CentOS-6.7 编译 hadoop-2.6.5

目录

1、阅读编译文档

1、准备一个 hadoop 源码包，我选择的 hadoop 版本是：hadoop-2.6.5-src.tar.gz，在 hadoop-2.6.5 的源码包的根目录下有一个文档叫做 BUILDING.txt，这其中说明了编译 hadoop 所需要的一些编译环境相关的东西。不同的 hadoop 版本的要求都不一样。对应的版本参照 BUILDING.txt

请仔细阅读：

```
Requirements:

* Unix System
* JDK 1.6+
* Maven 3.0 or later
* Findbugs 1.3.9 (if running findbugs)
* ProtocolBuffer 2.5.0
* CMake 2.6 or newer (if compiling native code), must be 3.0 or newer on Mac
* Zlib devel (if compiling native code)
* openssl devel ( if compiling native hadoop-pipes )
* Internet connection for first build (to fetch all Maven and Hadoop dependencies)
```

2、对应以上需求，我们准备好所要求版本的这些软件

1、准备一台 Unix 类型操作系统，在这里我们选用的是 CentOS-6.7，初次编译要求必须

联网，切记

以下这些东西都是需要的，详细安装在下面，这里只介绍我准备这些软件所选择的版本
2、 安装 openssl-devel
3、 安装 gcc
4、 安装 gcc-c++
5、 JDK：jdk-7u80-linux-x64.tar.gz
6、 Maven：apache-maven-3.3.3-bin.tar.gz
7、 Ant：apache-ant-1.9.4-bin.tar.gz
8、 FindBugs：findbugs-3.0.0.tar.gz
9、 Cmake：cmake-2.8.12.2.tar.gz
10、Protobuf：protobuf-2.5.0.tar.gz
11、Snappy：snappy-1.1.1.tar.gz

2、准备编译环境

安装以下软件：
yum -y install svn
yum -y install autoconf automake libtool cmake
yum -y install ncurses-devel
yum -y install openssl-devel

3、安装 gcc

先使用命令检测一下看 gcc 是否已经安装过了
[root@compile_hadoop soft]# **gcc -v**

gcc version 4.4.7 20120313 (Red Hat 4.4.7-16) (GCC)
如果最后一行出现如上的 gcc 版本信息日志，表示已经安装成功过了。

不然使用命令安装：
yum install gcc

4、安装 gcc-c++

直接使用命令安装：**yum install gcc-c++**

5、安装 make

先检测以下系统是否安装了 make 工具：**make -version**

```
[root@hadoop apps]# make -version
GNU Make 3.81
Copyright (C) 2006  Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for x86_64-redhat-linux-gnu
[root@hadoop apps]#
```

如果没有安装过 make，那么请使用命令安装：**yum install -y make**

6、安装 maven（必须）

1、 解压

tar -zxvf /root/soft/apache-maven-3.3.3-bin.tar.gz -C /root/apps/

2、 修改配置文件（如果需要更改默认的 maven 仓库路径的话）

1、 进入到 maven 安装包的 conf 目录：**/root/apps/apache-maven-3.3.3/conf**

2、 修改配置文件 settings.xml

```
  <!-- localRepository
   | The path to the local repository maven will use to store artifacts.
   |
   | Default: ${user.home}/.m2/repository
  <localRepository>/path/to/local/repo</localRepository>
  -->

<localRepository>/root/mavenrepo/</localRepository>
```

在配置文件的中部找到 localRepository 这个标签，它本来是注释了的，并且有一个默认仓库路径，我们最好自己设置一个，所以我自己加了一个，我的路径是：

<localRepository>/root/mavenrepo/</localRepository>

3、 配置环境变量

export M2_HOME=/root/apps/apache-maven-3.3.3

export PATH=$PATH:$M2_HOME/bin

4、 检测是否成功

source /etc/profile

mvn -version

```
[root@hadoop apps]# mvn -version
Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T19:57:37+08:00)
Maven home: /root/apps/apache-maven-3.3.3
Java version: 1.7.0_80, vendor: Oracle Corporation
Java home: /root/apps/jdk1.7.0_80/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "2.6.32-573.el6.x86_64", arch: "amd64", family: "unix"
[root@hadoop apps]#
```

