Installation Guide

Community 1.5.1 release

This document details step-by-step deployment procedures, system and environment requirements to assist Jumbune deployment.

Table of Contents

Introduction

Big Data processing can run into tens or hundreds of machine clusters. Parallel computing on this scale brings a number of programming challenges, including identification of faults across MapReduce logic, discrepancies in working data and analytics logic, and identification of unhealthy nodes.

MapReduce developers need to profile your code, validate input/output data set, and understand MapReduce implementation across Hadoop cluster for debugging purposes. All of these tasks can be extremely labor-intensive, time-consuming, and painful.

Jumbune, the first MapReduce profiler is intended to help MapReduce developers, DevOps, and Hadoop cluster administrators to:

- Analyze cluster-wide Hadoop MapReduce job flow execution
- Profile MapReduce jobs
- Monitor Hadoop clusters
- Validate HDFS data

Jumbune gives freedom to Hadoop administrators and developers to monitor/operate their Hadoop cluster from remote machines by introducing concept of Jumbune Agent.

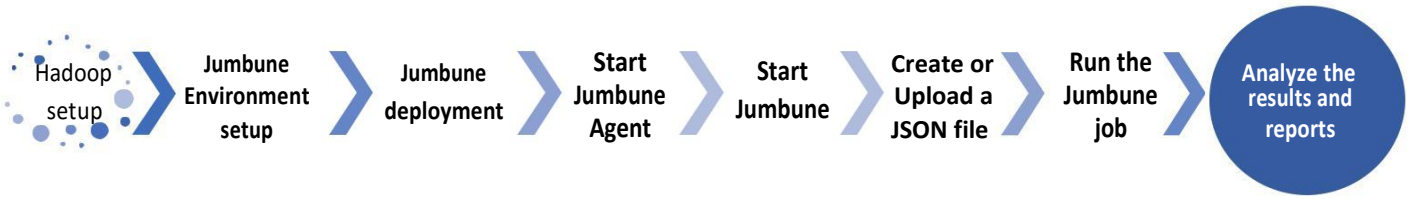
Jumbune System Requirements

- *NIX systems such as Linux
- Sun Hotspot JRE 1.7 or OpenJDK6 +
- At least 4GB RAM
- Firefox, Chrome

Supported Hadoop Distributions

- Apache Hadoop 2.x, 0.23.x, 1.2.x
- CDH 5.x
- MapR 3.x

A brief overview of the steps involved in executing MapReduce job(s) using Jumbune is depicted below:



Environment setup

Jumbune can be used on pseudo-distributed or multi-node Hadoop cluster. It requires an installation directory to be set and the environment variable $**JUMBUNE_HOME** should point to that.

For example

Add the following lines to the etc/environment or the .bashrc file,

```
export JUMBUNE_HOME= <PATH_TO_JUMBUNE_DIR>
```

Non YARN clusters

To set the environment for Jumbune on **Non-YARN** clusters, perform the following steps:

1. Set environment variable **$JAVA_HOME** to point location of the JDK using a logic similar to the one mentioned above.
2. Set environment variable **$HADOOP_HOME** to point the location of the Hadoop distribution.
3. Edit the following attributes of **Hadoop-env.sh** file present in the /conf directory present in the

```
export HADOOP_NAME NODE_OPTS="-Dcom.sun.management.jmxremote.ssl=false
- Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5677 $HADOOP_NAME NODE_OPTS"
export HADOOP_TASKTRACKER_OPTS="-Dcom.sun.management.jmxremote.ssl=false
- Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5678" export HADOOP_DATANODE_OPTS="-
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5679 $HADOOP_DATANODE_OPTS"
export HADOOP_JOBTRACKER_OPTS="-Dcom.sun.management.jmxremote.ssl=false
- Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5680"
```

   Hadoop distribution:

4. Restart Hadoop

YARN clusters

To set the environment for Jumbune on **YARN** clusters, perform the following steps:

1. Set environment variable **$JAVA_HOME** to point location of the JDK using a logic similar to the one mentioned above.
2. Set environment variable **$HADOOP_HOME** to point the location of the Hadoop distribution.
3. Edit the following attributes of **Hadoop-env.sh** file present in the **etc/Hadoop/** directory present in the Hadoop distribution:

```
export HADOOP_NAME NODE_OPTS="-
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5677 $HADOOP_NAME NODE_OPTS"
export HADOOP_DATANODE_OPTS="-Dcom.sun.management.jmxremote.ssl=false
- Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5679 $HADOOP_DATANODE_OPTS"
```

