Hadoop 部署、配置与运行

扉言:此文档为自己部署过程中的记录。配置后演示了单节点、单机伪分布和两台机器之间的分布运行、并对伪分布和完全分布做了初步对比以增进理解，最后演示了在 eclipse 下运行 hadoop 自带例子 wordcount 的步骤。

==========

系统配置

==========

（一）资源需求

- **Linux Ubuntu 9.10**

  //最新版本可上官方网站免费下载 www.ubuntulinux.org

  //也可以向 Ubuntu 社区申请免费安装 shipit.ubuntu.com

- **Hadoop 0.20.0** 包

  //最新版本可在 Apache 提供的镜像服务器下载

  //www.apache.org → download → 镜像服务器 →hadoop

- **Sun-java6-jdk** 包

  //在终端机里输入:apt-get install sun-java6-jdk

  //系统会自动下载包以及所有的依存包,同时进行包的安装

- **SSH** 包(为远程登录会话提供安全性协议)

  //在终端机里输入:apt-get install ssh

- **Eclipse** 包

  //官方下载最新版本:www.eclipse.org/downloads/

（二）配置流程

1. 安装 ubuntu 9.04
2. 更新 deb 软件包列表

   ```
   $ sudo apt-get update
   ```

3. 安装系统更新

   ```
   $ sudo apt-get upgrade
   ```

4. 安装 JDK

```
$ sudo apt-get install sun-java6-jdk
                    //默认路径在/usr/lib/jvm,安装时需要 TAB 键选择 OK
```

5. 设置 java-6-sun 为默认的 java 程序

```
$ sudo update-alternatives --config java   //JDK 唯一,不需选择
$ sudo update-java-alternatives -s java-6-sun
```

6. 设置 CLASSPATH 和 JAVA_HOME 系统环境变量

```
$ sudo gedit /etc/environment
```

添加以下两行内容:

```
CLASSPATH=".:/usr/lib/jvm/java-6-sun/lib"
JAVA_HOME="/usr/lib/jvm/java-6-sun"
```

7. 调整系统虚拟机的优先顺序

```
$ sudo gedit /etc/jvm
```

在文件顶部添加一行

```
/usr/lib/jvm/java-6-sun
```

如果文件/etc/jvm 不存在则自己新建

8. 多节点分布式环境下的两个必要条件

a、每个节点有相同的用户名,如 shiep205

b、hadoop 文件路径相同,如/home/shiep205/hadoop

9. 下载 hadoop-*.tar.gz 至 /home/shiep205/

```
$ cd ~                                          //选择默认路径
$ sudo tar xzf hadoop-0.20.0.tar.gz             //解压至当前路径
$ mv hadoop-0.20.0 hadoop                       //重命名为 hadoop
$ sudo chown -R shiep205:shiep205 hadoop
                                                //赋予 shiep205 权限
```

10. 更新 hadoop 环境变量

```
$ gedit hadoop/conf/hadoop-env.sh
```

将 #export JAVA_HOME=/usr/lib/jvm/java-6-sun

改为 export JAVA_HOME=/usr/lib/jvm/java-6-sun

11. 配置 SSH

```
$ sudo apt-get install ssh
$ sudo apt-get install rsync  //远程同步,可能已经安装了最新版本
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
```

```
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
$ ssh localhost                    //验证配置成功与否
```

==========

单节点配置

==========

在前面工作已经做好的基础上,单节点的运行,运行在非分布模式,hadoop 作为单个 java 进程。运行命令,查看 hadoop 的使用文档

Bin/hadoop

以下例子复制压缩的 conf 目录作为输入,查找并显示正规式的匹配。输出写到 output 目录

```
$ mkdir input
$ cp conf/*.xml input
$ bin/hadoop jar hadoop-*-examples.jar grep input output
'dfs[a-z.]+'
$ cat output/*
```

==========

单机伪分布

==========

伪分布运行模式是在运行在单个机器之上,每一个 hadoop 的守护进程为一个单独的 java 进程。

(一)配置三个文件

conf/core-site.xml:

```
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

conf/hdfs-site.xml:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

conf/mapred-site.xml:

```
<configuration>
    <property>
        <name>mapred.job.tracker</name>
      <value>localhost:9001</value>
    </property>
</configuration>
```

(二)格式化 HDFS

进入 hadoop 的 bin 目录,运行命令:

```
$ sudo bin/hadoop namenode -format
10/02/21 00:15:08 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 0.20.0
STARTUP_MSG:   build =
https://svn.apache.org/repos/asf/hadoop/core/branches/branch-0.20
-r 763504; compiled by 'ndaley' on Thu Apr  9 05:18:40 UTC 2009
************************************************************/
10/02/21 00:15:09 INFO namenode.FSNamesystem: fsOwner=root,root
10/02/21 00:15:09 INFO namenode.FSNamesystem:
supergroup=supergroup
10/02/21 00:15:09 INFO namenode.FSNamesystem:
isPermissionEnabled=true
10/02/21 00:15:09 INFO common.Storage: Image file of size 94
saved in 0 seconds.
10/02/21 00:15:09 INFO common.Storage: Storage directory
/tmp/hadoop-root/dfs/name has been successfully formatted.
10/02/21 00:15:09 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master/127.0.1.1
************************************************************/
```

（三）启动 hadoop 监护进程

命令 `$ bin/start-all.sh`

```
starting namenode, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-namenode-
master.out
localhost: starting datanode, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-datanode-
```

```
master.out
localhost: starting secondarynamenode, logging to /home/shiep205/
hadoop/bin/../logs/hadoop-shiep205-secondarynamenode-master.out
starting jobtracker, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-jobtracker-
master.out
localhost: starting tasktracker, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-tasktracker-
master.out
```

（四）复制输入文件到 HDFS

命令:$ bin/hadoop dfs -put conf input

//在HDFS下创建input目录，将hadoop/conf下的文件上传到input下

//可以通过 bin/hadoop dfs -ls input 查看文件夹中的内容

（五）运行例子

命令:$ bin/hadoop jar hadoop-*-examples.jar grep input output 'dfs[a-z].+'

```
10/02/21 00:06:13 INFO mapred.FileInputFormat: Total input paths
to process : 19
10/02/21 00:06:13 INFO mapred.JobClient: Running job:
job_201002202351_0001
10/02/21 00:06:14 INFO mapred.JobClient:  map 0% reduce 0%
10/02/21 00:06:27 INFO mapred.JobClient:  map 10% reduce 0%
10/02/21 00:06:33 INFO mapred.JobClient:  map 21% reduce 0%
10/02/21 00:06:36 INFO mapred.JobClient:  map 31% reduce 7%
10/02/21 00:06:39 INFO mapred.JobClient:  map 42% reduce 7%
10/02/21 00:06:42 INFO mapred.JobClient:  map 52% reduce 7%
10/02/21 00:06:45 INFO mapred.JobClient:  map 63% reduce 10%
10/02/21 00:06:48 INFO mapred.JobClient:  map 73% reduce 10%
10/02/21 00:06:51 INFO mapred.JobClient:  map 84% reduce 17%
10/02/21 00:06:54 INFO mapred.JobClient:  map 94% reduce 17%
```

