

Ubuntu 14.04下Hadoop集群安装

最近在自学 Hadoop，于是乎跟着一起玩玩，在此为她整理一篇基础搭建的博客，希望对她能有所帮助。同样，开始之前先来了解下，什么是 Hadoop。

Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构。它是根据 Google 公司发表的 MapReduce 和 Google 文件系统的论文自行实现而成。Hadoop 框架透明地为应用提供可靠性和数据移动。它实现了名为 MapReduce 的编程范式：应用程序被分区成许多小部分，而每个部分都能在集群中的任意节点上运行或重新运行。

Hadoop 实现了一个分布式文件系统（Hadoop Distributed File System），简称 HDFS。HDFS 有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS 放宽了（relax）POSIX 的要求，可以以流的形式访问（streaming access）文件系统中的数据。

用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop 的框架最核心的设计就是 HDFS 和 MapReduce。HDFS 为海量的数据提供了存储，则 MapReduce 为海量的数据提供了计算。

-----分割线-----

下面关于 **Hadoop** 的文章您也可能喜欢，不妨看看：

Ubuntu14.04 下 Hadoop2.4.1 单机/伪分布式安装配置教程 <http://www.linuxidc.com/Linux/2015-02/113487.htm>

CentOS 6.3 下 Hadoop 伪分布式平台搭建 <http://www.linuxidc.com/Linux/2016-11/136789.htm>

Ubuntu 14.04 LTS 下安装 Hadoop 1.2.1（伪分布模式） <http://www.linuxidc.com/Linux/2016-09/135406.htm>

Ubuntu 上搭建 Hadoop 环境（单机模式+伪分布模式） <http://www.linuxidc.com/Linux/2013-01/77681.htm>

实战 CentOS 系统部署 Hadoop 集群服务 <http://www.linuxidc.com/Linux/2016-11/137246.htm>

Hadoop 2.6.0 HA 高可用集群配置详解 <http://www.linuxidc.com/Linux/2016-08/134180.htm>

Spark 1.5、Hadoop 2.7 集群环境搭建 <http://www.linuxidc.com/Linux/2016-09/135067.htm>

在 Ubuntu X64 上编译安装 Hadoop <http://www.linuxidc.com/Linux/2016-12/138568.htm>

CentOS 6.7 安装 Hadoop 2.7.3 <http://www.linuxidc.com/Linux/2017-01/139089.htm>

CentOS7+Hadoop2.5.2+Spark1.5.2 环境搭建 <http://www.linuxidc.com/Linux/2017-01/139364.htm>

-----分割线-----

搭建

搭建集群的话，需要最少两台机器来搭建一个多节点的 Hadoop 环境，我这里使用 Hadoop 最新稳定的 2.7.3 版本和三台云主机（1主两从，Ubuntu 14.04 LTS）。

1.修改 hosts 文件

保证三台机器的网络是可达的前提下，更改主机名，并修改 hosts 文件：

```
# hostnamectl set-hostname master // 在 master 节点上执行
# hostnamectl set-hostname slave-1 // 在 slave-1 节点上执行
# hostnamectl set-hostname slave-2 // 在 slave-2 节点上执行
分别把三台机器的 hosts 文件进行修改：
# vim /etc/hosts
192.168.1.2 master
192.168.1.3 slave-1
192.168.1.4 slave-2
```

2.在 master 和 slave 节点上安装 java:

```
# add-apt-repository ppa:webupd8team/java //添加 ppa
# apt-get update
# apt-get install oracle-java8-installer
# java -version //检验 Java 版本
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
```

3.禁用 IPv6

现在 Hadoop 目前对 IPv6 不是很好, 在一些 Linux 发行版上可能造成一些未知 bug。在 Hadoop Wiki 上提供了方法来禁用, 我这里修改 sysctl.conf 文件, 添加以下几行:

```
# vim /etc/sysctl.conf
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
# sysctl -p //使其立即生效
```

4.创建 Hadoop User

在 master 和 slave 节点上执行:

```
# addgroup hdgroup //创建 hadoop group
# adduser --ingroup hdgroup hduser //创建 Hadoop User 并加入 Hadoop group
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hdgroup' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:          //输入密码之后一路回车即可
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
```