7、安装 jdk（必须）

1、 解压

tar -zxvf /root/soft/jdk-7u80-linux-x64.tar.gz -C /root/apps/

2、 配置环境变量

export JAVA_HOME=/root/apps/jdk1.7.0_80

export PATH=$PATH:$JAVA_HOME/bin

export CLASSPATH=.:/root/apps/jdk1.7.0_80/lib/dt.jar:/root/apps/jdk1.7.0_80/lib/tools.jar

3、 检测是否安装

source /etc/profile

java -version

```
[root@hadoop ~]# java -version
java version "1.7.0_80"
Java(TM) SE Runtime Environment (build 1.7.0_80-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.80-b11, mixed mode)
[root@hadoop ~]#
```

8、安装 ant（重要）

1、 解压

tar -zxvf /root/soft/apache-ant-1.9.4-bin.tar.gz -C /root/apps/

2、 配置环境变量

export ANT_HOME=/root/apps/apache-ant-1.9.4

export PATH=$PATH:$ANT_HOME/bin

3、 检测是否安装成功

source /etc/profile

ant -version

```
[root@hadoop ~]# ant -version
Apache Ant(TM) version 1.9.4 compiled on April 29 2014
[root@hadoop ~]#
```

9、安装 findbugs（可选，最好装）

1、 解压

tar -zxvf /root/soft/findbugs-3.0.0.tar.gz -C /root/apps/

2、 配置环境变量

export FINDBUGS_HOME=/root/apps/findbugs-3.0.0

export PATH=$PATH:$FINDBUGS_HOME/bin

3、 检测是否安装成功

[root@hadoop apps]# **findbugs -version**

```
[root@hadoop ~]# findbugs -version
3.0.0
[root@hadoop ~]#
```

10、安装 cmake（重要）

1、 解压

tar -zxvf /root/soft/cmake-2.8.12.2.tar.gz -C /root/apps/

2、 安装

首先进入到根目录：

cd /root/apps/cmake-2.8.12.2/

然后依次执行以下命令：

./bootstrap

gmake & gmake install

3、 检测是否安装成功

cmake -version

```
[root@hadoop ~]# cmake -version
cmake version 2.8.12.2
[root@hadoop ~]#
```

11、安装 protobuf（重要）

1、 解压

tar -zxvf /root/soft/protobuf-2.5.0.tar.gz -C /root/apps/

2、 安装

首先进入到根目录：

cd /root/apps/protobuf-2.5.0/

然后依次执行以下命令：

./configure --prefix=/root/apps/protobuf （表示安装到这个目录）

make

make check

make install

3、 配置环境变量

export PROTOBUF_HOME=/root/apps/protobuf

export PATH=$PATH:$PROTOBUF_HOME/bin

注意：PROTOBUF_HOME 就是我们在执行 configure 命令的时候指定的目录

4、 检测是否安装成功

[root@hadoop ~]# protoc --version

```
[root@hadoop ~]# protoc --version
libprotoc 2.5.0
[root@hadoop ~]#
```

12、安装 Snappy

1、使用 root 用户安装 Snappy：

tar -zxvf /root/soft/snappy-1.1.1.tar.gz -C /root/apps/

2、编译安装：

cd snappy-1.1.1/

./configure

make

make install

3、查看 snappy 库文件

ls -lh /usr/local/lib |grep snappy

```
[root@hadoop ~]# ls -lh /usr/local/lib |grep snappy
-rw-r--r--. 1 root root 228K Sep  2 19:28 libsnappy.a
-rwxr-xr-x. 1 root root  953 Sep  2 19:28 libsnappy.la
lrwxrwxrwx. 1 root root   18 Sep  2 19:28 libsnappy.so -> libsnappy.so.1.2.0
lrwxrwxrwx. 1 root root   18 Sep  2 19:28 libsnappy.so.1 -> libsnappy.so.1.2.0
-rwxr-xr-x. 1 root root 145K Sep  2 19:28 libsnappy.so.1.2.0
[root@hadoop ~]#
```

13、编译 hadoop

1、 解压 hadoop 源码包

tar -zxvf /root/soft/hadoop-2.6.5-src.tar.gz -C /root/apps/

2、 在编译之前防止 java.lang.OutOfMemoryError: Java heap space 堆栈问题,在 centos 系统中执行命令：**export MAVEN_OPTS="-Xms256m -Xmx512m"**

