Running Speech Recognition Experiments on XSEDE

Mark Hasegawa-Johnson

1/28/2014
Overview

1. What is XSEDE?
2. Logging in
3. Modules
4. Installing kaldi
5. “Hello world” as a qsub job
6. Uploading your data
7. Kaldi tutorial
1. What is XSEDE?

XSEDE is an NSF-funded network of “supercomputers” (each server is >1000 cpus)
How do you acknowledge XSEDE?

Every paper you write should acknowledge:

- Grants that pay your stipend and/or your airfare to the conference
- Grant covering each co-author who contributed >4 hours on external grant support
- Sometimes: your data (e.g., IARPA OpenKWS)
- Your compute time, XSEDE or otherwise:

“This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. All opinions and conclusions are those of the authors, and are not endorsed by NSF.”
2. Logging in
Logging in: (1) Getting an account

- Portal.xsede.org
- Create an account for yourself
(2) Getting an allocation

- Faculty can apply for grants of compute time.
- Faculty can add students to their grants. Send me e-mail.
Logging in: (3) ssh

Two steps:
1. ssh to login.xsede.org using your login ID and PWD
2. gsissh to the machine you want
Logging in: (4) gsissh

We have three machines available on the LBMAVSR allocation. All are unix, but with different storage setup, different libraries.
Logging in: (4) gsissh

Kaldi has only been successfully compiled on these two machines.
3. Modules
Libraries, environment variables and software can be added using “module load.” Available modules listed by “module avail” may include:

- matlab and/or octave
- R
- scipy
- fftw
- lapack
4. Installing kaldi
Installing kaldi: (1) what is kaldi?

- kaldi is an open-source ASR toolbox resulting from an NSF/DARPA/FRTI project, based at JHU
- C++ code well supported, easy to read and revise
- lots of tutorials on line
- OpenFST-based
Installing kaldi on trestles: (2) svn

• Create a directory where you want to put the source code

• `svn co http://svn.code.sf.net/p/kaldi/code/trunk`

...and lots more stuff gets checked out.
Installing kaldi on lonestar: cd $WORK; mkdir kaldi; cd kaldi; svn

Lonestar home directory has no space. You need to use $WORK directory.
Make kaldi

cd $WORK/kaldi/trunk/tools;
make;
module load lapack;
cd $WORK/kaldi/trunk/src;
./configure;
make depend;
make;
make;
Make your own ATLAS, if libatlas.a can’t be found

Not recommended because it’s 4X slower than running ATLAS native to the machine, but this is the only method that I’ve managed to get working.
5. Running “Hello world” as a qsub job
Running “Hello world” as a qsub job

• XSEDE computers have three run modes:
  1. Interactive command line. Type the command, it runs. Log out, it dies.
  2. Debug batch mode:
     1. qsub –q development jobname.sh (on lonestar) or
     2. qsub –q shared jobname.sh (on trestles)
  3. Long-run batch mode:
     1. qsub –q normal jobname.sh
Create a “Hello world” program

- I usually use ctrl-D to end cat, rather than the weird `<<EOF...EOF` syntax, but ctrl-D is hard to show you in a powerpoint figure.
- You can also use “emacs hello.cpp” or “vi hello.cpp” to create the file. Uh, that’s probably easier if you know emacs or vi.
Create a “run_hello.sh” script

- qsub requires a plaintext script file as argument, e.g., a bash script, so you’d better create one.
- qsub run_hello.sh; is the same as qsub –V –l h_rt=01:00:00 ... run_hello.sh;
  In other word, options specified in $# lines are given as options to qsub.
- The program (hello) is given with full path, in case –V doesn’t work.
$# options in a shell script that you want qsub to run

• ...are listed in great detail if you type “man qsub” in the ssh window.
• ...are more usefully exemplified in the very useful tutorial, http://portal.xsede.org/web/xup/documentation-overview
• $# -V means “run with current directory and environment variables” on lonestar; not sure if it works on trestles.
Running “Hello world” as a qsub job

• Type: qsub run_hello.sh
• Type: qstat –u myname
• When it finishes, you should see run_hello.eXXXX and run_hello.oXXXXX in either your current directory or your home directory. Look in .eXXXXX to find error messages. Look in .oXXXXX to find “Hello, world!”
6. Uploading your data
Uploading your data

- XUP java applet crashed on me.
- I haven’t tried globus.
- sftp should work, if you’re careful to first cd to a directory where you have enough space (not home directory on lonestar). For me, unfortunately, it crashed.
Uploading your data: three-step method. (1) create a web page.

- ssh to mickey.ifp.illinois.edu
- mkdir public_html
- cd public_html;
- echo “Look ma, a web page” > foo.html;
- chmod a+r foo.html
- Go to your web browser, and load your new web page. If you have an IFP account, it should show up under http://ifp.illinois.edu/~yourname or http://isle.illinois.edu/~yourname. If not, ask Paritosh; might take an extra day.
Uploading your data: three-step method. (2) upload to isle

On your own PC:
# tar up the complete Resource Management corpus
tar czvf rm_comp.tgz rm_comp;
# Upload it to mickey
sftp myname@mickey.ifp.illinois.edu;
cd public_html;
put mydata.tgz
exit;
# Make it world-readable
ssh myname@mickey.ifp.illinois.edu;
chmod a+r public_html/mydata.tgz;
Uploading your data: three-step method. (3) download to XSEDE

• On the XSEDE machine of your choice, in a directory whose quota permits you to do this:

```bash
wget http://isle.illinois.edu/~yourname/rm_comp.tgz;
tar xzvf rm_comp.tgz
```

• Be sure to remove rm_comp.tgz from your mickey.isle.edu directory afterward, since your disk quota is probably exceeded because of that file.
7. kaldi tutorial
Kaldi tutorial: overview of the distribution

• Start your kaldi tutorial with the web page titled "overview of the distribution"
• Run all of the commands listed here in interactive mode, in your ssh window, to see the results
Kaldi tutorial: overview of the distribution

It’s useful to know the full pathname of the commands you’re using, e.g.,
/work/00680/tg458371/kaldi/trunk/tools/openfst/bin/fstcompile
kaldi tutorial: Resource Management

Next, do “Running the example scripts” using the Resource Management corpus.
Oops, look at the time...

- That’s as far as I’ve managed to get.
- Obviously, running kaldi on XSEDE is a little harder than running it on your laptop, but...
- If you run it on XSEDE, you can run on 48 cpus at once. For a large enough experiment, that is a pretty big reason to use XSEDE.