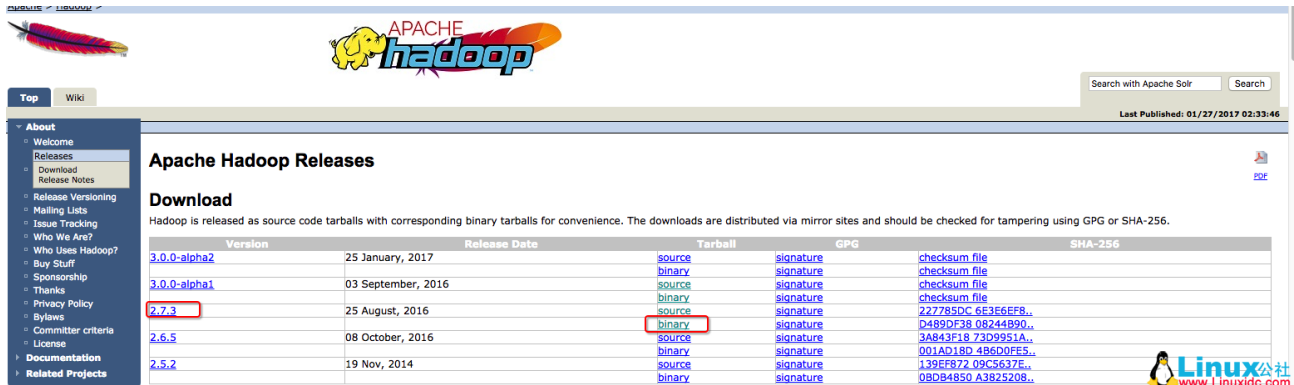
Hadoop 要求无密码登录, 所以需要生成秘钥, 这里注意要用刚才创建的普通 hduser 用户, 分别在 master 和 slave 上执行如下操作:

```
# su - hduser
$ ssh-keygen -N ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
5b:ae:c6:5a:ce:66:51:d3:6c:14:9b:b2:8a:da:e9 hduser@master
The key's randomart image is:
+---[ RSA 2048]-----+
|      .. |
|      .o |
|      .o= |
|      oo* |
|      S.o+ |
|      ..= |
|      ..+.. |
|      o ==. |
|      ..E=+ |
+-----+
$ ssh-copy-id hduser@master
```

```
$ ssh-copy-id hduser@slave-1
$ ssh-copy-id hduser@slave-2
```

