

HDFS+MapReduce+Hive+HBase 十分钟快速入门

目录

1.前言	1
2.Hadoop 家族.....	2
3.演练环境.....	2
3.1. 操作系统.....	3
3.2. Hadoop.....	3
3.3. Hive	3
3.4. Java	3
3.5. 机器	3
3.6. 安装列表.....	4
3.7. 安装路线.....	4
4.配置 ssh2 和修改/etc/hosts.....	4
4.1. 配置 ssh2	5
4.2.修改/etc/hosts	5
5.安装 Java.....	5
6.安装 Hadoop.....	6
7.配置 HDFS.....	6
7.1. 修改 hadoop-env.sh.....	6
7.2. 修改 core-site.xml.....	7
7.2.1.hadoop.tmp.dir	7
7.2.2.fs.default.name	7
7.3.修改 masters.....	8
7.4.修改 slaves.....	8
7.5.修改 hdfs-site.xml.....	9
8.配置 MapReduce	9
8.1.修改 mapred-default.xml.....	9
9.安装 Hive	10
9.1. 安装 ant	10
9.2. 安装 ivy	10
9.3. 编译 Hive	11
10.安装 HBase	11
11.体验	12

1.前言

本文的目的是让一个从未接触 Hadoop 的人，在很短的时间内快速上手，掌握编译、安装和

简单的使用。

2.Hadoop 家族

整个 Hadoop 家族由以下几个子项目组成：

成员名 用途

Hadoop Common Hadoop 体系最底层的一个模块，为 Hadoop 各子项目提供各种工具，如：配置文件和日志操作等。

Avro Avro 是 doug cutting 主持的 RPC 项目，有点类似 Google 的 protobuf 和 Facebook 的 thrift。avro 用来做以后 hadoop 的 RPC，使 hadoop 的 RPC 模块通信速度更快、数据结构更紧凑。

Chukwa Chukwa 是基于 Hadoop 的大集群监控系统，由 yahoo 贡献。

HBase 基于 Hadoop Distributed File System，是一个开源的，基于列存储模型的分布式数据库。

HDFS 分布式文件系统

Hive hive 类似 CloudBase，也是基于 hadoop 分布式计算平台上的提供 data warehouse 的 sql 功能的一套软件。使得存储在 hadoop 里面的海量数据的汇总，即席查询简单化。hive 提供了一套 QL 的查询语言，以 sql 为基础，使用起来很方便。

MapReduce 实现了 MapReduce 编程框架

Pig Pig 是 SQL-like 语言，是在 MapReduce 上构建的一种高级查询语言，把一些运算编译进 MapReduce 模型的 Map 和 Reduce 中，并且用户可以定义自己的功能。Yahoo 网络运算部门开发的又一个克隆 Google 的项目 Sawzall。

ZooKeeper Zookeeper 是 Google 的 Chubby 一个开源的实现。它是一个针对大型分布式系统的可靠协调系统，提供的功能包括：配置维护、名字服务、分布式同步、组服务等。ZooKeeper 的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。

3.演练环境

3.1. 操作系统

SuSE10, Linux2.6.16, 32 位版本。

3.2. Hadoop

Hive 通过 SVN 直接从网上下载原代码, 其它采用 `hadoop-0.20.0.tar.gz` 二进制安装包, 所以只需要编译 Hive。

另外, 请注意不要下载 2009-8-19 号的 Hive, 2009-8-19 提交的 Hive 版本在 `hadoop-0.20.0.tar.gz` 上编译通不过, 原因是:

Hive 源代码根目录 `\ql\src\java\org\apache\hadoop\hive\ql\exec\FileSinkOperator.java` 文件中多了一行:

```
import org.apache.hadoop.io.compress.LzoCodec;
```

而应当下载 2009-8-4 号和 2009-8-19 之间的版本, 或 2009-8-19 之后的版本。

3.3. Hive

在 0.20.0 版本的 Hadoop 二进制安装包中, 并没有集成 Hive, 所以 Hive 需要自己编译, 编译时需要使用到 `ant` 和 `ivy` 两个工具, 而且依赖于 Hadoop。

3.4. Java

运行 Hadoop 和编译 Hive, 都需要使用到 Java, 所以需要第一个安装的是 Java。

3.5. 机器

演练时, 在 3 台机器上搭建 Hadoop 集群, 假设这 3 台机器如下表所示:

机器名	机器 IP	用途
-----	-------	----

Hadoop-A 192.168.0.1 用作 Master

Hadoop-B 192.168.0.2 用作 Slave

Hadoop-C 192.168.0.3 用作 Slave

注意事项：机器名中不能包括下划线“_”，但可以包括横线。另可使用不带参数的 `hostname` 来查看机器名，修改机器名方法为：`hostname 机器名`，如：`hostname Hadoop-A`。另外，集群内所有的机器登录用户名和密码要保持相同。

