



链滴

CDH5.4.3 离线安装

作者: [MrWang](#)

原文链接: <https://ld246.com/article/1474944657647>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>Cloudera Manager 下载地址: </p>

<p>http://archive.cloudera.com/cm5/cm/5/cloudera-manager-el6-cm5.4.3_x86_64.ar.gz , </p>

<p>CDH 安装包地址: </p>

<p>http://archive.cloudera.com/dh5/parcels/5.4.3/ , 由于我们的操作系统为 CentOS6.5, 需要下载以下文件: </p>

<p>· CDH-5.4.3-1.cdh5.4.3.p0.6-el6.parcel </p>

<p>· CDH-5.4.3-1.cdh5.4.3.p0.6-el6.parcel.sha1 </p>

<p>· manifest.json </p>

<p>准备工作: 系统环境搭建</p>

<p>以下操作均用 root 用户操作。</p>

<p>1.网络配置(所有节点)</p>

<p>vi /etc/sysconfig/network 修改 hostname: </p>

<p>NETWORKING=yes</p>

<p>HOSTNAME=hadoop.master</p>

<p>通过 service networkrestart 重启网络服务生效。</p>

<p>vi /etc/hosts ,修改 ip 与主机名的对应关系</p>

<p>192.168.226.17 hadoop.master master</p>

<p>192.168.226.18 hadoop.slave1 slave1</p>

<p>192.168.226.19 hadoop.slave2 slave2</p>

<p>192.168.226.20 hadoop.slave3 slave3</p>

<p>注意: 这里需要将每台机器的 ip 及主机名对应关系都写进去, 本机的也要写进去, 否则启动 Agent 的时候会提示 hostname 解析错误。</p>

<p>2.打通 SSH, 设置 ssh 无密码登陆 (所有节点) </p>

<p>在主节点上执行 ssh-keygen -trsa 一路回车, 生成无密码的密钥对。</p>

<p>将公钥添加到认证文件中: cat ~/.ssh/id_rsa.pub>> ~/.ssh/authorized_keys , 并设置 authorized_keys 的访问权限: chmod 600 ~/.ssh/authorized_keys 。</p>

<p>scp 文件到所有 datenode 节点: </p>

<p>scp ./ssh/authorized_keys root@slave1:./ssh/</p>

<p>测试: 在主节点上 ssh slave1, 正常情况下, 不需要密码就能直接登陆进去了。</p>

<p>3.安装 Oracle 的 Java (所有节点) </p>

<p>CentOS, 自带 OpenJdk, 不过运行 CDH5 需要使用 Oracle 的 Jdk, 需要 Java 7 的支持。</p>

<p>卸载自带的 OpenJdk, 使用 rpm -qa | grepjava 查询 java 相关的包, 使用 rpm -e --nodeps 名 卸载 (凡是 java 开头 都卸载) 。</p>

<p>去 Oracle 的官网下载 jdk 的 rpm 安装包, 并使用 rpm -ivh 包名 安装之。</p>

<p>由于是 rpm 包并不需要我们来配置环境变量, 我们只需要配置一个全局的 JAVA_HOME 变量即, 执行命令: </p>

<p>echo "JAVA_HOME=/usr/java/latest/" >> /etc/profile 注意: latest 是 jdk 目录软连接, 所以你使用它或者直接写 jdk 的目录都可以</p>

