Pig Quick Start

Table of contents

1. Requirements

Unix and **Windows** users need the following:

1. **Hadoop 18**: http://hadoop.apache.org/core/
2. **Java 1.6**, preferably from Sun: http://java.sun.com/javase/downloads/index.jsp. Set JAVA_HOME to the root of your Java installation.
3. **Ant** for builds: http://ant.apache.org/.
4. **JUnit** for unit tests: http://junit.sourceforge.net/.

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

2. Building Pig

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. Running Pig

3.1. Overview

This section discusses the Pig run modes and the different ways you can run Pig using these modes.

3.1.1. Run Modes

Pig has two run modes or exectypes, local and hadoop (mapreduce).

- **Local Mode**: To run Pig in local mode, you need access to a single machine.
- **Hadoop (mapreduce) Mode**: To run Pig in hadoop (mapreduce) mode, you need access to a Hadoop cluster and HDFS installation.

To get a listing of all Pig commands, including the run modes, use:

```
$ pig -help
```

3.1.2. Run Ways

You can run Pig three ways – using either local mode or hadoop (mapreduce) mode:

- **Grunt Shell**: Enter Pig commands manually using Pig's interactive shell, Grunt.
- **Script File**: Place Pig commands in a script file and run the script.
- **Embedded Program**: Embed Pig commands in a host language and run the program.

Note: Also see the Pig Latin exec and run commands.

3.1.3. Sample Code

The examples in this section are based on these Pig Latin statements, which extract all user IDs from the /etc/passwd file.

To set environment variables, use the right command for your shell:

- setenv PIGDIR /pig (tcsh, csh)
- export PIGDIR=/pig (bash, sh, ksh)

The examples in the Running Pig section use export.

3.1.3.1. id.pig

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

3.1.3.2. idlocal.java

```
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");
 }
}
```

3.1.3.3. idhadoop.java

```
import java.io.IOException;
import org.apache.pig.PigServer;
public class idhadoop {
   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");
   }
}
```

3.2. Local Mode

This section shows you how to run Pig in local mode, using the Grunt shell, a Pig script, and an embedded program.

To run Pig in local mode, you only need access to a single machine. To make things simple, copy these files to your current working directory (you may want to create a temp directory and move to it):

- The /etc/passwd file
- The pig.jar file, created when you build Pig.
- The sample code files (id.pig and idlocal.java) located on this page

3.2.1. Grunt Shell

To run Pig's Grunt shell in local mode, follow these instructions.

First, point $PIG_CLASSPATH to the pig.jar file (in your current working directory):

```
$ export PIG_CLASSPATH=./pig.jar
```

From your current working directory, run:

```
$ pig -x local
```

The Grunt shell is invoked and you can enter commands at the prompt.

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

3.2.2. Script File

To run a Pig script file in local mode, follow these instructions (which are the same as the Grunt Shell instructions above – you just include the script file).

First, point $PIG_CLASSPATH to the pig.jar file (in your current working directory):

```
$ export PIG_CLASSPATH=./pig.jar
```

From your current working directory, run:

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

The Pig Latin statements are executed and the results are displayed to your terminal screen.

3.2.3. Embedded Program

To compile and run an embedded Java/Pig program in local mode, follow these instructions.

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.

3.3. Hadoop Mode

This section shows you how to run Pig in hadoop (mapreduce) mode, using the Grunt shell, a Pig script, and an embedded program.

To run Pig in hadoop (mapreduce) mode, you need access to a Hadoop cluster. You also need to copy these files to your home or current working directory.

- The /etc/passwd file

- The pig.jar file, created when you build Pig.
- The sample code files (id.pig and idhadoop.java) located on this page

3.3.1. Grunt Shell

To run Pig's Grunt shell in hadoop (mapreduce) mode, follow these instructions. When you begin the session, Pig will allocate a 15-node cluster. When you quit the session, Pig will deallocate the nodes.

From your current working directory, run:

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

The Grunt shell is invoked and you can enter commands at the prompt.

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

3.3.2. Script File

To run Pig script files in hadoop (mapreduce) mode, follow these instructions (which are the same as the Grunt Shell instructions above – you just include the script file). Again, Pig will automatically allocate and deallocate a 15-node cluster.

From your current working directory, run:

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

The Pig Latin statements are executed and the results are displayed to your terminal screen.

3.3.3. Embedded Program

To compile and run an embedded Java/Pig program in hadoop (mapreduce) mode, follow these instructions.

First, 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 idhadoop.java
```

Note: idhadoop.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 idhadoop
Cygwin: $ java –cp '.;pig.jar;$HADOOPDIR' idhadoop
```

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