# The environment preparation before using iServer built-in Spark

After copy the SuperMap iServer package to the cluster member machine Then you can configure the environment before useing. **If you are using the iServer package on Windows, it only needs to open the port.**

## Recommended hardware requirements

The recommended hardware configuration for the iServer distributed analysis service includes:

- Processor: 8-cores
- RAM: 16 GB or higher
- Network Adapter: 1 Gb

## Open Port

In order to work properly of the distributed cluster, Windows and Linux system need to modify the firewall configuration, and open the following port:

1. iServer default port
2. The default port used in Spark distributed cluster system:
    - 8080: Access the default port of Web UI of Master node
    - 7077: Start the default port of Master node
    - 22: The default communication port of SSH

## Linux Package


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

If you are using Linux package, except the needed port, you need to modify the Linux host name. Here it takes three Ubuntu virtual machines as an example. The IP is

Master：192.168.177.136

Worker1： 192.168.177.135

Worker2：192.168.177.137

The following processes should be performed as root. You can switch the general user to root user through the following command:

```
sudo -i
```

Input the current user password.

## Modify Machine Host Name

Open the hostname configuration file:

```
vi /etc/hostname
```

Modify the host name to sparkmaster. The host name in Worker node is modified to sparkworker1 and sparkworker2.

After save and ecit the editing, view the IP address through the following command:

```
ifconfig
```

Set the corresponding relationship between the host name and IP. Open the /etc/hosts file:

```
vi /etc/hosts
```

Add the following configuration:

```
192.168.177.136        sparkmaster
192.168.177.135        sparkworker1
192.168.177.137        sparkworker2
```

The hosts files in three nodes are the same. Save and exit the editing. Start the virtual machine again.

You can view whether the current host name is modified successfully throgh the following command:

```
hostname
```

View whether the relationship is correct between the host name and IP:

```
ping [hostname]
```

Possible use case:

```
ping sparkmaster
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
root@sparkMaster:/# ping sparkmaster
PING sparkMaster (192.168.177.136) 56(84) bytes of data.
64 bytes from sparkMaster (192.168.177.136): icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=3 ttl=64 time=0.145 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=6 ttl=64 time=0.030 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=7 ttl=64 time=0.072 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=8 ttl=64 time=0.089 ms
64 bytes from sparkMaster (192.168.177.136): icmp_seq=9 ttl=64 time=0.074 ms
```

## war Package

Please confirm whether the environment configuration of war package is correct. After the war deploy is success, you also need to configure the environment for distributed analysis services by performing the following operations on each platform:

### Unzip support _win_x64.zip or support _linux_x64.tar.gz

1. Unzip the support decompression file and you can see the hadoop and spark folders.

2. Create a new support in the folder higher lever of iserver. Put the hadoop and spark folders in it.

### Set Environment Variables

1. Add the HADOOP_HOME to the environment variable. It can be set to iServer built-in hadoop directory. And add the %HADOOP_HOME%\bin to the system PATH. (there is no need to set the hadoop environment variable on Linux)

2. Add the SPARK_HOME to the environment variable. It can be set to iServer built-in spark directory. And add the %SPARK_HOME%\bin to the system PATH.

3. Add the SPARK_SCALA_VERSION to the environment variable. The valus is 2.11.8.

# Building Spark distributed cluster



Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

SuperMap iServer provides a built-in Spark installation package. You also can build Spark distributed cluster on other computer. Note: The version of Spark should be consistent with the one in iServer, and the version of Hadoop should be a corresponding version. **Currently the Spark version in iServer is spark-2.1.0-bin-hadoop2.7, so the Hadoop version should be 2.7.x.** It will introduce how to build Spark cluster on a Linux computer as follows.

The sample will use VMware Workstation 11.0.0 build-2305329, and create 3 Ubuntu 14.04 VM with the same configuration, one of them is the Master node, and the others are Worker nodes. The three VMs are named as Master, Worker1, Worker2 (the names are the VM names, not the host name of the system in the VM), and their IPs are Master: 192.168.177.136, Worker1: 192.168.177.135, Worker2: 192.168.177.137.

# Software requirements

It needs Java environment (the JDK download like is http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html#javasejdk, it's suggested to use JDK 8 or above), and to configure SSH and Spark (the download link is: http://spark.apache.org/downloads.html).

Here it uses:

- jdk-8u111-linux-x64.tar.gz
- spark-2.1.0-bin-hadoop2.7.tgz
- openssh-server

# Steps

The following will take the Master node as an example, which is the same with the Worker node.

All operations should use root user. You can use the following command to convert a common user to the root user:

```
sudo -i
```

Input the current user password again.