4. Add the following line to the **YARN-env.sh** file present in the same directory.

```
export YARN_NODEMANAGER_OPTS="-Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5678 $YARN_NODEMANAGER_OPTS"

export YARN_RESOURCEMANAGER_OPTS="-Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.port=5680 $YARN_RESOURCEMANAGER_OPTS"
```

5. Restart Hadoop.

Jumbune works with help of an agent on the Name Node, which is light weight and non-obtrusive. Perform the following steps when Jumbune and Hadoop run on *same* machine or on *different* machines:

Picking the correct distribution for your cluster:

There are two types of Jumbune distributions, one each for YARN and non-YARN based clusters. The appropriate version of the distribution can be obtained from the website.

- For Apache 2.x, Apache 0.23.x, CDH5 based clusters – pick YARN distribution of Jumbune
- For Apache 1.x, MapR3.x – pick non-YARN distribution of Jumbune

Unilateral Hadoop and Jumbune Deployment *(Name Node and Jumbune on same machine)*

1. Create a working directory for Jumbune agent and set environment variable **$AGENT_HOME** to point to that.

Add the following lines to the /etc/environment or the .bashrc file,

```
export AGENT_HOME= <PATH_TO_AGENT_DIR>
```

2. Set an environment variable **$HADOOP_HOME** to point the Hadoop distribution directory.

```
export HADOOOP_HOME= <PATH_TO_HADOOP_DIR>
```

3. There are two ways to deploy jumbune from the distribution jar file.

Method I:

Run Jumbune distribution jar file using the following command:

```
java –jar <location of the distribution jar file>
```

To deploy Jumbune in verbose mode use,

```
java –jar <location of the distribution jar file > -verbose
```

In case the distribution is for YARN based Hadoop distributions, following message will appear.

```
--Jumbune built for [YARN based Hadoop] distributions-                    .
```

In case the distribution is for Non YARN based Hadoop distributions, following message will appear.

```
--Jumbune built for [Non-YARN based Hadoop] distributions-                .
```

Picked JAVA_HOME and JUMBUNE_HOME will appear, can be like below,

```
JAVA_HOME linked to: - /usr/lib/java                                     .
JUMBUNE_HOME linked to: - <path to JUMBUNE_HOME directory>               .
```

User will be prompted to choose the Hadoop distribution on which Jumbune will be used.

In case the distribution is for YARN based Hadoop distributions

```
Choose the Hadoop Distribution Type: (a)Apache | (c)Cloudera |(h)HortonWork                    .
```

In case the distribution is for non-YARN based Hadoop distributions

```
Choose the Hadoop Distribution Type: (a)Apache | (m)MapR                                      .
```

Jumbune will prompt for the IP of the Name Node, followed by the username, password and the private key file path to access the remote machine

```
Jumbune needs to calibrate itself according to the installed Hadoop distribution, please provide  .
Details about Hadoop namenode machine                                                              .
                                                                                                   .
IP address of the namenode machine [127.0.0.1]                                                     .
                                                                                                   .
Username of the namenode machine:[user]                                                            .
                                                                                                   .
Do we have passwordless SSH between [127.0.0.1]-[127.0.0.1]machines? (y)/(n)                       .
n                                                                                                  .
Password of the namenode machine:                                                                  .
                                                                                                   .
Please provide private key file path [/home/user/.ssh/id_rsa]                                      .
                                                                                                   .
Extracting Jumbune                                                                                 .
Using Hadoop: [/usr/lib/Hadoop]                                                                    .
Syncing Jars from Hadoop to Jumbune...                                                             .
Done.                                                                                              .
```

A message is displayed to show that Jumbune deployment has been completed successfully.

```
!!! Jumbune successfully deployed at [<JUMBUNE_HOME>] !!!                                        .
```

Method II:

Jumbune provides dynamic argument facility so that user can specify arguments at the deployment time:

```
java –jar <location of the distribution jar file> <command line options>
```

To deploy Jumbune in verbose mode use,

```
java –jar <location of the distribution jar file> <command line options> -verbose
```

Example 1:

```
java -jar <location of the distribution jar file> -DnamenodeIP= 127.0.0.1
```

In the above command, user will not be asked for name node IP later.

