



UNIVERSIDAD
**PABLO DE
OLAVIDE**
S E V I L L A



**DATA SCIENCE
& BIG DATA**
RESEARCH LAB
PABLO DE OLAVIDE UNIVERSITY

LadonSpark

Manual

Antonio M. Fernández-Gómez,
David Gutiérrez-Avilés, Alicia Troncoso,
Francisco Martínez-Álvarez

Version 3.0, 24/03/2020

Revision History

Revision	Date	Author(s)	Description
1.0	20/06/2016	AMFG	First version of the document
1.5	04/04/2017	AMFG	Minor revision of the document
2.0	05/05/2018	DGA	Mayor revision of the document
2.5	10/03/2020	AMFG	Mayor revision of the document
3.0	24/03/2020	DGA	Mayor revision of the document

Contents

1	Introduction	3
2	Obtaining the resources	3
3	Pre-requisites	4
4	Installation guide	5
4.1	MYSQL Configuration	5
4.2	Master and Hadoop configuration	5
4.3	Slaves and Hadoop configuration	6
4.4	Starting Hadoop process	6
4.5	Tomcat deployment	6
5	User guide	7
5.1	Home page	7
5.2	Cluster settings	7
5.2.1	All hosts in the network	8
5.2.2	Selected hosts	9
5.3	Algorithm	9
5.3.1	Add new Algorithm	10
5.3.2	Launch algorithm	11
6	Contact	12

List of Tables

List of Figures

1	Home page	7
2	Cluster page.	7
3	Cluster page.	8
4	All host in the network page.	8
5	Hosts Selected page.	9
6	Algorithm page.	10
7	Add algorithm page.	10

1 Introduction

The LadonSpark tool offers an open-source and non-commercial solution to automatically configure and deploy a Spark cluster. Besides, the main advantage that a potential user acquires when he installs this system is to avoid the necessity to collaborate with an administrator role. Therefore, any user that have several machines connected by a network can configure and deploy a Spark cluster in a user-friendly, and free of charge way without any system administrator capabilities. Note that this fact means a great advantage, for instance, for small-medium data science research groups, as well as another more kind of users. The application has also been designed to easily integrate new algorithms by just uploading executable files and configuring the inputs. As a sample usage, the tool incorporates some algorithms of the machine learning library (MLlib) of Spark, in particular, Kmeans (clustering), Generalized linear models (regression), and FP-Growth (pattern extraction).

2 Obtaining the resources

The resources of the LadonSpark tool are available in GitHub at <https://github.com/datascienceresearchlab/LadonSpark>

3 Pre-requisites

This section describes the prerequisites needed for the proper functioning of the proposed approach. In particular, the minimal prerequisites for the cluster launching can be summarized as follows:

1. Shared dataset. The dataset to be processed by an algorithm has to be shared for all nodes of the cluster. Currently, there are two different ways to share it:
 - (a) HDFS System. This system distributes a dataset in all nodes of the cluster. The LadonSpark application integrates the HDFS, which can be started up using a script that has been developed to install it across the cluster easily.
 - (b) File repository. The dataset is replicated in every node at a specific folder. That way, Spark can access to the required specific data blocks in every node. This option reduces the computational time, but it requires much space in memory for each node.
2. RSA ring. RSA keys are necessary for the exchange of information between nodes without having to enter credentials for each connection.
3. Global user. It is necessary to facilitate the RSA ring. Hence, access to the path of the files is greatly simplified through the same user and password for all nodes.
4. Nmap. This is a critical prerequisite because this application sniffs the network and creates the nodes list. Nmap must be installed in the master node, enabling it to discover new potential nodes to be part of the cluster.
5. Spark package. This package must be downloaded and unzipped in the specific path `/home/username`.
6. Scala package. As happens with the Spark package, the Scala package must be downloaded and unzipped for the proper execution of an algorithm, which has been developed in the Scala programming language.

Finally, two new libraries have been included in the last update of the LadonSpark, and therefore, their installation will be required to execute an algorithm in both R or Python languages supported by Spark.

1. R-base. This library allows executing R code from Spark. This language has been included because it is one of the most used languages for data analysis currently.
2. Python. This language is a pervasive and popular programming language nowadays. For that reason, this library has been included in developing algorithms using Python from Spark.

4 Installation guide

This section defines the process of configuring and installing the tools required for Ladon Spark. To begin with, we have to configure the IP of each node to be fixed so that we know for sure that IP has each of the nodes. We must also have a common username and password for each node. To complete the first part of the installation, download the Hadoop and spark fonts and place them in a path.