<p>4.安装配置 MySQL (主节点) </p>

<p>安装 mysql 请看 <a href="https://ld246.com/forward?goto=http%3A%2F%2Fblog.csdn.net%2Fwanghui19931015%2Farticle%2Fdetails%2F52663862" target="_blank" rel="nofollow u

c"><http://blog.csdn.net/wanghui19931015/article/details/52663862> </p>
<p>mysql -uroot -p123456 进入 mysql 命令行, 创建以下数据库: </p>
<p>#hive</p>
<p>create database hiveDEFAULT CHARSET utf8COLLATEutf8_general_ci;</p>
<p>#activity monitor</p>
<p>create database amonDEFAULTCHARSET utf8COLLATE utf8_general_ci;</p>
<p>设置 root 授权访问以上所有的数据库: </p>
<p>#授权 root 用户在主节点拥有所有数据库的访问权限</p>
<p>grant all privileges on *.*to'root'@'master' identifiedby '123456'with grant option;</p>
<p>grant all privileges on *.*to 'scm'@'%' identifiedby'scm'with grant option;</p>
<p>flush privileges;</p>
<p>5.关闭防火墙和 SELinux</p>
<p>注意: 需要在所有的节点上执行, 因为涉及到的端口太多了, 临时关闭防火墙是为了安装起来方便, 安装完以后可以根据需要设置防火墙策略, 保证集群安全。 </p>
<p>关闭防火墙: </p>
<p>service iptables stop (临时关闭) </p>
<p>chkconfig iptables off (重启后生效) </p>
<p>关闭 SELINUX (实际安装过程中发现没有关闭也是可以的, 不知道会不会有问题, 还需进一步验证) :</p>
<p>setenforce 0 (临时生效) </p>
<p>修改 /etc/selinux/config 下的 SELINUX=disabled (重启后永久生效) </p>
<p>6.所有节点配置 NTP 服务</p>
<p>集群中所有主机必须保持时间同步, 如果时间相差较大会引起各种问题。具体思路如下: </p>
<p>master 节点作为 ntp 服务器与外界对时中心同步时间, 随后对所有 datanode 节点提供时间同服务。 </p>
<p>所有 datanode 节点以 master 节点为基础同步时间。 </p>
<p>所有节点安装相关组件: yum install ntp 。完成后, 配置开机启动: chkconfig ntpdon ,检查否设置成功: chkconfig --list ntpd 其中 2-5 为 on 状态就代表成功。 </p>
<p>主节点配置</p>
<p>在配置之前, 先使用 ntpdate 手动同步一下时间, 免得本机与对时中心时间差距太大, 使得 ntpd 不能正常同步。这里选用 master 作为对时中心, ntpdate -u master 。 master 主机添加下面信息到/etc/ntp.conf</p>
<p>server 127.127.1.0</p>
<p>fudge 127.127.1.0 stratum 2</p>
<p>restrict 192.168.226.0 mask 255.255.255.0 nomodify notrap</p>
<p>集群其他节点修改/etc/ntp.conf,如下: </p>
<h2 id="Pleaseconsider-joining-the-pool--http---www-pool-ntp-org-join-html--">Pleaseconsider joining the pool (http://www.pool.ntp.org/join.html.</h2>
<p>#server0.centos.pool.ntp.org iburst</p>
<p>#server1.centos.pool.ntp.org iburst</p>
<p>#server2.centos.pool.ntp.org iburst</p>
<p>#server3.centos.pool.ntp.org iburst</p>
<p>server master</p>
<p>配置文件完成, 保存退出, </p>
<p>设置开机启动 chkconfig ntpd on</p>
<p>启动服务, 执行如下命令: service ntpdstart</p>
<p>正式开工</p>
<p>安装 Cloudera Manager Server 和 Agent</p>
<p>主节点解压安装</p>
<p>cloudera manager 的目录默认位置在/opt 下, 解压: tar xzvfcloudera-manager*.tar.gz 将压后的 cm-5.4.3 和 cloudera 目录放到/opt 目录下。 </p>
<p>为 Cloudera Manager 5 建立数据库</p>