3、 进入到源码包根目录下

cd /root/apps/hadoop-2.6.5-src

4、 执行命令编译

Create binary distribution with native code and with documentation:

mvn package -Pdist,native,docs -DskipTests –Dtar

如果中途编译失败，并且不要文档的话，请使用这个命令：

mvn clean package -Pdist,native -DskipTests -Dtar -Dsnappy.lib=/usr/local/lib -Dbundle.snappy -Drequire.openssl

PS：tar 和 **dist** 表示用 maven 项目管理工具对 hadoop 进行编译，编译好了之后会打成 tar.gz 包放到 hadoop-dist 目录下，**native** 和 **docs** 表示编译出来会编译出来本地库，并且把文档打包到该.tar.gz 文件中，**skipTests** 表示忽略测试

5、 静静等待编译…… 第一次编译预估一个小时左右。

6、 编译成功了的话，最后的日志信息：

```
[INFO] Reactor Summary:
[INFO]
[INFO] Apache Hadoop Main ..... SUCCESS [05:52 min]
[INFO] Apache Hadoop Project POM ..... SUCCESS [02:23 min]
[INFO] Apache Hadoop Annotations ..... SUCCESS [01:11 min]
[INFO] Apache Hadoop Assemblies ..... SUCCESS [    0.267 s]
[INFO] Apache Hadoop Project Dist POM ..... SUCCESS [01:08 min]
[INFO] Apache Hadoop Maven Plugins ..... SUCCESS [01:13 min]
[INFO] Apache Hadoop MiniKDC ..... SUCCESS [06:36 min]
[INFO] Apache Hadoop Auth ..... SUCCESS [04:32 min]
[INFO] Apache Hadoop Auth Examples ..... SUCCESS [ 22.255 s]
[INFO] Apache Hadoop Common ..... SUCCESS [08:12 min]
[INFO] Apache Hadoop NFS ..... SUCCESS [    8.060 s]
[INFO] Apache Hadoop KMS ..... SUCCESS [01:37 min]
[INFO] Apache Hadoop Common Project ..... SUCCESS [    0.039 s]
[INFO] Apache Hadoop HDFS ..... SUCCESS [05:54 min]
[INFO] Apache Hadoop HttpFS ..... SUCCESS [01:24 min]
[INFO] Apache Hadoop HDFS BookKeeper Journal ..... SUCCESS [01:11 min]
[INFO] Apache Hadoop HDFS-NFS ..... SUCCESS [    4.370 s]
[INFO] Apache Hadoop HDFS Project ..... SUCCESS [    0.048 s]
[INFO] hadoop-yarn ..... SUCCESS [    0.040 s]
[INFO] hadoop-yarn-api ..... SUCCESS [01:26 min]
```

```
[INFO] hadoop-yarn-common ..... SUCCESS [02:11 min]
[INFO] hadoop-yarn-server ..... SUCCESS [    0.048 s]
[INFO] hadoop-yarn-server-common ..... SUCCESS [ 32.832 s]
[INFO] hadoop-yarn-server-nodemanager ..... SUCCESS [03:11 min]
[INFO] hadoop-yarn-server-web-proxy ..... SUCCESS [    2.603 s]
[INFO] hadoop-yarn-server-applicationhistoryservice ..... SUCCESS [    6.006 s]
[INFO] hadoop-yarn-server-resourcemanager ..... SUCCESS [ 19.331 s]
[INFO] hadoop-yarn-server-tests ..... SUCCESS [    5.008 s]
[INFO] hadoop-yarn-client ..... SUCCESS [    7.812 s]
[INFO] hadoop-yarn-applications ..... SUCCESS [    0.049 s]
[INFO] hadoop-yarn-applications-distributedshell ..... SUCCESS [    2.211 s]
[INFO] hadoop-yarn-applications-unmanaged-am-launcher ..... SUCCESS [    1.773 s]
[INFO] hadoop-yarn-site ..... SUCCESS [    0.033 s]
[INFO] hadoop-yarn-registry ..... SUCCESS [    4.927 s]
[INFO] hadoop-yarn-project ..... SUCCESS [    2.988 s]
[INFO] hadoop-mapreduce-client ..... SUCCESS [    0.052 s]
[INFO] hadoop-mapreduce-client-core ..... SUCCESS [ 22.567 s]
[INFO] hadoop-mapreduce-client-common ..... SUCCESS [ 16.208 s]
[INFO] hadoop-mapreduce-client-shuffle ..... SUCCESS [    4.122 s]
[INFO] hadoop-mapreduce-client-app ..... SUCCESS [    9.984 s]
[INFO] hadoop-mapreduce-client-hs ..... SUCCESS [    7.873 s]
[INFO] hadoop-mapreduce-client-jobclient ..... SUCCESS [ 31.351 s]
[INFO] hadoop-mapreduce-client-hs-plugins ..... SUCCESS [    1.769 s]
[INFO] Apache Hadoop MapReduce Examples ..... SUCCESS [    5.106 s]
[INFO] hadoop-mapreduce ..... SUCCESS [    2.335 s]
[INFO] Apache Hadoop MapReduce Streaming ..... SUCCESS [ 11.850 s]
[INFO] Apache Hadoop Distributed Copy ..... SUCCESS [ 33.553 s]
[INFO] Apache Hadoop Archives ..... SUCCESS [    1.983 s]
[INFO] Apache Hadoop Rumen ..... SUCCESS [    5.720 s]
[INFO] Apache Hadoop Gridmix ..... SUCCESS [    4.326 s]
[INFO] Apache Hadoop Data Join ..... SUCCESS [    2.572 s]
[INFO] Apache Hadoop Ant Tasks ..... SUCCESS [    1.878 s]
[INFO] Apache Hadoop Extras ..... SUCCESS [    2.936 s]
[INFO] Apache Hadoop Pipes ..... SUCCESS [    6.255 s]
[INFO] Apache Hadoop OpenStack support ..... SUCCESS [    4.872 s]
[INFO] Apache Hadoop Amazon Web Services support ..... SUCCESS [05:32 min]
[INFO] Apache Hadoop Client ..... SUCCESS [    4.333 s]
[INFO] Apache Hadoop Mini-Cluster ..... SUCCESS [    0.094 s]
[INFO] Apache Hadoop Scheduler Load Simulator ..... SUCCESS [    3.894 s]
[INFO] Apache Hadoop Tools Dist ..... SUCCESS [    6.618 s]
[INFO] Apache Hadoop Tools ..... SUCCESS [    0.047 s]
[INFO] Apache Hadoop Distribution ..... SUCCESS [ 13.077 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
```

```
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 01:01 h
[INFO] Finished at: 2016-12-31T01:43:51-08:00
[INFO] Final Memory: 93M/395M
[INFO] ------------------------------------------------------------------------
```