You can download the sources from the following website:

- Apache Hadoop: <https://archive.apache.org/dist/hadoop/common/hadoop-2.7.6/hadoop-2.7.6.tar.gz>
- Apache Spark: <https://archive.apache.org/dist/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>

To help in this part you can rely on the following commands:

- Apache Spark: Assuming that the downloads have been made in the Downloads folder.
 1. `tar -xvf /home/$USER/Downloads/spark-2.3.1-bin-hadoop2.7.tgz`
 2. `mv /home/$USER/Downloads/spark-2.3.1-bin-hadoop2.7 /home/$USER/spark`
- Apache Hadoop: Assuming that the downloads have been made in the Downloads folder.
 1. `tar -xvf /home/$USER/Downloads/hadoop-2.7.6.tar.gz`
 2. `sudo mv /home/$USER/Downloads/hadoop-2.7.6 /opt/hadoop`

4.1 MYSQL Configuration

The configuration of MYSQL is done through the file `ladonspark.sql`, which generates the user and database necessary for the operation of the system. Depending on the installation mode, we will launch the SQL files using the command line interface or the PHPMYADMIN management web.

E.g.: `mysql -uroot -p j ladonspark.sql`

4.2 Master and Hadoop configuration

In the package folder, the `master.sh` script can be found. The `master.sh` installs the source package necessary to correct work. The next command should write in the terminal.

1. `chmod +x master.sh`
2. `./master.sh`

You can also find the script `Hadoop.Setting.sh`. With this script, we configure the Hadoop system to create an HDFS system with which to share the data sets. In the same way, you need the master's IP address for the correct one installation and the version of java, please note that only we need the version number of java, we will use the following command.

1. `chmod +x Hadoop_Setting.sh`
2. `./Hadoop_Setting.sh 192.168.10 8`

4.3 Slaves and Hadoop configuration

The slave nodes are all nodes to use in a cluster that is different from the master node. The script file should be launch is `slave.sh`. This script configures the necessary folders for the correct functioning of the application by the slave node. You need a parameter that is the master's IP address.

1. `chmod +x slave.sh`
2. `./slave.sh 192.168.1.10`

We can also need to configure Hadoop in the slaves. With the `Hadoop_Setting.sh`, we configure the Hadoop system to create an HDFS system with which to share the data sets. In the same way, you need the master's IP address for the correct one installation and the version of java, please note that only we need the version number of Java, we will use the following command.

1. `chmod +x Hadoop_Setting.sh`
2. `Hadoop_Setting.sh 192.168.156 8`

4.4 Starting Hadoop process

To complete the installation process, we must run the last script that deploys the Hadoop services and configures the HDFS services.

Hadoop.setting2.sh: This script is in charge of configuring the HDFS to support the file transfer, just run, and you're done.

1. `chmod +x Hadoop.setting2.sh`
2. `./Hadoop.setting2.sh`

In addition to the HDFS services, it is also included to facilitate the upload of data sets to the HDFS system.

We must copy the `upfile.sh` script in the folder where the datasets are located.

upfile.sh: A script in charge of uploading files to HDFS, for them we pass as parameter the file name along with the path.

1. `chmod +x upfile.sh`
2. `./upfile.sh /home/$USER/dataset/dataset.csv`

4.5 Tomcat deployment

As LadonSpark is a web application we need to deploy it in an application server. To do that, we need to install Tomcat 8. Inside the resources, we can find a script that automatize the LadonSpark deployment.

1. `chmod +x tomcat_deployment.sh`
2. `./tomcat_deployment.sh username password ladonSpark.Path`

5 User guide

In this section, we describe the functionalities of LadonSpark. We assume that the installation manual has been completed, and Ladon Spark is working correctly; otherwise, go to Section 4 and complete the installation.

5.1 Home page

This is the main page of the application where information about the website and the developer is displayed.