<p>首先需要去 MySQL 的官网下载 JDBC 驱动， http://dev.mysql.com/downloads/connector/j/ ， 解压后， 找到 mysql-connecto-java-5.1.39-bin.jar， 放到/opt/cm-5.4.3/share/cm5/lib/中。 </p>
<p>同时 chmod 755mysql-connector-java-5.1.39-bin.jar</p>
<p>在主节点初始化 CM5 的数据库： </p>
<p>/opt/cm-5.4.3/share/cm5/schema/scm_prepare_database.sh mysql cm -hmaster -uroot -123456 --scm-host master scm scm scm</p>
<p>Agent 配置</p>
<p>修改/opt/cm-5.4.3/etc/cloudera-scm-agent/config.ini 中的 server_host 为主节点的主机名</p>
<p>修改/opt/cm-5.4.3/lib64/cm5/agent/src/cm5/util.py 文件。 </p>
<p>将其中的代码： </p>
<p>pipe = subprocess.Popen(['/bin/bash','-c', ". %s; %s; env" % (path, command)],</p>
<pre><code class="highlight-chroma"> stdout=subprocess.PIPE,env=caller_env)</code></pre>
<p>修改为： </p>
<p>pipe = subprocess.Popen(['/bin/bash','-c', ". %s; %s; env | grep -v { | grep -v }" % (path, command)],</p>
<pre><code class="highlight-chroma"> stdout=subprocess.PIPE, env=caller_env)</code></pre>
<p>同步 Agent 到其他节点</p>
<p>scp -r/opt/cm-5.4.3 root@slave1:/opt/</p>
<p>在所有节点创建 cloudera-scm 用户</p>
<p>useradd --system --home=/opt/cm-5.4.3/run/cloudera-scm-server/ --no-create-home --hell=/bin/false --comment "Cloudera SCM User" cloudera-scm</p>
<p>准备 Parcels， 用以安装 CDH5</p>
<p>将 CHD5 相关的 Parcel 包放到主节点的/opt/cloudera/parcel-repo/目录中（parcel-repo 需手动创建）。 </p>
<p>相关的文件如下： </p>
<p>· CDH-5.4.3-1.cdh5.4.3.p0.12-el6.parcel</p>
<p>· CDH-5.4.3-1.cdh5.4.3.p0.12-el6.parcel.sha1</p>
<p>· manifest.json</p>
<p>最后将 CDH-5.4.3-1.cdh5.4.3.p0.12-el6.parcel.sha1， 重命名为 CDH-5.4.3-1.cdh5.4.3.p0.12-l6.parcel.sha， 这点必须注意， 否则， 系统会重新下载 CDH-5.4.3-1.cdh5.4.3.p0.12-el6.parcel 文。 </p>
<p>相关启动脚本</p>
<p>通过 /opt/cm-5.4.3/etc/init.d/cloudera-scm-server start 启动主节点服务端。 </p>
<p>通过 /opt/cm-5.4.3/etc/init.d/cloudera-scm-agent start 启动其他节点 Agent 服务。 </p>
<p>注意： 两个脚本都启动成功， 正常运行</p>
<p>需要停止服务将以上的 start 参数改为 stop 就可以了， 重启是 restart。 </p>
<p>CDH5 的安装配置</p>
<p>Cloudera ManagerServer 和 Agent 都启动以后， 就可以进行 CDH5 的安装配置了。 </p>
<p>这时可以通过浏览器访问主节点的 7180 端口测试一下了（由于 CM Server 的启动需要花点时， 这里可能要等待一会才能访问）， 默认的用户名和密码均为 admin： </p>
<p>1、 选择免费版本的 CM5。 </p>
<p>2、 各个 Agent 节点正常启动后， 可以在当前管理的主机列表中看到对应的节点。 选择要安装的点， 点继续。 </p>
<p>3、 接下来， 出现以下包名， 说明本地 Parcel 包配置无误， 直接点继续就可以了。 </p>
<p>4、 点击， 继续。 </p>
<p>5、 接下来是服务器检查， 可能会遇到以下问题： </p>
<p>Cloudera 建议将 /proc/sys/vm/swappiness 设置为 0。 当前设置为 60。 使用 sysctl 命令在运