```
10/02/21 00:06:57 INFO mapred.JobClient:  map 100% reduce 31%
10/02/21 00:07:06 INFO mapred.JobClient:  map 100% reduce 100%
10/02/21 00:07:08 INFO mapred.JobClient: Job complete:
job_201002202351_0001
10/02/21 00:07:08 INFO mapred.JobClient: Counters: 18
10/02/21 00:07:08 INFO mapred.JobClient:   Job Counters
10/02/21 00:07:08 INFO mapred.JobClient:     Launched reduce
tasks=1
10/02/21 00:07:08 INFO mapred.JobClient:     Launched map
tasks=19
10/02/21 00:07:08 INFO mapred.JobClient:     Data-local map
tasks=19
10/02/21 00:07:08 INFO mapred.JobClient:   FileSystemCounters
10/02/21 00:07:08 INFO mapred.JobClient:     FILE_BYTES_READ=114
10/02/21 00:07:08 INFO mapred.JobClient:
HDFS_BYTES_READ=23954
10/02/21 00:07:08 INFO mapred.JobClient:
FILE_BYTES_WRITTEN=944
10/02/21 00:07:08 INFO mapred.JobClient:
HDFS_BYTES_WRITTEN=206
10/02/21 00:07:08 INFO mapred.JobClient:   Map-Reduce Framework
10/02/21 00:07:08 INFO mapred.JobClient:     Reduce input
groups=2
10/02/21 00:07:08 INFO mapred.JobClient:     Combine output
records=2
10/02/21 00:07:08 INFO mapred.JobClient:     Map input
records=709
10/02/21 00:07:08 INFO mapred.JobClient:     Reduce shuffle
bytes=222
10/02/21 00:07:08 INFO mapred.JobClient:     Reduce output
records=2
```

```
10/02/21 00:07:08 INFO mapred.JobClient:     Spilled Records=4
10/02/21 00:07:08 INFO mapred.JobClient:     Map output bytes=104
10/02/21 00:07:08 INFO mapred.JobClient:     Map input
bytes=23954
10/02/21 00:07:08 INFO mapred.JobClient:     Combine input
records=2
10/02/21 00:07:08 INFO mapred.JobClient:     Map output records=2
10/02/21 00:07:08 INFO mapred.JobClient:     Reduce input
records=2
10/02/21 00:07:08 WARN mapred.JobClient: Use GenericOptionsParser
for parsing the arguments. Applications should implement Tool for
the same.
10/02/21 00:07:08 INFO mapred.FileInputFormat: Total input paths
to process : 1
10/02/21 00:07:09 INFO mapred.JobClient: Running job:
job_201002202351_0002
10/02/21 00:07:10 INFO mapred.JobClient:  map 0% reduce 0%
10/02/21 00:07:18 INFO mapred.JobClient:  map 100% reduce 0%
10/02/21 00:07:30 INFO mapred.JobClient:  map 100% reduce 100%
10/02/21 00:07:32 INFO mapred.JobClient: Job complete:
job_201002202351_0002
10/02/21 00:07:32 INFO mapred.JobClient: Counters: 18
10/02/21 00:07:32 INFO mapred.JobClient:   Job Counters
10/02/21 00:07:32 INFO mapred.JobClient:     Launched reduce
tasks=1
10/02/21 00:07:32 INFO mapred.JobClient:     Launched map tasks=1
10/02/21 00:07:32 INFO mapred.JobClient:     Data-local map
tasks=1
10/02/21 00:07:32 INFO mapred.JobClient:   FileSystemCounters
10/02/21 00:07:32 INFO mapred.JobClient:     FILE_BYTES_READ=114
10/02/21 00:07:32 INFO mapred.JobClient:     HDFS_BYTES_READ=206
```

```
10/02/21 00:07:32 INFO mapred.JobClient:
FILE_BYTES_WRITTEN=260
10/02/21 00:07:32 INFO mapred.JobClient:
HDFS_BYTES_WRITTEN=92
10/02/21 00:07:32 INFO mapred.JobClient:   Map-Reduce Framework
10/02/21 00:07:32 INFO mapred.JobClient:     Reduce input
groups=1
10/02/21 00:07:32 INFO mapred.JobClient:     Combine output
records=0
10/02/21 00:07:32 INFO mapred.JobClient:     Map input records=2
10/02/21 00:07:32 INFO mapred.JobClient:     Reduce shuffle
bytes=114
10/02/21 00:07:32 INFO mapred.JobClient:     Reduce output
records=2
10/02/21 00:07:32 INFO mapred.JobClient:     Spilled Records=4
10/02/21 00:07:32 INFO mapred.JobClient:     Map output bytes=104
10/02/21 00:07:32 INFO mapred.JobClient:     Map input bytes=120
10/02/21 00:07:32 INFO mapred.JobClient:     Combine input
records=0
10/02/21 00:07:32 INFO mapred.JobClient:     Map output records=2
10/02/21 00:07:32 INFO mapred.JobClient:     Reduce input
records=2
```

（六）将文件输出

1、将输出文件从分布式文件系统拷贝到本地文件系统查看

```
$ bin/hadoop dfs -get output output
$ cat output/*
```

```
cat: output/_logs: 是一个目录
1  dfsadmin and mradmin commands to refresh the security policy
in-effect.
1  dfsmetrics.log
```

2、在分布式文件系统上查看输出文件

```
$ bin/hadoop fs -cat output/part-*
1  dfsadmin and mradmin commands to refresh the security policy
in-effect.
1  dfsmetrics.log
```

（七）停止 hadoop 系统

```
$ bin/stop-all.sh
stopping jobtracker
localhost: stopping tasktracker
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode
```

==========

两台机器间的分布实现

==========

前提:已完成单节点配置

(一)系统规划

Node	User	IP address	备注
Namenode	shiep205	192.168.0.154	NameNode和JobTracker为同一台主机
Jobtracker	shiep205	192.168.0.154	
Datanode	shiep205	192.168.0.136	

(二)修改 hosts,将 IP 与主机名对应上(ifconfig 命令查看 IP)

```
$ sudo gedit /etc/hosts
```

添加两行数据

```
192.168.0.154 master        //保证一个主机名对应一个 IP
192.168.0.136 slave
```

(三)配置 ssh(保证 masters 无需密码可 SSH 到 slaves)

i) 在所有 **slave** 节点上执行命令:

scp 远程用户名@IP 地址:文件名 1 本地用户名@IP 地址:文件名 2

