注：hadoop 是 java 编写的，可以通过 java 直接编译出 hadoop 包。因此，操作系统影响不大，只是需要剥去 linux 这个外壳。

材料：

1. JDK 版本"1.7.0_01"
2. Cygwin——在线安装
3. Eclipse SDK 版本 3.3.2
4. Hadoop 版本 hadoop-0.20.2
5. windows 7

本教程适合以上版本，特别是 hadoop 版本和 Eclipse 版本。如果换别的搭配，不一定成功。提示：hadoop0.20.2 版本下的 Eclipse 插件只适合 3.3.2 的 eclipse，Eclispe 版本高不能运行。另外，hadoop 0.20.203 版本在和 cygwin 搭配上有 bug，不能启动 jodtracker。

实施：

1. 安装 JDK 和 Cygwin

Jdk 详细步骤略去。要注意的是：jdk 的安装目录尽量简洁，中间不要带空格。

1. 下载 cygwin 的 setup.exe 进行在线安装，setup 版本：2.764

1.1 install from Internet

1.2 设置 cygwin 最好简短，地址中间不带空格

1.3 Direct Connection

1.4 选择 down site (网易 163： mirrors.163.com)

1.5 开始选择需要安装的组件

1.5.1 Net Category 下的两个 OpenSSL(重要)



1.5.2 Base Category 下的 sed



1.5.3 Editor Category 下的 vim



1.5.4 Devel Category 下的 subversion



1.5.5 点击下一步开始下载并安装，一气呵成，否则重新安装。

1.6 配置系统环境变量

JAVA_HOME=C:\Java\jdk1.7.0_01

CLASSPATH=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar

PATH=%JAVA_HOME%/bin;%JAVA_HOME%/jre/bin;C:\windows\system32\;

1.7 BINGO 用户中的变量

PATH=D:\study\cygwin\bin;D:\study\cygwin\usr\bin

2. 在 cygwin 中安装 ssh

2.1 打开 cygwin 会提示用户没有权限的问题，按照提示敲入相应两个命令解决，待参数 -l

2.2 敲入命令 ssh-host-config 开始配置 ssh。遇到问题“should privilege separation be used?”回答 no，其他的按默认，它会询问是否注册为系统服务，回答是。

2.3 在“管理”-“服务”里开启 CYGWIN sshd 服务。

2.4 配置 SSH 空密码登录

执行以下命令：

ssh-keygen　生成本机密钥

cd　~/.ssh/

cp id_rsa.pub authorized_keys

关闭 cygwin，重新打开 cygwin。尝试 ssh localhost 连接。

2.5 但是，在 cygwin 中无法执行命令 ssh localhost，提示，Connection closed by 1. 这是权限问题。

解决方案：1、开端——运行——services.msc2、右键 CYGWIN sshd——属性——登录选项卡——选择“此账户”——浏览——高级——立即查找——选择你的账户名（必须为管理员权限）——输进密码（必须 要有，空密码不将不被接受）——确定。3、重启 CYGWIN sshd 即可。

2.6 执行命令 ssh localhost 正常

3. Hadoop

入门：cygwin 中看到的 windows 文件全部都在/cygwin/目录下。这个目录下的文件夹对应这 windows 系统中的 c 盘，d 盘…

因此，如果在 windows 目录下的路径是 c:\1.txt

那么，在 linux 目录下的路径是/cygdrive/c/1.txt

3.1 解压 hadoop 到 D:\study\hadoop-0.20.2

3.2 配置 hadoop

3.2.1 配置 conf 下的 hadoop-env.sh

The java implementation to use.　Required.

export JAVA_HOME=/cygdrive/c/Java/jdk1.7.0_01

记得去掉 export 前面的注释

3.2.2 分别从 src 下的 core，hdfs 和 mapred 三个目录下的 core-default.xml，hdfs-default.xml 和 mapred-default.xml 拷贝到 conf 目录中；分别替换掉目录下的 core-site.xml，hdfs-site.xml 和 mapred-site.xml

3.2.3 编辑 core-site.xml

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

<value>hdfs://localhost:8888</value>

3.2.4 编辑 mapred-site.xml

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

<value>localhost:9999</value>

3.3 格式化 namenode，cd 到 bin 目录下

命令：

$ ./hadoop namenode –format

3.4 开启 hadoop

命令：

./start-all.sh

3.5 查询 hdfs 文件

./hadoop fs –ls /

3.6 为了测试每个组件是否正常工作，可以开启五个 cygwin 窗口依次执行

./hadoop namenode

./hadoop secondarynamenode

./hadoop jobtracker

./hadoop datanode

./hadoop tasktracker

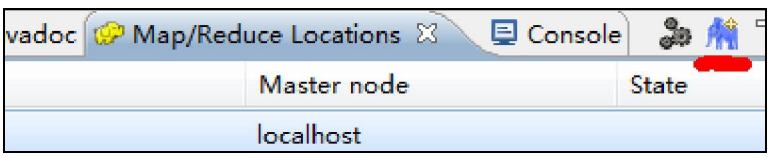
如果 datanode 出问题，直接进 D:\tmp\hadoop-BinGo\dfs\data 把这个目录删了！重新启动。