5. 下载和安装 Hadoop

登录 Hadoop 的[官方下载地址](#)，选择你需要的版本，复制下载链接，这里我选用最新的 2.7.3 版本：



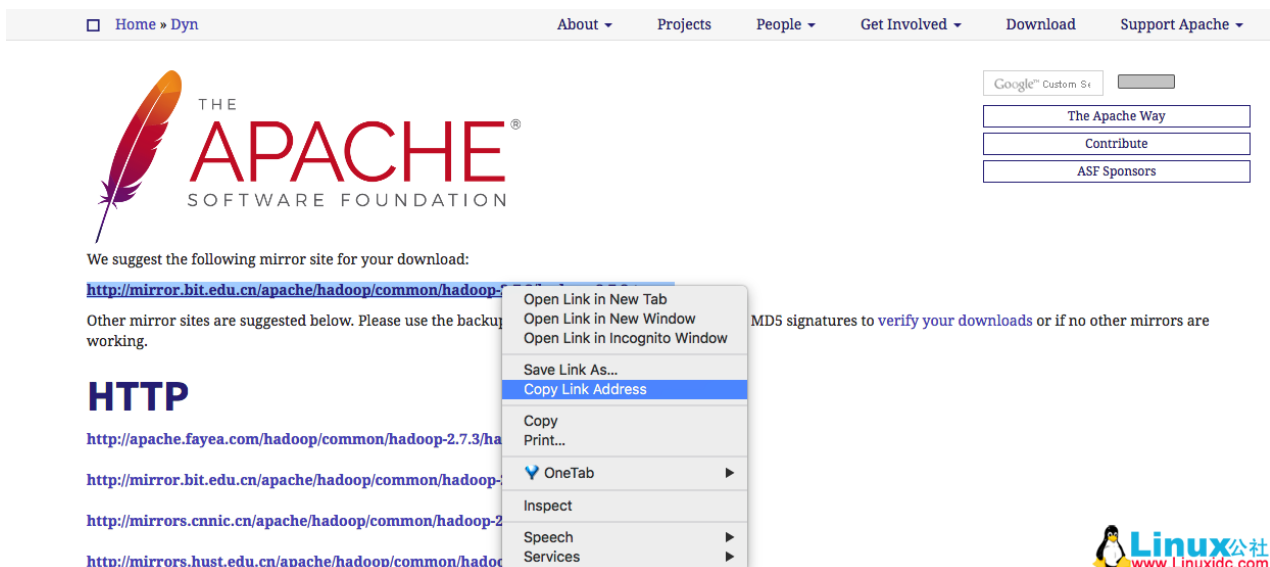
Apache Hadoop Releases

Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-256.

Version	Release Date	Tarball	GPG	SHA-256
3.0.0-alpha2	25 January, 2017	source	signature	checksum file
3.0.0-alpha1	03 September, 2016	source	signature	checksum file
2.7.3	25 August, 2016	source	signature	checksum file
2.6.5	08 October, 2016	source	signature	checksum file
2.5.2	19 Nov, 2014	source	signature	checksum file

打开链接之后，右键复制链接地址：



THE APACHE SOFTWARE FOUNDATION

We suggest the following mirror site for your download:

<http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

Other mirror sites are suggested below. Please use the backup mirror sites if the primary mirror is not working.

HTTP

<http://apache.fayea.com/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://mirrors.cnnic.cn/apache/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

<http://mirrors.hust.edu.cn/apache/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

在 master 和 slave 上分别执行（你也可以在一台机器上下载完之后拷贝到另外两台上）：

```
$ cd /home/hduser
$ wget -c
$ tar -zxvf hadoop-2.7.3.tar.gz
$ mv hadoop-2.7.3 hadoop
```

6. 更改环境变量

首先确定之前安装的 java home 目录，查找办法如下（在任意一台机器上执行）：

```
hduser@master:~$ env | grep -i java
JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

分别在 master 和 slave 节点上执行以下操作，编辑".bashrc"文件，添加如下几行：

```
$ vim .bashrc //编辑文件，添加如下几行
export HADOOP_HOME=/home/hduser/hadoop
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
$ source .bashrc //source 使其立即生效
```

分别在 master 和 slave 节点上执行以下操作，更改 Hadoop-env 的 JAVA_HOME：

```
$ vim /home/hduser/hadoop/etc/hadoop/hadoop-env.sh
#export JAVA_HOME=${JAVA_HOME} //更改此行，或者注释掉新加以下一行
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

7.Hadoop 配置

Hadoop 的配置这里主要涉及四个配置文件：etc/hadoop/core-site.xml，etc/hadoop/hdfs-site.xml，etc/hadoop/yarn-site.xml and etc/hadoop/mapred-site.xml.

这里摘录网络上的一段话，在继续下面的操作之前一定要阅读这段，以便更好的理解：

Hadoop Distributed File System: A distributed file system that provides high-throughput access to application data. A HDFS cluster primarily consists of a NameNode that manages the file system metadata and DataNodes that store the actual data. If you compare HDFS to a traditional storage structures (e.g. FAT, NTFS), then NameNode is analogous to a Directory Node structure, and DataNode is analogous to actual file storage blocks.

Hadoop YARN: A framework for job scheduling and cluster resource management.

Hadoop MapReduce: A YARN-based system for parallel processing of large data sets.