```
$ scp shiep205@NameNodeIP:/home/shiep205/.ssh/id_dsa.pub /home/shiep205/.ssh/IP1_dsa.pub
$ scp shiep205@JobTrackerIP:/home/shiep205/.ssh/id_dsa.pub /home/shiep205/.ssh/IP2_dsa.pub
$ cat ~/.ssh/IP1_dsa.pub >> ~/.ssh/authorized_keys
$ cat ~/.ssh/IP2_dsa.pub >> ~/.ssh/authorized_keys
```

此例中 192.168.0.154 是 NameNode 的 IP

```
$ scp shiep205@192.168.0.154:/home/shiep205/.ssh/id_dsa.pub /home/shiep205/.ssh/154_dsa.pub
        //该命令将 NameNode 上的公钥远程拷贝到本地,并更名为 154_dsa.pub
shiep205@192.168.0.154's password:
id_dsa.pub                    100%  615     0.6KB/s   00:00
```

将 NameNode 的公钥加入到受信列表:

```
$ cat ~/.ssh/154_dsa.pub >> ~/.ssh/authorized_keys
```

在 JobTracker 上执行命令:

ii)在 JobTracker 上执行命令:

```
//因为NameNode要在JobTracker上启动 SecondaryNameNode
$ scp shiep205@NameNodeIP:/home/shiep205/.ssh/id_dsa.pub /home/shiep205/.ssh/IP_dsa.pub
$ cat ~/.ssh/IP1_dsa.pub >> ~/.ssh/authorized_keys
//本例中,由于 JobTracker 和 NameNode 为同一台主机,所以等
//于将自己的id_dsa.pub 追加到自己的 authorized_keys 中。
$ scp shiep205@192.168.0.154:/home/shiep205/.ssh/id_dsa.pub /home/shiep205/.ssh/154_dsa.pub
$ cat ~/.ssh/154_dsa.pub >> ~/.ssh/authorized_keys
```

(四)配置 conf/masters 、conf/slaves

在所有节点上:

在<HADOOP_INSTALL>/conf/masters 中加入 NameNode IP、Jobtracker IP

在<HADOOP_INSTALL>/conf/slaves 中加入 slaveIPs

(五)配置 core-site.xml、hdfs-site.xml、mapred-site.xml

conf/core-site.xml:

```
<configuration>
   <property>
      <name>fs.default.name</name>
      <value>hdfs://NameNodeIP:9000</value>
   </property>
</configuration>
```

conf/hdfs-site.xml:

```
<configuration>
   <property>
      <name>dfs.replication</name>
      <value>1</value>
   </property>
```

```
        <property>
            <name>dfs.name.dir</name>
            <value>/home/shiep205/hdfs/name</value>
        </property>
        <property>
            <name>dfs.data.dir</name>
            <value>/home/shiep205/hdfs/data</value>
        </property>
    </configuration>
```

conf/mapred-site.xml:

```
    <configuration>
        <property>
            <name>mapred.job.tracker</name>
            <value>JobTrackerIP:9001</value>
        </property>
    </configuration>
```

（六）运行

1）格式化分布式文件系统,在 NameNode 上

```
                                    //一定要在 NameNode 上格式化 HDFS
    $ sudo bin/hadoop namenode -format
    10/02/17 22:11:24 INFO namenode.NameNode: STARTUP_MSG:
    /************************************************************
    STARTUP_MSG: Starting NameNode

  STARTUP_MSG:    host = master/192.168.0.154
  STARTUP_MSG:    args = [-format]
  STARTUP_MSG:    version = 0.20.0
  STARTUP_MSG:    build =
https://svn.apache.org/repos/asf/hadoop/
  core/branches/branch-0.20 -r 763504; compiled by 'ndaley' on
```

```
Thu Apr   9 05:18:40 UTC 2009
************************************************************/
Re-format filesystem in /home/shiep205/hdfs/name ? (Y or N) y
Format aborted in /home/shiep205/hdfs/name
10/02/17 22:11:30 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.0.154
************************************************************/
```

2) 启动 HDFS,在 NameNode 上:

```
$ bin/start-dfs.sh
```

//该命令将访问 NameNode 上的 conf/slaves 文件,在本机上启动 NameNode,在本机和 JobTracker 上启动 SecondaryNameNode,在 conf/slaves 文件里的所有主机上启动 DataNode

```
starting namenode, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-namenode-
master.out
192.168.0.136: starting datanode, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-datanode-
slave.out
192.168.0.154: starting secondarynamenode, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-
secondarynamenode-master.out
```

3) 启动 Map-Reduce,在 JobTracker 上:

```
$ bin/start-mapred.sh
```

//该命令将访问 JobTracker 上的 conf/slaves 文件,在本机上启动 //Jobtracker, 在 conf/slaves 文件里的所有主机上启动 TaskTracker

```
starting jobtracker, logging to
/home/shiep205/hadoop/bin/../logs/hadoop-shiep205-jobtracker-
master.out
192.168.0.136: starting tasktracker, logging to
```

```
/home/shiep205/
   hadoop/bin/../logs/hadoop-shiep205-tasktracker-slave.out
```

（七）运行例子

在/home/shiep205 文件夹下新建一文件 a,里面输入若干单词。

1)将本地文件 a 上传到 hdfs:

```
$ bin/hadoop dfs -put ~/a test1/a
                    //删除命令为 $ bin/hadoop dfs -rmr test1
```

2)执行例子

```
$ bin/hadoop jar hadoop-*-examples.jar wordcount test1/ test2/
```

```
   10/02/17 22:24:24 INFO input.FileInputFormat: Total input
paths
   to process : 1
   10/02/17 22:24:25 INFO mapred.JobClient: Running job:
   job_201002172218_0002
   10/02/17 22:24:26 INFO mapred.JobClient:   map 0% reduce 0%
   10/02/17 22:24:42 INFO mapred.JobClient:   map 100% reduce 0%
   10/02/17 22:24:54 INFO mapred.JobClient:   map 100% reduce
100%
   10/02/17 22:24:56 INFO mapred.JobClient: Job complete:
   job_201002172218_0002
   10/02/17 22:24:56 INFO mapred.JobClient: Counters: 17
   10/02/17 22:24:56 INFO mapred.JobClient:    Job Counters
   10/02/17 22:24:56 INFO mapred.JobClient:      Launched reduce
   tasks=1
   10/02/17 22:24:56 INFO mapred.JobClient:      Launched map
   tasks=1
   10/02/17 22:24:56 INFO mapred.JobClient:      Data-local map
  tasks=1
  10/02/17 22:24:56 INFO mapred.JobClient: FileSystemCounters
  10/02/17 22:24:56 INFO mapred.JobClient:
  FILE_BYTES_READ=285
```