4. 配置 eclipse 插件

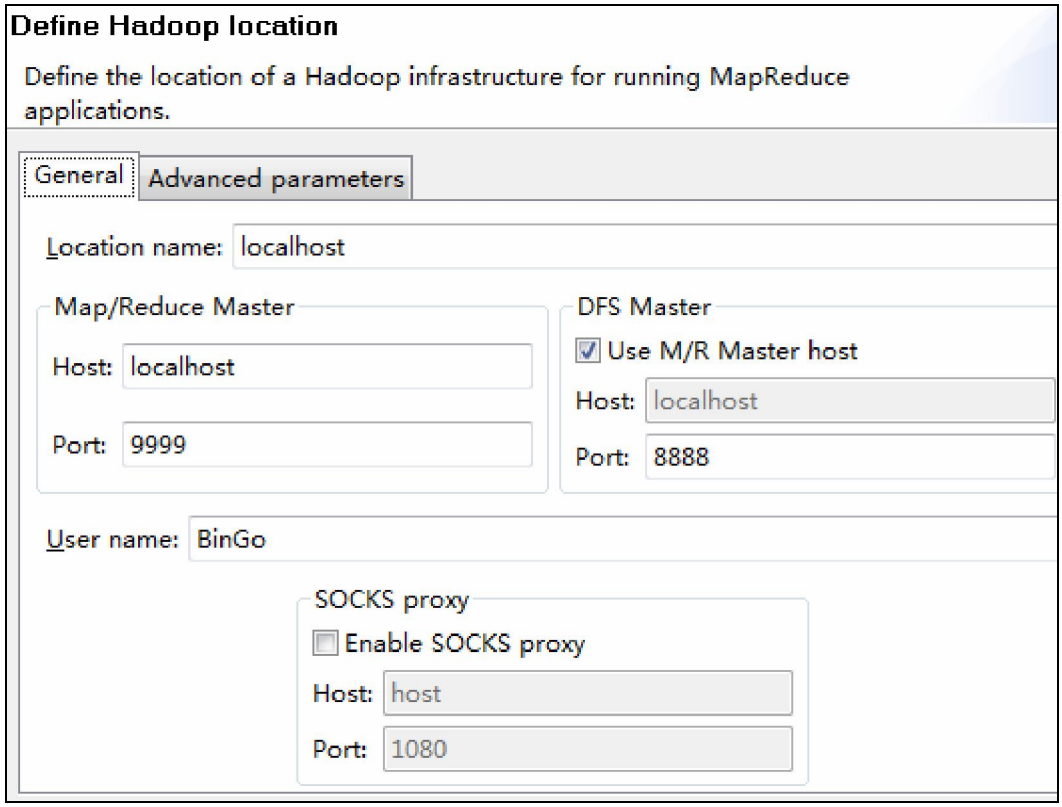
在 hadoop 的 hadoop-0.20.2\contrib\eclipse-plugin 目录下找到 hadoop-0.20.2-eclipse-plugin.jar 插件