①在 master 和 slave 节点上更改"core-site.xml"文件，master 和 slave 节点应该使用相同"fs.defaultFS"值，而且必须指向 master 节点;在"configuration"中间添加如下配置：

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hduser/tmp</value>
  <description>Temporary Directory.</description>
</property>

<property>
  <name>fs.defaultFS</name>
  <value>hdfs://master:54310</value>
  <description>Use HDFS as file storage engine</description>
</property>
```

最终 core-site.xml 配置文件如下图所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hduser/tmp</value>
  <description>Temporary Directory.</description>
</property>

<property>
  <name>fs.defaultFS</name>
  <value>hdfs://master:54310</value>
  <description>Use HDFS as file storage engine</description>
</property>
</configuration>
```



如果 tmp 目录不存在，需要手动创建一个：

```
$ mkdir /home/hduser/tmp
$ chown -R hduser:hdgroup /home/hduser/tmp //非 hduser 用户创建虚赋权
```

②只在 master 节点上更改"mapred-site.xml"文件，由于没有这个文件，需要复制那个 template 文件生成一个：

```
$ cd /home/hduser/hadoop/
$ cp -av etc/hadoop/mapred-site.xml.template etc/hadoop/mapred-site.xml
```

编辑 xml 配置文件，在"configuration"中间添加如下配置：

```
<property>
<name>mapreduce.jobtracker.address</name>
<value>master:54311</value>
<description>The host and port that the MapReduce job tracker runs
  at. If "local", then jobs are run in-process as a single map
  and reduce task.
</description>
</property>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
<description>The framework for running mapreduce jobs</description>
</property>
```

③在 master 和 slave 节点上更改"hdfs-site.xml"文件，在"configuration"中间添加如下配置：

```
<property>
<name>dfs.replication</name>
<value>2</value>
<description>Default block replication.
The actual number of replications can be specified when the file is created.
The default is used if replication is not specified in create time.
</description>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/data/hduser/hdfs/namenode</value>
<description>Determines where on the local filesystem the DFS name node should store the name table(fsimage).
If this is a comma-delimited list of directories then the name table is replicated in all of the directories, for
redundancy.
</description>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/data/hduser/hdfs/datanode</value>
<description>Determines where on the local filesystem an DFS data node should store its blocks. If this is a
comma-delimited list of directories, then data will be stored in all named directories, typically on different devices.
Directories that do not exist are ignored.
</description>
```

更多详情见请继续阅读下一页的精彩内容：<http://www.linuxidc.com/Linux/2017-02/140783p2.htm>

并创建刚才配置文件中指定的目录：

```
$ mkdir -p /home/hduser/data/hduser/hdfs/{namenode,datanode}
$ chown -R hduser:hdgroup /home/hduser/data/ //如果非 hduser 用户创建的需要赋权
```

1).这里 dfs.replication 的默认值为 3，这里我设置了 2 个副本，代表每个存储到 HDFS 的文件都有额外一个副本，这里的值可以视集群规模而定。

2)dfs.namenode.name.dir 和 dfs.datanode.name.dir 是 namenode 和 datanode 存放 HDFS 数据块文件的位置，如果没有需要手动创建。

④在 master 和 slave 节点上更改"yarn-site.xml"文件，master 和 slave 节点应该使用相同的值，并且指向 master 节点。在"configuration"中间添加如下配置：

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8032</value>
</property>
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>master:8088</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8031</value>
</property>
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>master:8033</value>
</property>
```


⑤更新 slave 文件

在 master 节点上修改 slave 文件,添加 master 和 slave 节点的主机名或者 ip 地址,并去掉"localhost":

```
$ vim /home/hduser/hadoop/etc/hadoop/slaves
master
slave-1
slave-2
```

⑥格式化 namenode:

在启动 cluster 之前需要先格式化 namenode, 在 master 上执行:

```
$ hdfs namenode -format
```

看到类似提示 INFO: "Storage directory /home/hduser/data/hduser/hdfs/namenode has been successfully formatted."表示格式化成功。

⑦启动服务

可以直接使用 Hadoop 提供的脚本"start-all.sh"启动所有服务,也可以把 dfs 和 yarn 分别启动。可以使用绝对路径:/home/hduser/hadoop/sbin/start-all.sh,也可以直接调用 start-all.sh 脚本(因为前面已经改过 PATH 的路径):