```
10/02/17 22:24:56 INFO mapred.JobClient:
HDFS_BYTES_READ=261
10/02/17 22:24:56 INFO mapred.JobClient:
FILE_BYTES_WRITTEN=602
10/02/17 22:24:56 INFO mapred.JobClient:
HDFS_BYTES_WRITTEN=259
10/02/17 22:24:56 INFO mapred.JobClient: Map-Reduce Framework
10/02/17 22:24:56 INFO mapred.JobClient:   Reduce input
groups=0
10/02/17 22:24:56 INFO mapred.JobClient:   Combine output
records=5
10/02/17 22:24:56 INFO mapred.JobClient:   Map input
records=8
10/02/17 22:24:56 INFO mapred.JobClient:   Reduce shuffle
bytes=0
10/02/17 22:24:56 INFO mapred.JobClient:   Reduce output
records=0
10/02/17 22:24:56 INFO mapred.JobClient:   Spilled Records=10
10/02/17 22:24:56 INFO mapred.JobClient:   Map output
bytes=293
10/02/17 22:24:56 INFO mapred.JobClient:   Combine input
records=8
10/02/17 22:24:56 INFO mapred.JobClient:   Map output
records=8
10/02/17 22:24:56 INFO mapred.JobClient:   Reduce input
records=5
```

3)查看结果:

```
$ bin/hadoop dfs -cat test2/part-r-00000
```

```
 map   4
 reduce   1
 hadoop   1
```

（八）关闭 Hadoop 进程

关闭 Map-Reduce,在 JobTracker 上:

```
$ bin/stop-mapred.sh
stopping jobtracker
192.168.0.136: stopping tasktracker
```

关闭 HDFS,在 NameNode 上:

```
$ bin/stop-dfs.sh
stopping namenode
192.168.0.136: stopping datanode
192.168.0.154: stopping secondarynamenode
```

（九）进程查看

可以使用 jps 命令查看系统目前运行的进程,可用来查看 start 过程中 java 进程的产生。

（十）HDFS 命令

```
$ bin/hadoop dfs  -command  [parameter]
```

command	usage
cat	显示文件
get	把 HDFS 上的文件下载到本地
put	向 HDFS 上载数据文件
rmr	删除
cp	复制
mkdir	新建目录

==========

对比伪分布与完全分布

==========

前言：为了简化理解，可以抛开 **SSH** 协议

1）Hadoop 进程启动

在伪分布中，masters 的 IP 为 localhost，slaves 的 IP 也为 localhost。与分布式相比，它的 datanode、secondarynamenode 和 tasktracker 进程都是在 localhost 创建，如图：

伪分布启动所有 Hadoop 进程

```
shiep205@master:~/hadoop$ bin/start-all.sh
starting namenode, logging to /home/shiep205/hadoop/bin/../logs/hadoop-shiep205-
namenode-master.out
localhost: starting datanode, logging to /home/shiep205/hadoop/bin/../logs/hadoo
p-shiep205-datanode-master.out
localhost: starting secondarynamenode, logging to /home/shiep205/hadoop/bin/../l
ogs/hadoop-shiep205-secondarynamenode-master.out
starting jobtracker, logging to /home/shiep205/hadoop/bin/../logs/hadoop-shiep20
5-jobtracker-master.out
localhost: starting tasktracker, logging to /home/shiep205/hadoop/bin/../logs/ha
doop-shiep205-tasktracker-master.out
```

而在完全分布中，datanode、tasktracker 由 slaves 中的主机创建，由 masters 中的 JobTracker 创建 secondarynamenode。

完全分布启动 DFS 和 MR 进程

```
shiep205@master:~/hadoop$ bin/start-dfs.sh
starting namenode, logging to /home/shiep205/hadoop/bin/../logs/hadoop-shiep205-
namenode-master.out
192.168.0.136: starting datanode, logging to /home/shiep205/hadoop/bin/../logs/h
adoop-shiep205-datanode-slave.out
192.168.0.154: starting secondarynamenode, logging to /home/shiep205/hadoop/bin/
../logs/hadoop-shiep205-secondarynamenode-master.out
shiep205@master:~/hadoop$ bin/start-mapred.sh
starting jobtracker, logging to /home/shiep205/hadoop/bin/../logs/hadoop-shiep20
5-jobtracker-master.out
192.168.0.136: starting tasktracker, logging to /home/shiep205/hadoop/bin/../log
s/hadoop-shiep205-tasktracker-slave.out
shiep205@master:~/hadoop$
```

2）/etc/hosts 文件参数比较

```
192.168.0.154  master      //完全分布
192.168.0.136  slave       //完全分布
#127.0.1.1  master      //伪分布  for Namenode & Jobtracker
```