3.6. 安装列表

安装 说明

Java 1.6 A、B 和 C 三台机器都需要安装

Hadoop 0.20.0 A、B 和 C 三台机器都需要安装

Hive 只需要在 B 和 C 两台机器上安装，也就是 Slave 机器不需要安装

ant 只需要在 B 和 C 两台机器上安装，也就是 Slave 机器不需要安装

ivy 只需要在 B 和 C 两台机器上安装，也就是 Slave 机器不需要安装

注意事项：为求简单所有操作均以 `root` 用户完成，实际上可以使用非 `root` 用户，但环境变量就不能直接在 `/etc/profile` 文件中添加了。

3.7. 安装路线

4.配置 ssh2 和修改/etc/hosts

需要在所有机器上配置 ssh2 和修改 `/etc/hosts`。

4.1. 配置 ssh2

1)mkdir ~/.ssh2 # 如果~/.ssh2 目录已经存在，则这一步可以跳过

2)cd ~/.ssh2

3)ssh-keygen2 -t rsa

4)echo "IdKey id_rsa_2048_a" > identification

5)echo "key id_rsa_2048_a.pub" > authorization

4.2.修改/etc/hosts

1)在/etc/hosts 文件中增加如下三行：

192.168.0.1 Hadoop-A

192.168.0.2 Hadoop-B

192.168.0.3 Hadoop-C

并根据上述关系，使用 hostname 命令修改三台机器的机器名。

5.安装 Java

1)从 http://www.java.com/zh_CN/download/manual.jsp 下载 jre，请注意是 Java1.6 版本，原因是 Hadoop 和 Hive 使用到了模板等特性

2)下载“Linux（自解压文件）”这个版本的 jre，如下图所示

3)将下载的 Java 二进制安装包 jre-6u15-linux-i586.bin 上传到/usr/local 目录

4)为 jre-6u15-linux-i586.bin 文件增加可执行权限：chmod +x jre-6u15-linux-i586.bin

5)运行 jre-6u15-linux-i586.bin：./jre-6u15-linux-i586.bin

6)用户协议接受，选择 yes

7)安装成功后，在/etc/profile 增加如下一行：

```
export PATH=/usr/local/jdk/bin:$PATH
```

6.安装 Hadoop

1) 从 <http://hadoop.apache.org/common/releases.html> 下载 Hadoop 二进制安装包 hadoop-0.20.0.tar.gz

2)将 hadoop-0.20.0.tar.gz 上传到/usr/local 目录

3)在/usr/local 目录，将 hadoop-0.20.0.tar.gz 解压

4)为 hadoop-0.20.0 建一个软链接：ln -s hadoop-0.20.0 hadoop

5)在/etc/profile 文件中，增加如下两行：

```
export HADOOP_HOME=/usr/local/hadoop
```

```
export PATH=$HADOOP_HOME/bin:$PATH
```

完成以上步骤后，Hadoop 就安装好了。

7.配置 HDFS

7.1. 修改 hadoop-env.sh

所有机器做同样操作。

hadoop-env.sh 位于 /usr/local/hadoop/conf 目录下，需要在这个文件中设置环境变量 JAVA_HOME，增加如下一行即可：

```
export JAVA_HOME=/usr/local/jre
```

7.2. 修改 core-site.xml

所有机器做同样操作。

在 `hadoop-0.20.0.tar.gz` 安装包中，`conf` 目录下的 `core-site.xml` 默认是空的，但 `src/core` 目录下有一个缺省的配置文件 `core-default.xml`，所以只需要将它复制到 `conf` 目录下，并重命名为 `core-site.xml` 即可，下述操作都是基于此操作的。

`core-site.xml` 文件位于 `/usr/local/hadoop/conf` 目录下。

7.2.1.hadoop.tmp.dir

设置临时文件目录参数 `hadoop.tmp.dir`，默认情况下 `master` 会将元数据等存在这个目录下，而 `slave` 会将所有上传的文件放在这个目录下，在这里数据目录统一为 `/usr/local/hadoop/data`：

```
<property>

<name>hadoop.tmp.dir</name>

<value>/usr/local/hadoop/data</value>

<description>A base for other temporary directories.</description>

</property>
```

注意事项：由于上传到 Hadoop 的所有文件都会被存放在 `hadoop.tmp.dir` 所指定的目录，所以要确保这个目录是足够大的。

7.2.2.fs.default.name

`master` 用这个参数，提供基于 `http` 协议的状态上报界面，而 `slave` 通过这个地址连接 `master`，设置如下：

```
<property>

<name>fs.default.name</name>

<value>hdfs://Hadoop-A:54310</value>
```

