

在Cloudera CDH上部署Zeppelin和SparkR Interpreter

IE
.....
:S Apache Zeppelin是强大的在线notebook工具，和ipython的notebook相似，但是支持更多的interpreter，如Python、Spark、Hive等。

IE
.....
默认Cloudera CDH是没有包含Zeppelin组件的，如果要部署则须要重新编译安装。

JT
.....
本文记录了我在已经部署好的CDH集群中安装Zeppelin和SparkR相关支持的流程，仅做参考。

环境

IB
.....
本文所有操作都在Ubuntu 14.04 Trusty环境下执行，使用Cloudera CDH5.7.1版本作为Hadoop集群版本。

Q
.....
Hadoop 2.6.0-cdh5.7.1

N
.....
Cloudera Spark 1.6.0 自编译版本，增加了SparkR的支持。

IE
.....
编译环境可以在其他机器上完成，我编译使用的机器同样是Ubuntu 14.04 Trusty的一台虚拟机。

编译需要JDK、Git、Maven、Nodejs、npm

编译Zeppelin

确保JDK和Git已经安装好，我使用的是

```
$ git version  
git version 1.9.1
```

```
$ javac -version  
javac 1.8.0_91
```

安装maven 3.3.9，如果已有可以跳过，我使用apt-get install maven得到的版本太旧，所以重新通过源码安装一份

```
$ curl -OL http://mirror.olsnehost.net/pub/apache/maven/maven-3/3.3.9/binaries/apach  
$ tar -zxf apache-maven-3.3.9-bin.tar.gz -C /usr/local/  
$ ln -s /usr/local/apache-maven-3.3.9/bin/mvn /usr/bin/mvn  
$ mvn -v  
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08  
Maven home: /usr/local/apache-maven-3.3.9  
Java version: 1.8.0_91, vendor: Oracle Corporation  
Java home: /usr/lib/jvm/java-8-oracle/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "3.13.0-32-generic", arch: "amd64", family: "unix"
```

从Github repo上抓取Zeppelin的源码，并且切换到对应的版本的branch

```
$ git clone https://github.com/apache/incubator-zeppelin.git
$ git checkout branch-0.6 # 最新的release版本是0.6，所以我们也用这个版本
                           # 之前忘记更换branch，一直编译master的版本，各种出错
$ git pull                 # 确保代码是最新的
```

确定Hadoop和Spark的版本号，在CDH集群中的任意一台机器上登录

```
$ hadoop version
Hadoop 2.6.0-cdh5.7.1
Subversion http://github.com/cloudera/hadoop -r ae44a8970a3f0da58d82e0fc65275fff8dea
Compiled by jenkins on 2016-06-01T23:26Z
Compiled with protoc 2.5.0
From source with checksum 298b68dc3b308983f04cb37e8416f13
This command was run using /opt/cloudera/parcels/CDH-5.7.1-1.cdh5.7.1.p0.11/jars/had

$ spark-shell --version
Welcome to
```

```
  ____
 /  __/  _  _  _  _  _  _  _  _
 \  \  _  \  _  \  _  \  _  \
/_  _/  .  _/\  _/\  _/\  _/\  _  version 1.6.0
 /  _/
```

我这边的Hadoop版本号是2.6.0-cdh5.7.1，Spark版本号是1.6.0，于是所有都准备好了，可以开始编译啦。

每个编译的参数都可以在这里查看到，我们主要有几个参数需要注意：

- -Pbuild-distr：打包成可以发布的版本，包含压缩包
- -Psparkr：开启SparkR的支持
- -Pvendor-repo：使用Cloudera的第三方repo

```
$ mvn clean package -Pbuild-distr -Pyarn -Pspark-1.6 -Dspark.version=1.6.0 -Phadoop-
```

编译需要一定的时间，如果中途出错，可以根据日志，修正以后，通过命令-rf :step来继续编译，比如我在过程中因为网络原因，卡在了:zeppelin-web这个步骤，就可以使用mvn clean package -Pbuild-distr -Pyarn -Pspark-1.6 -Dspark.version=1.6.0 -Phadoop-2.6 -Dhadoop.version=2.6.0-cdh5.7.1 -Ppyspark -Psparkr -Pvendor-repo -DskipTests -rf :zeppelin-web继续刚才的编译。