时更改该设置并编辑/etc/sysctl.conf 以在重启后保存该设置。您可以继续进行安装，但可能会遇到问题，Cloudera Manager 报告您的主机由于交换运行状况不佳。以下主机受到影响：

<p>通过 echo 0 >/proc/sys/vm/swappiness 即可解决。 </p>

<p>6、接下来是选择安装服务： </p>

<p>服务配置，一般情况下保持默认就可以了（Cloudera Manager 会根据机器的配置自动进行配置如果需要特殊调整，自行进行设置就可以了）： </p>

<p>接下来是数据库的设置，检查通过后就可以进行下一步的操作了： </p>

<p>7、集群设置的审查页面，保持默认配置的： </p>

<p>终于到安装各个服务的不地方了，注意，这里安装 Hive 的时候可能会报错，因为我们使用了 MySQL 作为 hive 的元数据存贮，hive 默认没有带 mysql 的驱动，通过以下命令拷贝一个就行了： </p>

<p>cp/opt/cm-5.4.3/share/cm/lib/mysql-connector-java-5.1.39-bin.jar/opt/cloudera/parcels/CDH-5.4.3-1.cdh5.1.3.p0.12/lib/hive/lib/</p>

<p>安装完成后，就可以进入集群界面看一下集群的当前状况了。 </p>

<p>测试</p>

<p>在集群的一台机器上执行以下模拟 Pi 的示例程序： </p>

<p>sudo -u hdfs hadoop jar/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi10100</p>

<p>执行过程需要花一定的时间，通过 YARN 的后台也可以看到 MapReduce 的执行状态： </p>

<p>MapReduce 执行过程中终端的输出如下： </p>

<p>Number of Maps =10</p>

<p>Samples per Map =100</p>

<p>Wrote inputfor Map#0</p>

<p>Wrote inputfor Map#1</p>

<p>Wrote inputfor Map#2</p>

<p>Wrote inputfor Map#3</p>

<p>Wrote inputfor Map#4</p>

<p>Wrote inputfor Map#5</p>

<p>Wrote inputfor Map#6</p>

<p>Wrote inputfor Map#7</p>

<p>Wrote inputfor Map#8</p>

<p>Wrote inputfor Map#9</p>

<p>Starting Job</p>

<p>14/10/1301:15:34 INFOclient.RMProxy: Connecting to ResourceManager at n1/192.168.1.61:8032</p>

<p>14/10/1301:15:36 INFOinput.FileInputFormat: Total input paths to process : 10</p>

<p>14/10/1301:15:37 INFOmapreduce.JobSubmitter: number of splits:10</p>

<p>14/10/1301:15:39 INFOmapreduce.JobSubmitter: Submitting tokens for job:job_141313207582_0001</p>

<p>14/10/1301:15:40 INFOimpl.YarnClientImpl: Submitted application application_141313237582_0001</p>

<p>14/10/1301:15:40 INFOmapreduce.Job: The url to track the job: http://n1:8088/proxy/application_1413132307582_0001/</p>

<p>14/10/1301:15:40 INFOmapreduce.Job: Running job: job_1413132307582_0001</p>

<p>14/10/1301:17:13 INFOmapreduce.Job: Job job_1413132307582_0001 running in uber mode : false</p>