<description>The name of the default file system. A URI whose

scheme and authority determine the FileSystem implementation. The

uri's scheme determines the config property (fs.SCHEME.impl) naming

the FileSystem implementation class. The uri's authority is used to

determine the host, port, etc. for a filesystem.</description>

</property>

备注：fs.default.name 指定 NameNode 的 IP 地址和端口号,缺省值是 file:///, 表示使用本地文件系统, 用于单机非分布式模式。

dfs.replication 指定 HDFS 中每 Block 被复制的次数, 起数据冗余备份的作用。在典型的生产系统中, 这个数常常设置为 3。

7.3.修改 masters

这一步只需要在 master 节点上执行。

这个文件存储 master 节点的 IP 或机器名, 建议使用机器名, 每行一个机器名。主 master 会将元数据等备份到备 master 节点, 文件位于 conf 目录下。

这里, 我们为 masters 文件增加如下一行即可:

Hadoop-A

7.4.修改 slaves

这一步也只需要在 master 节点上执行, 以便 master 以 ssh2 方式去启动所有的 slave 节点。

这个文件存储 slave 节点的 IP 或机器名, 建议使用机器名, 每行一个机器名。这里我们增加如下两行:

Hadoop-B

Hadoop-C

7.5.修改 hdfs-site.xml

所有机器做同样操作。

从 src/hdfs 目录下，将 hdfs-default.xml 复制到 conf 目录，并重命名成 hdfs-site.xml，这里不需要修改此文件。

8.配置 MapReduce

8.1.修改 mapred-default.xml

所有机器做同样操作。

从 src/mapred 目录下，将 mapred-default.xml 复制到 conf 目录，并重命名成 mapred-site.xml。

```
<property>
```

```
<name>mapred.job.tracker</name>
```

```
<value>Hadoop-A: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>
```

备注：mapred.job.tracker 指定 JobTracker 的 IP 地址和端口号。缺省值是 local，表示在本地同一 Java 进程内执行 JobTracker 和 TaskTracker，用于单机非分布式模式。

9.安装 Hive

hadoop-0.20.0 中并没有集成二进制的 Hive，所以需要通过源代码编译一个，但是 2009-8-19 日提交的 Hive 并不能在 hadoop-0.20.0 上编译，而应当使用 2009-8-4 之后和 2009-8-19 之间的版本。

9.1. 安装 ant

1)从 <http://ant.apache.org/>下载 ant 二进制安装包，选择 1.7.1 版本

2)下载后，将包 apache-ant-1.7.1-bin.zip 上传到/usr/local 目录

3)进入/usr/local 目录，将 apache-ant-1.7.1-bin.zip 解压：unzip apache-ant-1.7.1-bin.zip

4)进入/usr/local 目录，为 ant 建一个软连接：ln -s apache-ant-1.7.1 ant

5)修改/etc/profile，增加如下行：

```
export PATH=/usr/local/ant/bin:$PATH
```

至此，ant 算是安装好了 。

9.2. 安装 ivy

1)从 <http://www.apache.org/dist/ant/ivy/>下载 ivy 二进制安装包，选择 2.1.0-rc2 版本

2)下载后，将包 apache-ivy-2.1.0-rc2-bin.tar.gz 上传到/usr/local 目录

3) 进 入 /usr/local 目 录 ， 将 apache-ivy-2.1.0-rc2-bin.tar.gz 解 压 ： tar xzf apache-ivy-2.1.0-rc2-bin.tar.gz

4)进入/usr/local 目录，为 ivy 建一个软连接：ln -s apache-ivy-2.1.0-rc2 ivy

6)修改/etc/profile，增加如下行：

```
export IVY_HOME=/usr/local/ivy
```

至此，ivy 算是安装好了。

9.3. 编译 Hive

在编译 Hive 之前，请确保 HADOOP_HOME 和 IVY_HOME 两个环境变量已经生效。

1)使用 svn 从 <http://svn.apache.org/repos/asf/hadoop/hive/trunk> 下载 Hive 源代码

2)将下载来的 Hive 源代码打包，然后上传到 Hadoop-A 机器

3)解压 Hive 源代码包

4)修改 shims/ivy.xml:

只保留 0.20.0 的配置，否则编译会出错，将原 ivy.xml 文件:

修改成如下:

5)运行 ant 开始编译:

```
ant -Dtarget.dir=/usr/local/hadoop/hive -Dhadoop.version=0.20.0 package
```

这步完成之后，Hive 会被安装到 /usr/local/hadoop/hive 目录下

6)添加 Hive 环境变量，在 /etc/profile 文件中增加如下两行:

```
export HIVE_HOME=/usr/local/hadoop/hive
```

```
export PATH=$HIVE_HOME/bin:$PATH
```