## The needed ports

It needs the following ports:


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

- 8080: the default port to access Web UI of the Master node
- 7077: the default port to start the Master node
- 22: the default communication port of SSH

Check whether the port is available as follows:

```
ufw status
```

Enable the port as follows:

```
ufw allow [port]
```

[port] is the open port number. E.g.:

```
ufw allow 8080
```

## Configure the Java environment

Here it saves the downloaded JDK package in /home/supermap/

Extract the JDK package

```
tar -xvf   jdk-8u111-linux-x64.tar.gz
```

Create java folder in /usr/lib/:

```
mkdir /usr/lib/java
```

Copy or move the extracted JDK files to /usr/lib/java, e.g.:

```
cp -r /home/supermap/jdk-8u111-linux-x64/ /usr/lib/java
```

Open the environment variable configuration file:

```
vi ~/.bashrc
```

Press i to edit. Write the following codes in the end of the configuration file:

```
export JAVA_HOME=/usr/lib/java/jdk1.8.0_111
export CLASSPATH=.:${JAVA_HOME}/lib:${JAVA_HOME}/jre/lib
export PATH=${JAVA_HOME}/bin:${JAVA_HOME}/jre/bin:$PATH
```

Press esc to exit the editing mode, input :wq to save an exit the editing.

Activate the new configuration with the following command:

```
source ~/.bashrc
```

Check whether Java is installed successfully:

```
java -version
```

If it shows the installed Java" version information, it means it is successful.

## Configure SSH

Spark uses SSH for communication. Input the following command to install SSH when the external Internet is connected.



Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
apt-get install open-ssh
```

When it uses internal network, it needs a file transfer tool such as XManager to transfer the downloaded .deb package to the VM to install, and the installation command is as follows:

```
dpkg -i /home/supermap/openssh-server_6.6p1-2ubuntu1_amd64.deb
```

Check whether SSH is installed successfully:

```
ps -e|grep ssh
```

If it shows the following information it means it is successful

```
root@sparkMaster:/# ps -e|grep ssh
 1148 ?        00:00:00 sshd
 2141 ?        00:00:00 ssh-agent
```

It can set SSH non-code validation to realize the communication without passwords.

Generate a private keygen, where '' means two ':

```
ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
```

Then it will generate two files: id_dsa and id_dsa.pub in /root/.ssh, where id_dsa is private, id_dsa.pub is public.

It needs to append id_dsa.pub to authorized_keys which is used to save all public keygen contents to allow accessing ssh client with current user role:

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

Check whether it can access SSH without password:

```
ssh localhost
```

Input yes to log in. After completed, input exit to exit localhost. It doesn't need a password for the next log-in.

Configure in the Worker node with the same steps.

To realize the communication without password between the nodes, it needs to copy the public key in the Worker node to Master, then append them to authorized_keys in Master.

In Worker1, copy the public key to Master:

```
scp   id_dsa.pub root@192.168.177.136:/root/.ssh/id_dsa.pub.Worker1
```

In Worker2, copy the public key to Master:

```
scp   id_dsa.pub root@192.168.177.136:/root/.ssh/id_dsa.pub.Worker2
```

Append the public key to authorized_keys in Master:

```
cat ~/.ssh/id_dsa.pub.Worker1 >> ~/.ssh/authorized_keys
~/.ssh/id_dsa.pub.Worker2 >> ~/.ssh/authorized_keys
```

In Master copy authorized_keys to two Worker nodes:

```
scp ~/.ssh/authorized_keys root@192.168.177.135:/root/.ssh/authorized_keys
```



Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
scp ~/.ssh/authorized_keys root@192.168.177.136:/root/.ssh/authorized_keys
```

Then it will not need a password between the Master and the Worker node.

## Install Spark

Download Spark and save it in /home/supermap/

Extract the Spark package:

```
tar -xvf spark-2.1.0-bin-hadoop2.7.tgz
```

Create Spark folder in /usr/local/:

```
mkdir /usr/local/spark
```

Copy the extracted spark-2.1.0-bin-hadoop2.7 to /usr/local/spark:

```
cp -r /home/supermap/spark-2.1.0-bin-hadoop2.7.tgz/ /usr/local/spark
```

**Configure Spark environment variable**

Open the environment variable configuration file with the following command:

```
vi ~/.bashrc
```

Press i to edit. Write the following codes in the end of the configuration file:

```
export SPARK_HOME=/usr/local/spark/spark-2.1.0-bin-hadoop2.7
```

Add the bin directory to PATH, then it will be like:

```
export
PATH=${JAVA_HOME}/bin:${JAVA_HOME}/jre/bin:${SPARK_HOME}/
bin:$PATH
```

Press esc to exit the editing mode, input :wq to save an exit the editing.

