GettingNutchRunningOnCygwin

Problems and workarounds for running nutch on cygwin

I followed the NutchHadoopTutorial and encountered a few problems, which after looking for solutions, seems to have affected others trying to do nutch on cygwin.

Line Endings

The tutorial mentions using dos2unix for all commands, i.e.

```
dos2unix /nutch/search/bin/*.sh /nutch/search/bin/hadoop /nutch/search/bin/nutch
dos2unix /nutch/search/conf/*.sh
```

But this also applies to the slave (and master) files. i.e.

```
dos2unix /nutch/search/conf/slaves
```

Without this, the hostname passed to ssh will include a trailing '\r', producing "no address assocaited with name" error.

Logging files

The "hostname" command is used to construct logfile names. This command is included in Windows and in cygwin. When the windows version is used, an additional '\r' is included in the command output, causing the logfile name to be an invalid filename. Errors such as "Head: cannot open <filename> for reading: no such file or directory" occur, even though the name looks ok.

You can see the problem first hand by running

```
ssh localhost "hostname | cat -v"
```

The output will include `^M` after the hostname.

The first cause of the problem is that hostname is not installed under cygwin. To get this, install "coreutils" from the "base" category.

The firther problem is the path setting for the sshd service. After ssh'ing in, the Windows hostname is still used (although on my installation $PATH appeared correct, e.g. `ssh localhost "echo $PATH; type hostname"`, showed a correct path with `/usr/bin:/bin` in the path before the windows directories, yet "type" was finding the windows version of the file.)

To fix the path setting, add the PATH environment variable to the sshd service. Under the key `HKLM\System\CurrentControlSet\Services\sshd\Parameters\Environment` create a new string value PATH, set to `/usr/bin:/usr/lib:/bin:%PATH%`. This prepends /user/bin etc. to the PATH defined in Windows. After restarting the sshd service, return the tests above, and the `^M` should no longer be present.

See also http://www.cygwin.com/ml/cygwin/2007-07/msg00045.html

Impersonated SSH Account

When using the cygwin sshd, it is necessary to first ssh in before running NDFS commands (e.g. bin/hadoop -put urls urls.) This is to ensure the current user account is constistent with later ssh sessions. (Even if you ssh in as the same user you are running locally, the sshd service may use a different user account.)

With my setup, I had a "nutch" shortcut to cygwin.bat that was started using runas.exe, to launch the nutch user. NDFS commands would then write files to /user/nutch. But, after running ssh, supposedly logging in as the same user, the NDFS stores files under "/user/sshd", because the current user account was in fact the sshd account.

On Windows 2003 Server, cygwin sshd is not able to log in users under their actual account. If you ssh in and enter the command

```
%SystemRoot%\System32\whoami.exe
```

It will display the account name running the sshd service, and not the user you expected (e.g. "nutch".) If you simply type

```
whoami
```

then it will print "nutch" or whichever user you ssh'ed in as. When HDFS runs, it's the native username that it sees, i.e. the account running the sshd service.

Logging in first with ssh before doing anything with NDFS ensures that all files are created using the same account name in the HDFS hierarchy (the sshd service account.) Not doing this, the first files created by local commands (e.g. hadoop dfs -put urls) will go to the "nutch" user (or the user running the cygwin shell) while subsequent commands run on remote machines will go under the sshd account folder in HDFS.

As a result of this, the sshd account also needs write access to the nutch folder.