<p>14/10/1301:17:13 INFOmapreduce.Job: map 0% reduce0%</p>

<p>14/10/1301:18:02 INFOmapreduce.Job: map 10% reduce0%</p>

<p>14/10/1301:18:25 INFOmapreduce.Job: map 20% reduce0%</p>

<p>14/10/1301:18:35 INFOmapreduce.Job: map 30% reduce0%</p>

<p>14/10/1301:18:45 INFOmapreduce.Job: map 40% reduce0%</p>

<p>14/10/1301:18:53 INFOmapreduce.Job: map 50% reduce0%</p>

<p>14/10/1301:19:01 INFOmapreduce.Job: map 60% reduce0%</p>

<p>14/10/1301:19:09 INFOmapreduce.Job: map 70% reduce0%</p>

<p>14/10/1301:19:17 INFOmapreduce.Job: map 80% reduce0%</p>

<p>14/10/1301:19:25 INFOmapreduce.Job: map 90% reduce0%</p>
<p>14/10/1301:19:33 INFOmapreduce.Job: map 100% reduce0%</p>
<p>14/10/1301:19:51 INFOmapreduce.Job: map 100% reduce100%</p>
<p>14/10/1301:19:53 INFOmapreduce.Job: Job job_1413132307582_0001 completed successfully</p>
<p>14/10/1301:19:56 INFOmapreduce.Job: Counters: 49</p>
<p>File System Counters</p>
<p>FILE: Number of bytesread=91</p>
<p>FILE: Number of bytes written=1027765</p>
<p>FILE: Number ofread operations=0</p>
<p>FILE: Number of largeread operations=0</p>
<p>FILE: Number ofwrite operations=0</p>
<p>HDFS: Number of bytesread=2560</p>
<p>HDFS: Number of bytes written=215</p>
<p>HDFS: Number ofread operations=43</p>
<p>HDFS: Number of largeread operations=0</p>
<p>HDFS: Number ofwrite operations=3</p>
<p>Job Counters</p>
<p>Launchedmap tasks=10</p>
<p>Launched reduce tasks=1</p>
<p>Data-local map tasks=10</p>
<p>Totaltime spent by all maps in occupied slots (ms)=118215</p>
<p>Totaltime spent by all reduces in occupied slots (ms)=11894</p>
<p>Totaltime spent by allmap tasks (ms)=118215</p>
<p>Totaltime spent by all reduce tasks (ms)=11894</p>
<p>Total vcore-seconds taken by allmap tasks=118215</p>
<p>Total vcore-seconds taken by all reduce tasks=11894</p>
<p>Total megabyte-seconds taken by allmap tasks=121052160</p>
<p>Total megabyte-seconds taken by all reduce tasks=12179456</p>
<p>Map-Reduce Framework</p>
<p>Map input records=10</p>
<p>Map output records=20</p>
<p>Map output bytes=180</p>
<p>Map output materialized bytes=340</p>
<p>Inputsplitted bytes=1380</p>
<p>Combine input records=0</p>
<p>Combine output records=0</p>
<p>Reduce input groups=2</p>
<p>Reduce shuffle bytes=340</p>
<p>Reduce input records=20</p>
<p>Reduce output records=0</p>
<p>Spilled Records=40</p>
<p>Shuffled Maps =10</p>
<p>Failed Shuffles=0</p>
<p>Merged Map outputs=10</p>
<p>GCtime elapsed (ms)=1269</p>
<p>CPUtime spent (ms)=9530</p>
<p>Physical memory (bytes) snapshot=3792773120</p>
<p>Virtual memory (bytes) snapshot=16157274112</p>
<p>Total committed heap usage (bytes)=2856624128</p>
<p>Shuffle Errors</p>
<p>BAD_ID=0</p>
<p>CONNECTION=0</p>
<p>IO_ERROR=0</p>

<p>WRONG_LENGTH=0</p>
<p>WRONG_MAP=0</p>
<p>WRONG_REDUCE=0</p>
<p>File Input Format Counters</p>
<p>Bytes Read=1180</p>
<p>File Output Format Counters</p>
<p>Bytes Written=97</p>
<p>Job Finished in262.659 seconds</p>
<p>Estimated value of Pi is3.14800000000000000000</p>
<p>最后 感谢 http://www.tuicool.com/articles/ENjmeaY/ 博主，因为有这个作为参考，可以让自己尽快的熟悉安装流程。具体要通过 日志来分析问题。</p>
<p>当然，离线安装后可能会因为一些原因想删除 CDH 重新安装，这里我会在下一篇文章中会体现</p>