουμε (σαν αντίγραφα των αντίστοιχων .template) και στη συνέχεια θα ρυθμίσουμε τα εξής αρχεία:

- spark-env.sh
- spark-defaults.conf: Ρυθμίσεις για την παραμετροποίηση του Spark
- slaves: τα host names των κόμβων που λειτουργούν ως compute slaves

Περισσότερες λεπτομέρειες για τα αρχεία παραμετροποίησης μπορούν να βρεθούν στη διεύθυνση: https://spark.apache.org/docs/2.4.4/spark-standalone.html#installing-spark-standalone-to-a-cluster

Κάνουμε edit τα αρχεία (vim <file-name>) και προσθέτουμε τα εξής configuration properties:

spark-env.sh

```
...
SPARK_WORKER_CORES=2
SPARK_WORKER_MEMORY=1g
export SPARK_MASTER_HOST=127.0.0.1
export SPARK_MASTER_PORT=7077
#export PYSPARK_PYTHON=python3.6 (export it to your system version or Python's path)
```

spark-defaults.conf

```
...
spark.master                     spark://master:7077
spark.submit.deployMode          client
spark.executor.instances         2
spark.executor.cores             1
spark.executor.memory            512m
spark.driver.memory              512m
```

slaves: αφαιρούμε το localhost και βάζουμε

```
localhost
```

1.6 Εκκίνηση του HDFS

Το πρώτο πράγμα που έχουμε να κάνουμε είναι format του HDFS. Δίνουμε την παρακάτω εντολή:

```
ssh user@master
hdfs namenode -format
```

Format κάνουμε την πρώτη φορά που εγκαθιστούμε το HDFS.

TIP: Σε περίπτωση που εμφανιστεί το εξής error:

```
kostas@kostas-HP-ProBook-450-G2:/opt# hdfs namenode -format
```

```
-bash: hdfs: command not found
```

σημαίνει ότι το σύστημα δεν έχει 'δει' ακόμα τις αλλαγές που κάναμε στο .bashrc αρχείο οπότε εκτελούμε

```
source ~/.bashrc
```

Στη συνέχεια ξεκινάμε το HDFS (NameNode + DataNodes) δίνοντας από τον master την εξής εντολή:

```
start-dfs.sh
```

Ελέγχουμε αν όλοι οι εικονικοί nodes εκκίνησαν κανονικά

```
kostas@kostas-HP-ProBook-450-G2:~# jps
5266 SecondaryNameNode
4982 NameNode
7863 Jps
5103 DataNode
```

Το σύστημα έχει ξεκινήσει και στα παρακάτω url μπορεί κανείς να βλέπει την κατάσταση του HDFS:

http://master:50070: Το url αυτό δείχνει την κατάσταση του NameNode, καθώς και τα περιεχόμενα του αποθηκευτικού συστήματος.

1.7 Εκκίνηση Spark

Για να ξεκινήσει στη συνέχεια το Spark (master+workers) εκτελούμε από τον master την εντολή:

```
start-all.sh
```

Παίρνουμε το παρακάτω output:

```
kostas@kostas-HP-ProBook-450-G2:~# /home/user/spark-2.4.4-bin-
hadoop2.7/sbin/start-all.sh
starting org.apache.spark.deploy.master.Master, logging to
/home/user/spark-2.4.4-bin-hadoop2.7/logs/spark-user-
org.apache.spark.deploy.master.Master-1-master.out
slave: starting org.apache.spark.deploy.worker.Worker, logging to
/home/user/spark-2.4.4-bin-hadoop2.7/logs/spark-user-
org.apache.spark.deploy.worker.Worker-1-slave.out
```

Ελέγχουμε να δούμε αν τρέχει το Spark:

```
kostas@kostas-HP-ProBook-450-G2:~# jps
5266 SecondaryNameNode
4982 NameNode
7863 Jps
5103 DataNode
```

Το σύστημα έχει ξεκινήσει και στα παρακάτω url μπορεί κανείς να βλέπει την κατάσταση του Spark node:

http://master:8080: Το url αυτό δείχνει την κατάσταση του cluster, καθώς και την κατάσταση και το history των jobs.

1.8 Χρήσιμες εντολές, αρχεία:

- Όπως είδαμε, με την εντολή jps μπορούμε να δούμε τις java διεργασίες που τρέχουν σ' ένα μηχάνημα. Κάνοντας kill -9 <pid> σκοτώνουμε κάποια διεργασία που δεν αποκρίνεται.
- Ο κατάλογος logs/ μέσα στους φακέλους εγκατάστασης περιέχει τα logfiles των υπηρεσιών του HDFS και του Spark. Είναι χρήσιμος καθώς εκεί μπορούμε να βλέπουμε πιθανά λάθη/προβλήματα που προκύπτουν.
- Στον κατάλογο tmp υπάρχουν τα lock files των προγραμμάτων του HDFS (με κατάληξη .pid). Καλό είναι, εάν έχουμε τερματίσει τον HDFS cluster με βίαιο τρόπο (πχ με kill -9 <pid>), να τα σβήνουμε.

2 Εκτέλεση παραδειγμάτων

Μετά την ολοκλήρωση της εγκατάστασης μπορούμε να τρέξουμε από τον master ένα έτοιμο παράδειγμα του spark υλοποιημένο σε java που υπολογίζει το π.

```
kostas@kostas-HP-ProBook-450-G2:~# spark-submit --class
org.apache.spark.examples.JavaSparkPi /home/user/spark-2.4.4-bin-
hadoop2.7/examples/jars/spark-examples_2.11-2.4.4.jar
…
Pi is roughly 3.1449
…
```

Στο φάκελο $SPARK_HOME/examples υπάρχουν γενικά πολλά παραδείγματα σε python και scala εκτός από java.

Για ετκέλεση python

```
kostas@kostas-HP-ProBook-450-G2:~# spark-submit mycode.py
```

Αν θέλουμε να δώσουμε είσοδο στο πρόγραμμά μας τότε θα πρέπει να ανεβάσουμε το αρχείο εισόδου στο hdfs cluster.

Φτιάχνουμε directories στο hdfs cluster και ύστερα ανεβάζουμε το αρχείο εισόδου

```
kostas@kostas-HP-ProBook-450-G2:~# hdfs dfs -mkdir /user
kostas@kostas-HP-ProBook-450-G2:~# hdfs dfs -mkdir /user/kostas/
kostas@kostas-HP-ProBook-450-G2:~# hdfs dfs -put input.txt
inputonHDFS.txt
```