```
#127.0.0.1  localhost   /*伪分布  for datanode & Tasktracker
                                $ secondarynamenode */
```

3) 3个核心配置文件 core-site.xml、hdfs-site.xml和mapred-site.xml

==========

Eclipse 配置

==========

在 eclipse 下运行 hadoop 自带例子 wordcount 的步骤。

1. 下载 **eclipse-SDK-*-linux-gtk.tar.gz** 到**/home/YourName**

2. 在**/home/[Your Name]**下解压

   ```
   $ sudo tar xzf eclipse-SDK-*-linux-gtk.tar.gz
   ```

3. 修改 eclipse 文件夹的权限,使其属于 **YourGroup** 组 **YourName** 用户

   ```
   $ sudo chown -R YourGroup:YourName eclipse
   ```

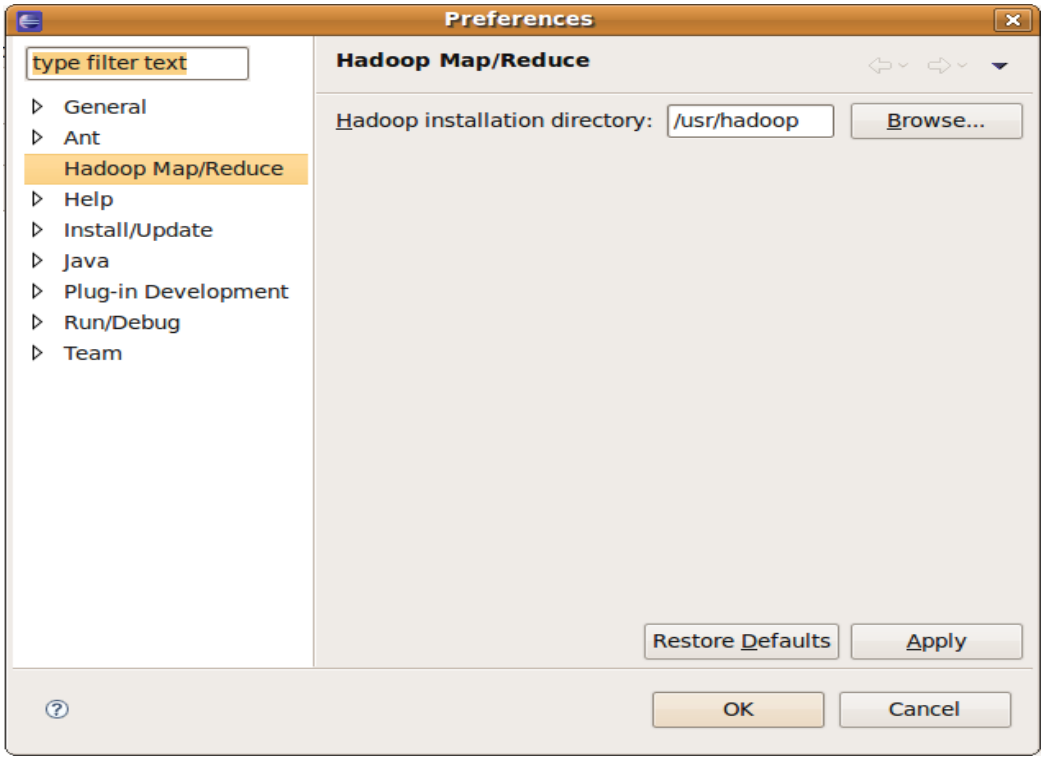
4. 将 **hadoop** 文件夹下的 **contrib/eclipse-plugin/hadoop-*-eclipse-plugin.jar** 拷贝到 **eclipse** 文件夹下的**/plugins** 文件夹里

5. 在**/home/YourName/testin** 下新建 2 个文本文件,里面各输入若干单词

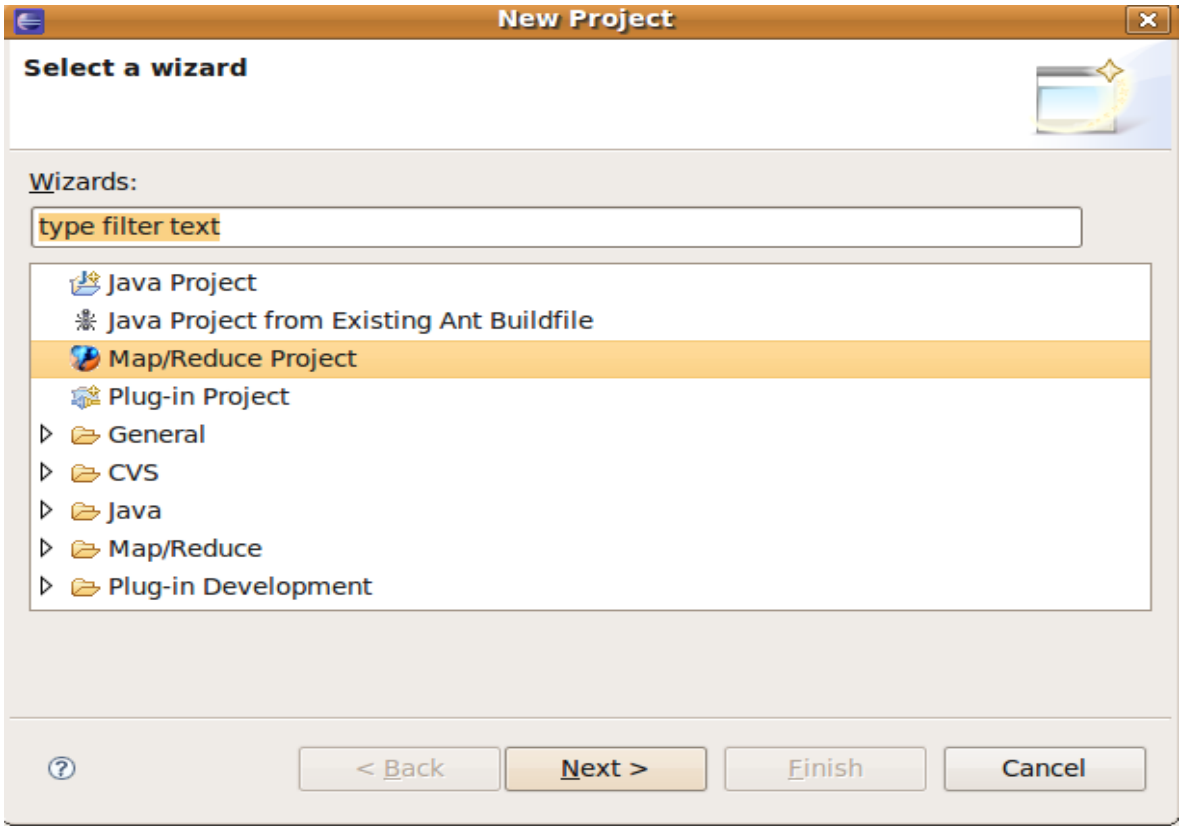
6. 启动 Eclipse