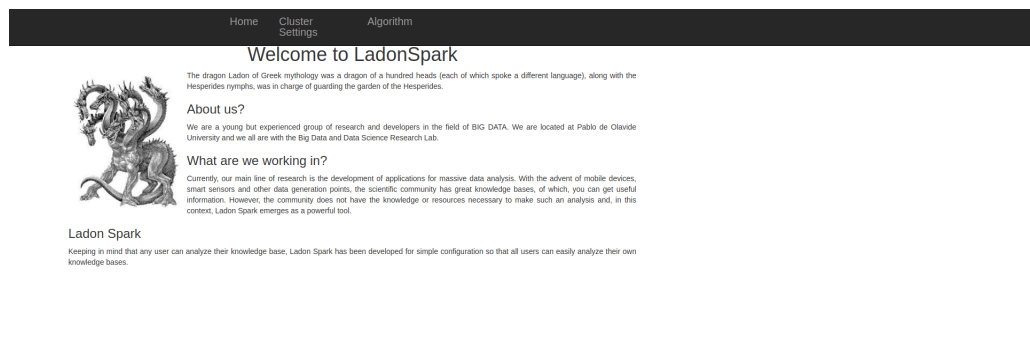


Figure 1: Home page

5.2 Cluster settings

To access the communications section, we can do so from the menu by clicking on the communications option. In this section, I would start with the cluster management functionality.

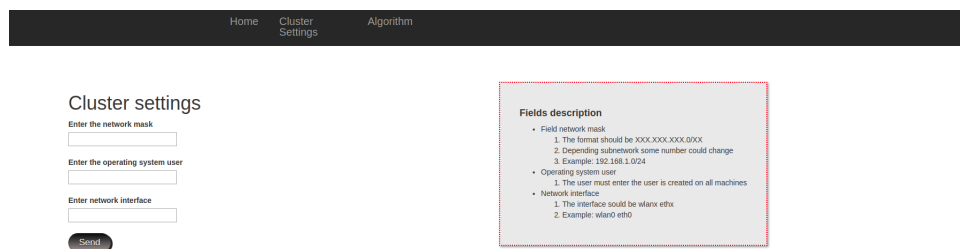


Figure 2: Cluster page.

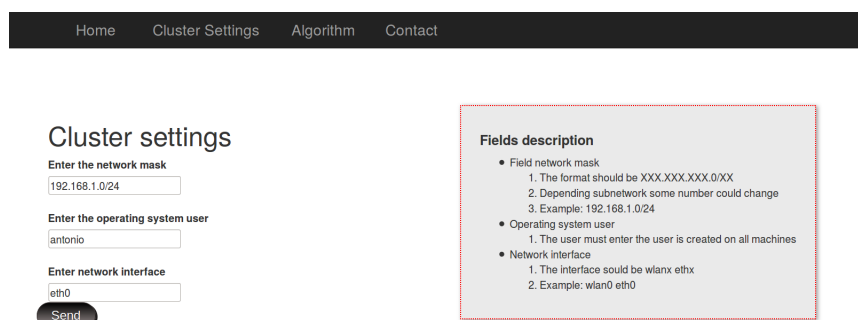
As we can see in Figure 2. On the right, we find a text box with a description of each field. On the left, the form is defined below.

- The network mask. In this field, we must introduce the network in the one that will find the nodes that will form the cluster. An example:

– 192.168.1.0/24

– 193.147.187.63/26

- System user. We enter the user name they must have in common all nodes of the system.
- Network interface. The name of the network interface with which we connect to the network. An example:
 - wlan0
 - eth0
 - eno1



Cluster settings

Enter the network mask
192.168.1.0/24

Enter the operating system user
antonio

Enter network interface
eth0

Send

Fields description

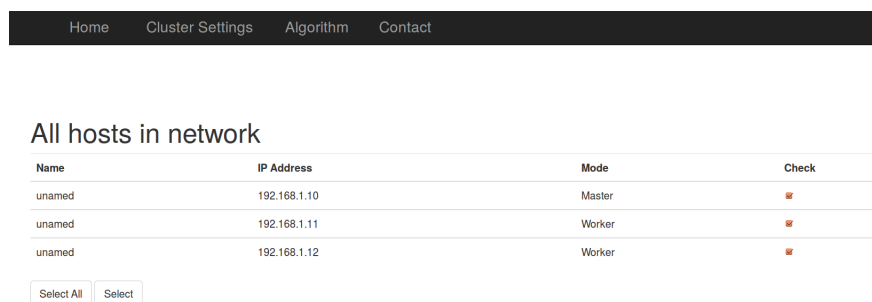
- Field network mask
 1. The format should be XXX.XXX.XXX.0/XX
 2. Depending subnetwork some number could change
 3. Example: 192.168.1.0/24
- Operating system user
 1. The user must enter the user is created on all machines
- Network interface
 1. The interface could be wlanx ethx
 2. Example: wlan0 eth0

Figure 3: Cluster page.

When you have entered all the data in the fields, as shown in Figure 3, click on the Send button.

