

Pig Setup

Table of contents

1 Overview.....	2
2 Beginning Pig.....	2
3 Advanced Pig.....	4
4 Sample Code.....	5

1. Overview

1.1. Requirements

Unix and Windows users need the following:

1. **Hadoop 0.20.2** - <http://hadoop.apache.org/common/releases.html>
2. **Java 1.6** - <http://java.sun.com/javase/downloads/index.jsp> (set JAVA_HOME to the root of your Java installation)
3. **Ant 1.7** - <http://ant.apache.org/> (optional, for builds)
4. **JUnit 4.5** - <http://junit.sourceforge.net/> (optional, for unit tests)

Windows users need to install Cygwin and the Perl package: <http://www.cygwin.com/>

2. Beginning Pig

2.1. Download Pig

To get a Pig distribution, download a recent stable release from one of the Apache Download Mirrors (see [Pig Releases](#)).

Unpack the downloaded Pig distribution. The Pig script is located in the bin directory (/pig-n.n.n/bin/pig).

Add /pig-n.n.n/bin to your path. Use export (bash,sh,ksh) or setenv (tcsh,csh). For example:

```
$ export PATH=/<my-path-to-pig>/pig-n.n.n/bin:$PATH
```

Try the following command, to get a list of Pig commands:

```
$ pig -help
```

Try the following command, to start the Grunt shell:

```
$ pig
```

2.2. Run Modes

Pig has two run modes or exectypes:

- **Local Mode** - To run Pig in local mode, you need access to a single machine.
- **Mapreduce Mode** - To run Pig in mapreduce mode, you need access to a Hadoop cluster and HDFS installation. Pig will automatically allocate and deallocate a 15-node cluster.

You can run the Grunt shell, Pig scripts, or embedded programs using either mode.

2.3. Grunt Shell

Use Pig's interactive shell, Grunt, to enter pig commands manually. See the [Sample Code](#) for instructions about the passwd file used in the example.

You can also run or execute script files from the Grunt shell. See the [run](#) and [exec](#) commands.

Local Mode

```
$ pig -x local
```

Mapreduce Mode

```
$ pig  
or  
$ pig -x mapreduce
```

For either mode, the Grunt shell is invoked and you can enter commands at the prompt. The results are displayed to your terminal screen (if DUMP is used) or to a file (if STORE is used).

```
grunt> A = load 'passwd' using PigStorage(':');  
grunt> B = foreach A generate $0 as id;  
grunt> dump B;  
grunt> store B;
```

2.4. Script Files

Use script files to run Pig commands as batch jobs. See the [Sample Code](#) for instructions about the passwd file and the script file (id.pig) used in the example.

Local Mode

```
$ pig -x local id.pig
```

Mapreduce Mode

```
$ pig id.pig  
or  
$ pig -x mapreduce id.pig
```

For either mode, the Pig Latin statements are executed and the results are displayed to your terminal screen (if DUMP is used) or to a file (if STORE is used).

3. Advanced Pig

3.1. Build Pig

To build pig, do the following:

1. Check out the Pig code from SVN: *svn co http://svn.apache.org/repos/asf/hadoop/pig/trunk.*
2. Build the code from the top directory: *ant*. If the build is successful, you should see the *pig.jar* created in that directory.
3. Validate your *pig.jar* by running a unit test: *ant test*

3.2. Environment Variables and Properties

See [Download Pig](#).

The Pig environment variables are described in the Pig script file, located in the `/pig-n.n.n/bin` directory.

The Pig properties file, `pig.properties`, is located in the `/pig-n.n.n/conf` directory. You can specify an alternate location using the `PIG_CONF_DIR` environment variable.

3.3. Run Modes

See [Run Modes](#).

3.4. Embedded Programs

Used the embedded option to embed Pig commands in a host language and run the program. See the [Sample Code](#) for instructions about the `passwd` file and java files (`idlocal.java`, `idmapreduce.java`) used in the examples.

Local Mode

From your current working directory, compile the program:

```
$ javac -cp pig.jar idlocal.java
```

Note: `idlocal.class` is written to your current working directory. Include “.” in the class path when you run the program.

From your current working directory, run the program:

```
Unix: $ java -cp pig.jar:. idlocal
```

```
Cygwin: $ java -cp `.;pig.jar` idlocal
```

To view the results, check the output file, id.out.

Mapreduce Mode

Point \$HADOOPDIR to the directory that contains the hadoop-site.xml file. Example:

```
$ export HADOOPDIR=/yourHADOOPsite/conf
```

From your current working directory, compile the program:

```
$ javac -cp pig.jar idmapreduce.java
```

Note: idmapreduce.class is written to your current working directory. Include “.” in the class path when you run the program.

From your current working directory, run the program:

```
Unix: $ java -cp pig.jar:.$HADOOPDIR idmapreduce  
Cygwin: $ java -cp `.;pig.jar;$HADOOPDIR` idmapreduce
```

To view the results, check the idout directory on your Hadoop system.

4. Sample Code

The sample code is based on Pig Latin statements that extract all user IDs from the /etc/passwd file.

Copy the /etc/passwd file to your local working directory.

id.pig

For the Grunt Shell and script files.

```
A = load 'passwd' using PigStorage(':');  
B = foreach A generate $0 as id;  
dump B;  
store B into 'id.out';
```

idlocal.java

For embedded programs.

```
import java.io.IOException;  
import org.apache.pig.PigServer;  
public class idlocal{  
public static void main(String[] args) {  
try {  
PigServer pigServer = new PigServer("local");
```

```

        runIdQuery(pigServer, "passwd");
    }
    catch(Exception e) {
    }
}
public static void runIdQuery(PigServer pigServer, String inputFile) throws
IOException {
    pigServer.registerQuery("A = load '" + inputFile + "' using
PigStorage(':');");
    pigServer.registerQuery("B = foreach A generate $0 as id;");
    pigServer.store("B", "id.out");
}
}
}

```

idmapreduce.java

For embedded programs.

```

import java.io.IOException;
import org.apache.pig.PigServer;
public class idmapreduce{
    public static void main(String[] args) {
        try {
            PigServer pigServer = new PigServer("mapreduce");
            runIdQuery(pigServer, "passwd");
        }
        catch(Exception e) {
        }
    }
    public static void runIdQuery(PigServer pigServer, String inputFile) throws
IOException {
        pigServer.registerQuery("A = load '" + inputFile + "' using
PigStorage(':');");
        pigServer.registerQuery("B = foreach A generate $0 as id;");
        pigServer.store("B", "idout");
    }
}
}

```