Example 2:

```
java -jar <location of the distribution jar file> -Ddistribution=apache -Dpassword=PnT0JaRmVaG -DnamenodeIP=
127.0.0.1 -Dusername=user
```

In the above command, the password is given in the encrypted form. The encrypted password can be generated by using below command:

```
java -jar <location of the distribution jar file> --encryption
```

Example 3:

```
java -jar <location of the distribution jar file> -Ddistribution=apache -Dusername=user -DnamenodeIP=127.0.0.1 -Dprivatekeypath=/home/user/.ssh/id_rsa
```

Password and Private Key file path can be provided alternatively. For instance in the above command, if user has password less ssh then private key file path needs to be provided.

Example 4:

```
java –jar <location of the distribution jar file> -Dpropertyfile=/home/user/myProperties.txt -Dusername=user -Ddistribution=apache
```

Alternatively, a configuration file containing all the above required arguments can also be provided like above.

where myProperties.txt contains

```
privatekeypath=/home/user/.ssh/id_rsa
namenodeIP=127.0.0.1
```

Use below help option to get more information about arguments

```
java –jar <location of the distribution jar file> --help
```

4. Start Jumbune server by running the startWeb script file.

5. Navigate to localhost:8080 (or any other port if you have specific in startWeb script)

Agent Deployment for unilateral deployment

Navigate to **<JUMBUNE_HOME>/agent-distribution** directory and use the following command:

```
java –jar <Jumbune agent jar> <port no>
```

To run agent in verbose mode use,

```
java –jar <Jumbune agent jar> <port no > -verbose
```

User will be prompted to choose the Hadoop distribution

In case the distribution is for YARN based Hadoop distributions, following message will appear.

```
Choose the Hadoop Distribution Type : (a)Apache |(c)Cloudera |(h)HortonWorks                    .
```

In case the distribution is for Non YARN based Hadoop distributions, following message will appear.

```
Choose the Hadoop Distribution Type : (a)Apache | (m)MapR
```

Enter the Hadoop installation directory if it's different from the one prompted

```
Please verify Hadoop installation directory
[/usr/lib/Hadoop]
```

In-case you have separate users for HDFS and YARN answer 'Y' in the prompt else 'N'.

```
Do you have separate users for MapReduce, YARN or HDFS (y)YES / (n) NO
Y
```

If your selection was 'Y' then enter the user name and password details for the respective users

```
Please provide the HDFS user name [HDFS]

Please provide the password for the HDFS user

Please provide the YARN user name [YARN

Please provide the password for the YARN user

Please provide the password for the Mapreduce user[mapred]
```

A message that the agent has started successfully would be displayed

```
Jumbune Agent has started successfully on port [xxxx]
```

Distributed Jumbune Deployment *(Name Node and Jumbune are on different machines)*

In multimode distribution, the Jumbune agent jar should be run on the node with the Name Node daemon running.

1. Create a working directory for Jumbune agent and set environment variable **$AGENT_HOME** to point to that.

   Add the following lines to the /etc/environment or the .bashrc file,

   ```
   export AGENT_HOME= <PATH_TO_AGENT_DIR>
   ```

2. Set an environment variable **$HADOOP_HOME** to point the Hadoop distribution directory.

```
export HADOOOP_HOME= <PATH_TO_HADOOP_DIR>
```

3. Run Jumbune distribution jar file using the following command:

```
java –jar <location of the distribution jar file>
```

To deploy Jumbune in verbose mode use,

```
java –jar <location of the distribution jar file > -verbose
```

In case the distribution is for YARN based Hadoop distributions, following message will appear.

```
--Jumbune built for [YARN based Hadoop] distributions—-
```

In case the distribution is for Non YARN based Hadoop distributions, following message will appear.

```
--Jumbune built for [Non-YARN based Hadoop] distributions--
```

Picked JAVA_HOME and JUMBUNE_HOME will appear, can be like below,

```
JAVA_HOME linked to: - /usr/lib/java
JUMBUNE_HOME linked to: - <path to JUMBUNE_HOME directory
```

User will be prompted to choose the Hadoop distribution on which Jumbune will be used.