5.2.1 All hosts in the network

In the user interface, all the nodes that exist in the network are displayed; you should check that the master node appears.



Name	IP Address	Mode	Check
unamed	192.168.1.10	Master	<input checked="" type="checkbox"/>
unamed	192.168.1.11	Worker	<input checked="" type="checkbox"/>
unamed	192.168.1.12	Worker	<input checked="" type="checkbox"/>

Select All Select

Figure 4: All host in the network page.

In Figure 4, an example is shown with the network 192.168.1.0.0/24, where we can select the nodes that want to participate in a cluster through a checkbox. We can also click on the Select all button, which marks all the nodes. Once you have selected the nodes you want, click on the Select button.

5.2.2 Selected hosts

This section is the last step before the cluster is deployed, in which we can see the hosts selected in the previous step, and a checkbox, which displays a small form that we describe below.

The screenshot shows the 'Selected Hosts' page. At the top is a navigation bar with links: Home, Cluster Settings, Algorithm, and Contact. Below the navigation bar is the title 'Selected Hosts'. Underneath is a table with three columns: Name, IP Address, and Mode. The table contains three rows: one Master node and two Worker nodes. Below the table is a section for 'Advanced configuration' with three input fields: 'SPARK worker cores' (set to 4), 'SPARK worker memory' (set to 2048GB), and 'SPARK worker instances' (set to 2). At the bottom of this section are 'Select' and 'Cancel' buttons.

Name	IP Address	Mode
unnamed	192.168.1.10	Master
unnamed	192.168.1.11	Worker
unnamed	192.168.1.12	Worker

☒ Advanced configuration

SPARK worker cores
4

SPARK worker memory
2048GB

SPARK worker instances
2

Select Cancel

Figure 5: Hosts Selected page.

This form is the parametrization of the Spark configuration file; the form is composed of the three primary parameters of Spark.

- **Cores.** Number of cores that we will use for each node.
- **RAM memory.** 1024 MB.
- **Instance number.** 1.

If the "Advanced configuration" option is not selected, the cluster has the default Spark configuration:

- **Cores.** 1.
- **RAM memory.** RAM memory to be used per node.
- **Instance number.** Number of instances that will be launched for each node.

Once this process is completed, you are redirected to the algorithms section. The menu undergoes a small modification: a button is added to turn off the cluster.

5.3 Algorithm

This section is composed of two parts, but first, we comment on the main interface of the algorithms section. In Figure 6, we can see the division of the interface into two parts, the left part is used to launch an algorithm, and the right part leads us to the section of adding an algorithm.

Figure 6: Algorithm page.

5.3.1 Add new Algorithm

In this subsection, we can add algorithms to the system using a form, which appears on the left side of the interface, and on the right side, we have a small description of the fields of the form.

Figure 7: Add algorithm page.

As shown in Figure 7, the form for adding an algorithm is simple.

- **Name of the Algorithm.** This field is for naming our algorithm.
- **Main Class.** We must indicate which is the main class of the algorithm.
- **Select binary file.** This field is for including the "jar" file with the source packages of the algorithm.
- **Number of parameters.** When we write the number of parameters, we will create the "Insert name and type" fields, since here we indicate the number of parameters that has our algorithm.
- **Name and type.** As its own name indicates, we must enter the name of the parameter and the type of data it uses.

5.3.2 Launch algorithm

First, we must select an algorithm from the list to auto-generate the fields. As shown in Figure 6, left section.

- Field select. Algorithms that we have stored in the system, which generate the parametrization fields of the system.
- Dataset path. Name of the dataSet containing the data we are going to work with.

Finally, click on the Select button to execute the algorithm. When the algorithm is finished, the system redirects to the last interface where the results are displayed.

6 Contact



UNIVERSIDAD
PABLO DE OLAVIDE
SEVILLA



**DATA SCIENCE
& BIG DATA**
RESEARCH LAB
PABLO DE OLAVIDE UNIVERSITY

Carretera de Utrera km. 1 - 41013 Sevilla - España
<http://datalab.upo.es>

Antonio M. Fernández-Gómez
Email: amfergom@alu.upo.es
Web: datalab.upo.es/fernandez

David Gutiérrez-Avilés
Email: dgutavi@upo.es
Web: datalab.upo.es/gutierrez

Alicia Troncoso
Email: atolor@upo.es
Web: datalab.upo.es/troncoso

Francisco Martínez-Álvarez
Email: fmaralv@upo.es
Web: datalab.upo.es/martinez