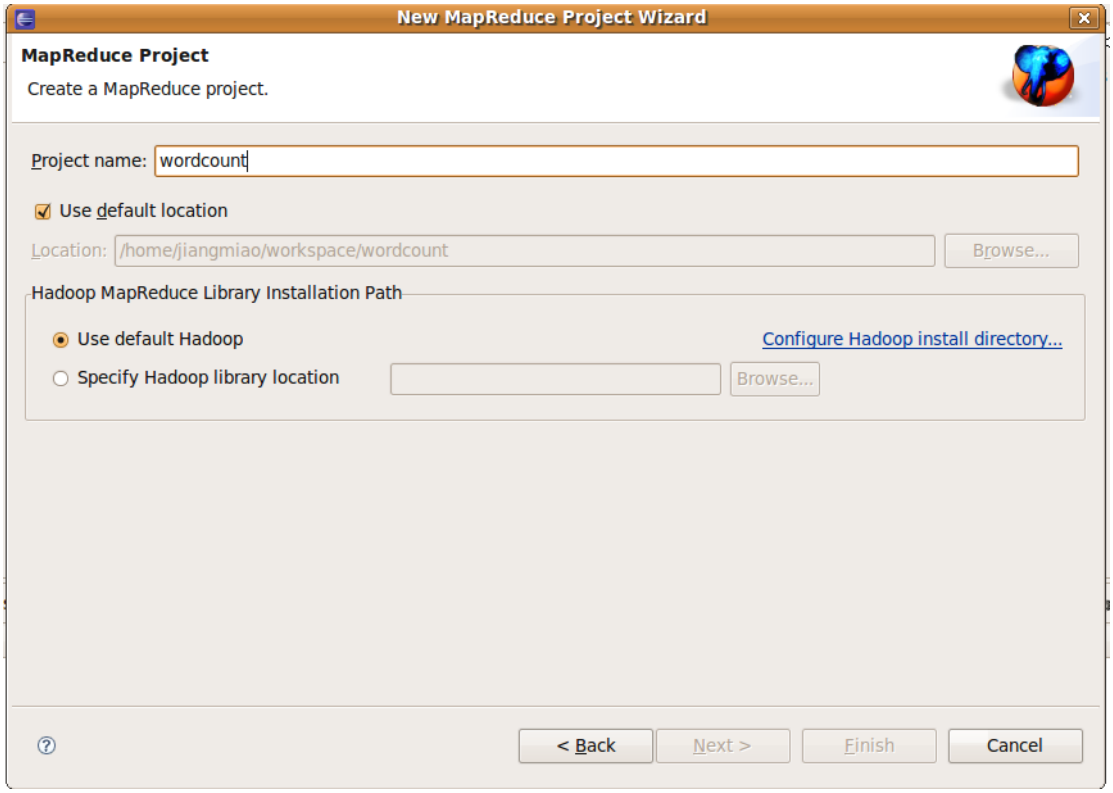
7. 设置 Hadoop 安装文件夹的路径

   Window->Preferences

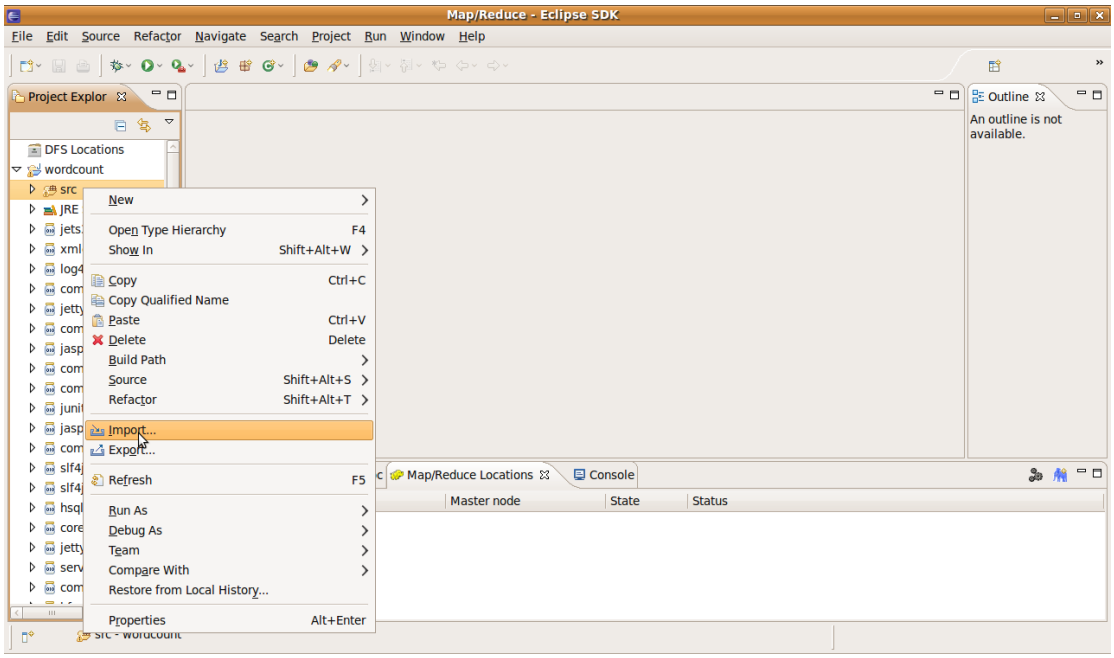


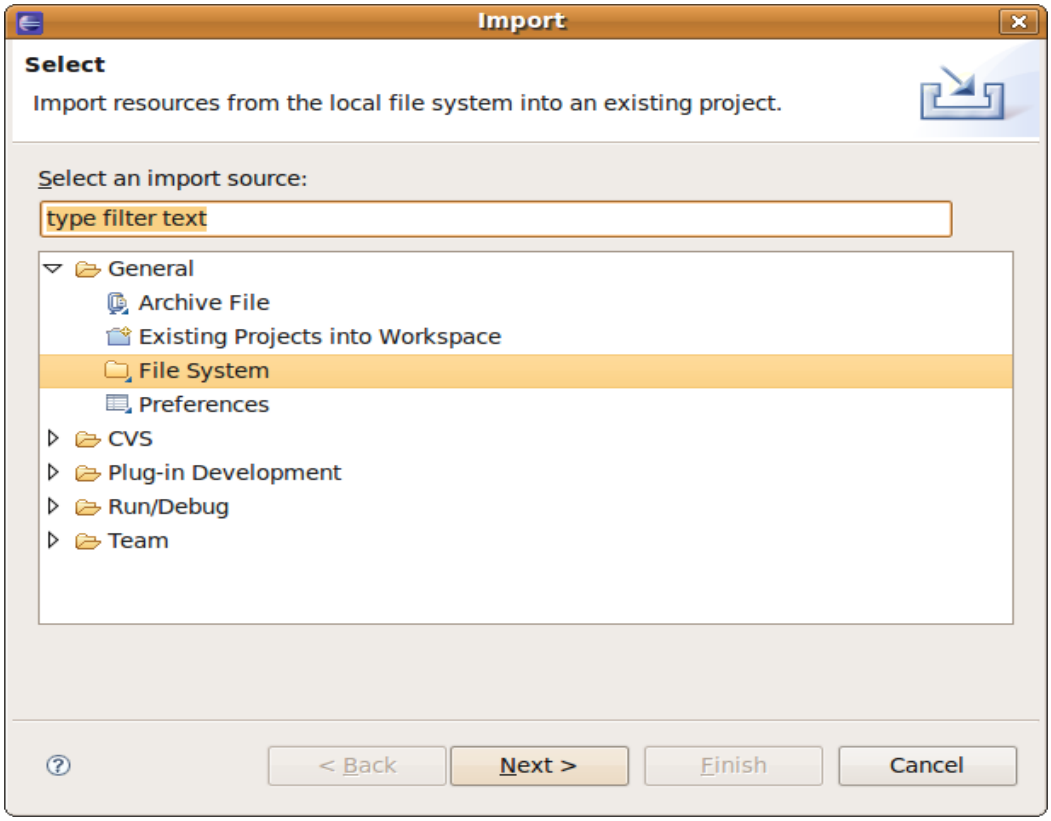
8.新建Map/Reduce项目

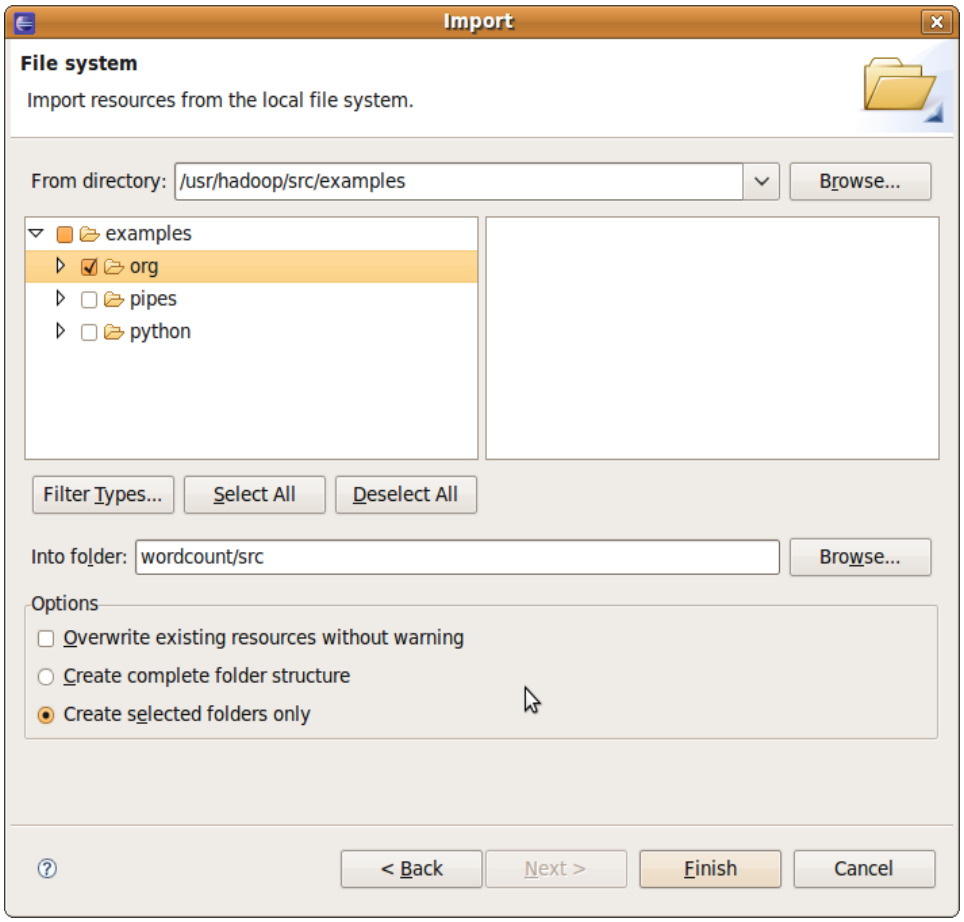




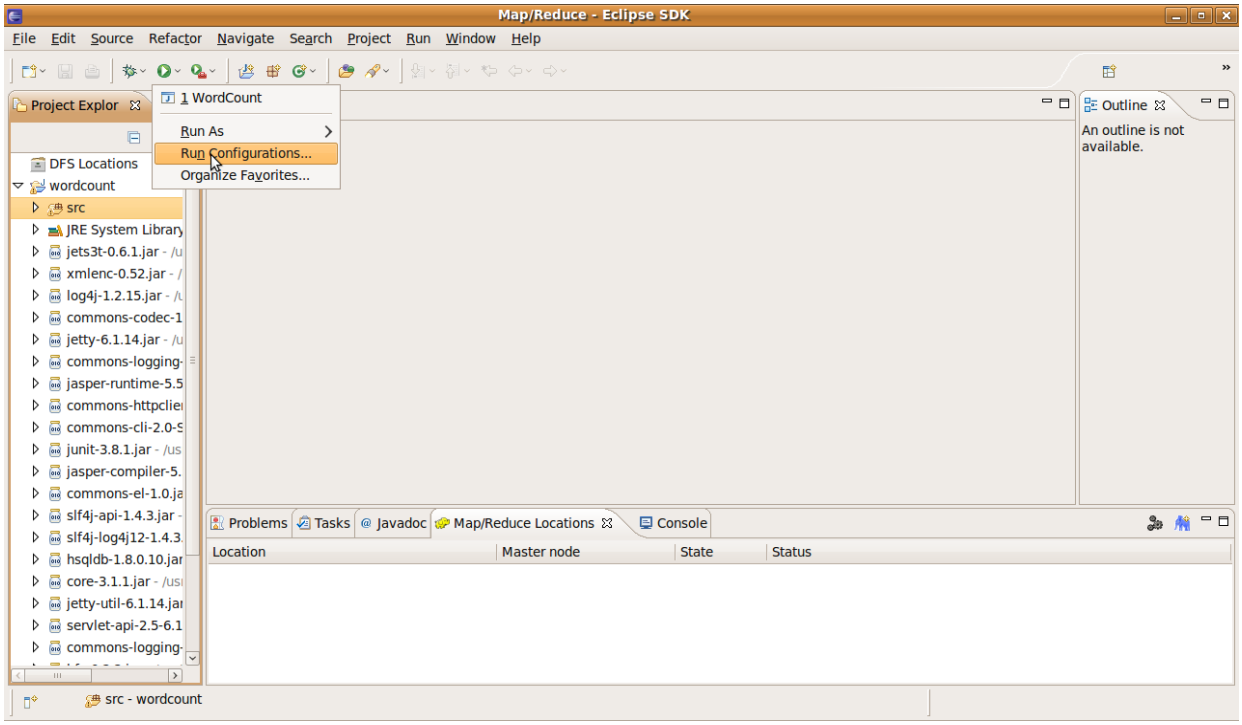
9.导入 Hadoop 自带例子

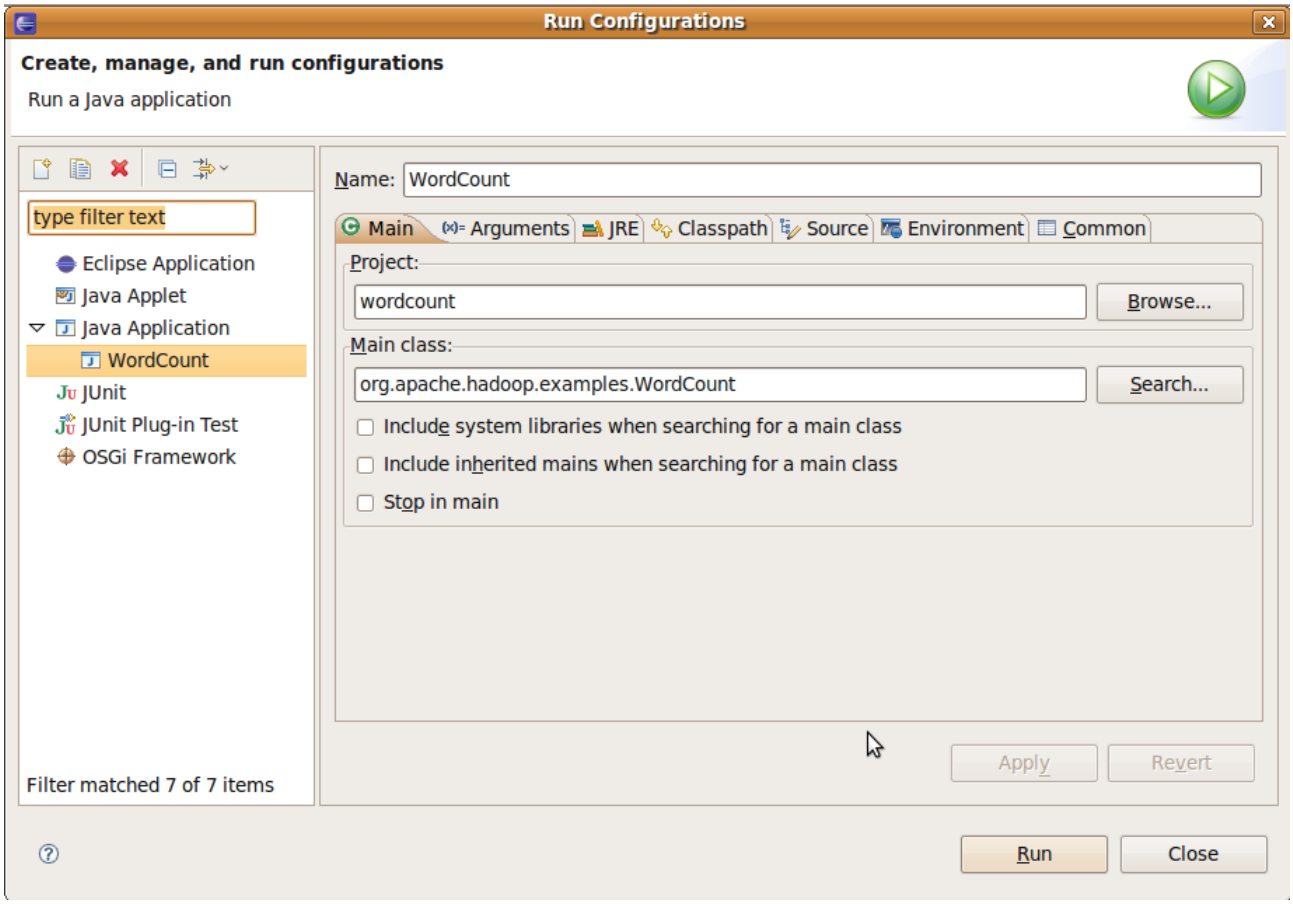


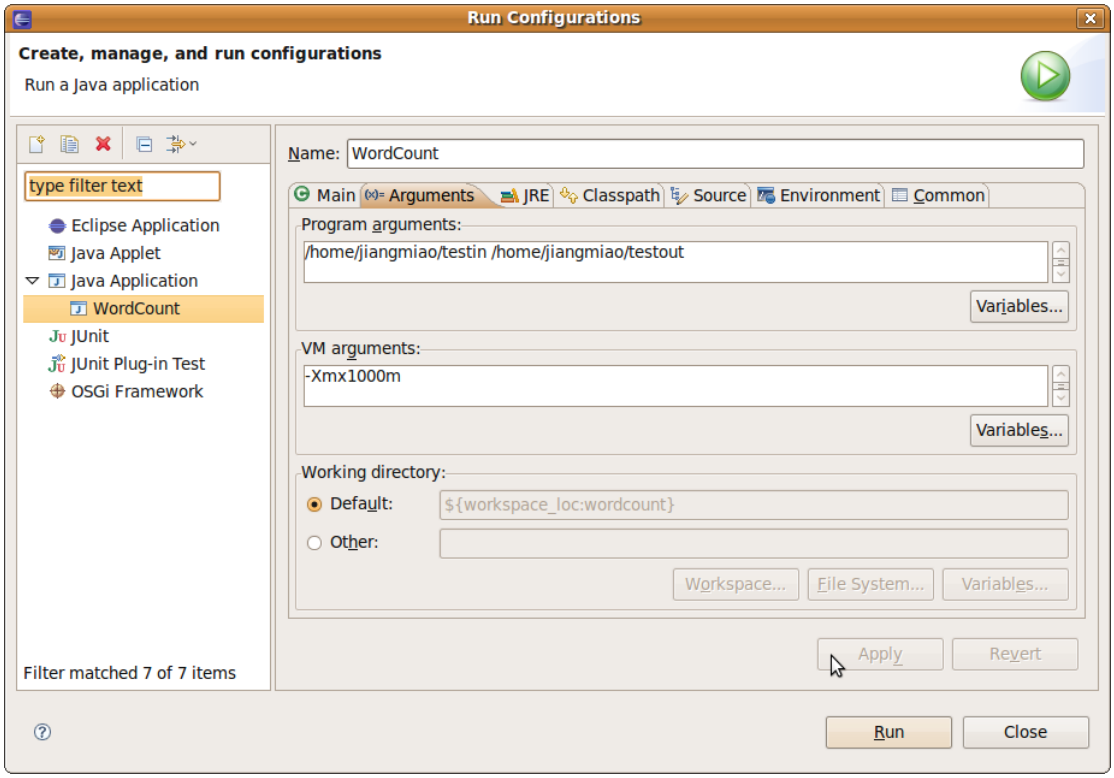

Import
Select
Import resources from the local file system into an existing project.
Select an import source:
type filter text
General
Archive File
Existing Projects into Workspace
File System
Preferences
CVS
Plug-in Development
Run/Debug
Team
< Back
Next >
Finish
Cancel


Import
File system
Import resources from the local file system.
From directory:
/usr/hadoop/src/examples
Browse...
examples
org
pipes
python
Filter Types...