In case the distribution is for YARN based Hadoop distributions

```
Choose the Hadoop Distribution Type: (a)Apache | (c)Cloudera |(h)HortonWorks
```

In case the distribution is for non-YARN based Hadoop distributions

```
Choose the Hadoop Distribution Type: (a)Apache | (m)MapR
```

4. Jumbune will prompt for the IP of the Name Node, followed by the username, password and the private key file path to access the remote machine

```
Jumbune needs to calibrate itself according to the installed Hadoop distribution, please
provide details about Hadoop namenode machine

IP address of the namenode machine [127.0.0.1]
192.168.10.2
Username of the namenode machine:[user]

Do we have passwordless SSH between [192.168.10.1]-[192.168.10.2] machines? (y)/(n)
n
Password of the namenode machine:

Please provide private key file path [/home/user/.ssh/id_rsa]

Extracting Jumbune...
Using Hadoop: [/usr/lib/Hadoop]
```

```
Syncing Jars from Hadoop to Jumbune...                                          .
Done.                                                                           .
```

5. A message is displayed to show that Jumbune deployment has been completed successfully.

```
!!! Jumbune successfully deployed at [<JUMBUNE_HOME>] !!!                       .
```

6. Start Jumbune server by running the startWeb script file on the Jumbune machine
7. Navigate to <ip of Jumbune machine>:8080 (or any other port if you have specific in startWeb script)

Agent deployment for distributed deployment

The Jumbune Agent jar file should be run on the machine with the **Name Node** daemon, the agent jar can be run from any directory on the machine. Use the following command to run the agent on the Name Node

```
java –jar <Jumbune agent jar> <port no>
```

User will be prompted to choose the Hadoop distribution

In case the distribution is for YARN based Hadoop distributions, following message will appear.

```
Choose the Hadoop Distribution Type : (a)Apache |(c)Cloudera |(h)HortonWorks
```

In case the distribution is for Non YARN based Hadoop distributions, following message will appear.

```
Choose the Hadoop Distribution Type : (a)Apache | (m)MapR
```

Enter the Hadoop installation directory if it's different from the one prompted

```
Please verify Hadoop installation directory
[/usr/lib/hadoop]
```

In-case you have separate users for HDFS and YARN answer 'Y' in the prompt else 'N'.

```
Do you have separate users for MapReduce, YARN or HDSFS (y)YES / (n) NO
Y
```

If your selection was 'Y' then enter the user name and password details for the respective users

```
Please provide the HDFS user name [HDFS]

Please provide the password for the HDFS user

Please provide the YARN user name [YARN]

Please provide the password for the YARN user

Please provide the password for the Mapreduce user[mapred]
```

A message that the agent has started successfully would be displayed

```
Jumbune Agent has started successfully on port [xxxx]
```

For CDH5.x and MapR 3.x based clusters, two specific jar files - *log4j-api-2.1.jar* & *log4j-core-2.1.jar* should be copied from the AGENT_HOME/lib directory to the lib directories of the respective Hadoop installations. Adequate access permissions (r) to the jar files should be provided to the user who is going to execute the job.

CDH5.x user can use the following commands.

```
cp $AGENT_HOME/lib/log4j-api-2.1.jar /usr/lib/hadoop-yarn/lib/
cp $AGENT_HOME/lib/log4j-core-2.1.jar /usr/lib/hadoop-yarn/lib/

sudo chmod o+r /usr/lib/hadoop-yarn/lib/log4j-api-2.1.jar
sudo chmod o+r /usr/lib/hadoop-yarn/lib/log4j-core-2.1.jar
```

MapR 3.x users can use the following commands.

```
cp $AGENT_HOME/lib/log4j-api-2.1.jar /opt/mapr/hadoop/hadoop-0.20.2/lib
cp $AGENT_HOME/lib/log4j-core-2.1.jar /opt/mapr/hadoop/hadoop-0.20.2/lib

sudo chmod o+r /opt/mapr/hadoop/hadoop-0.20.2/lib/log4j-api-2.1.jar
sudo chmod o+r /opt/mapr/hadoop/hadoop-0.20.2/lib/log4j-core-2.1.jar
```