```
$ start-all.sh
```

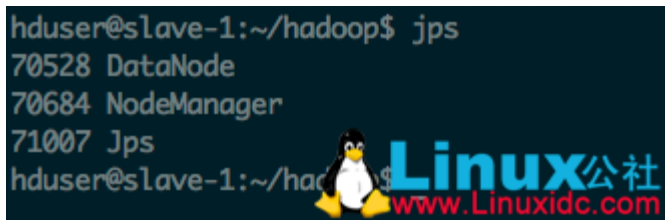
如下图所示没有看到任何错误信息,表示集群已成功启动:

```
hduser@master:~/hadoop$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /home/hduser/hadoop/logs/hadoop-hduser-namenode-master.out
slave-1: starting datanode, logging to /home/hduser/hadoop/logs/hadoop-hduser-datanode-slave-1.out
slave-2: starting datanode, logging to /home/hduser/hadoop/logs/hadoop-hduser-datanode-slave-2.out
master: starting datanode, logging to /home/hduser/hadoop/logs/hadoop-hduser-datanode-master.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is f7:a8:51:44:b3:6f:13:a0:b6:eb:b3:f2:4d:3f:1f:d5. 输入yes即可
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /home/hduser/hadoop/logs/hadoop-hduser-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /home/hduser/hadoop/logs/yarn-hduser-resourcemanager-master.out
slave-2: starting nodemanager, logging to /home/hduser/hadoop/logs/yarn-hduser-nodemanager-slave-2.out
slave-1: starting nodemanager, logging to /home/hduser/hadoop/logs/yarn-hduser-nodemanager-slave-1.out
master: starting nodemanager, logging to /home/hduser/hadoop/logs/yarn-hduser-nodemanager-master.out
```

⑧验证查看

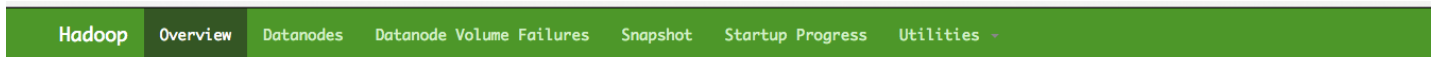
使用 jps 命令分别在 master 和 slave 上查看启动的服务

```
hduser@master:~/hadoop$ jps
42608 NodeManager
41912 DataNode
42122 SecondaryNameNode
42844 Jps
41741 NameNode
42286 ResourceManager
hduser@master:~/hadoop$
```



网页验证:

浏览器打开: <http://master:50070>



Overview 'master:54310' (active)

Started:	Sat Feb 11 20:17:48 CST 2017
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-8ab83c9b-a827-4ee0-880c-6f14515222c3
Block Pool ID:	BP-10524817-118.193.22.162-1486815042264

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 72.91 MB of 199 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 43.27 MB of 44.16 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	173.13 GB
DFS Used:	84 KB (0%)
Non DFS Used:	17.1 GB
DFS Remaining:	156.03 GB (90.12%)
Block Pool Used:	84 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)



查看 yarn web console: <http://master:8088/cluster/nodes>

如果所有 node 均正常启动, 这里能够全部显示:

Nodes of the cluster

Cluster

- About
- Nodes
- Node Labels
- Applications
- NEW
- NEW SAVING
- SUBMITTED
- ACCEPTED
- RUNNING
- FINISHED
- FAILED
- KILLED
- Scheduler

Tools

Cluster Metrics

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	0 B	24 GB	0 B	0	24	0	3	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Show 20 entries

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
/default-rack		RUNNING	slave-2:45504	slave-2:8042	Sat Feb 11 20:32:14 +0800 2017		0	0 B	8 GB	0	8	2.7.3
/default-rack		RUNNING	slave-1:40929	slave-1:8042	Sat Feb 11 20:32:14 +0800 2017		0	0 B	8 GB	0	8	2.7.3
/default-rack		RUNNING	master:39928	master:8042	Sat Feb 11 20:32:16 +0800 2017		0	0 B	8 GB	0	8	2.7.3

Showing 1 to 3 of 3 entries

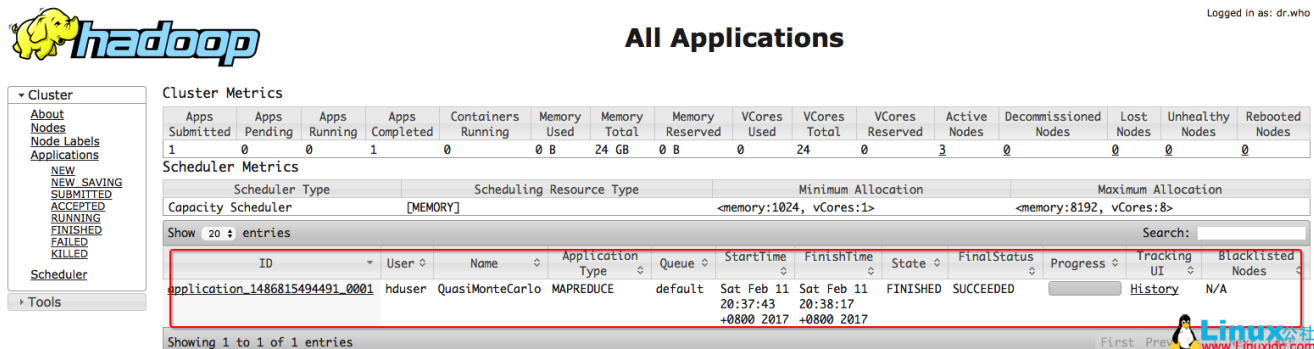


Hadoop 解压的 share 目录里给我们提供了几个 example 的 jar 包, 我们执行一个看下效果:


```
$ hadoop jar /home/hduser/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar pi 30 100
```

执行之后通过浏览器访问：<http://master:8088/cluster/apps>

能够看到当前执行的任务：



The screenshot shows the Hadoop cluster management interface. On the left is a sidebar with navigation links like 'Cluster', 'About Nodes', 'Node Labels', and 'Applications'. The main area is titled 'All Applications' and displays 'Cluster Metrics' and 'Scheduler Metrics'. Below these, there is a table of applications. One application, 'application_1486815494491_0001', is highlighted with a red box. It is a MAPREDUCE job run by user 'hduser' named 'QuasiMonteCarlo' in the 'default' queue. The job is in a 'FINISHED' state with a 'SUCCEEDED' final status. The progress bar is at 100%. The start time is Sat Feb 11 20:37:43 +0800 2017 and the finish time is Sat Feb 11 20:38:17 +0800 2017. The tracking UI is 'History' and blacklisted nodes are 'N/A'.

写在最后：

如果在新增了节点或者删除节点的时候出现了问题，首先删除 slave 的 `hadoop.tmp.dir`，然后重新启动试试看，如果还是不行,尝试把 Master 的 `hadoop.tmp.dir` 删除（意味着 dfs 上的数据也会丢失）然后需要重新 `namenode -format`。

如果有任何报错信息记得去查 log 日志，文件位置在 Hadoop 的安装目录 logs 文件夹下。

更多 Hadoop 相关信息见 [Hadoop 专题页面](http://www.linuxidc.com/topicnews.aspx?tid=13) <http://www.linuxidc.com/topicnews.aspx?tid=13>

本文永久更新链接地址：<http://www.linuxidc.com/Linux/2017-02/140783.htm>

欢迎点击这里的链接进入精彩的[Linux公社](http://www.Linuxidc.com)网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

[Linux公社](http://www.Linuxidc.com)是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：[Ubuntu 专题](#) [Fedora 专题](#) [Android 专题](#) [Oracle 专题](#) [Hadoop 专题](#)
[RedHat 专题](#) [SUSE 专题](#) [红旗 Linux 专题](#) [CentOS 专题](#)



Linux 公社微信公众号：[linuxidc_com](#)



微信扫一扫

Linuxidc.com

订阅专业的最新Linux资讯及开源技术教程。

搜索微信公众号：[linuxidc_com](#)