7、 编译成功之后，hadoop-2.6.5.tar.gz 位于/root/apps/hadoop-2.6.5-src/hadoop-dist/target 目录下，这是编译后该文件夹的状态：

```
[root@hadoop target]# pwd
/root/apps/hadoop-2.6.5-src/hadoop-dist/target
[root@hadoop target]# ll
total 538620
drwxr-xr-x. 2 root root      4096 Sep  3 11:19 antrun
drwxr-xr-x. 3 root root      4096 Sep  3 11:19 classes
-rw-r--r--. 1 root root      1867 Sep  3 11:19 dist-layout-stitching.sh
-rw-r--r--. 1 root root       640 Sep  3 11:19 dist-tar-stitching.sh
drwxr-xr-x. 9 root root      4096 Sep  3 11:19 hadoop-2.6.5
-rw-r--r--. 1 root root 183380862 Sep  3 11:19 hadoop-2.6.5.tar.gz
-rw-r--r--. 1 root root     26206 Sep  3 11:19 hadoop-dist-2.6.5.jar
-rw-r--r--. 1 root root 368044899 Sep  3 11:20 hadoop-dist-2.6.5-javadoc.jar
-rw-r--r--. 1 root root     23760 Sep  3 11:19 hadoop-dist-2.6.5-sources.jar
-rw-r--r--. 1 root root     23760 Sep  3 11:19 hadoop-dist-2.6.5-test-sources.jar
drwxr-xr-x. 2 root root      4096 Sep  3 11:20 javadoc-bundle-options
drwxr-xr-x. 2 root root      4096 Sep  3 11:19 maven-archiver
drwxr-xr-x. 3 root root      4096 Sep  3 11:19 maven-shared-archive-resources
drwxr-xr-x. 3 root root      4096 Sep  3 11:19 test-classes
drwxr-xr-x. 2 root root      4096 Sep  3 11:19 test-dir
[root@hadoop target]#
```

至此，大功告成。恭喜。