编译好以后可以去zeppelin-distribution/target目录中看到zeppelin-0.6.1-SNAPSHOT.tar.gz这个压缩包（如果版本不同，文件名中的版本号则不同），将该文件传到CDH集群的某个节点上即可。

部署Zeppelin

将Zeppelin解压到指定的目录中

```
$ tar xzf zeppelin-0.6.1-SNAPSHOT.tar.gz -C /opt/
$ mv -r /opt/zeppelin-0.6.1-SNAPSHOT /opt/zeppelin
```

配置Zeppelin

```
$ mv /opt/zeppelin/conf /etc/zeppelin/conf
$ cd /opt/zeppelin
$ ln -s /etc/zeppelin/conf conf
$ cd /etc/zeppelin/conf
$ cp zeppelin-env.sh{.template,}
$ cp zeppelin-site.xml{.template,}
```

修改zeppelin-env.sh文件，包含以下内容：

```
export JAVA_HOME=/usr/java/jdk1.8.0_77
export MASTER=yarn-client
export ZEPPELIN_JAVA_OPTS="-Dmaster=yarn-client -Dspark.yarn.jar=/opt/zeppelin/inter
export DEFAULT_HADOOP_HOME=/opt/cloudera/parcels/CDH-5.7.1-1.cdh5.7.1.p0.11/lib/hado
export SPARK_HOME=/opt/cloudera/parcels/CDH-5.7.1-1.cdh5.7.1.p0.11/lib/spark
export HADOOP_HOME=${HADOOP_HOME:-$DEFAULT_HADOOP_HOME}
if [ -n "$HADOOP_HOME" ]; then
    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${HADOOP_HOME}/lib/native
fi
export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-/etc/hadoop/conf}

export ZEPPELIN_LOG_DIR=/var/log/zeppelin
export ZEPPELIN_PID_DIR=/var/run/zeppelin
export ZEPPELIN_WAR_TEMPDIR=/var/tmp/zeppelin
```

根据配置检查每个路径是否正确，然后新建对应的目录：

```
$ mkdir /var/log/zeppelin
$ mkdir /var/run/zeppelin
$ mkdir /var/tmp/zeppelin
```

为Zeppelin新建一个用户，并且处理相关的路径权限

```
$ useradd zeppelin

$ chown -R zeppelin:zeppelin /opt/zeppelin/notebook
$ chown zeppelin:zeppelin /etc/zeppelin/conf/interpreter.json
$ chown -R zeppelin:zeppelin /var/log/zeppelin
$ chown -R zeppelin:zeppelin /var/run/zeppelin
$ chown -R zeppelin:zeppelin /var/tmp/zeppelin

$ su hdfs
```

```
$ hadoop fs -mkdir /user/zeppelin # 为用户建立hdfs的目录
$ hadoop fs -chmod 777 /user/zeppelin
```

一切搞定，启动zeppelin

```
$ su zeppelin
$ cd /opt/zeppelin/
$ bin/zeppelin-daemon.sh start
```


启动完毕，浏览器打开启动节点的8080端口即可。

SparkR Interpreter

在使用SparkR之前，需要安装R语言的knitr库

```
$ R
$ install.packages('knitr', dependencies = TRUE)
```

打开Zeppelin的R Tutorial笔记，可以测试啦



The screenshot shows a Zeppelin notebook interface. At the top left is the title 'Hello R'. At the top right are icons for 'FINISHED', a play button, a zoom icon, a book icon, and a settings gear. The main area contains R code in a light blue font: `%r`, `foo <- TRUE`, `print(foo)`, `bare <- c(1, 2.5, 4)`, `print(bare)`, `double <- 15.0`, and `print(double)`. Below the code, the output is displayed in a light blue font: `[1] TRUE`, `[1] 1.0 2.5 4.0`, and `[1] 15`. At the bottom, a status message reads: 'Took a few seconds. Last updated by anonymous at July 15 2016, 10:23:11 AM.'

其他的测试运行中可能会有库不存在导致的错误，自行使用R的shell进行安装即可。

Free / 2016/7/15

Published under (CC) BY-NC-SA in categories [technology](#)