Select All
Deselect All
Into folder:
wordcount/src
Browse...
Options
Overwrite existing resources without warning
Create complete folder structure
Create selected folders only
< Back
Next >
Finish
Cancel

10.运行例子


Map/Reduce - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explor
1 WordCount
Run As
Run Configurations...
Organize Favorites...
DFS Locations
wordcount
src
JRE System Library
Outline
An outline is not available.
Problems Tasks Javadoc Map/Reduce Locations Console
Location Master node State Status
src - wordcount


Run Configurations
Create, manage, and run configurations
Run a Java application
type filter text
Eclipse Application
Java Applet
Java Application
WordCount
JUnit
JUnit Plug-in Test
OSGi Framework
Filter matched 7 of 7 items
Name: WordCount
Main Arguments JRE Classpath Source Environment Common
Project:
wordcount
Browse...
Main class:
org.apache.hadoop.examples.WordCount
Search...
Include system libraries when searching for a main class
Include inherited mains when searching for a main class
Stop in main
Apply
Revert
Run
Close


Run Configurations
Create, manage, and run configurations
Run a Java application
type filter text
Eclipse Application
Java Applet
Java Application
WordCount
JUnit
JUnit Plug-in Test
OSGi Framework
Filter matched 7 of 7 items
Name: WordCount
Main
Arguments
JRE
Classpath
Source
Environment
Common
Program arguments:
/home/jiangmiao/testin /home/jiangmiao/testout
Variables...
VM arguments:
-Xmx1000m
Variables...
Working directory:
Default:
${workspace_loc:wordcount}
Other:
Workspace...
File System...
Variables...
Apply
Revert
Run
Close