4.1 将插件拷贝至 eclipse 的 plugins 目录下。

4.2 打开 eclipse，Window -> Open Perspective -> Other 选择 Map/Reduce。下方将出现：



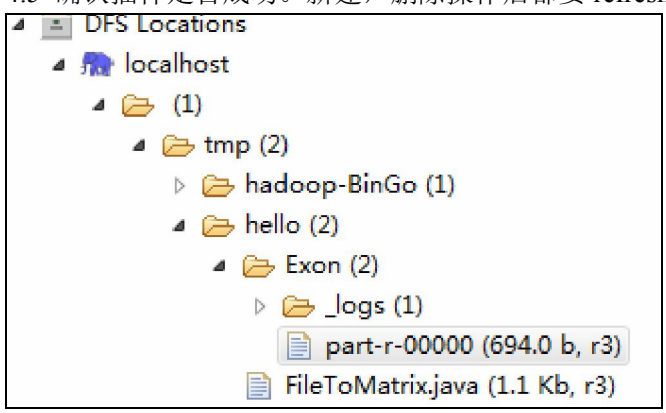
4.3 点击右上方的 New Hadoop location...配置一下信息



4.4 Window -> Preferences 选择 “Hadoop Map/Reduce”，点击“Browse...”选择 Hadoop 文件夹的路径。

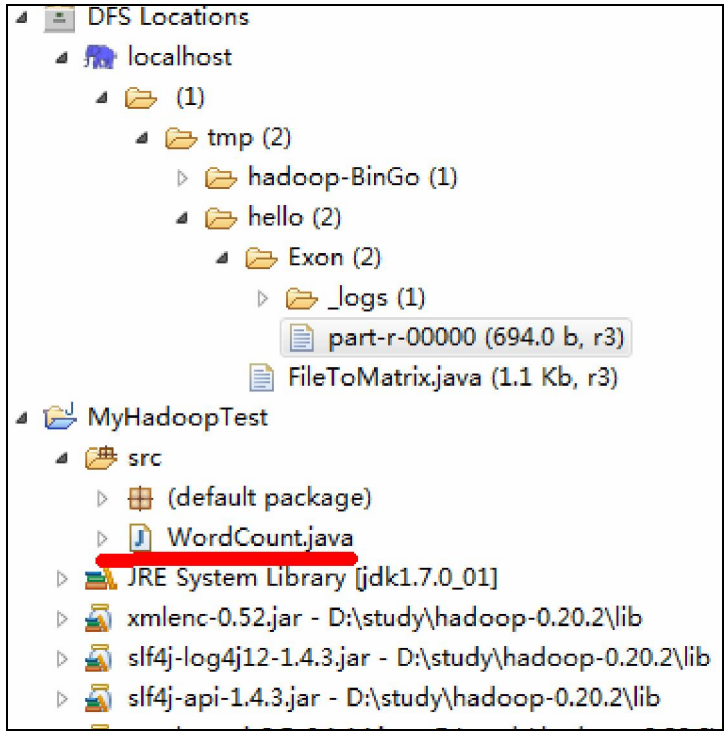
这个步骤与运行环境无关，只是在新建工程的时候能将 hadoop 根目录和 lib 目录下的所有 jar 包自动导入。

4.5 确认插件是否成功。新建，删除操作后都要 refresh 一次目录



5. 测试 hadoop 环境

5.1 新建一个 Map/Reduce 项目，测试 WordCount.java。注：统计文件的单词数量



WordCount.java 代码:

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
```

```
    }
  }
}

public static class IntSumReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}

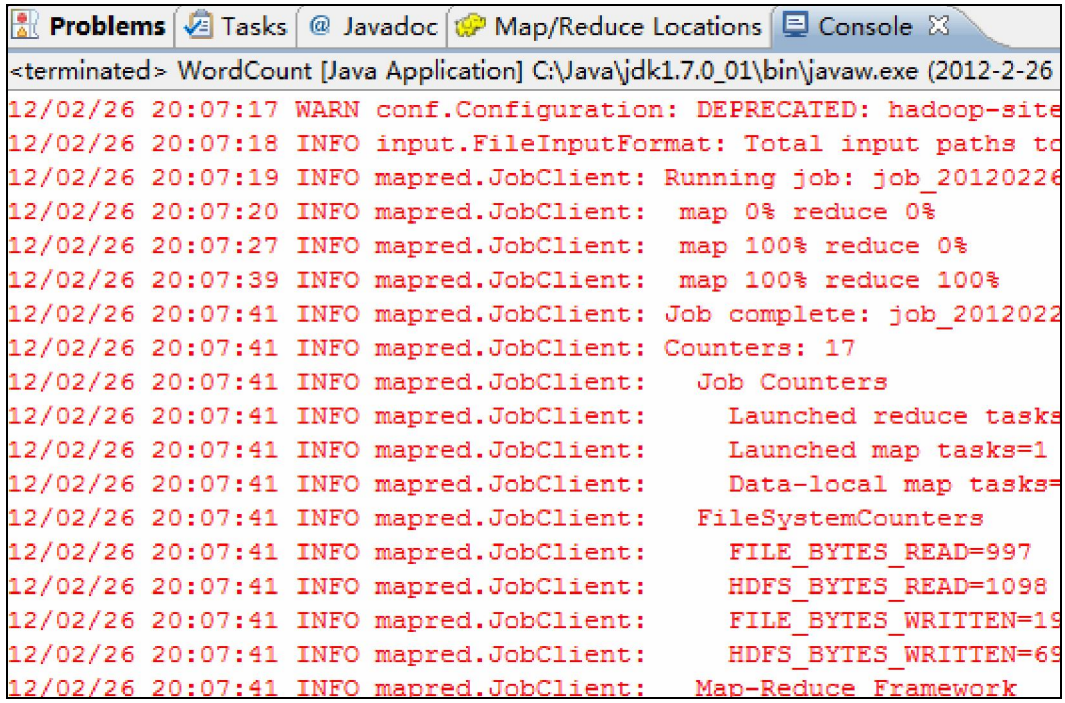
public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  String[] otherArgs = new GenericOptionsParser(conf,
args).getRemainingArgs();
  if (otherArgs.length != 2) {
    System.err.println("Usage: wordcount <in> <out>");
    System.exit(2);
  }
  Job job = new Job(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
  FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```

5.2 配置运行参数

Run As -> Open Run Dialog... 选择 WordCount 程序，在 Arguments 中配置运行参数：/tmp/hello/FileToMatrix.java /tmp/hello/Exon

5.3 运行

Run As -> Run on Hadoop 选择之前配置好的 MapReduce 运行环境，点击“Finish”运行



6. 参考文章

http://www.cnblogs.com/hiddenfox/archive/2011/11/26/2264461.html
http://blog.csdn.net/yanical/article/details/4474991
http://blog.csdn.net/hongweigg/article/details/7197662
http://www.tech126.com/eclipse-hadoop/
Hadoop 开发者入门专刊.pdf

2/26/2012
Written by BinGo