Activate the new configuration with the following command:

```
source ~/.bashrc
```

**Configure the Master node for Spark**

Go to the Spark configuration file directory:

```
cd /usr/local/spark/spark-2.1.0-bin-hadoop2.7/conf
```

There is no spark-env.sh file in current directory, it needs to modify spark-env.sh.template to spark-env.sh. Copy spark-env.sh.template to spark-env.sh with the following command:

```
cp spark-env.sh.template spark-env.sh
```

Open spark-env.sh

```
vi spark-env.sh
```

Press i to edit, add the following codes in the end of the file

```
if [ -z "${SPARK_HOME}" ]; then
  export SPARK_HOME="$(cd "`dirname "$0"`"/..; pwd)"
fi
```



Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
export JAVA_HOME=$SPARK_HOME/../jre
export UGO_HOME=$SPARK_HOME/../objectsjava
export PATH=$JAVA_HOME/bin:$PATH
export LD_LIBRARY_PATH=$UGO_HOME/bin:$LD_LIBRARY_PATH
export
SPARK_CLASS_PATH=$SPARK_HOME/../../webapps/iserver/WEB-INF/io
bjects-spark/com.supermap.bdt.core-9.0.0-14819.jar
```

Where:

- JAVA_HOME is Java installation directory;
- SPARK_MASTER_IP is IP of the Master node in the Spark cluster;
- SPARK_WORKER_MEMORY is the maximum memory for Executors provided by the Worker node. It can be set to the memory of the computer to use the resources fully.

In addition, you can set the port of the Master node, worker node, Master Web UI, etc.

If you want to customize the port, please refer to check whether the firewall has opened the port.

Press esc to exit the editing mode, input :wq to save an exit the editing.

**Modify host name**

Please refer to Modify host name.

**Configure the Worker node for Spark**

Each Worker node should be configured in the slaves file. The slaves file is not provided in Spark installation package directly, but slaves.template is provided in the configuration file directory, so the way how to generate the slaves file is the same with spark-env.sh:

```
cp slaves.template slaves
```

Open the slaves file.

```
vi slaves
```

Delete the default node “localhost”, input the host name of each node in Spark cluster:

```
sparkmaster
sparkworker1
sparkworker2
```

The configuration contents of the three nodes are the same. Save and exit the editing.

# Start Spark cluster


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

Start the Master node

Execute the following commands in the computer the Master node is on:

```
cd /usr/local/spark/spark-2.1.0-bin-hadoop2.7/sbin
./start-master.sh
```

Start the Worker node

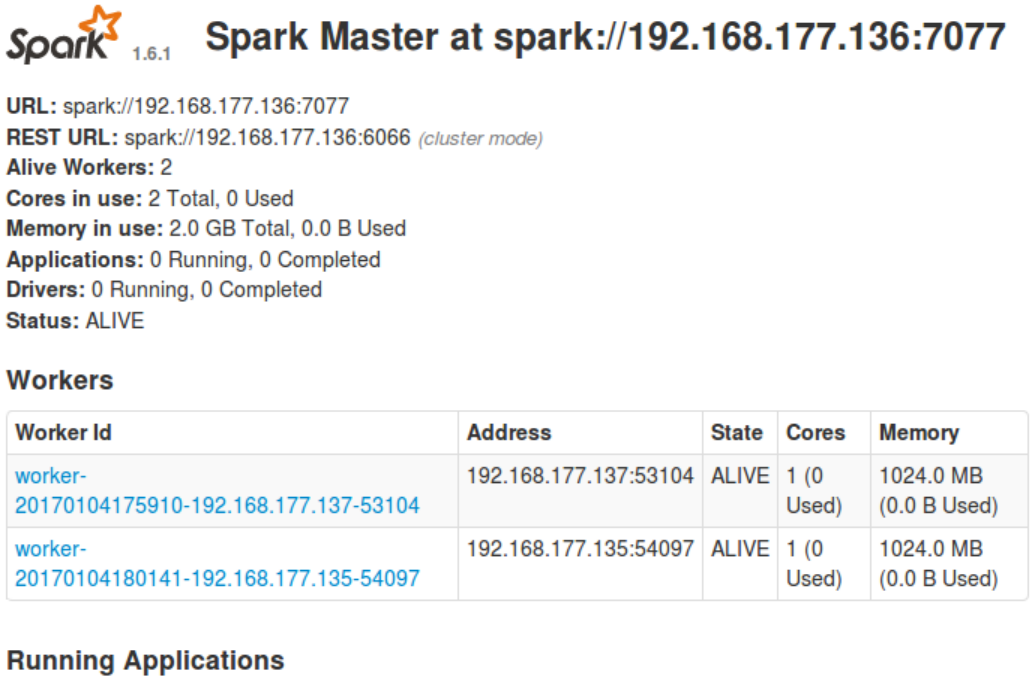
Execute the following commands in the computer the Worker node is on:

```
cd /usr/local/spark/spark-2.1.0-bin-hadoop2.7/sbin
./start-slave.sh --webui-port 8081 spark://sparkmaster:7077
```

Where:

- --webui-port specifies the Web UI port to access current Worker node, if not specified, it will use a random port.
- spark://sparkmaster:7077 indicates the master node address of the cluster that needs to be connected.

After starting, enter http: // sparkmaster: 8080 in the browser to view the cluster.




# Spark cluster optimization configuration



Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

iServer's distributed analysis service is based on the Spark computing platform, providing GIS distributed analysis and processing capabilities. Different hardware environments, Spark cluster environments, and analysis data of different sizes all affect the performance of distributed analytics. To achieve the best performance, you need to optimize the configuration according to different scenarios. Here are some commonly used optimization methods:

1. **When Spark is running, it will start the executor to perform the task, and you can optimize the data processing efficiency by adjusting the memory allocated to the executor (the default is 4G) in the Spark configuration file according to the actual situation of your machine. The method is:**

   - Enter the conf directory of the Spark installation package, such as 【iServer installation directory】 support \ spark \ conf, rename the spark-defaults.conf.template file to spark-defaults.conf
   - Use the document editor tool to open spark-defaults.conf, add: spark.executor.memory 8g

2. **When the result data of the analysis is relatively large, it will consume many system hardware resources when the spark cluster master node collects the result data from the various sub nodes and then stores the data in the local file or iServer DataStore. In order to improve the efficiency of analysis, you can take the following optimal configuration method:**

   - Modify the spark driver memory
     - Enter the conf directory of the Spark installation package, such as 【iServer installation directory】 support \ spark \ conf, rename the spark-defaults.conf.template file to spark-defaults.conf
     - Use the document editor tool to open spark-defaults.conf, add: spark.driver.memory 5g
   - If you are using the built-in spark in iServer, you can set the Spark master node not to participate in the analysis task
     - Click "Cluster", "Join cluster" in the iServer where the spark master node is located, cancel the cluster reporter in Whether to be the Distributed Analysis node.


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

# Starting local Spark cluster service

## Start iServer Built-in Spark Cluster Service

The master and child nodes need to be configured separately. Here are the steps:

### Start the cluster master node

1. After starting iServer in the master node machine, enter the cluster configuration page
2. In the iServer Service Manager (http://supermapiserver:8090/iserver/manager), click the Configure Clusters tab
3. Check "Enable", select "Use the local machine's spark cluster service (default)", and click Save. The system will automatically start the master node of the Spark distributed cluster, and the service list page will also display the distributed analysis services distributedanalyst/rest. You can view the status of the Spark cluster by using Spark's WebUI (http://{ip}:8080)
4. If you want the master node to participate in the data processing, you need to add the master node to the cluster. For details, please refer to Add Child Nodes to Cluster.

### Add Child Nodes to Cluster

1. After starting iServer in the child node machine, enter the cluster configuration page
2. Click the Join Clusters tab, and click Add Reporter
3. Fill in the "Cluster service address" tab with: http://{master node IP}:{port}/iserver/services/cluster. You need to check the two options: “whether to do distributed analysis on node" and "whether the reporter is enabled”
4. If the added cluster has its Security Control enabled, then fill in the "Security Token”
5. When you click Save, you can see which Spark Workers joined the Spark cluster via Spark's WebUI (http://{ip}:8080).


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

6. You need to create a task in the master node.

**Note: You must start the cluster master node first. If you add a child node first, then the child node cannot be added to the Spark cluster. You need to remove the added reporter first, and start the master node, then add the reporter for the child node.**

If you have built your own Spark cluster, iServer supports the use of distributed analytics and real-time data analysis through the option of "Use other Spark cluster services".

## Start Other Spark Cluster Services

1. Enter iServer service manager, such as: http://supermapiserver:8090/iserver/manager
2. Go to the Configure Clusters page, such as: http://supermapiserver:8090/iserver/manager/clustermembers
3. Check "Enable", select "Use other spark cluster services", enter the master node service address of the Spark distributed cluster you have built, and click Save
4. You can use the distributed analytics service by accessing the distributed analysis service distributedanalyst/rest in the service list page (http://supermapiserver:8090/iserver/services) without any other configuration.


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com