10. 安装 HBase

1)从 <http://svn.apache.org/repos/asf/hadoop/hbase/trunk> 下载最新的 HBase 源代码

2)将 HBase 源代码打包，并上传到 Linux 上

3)解压 HBase 源代码包

4)编译 HBase:

`ant -Dtarget.dir=/usr/local/hadoop/hbase -Dhadoop.version=0.20.0 package`

5)编译成功之后，HBase 可能并不象 Hive 一样自动安装到/usr/local/hadoop/hbase 目录下，这个时候需要手工复制到 HBase 安装目录下：将 build/hbase-0.21.0-dev 整个目录复制到 /usr/local/hadoop 目录下，并将 hbase-0.21.0-dev 重命名成 hbase 即可

6)进入 /usr/local/hadoop/hbase/conf 目录，将 hbase-default.xml 复制一份，并命名成 hbase-site.xml

7)修改 hbase-site.xml:

设置 hbase.rootdir 的值为: `hdfs://Hadoop-A:54310/hbase;`

设置 hbase.master (hbase.master 可能为 hbase.master.port) 的值为: `Hadoop-A:60000`

8)修改 hbase-env.sh:

设置环境变量 JAVA_HOME: `export JAVA_HOME=/usr/local/jre`

9)在 Master 节点，还需要修改 regionserver，在这个文件中列出所有的 slave 机器，一行一个机器名:

Hadoop-B

Hadoop-C

这一步不用在 slave 节点上操作。

10)通过以上操作，HBase 已经安装和配置好，然后应当打包，将它部署到集群的所有节点上

11.体验

11.1.启动和停止

11.1.1.hadoop

在启动 Hadoop 集群之前，需要先格式化，在 master 节点上执行下面的命令即可:

`hadoop namenode -format`

11.1.2.start-all.sh

这个脚本用来启动 Hadoop。

可以通过 `http://172.25.38.127:50070` 来查看 HDFS 的启动情况。

可以通过 `http://172.25.38.127:50030` 来查看 MapReduce 的启动情况。

11.1.3.stop-all.sh

这个脚本用来停止 Hadoop。

11.2.体验 HDFS

HDFS 的使用和普通的 Linux 命令差不多，只不过各类操作都必须作为 `hadoop` 命令的参数，如在 `hadoop` 上执行 `ls` 操作:

`hadoop fs -ls /`

这条命令相当于 Linux 下的 `ls /`。

11.3.体验 MapReduce

体验 MapReduce，可以使用 Hadoop 自带的 WordCount，如:

`hadoop jar wordcount.jar /x/x /z`

其中 `wordcount.jar` 是 WordCount 的可执行包，`/x/x` 是源文件，是一段以逗号分隔的英文片断，而 `/z` 是结果存放的目录。

11.4.体验 Hive

Hive 的使用非常简单，照着 <http://wiki.apache.org/hadoop/Hive/GettingStarted> 上说的来操作就可以了。

12.FAQ

12.1.如何查看 Hadoop 进程

如果安装了 JDK，则在 JDK 的 bin 目录下有一个 jps 命令，可以用来查看 java 进程，如：

```
# jps
```

```
27612 NameNode
```

```
17369 Jps
```

```
16206 HQuorumPeer
```

```
15423 HMaster
```

```
27761 SecondaryNameNode
```

```
27839 JobTracker
```

其中，第一列为进程号，第二列为进程名称。

12.2.ssh 端口问题

如果 ssh 不是使用默认端口，则需要修改 `hadoop-env.sh` 文件中的 `HADOOP_SSH_OPTS` 环境变量，假设 ssh 端口号为 8000，则可以简单设置为：`export`

```
HADOOP_SSH_OPTS="-p 8000"
```

如果安装了 HBase，还应当修改 `hbase-env.sh` 文件中的 `HBASE_SSH_OPTS`。

12.3.首次 ssh 登录问题

首次通过 ssh 登录另一台机器时，可能会遇到一个 yes 确认过程，因此在启动之前，应当先手工或使用其它脚本 ssh 成功登录一次，否则容易遇到如下错误：

```
r# ./start-hbase.sh
```

```
DOSS38-127-sles10: Host key not found from database.
```

```
DOSS38-127-sles10: Key fingerprint:
```

```
DOSS38-127-sles10: xuror-ledab-buhim-zohok-tanop-cyrig-tysac-gyhyp-refan-semim-pyxex
```

```
DOSS38-127-sles10: You can get a public key's fingerprint by running
```

```
DOSS38-127-sles10: % ssh-keygen -F publickey.pub
```

```
DOSS38-127-sles10: on the keyfile.
```

```
DOSS38-127-sles10: warning: tcgetattr failed in ssh_rl_set_tty_modes_for_fd: fd 1: Invalid argument
```

本文转载自：<http://baiyunl.javaeye.com/blog/626117>