

## QUICK START GUIDE TO THE SCHOOL OF MATH CLUSTER – BALSA

Updated 10-27-2009

1. Change your password in 24 hours. Email [support@math.gatech.edu](mailto:support@math.gatech.edu) if you have any questions.
2. From outside the math network, you can connect to <ssh.math.gatech.edu> using ssh command line clients, SecureCRT, Putty, etc. This host can also be used for ssh tunneling. Note that your home directory on balsa is different than your home directory on the math network. In particular, you may need to copy files around a bit:

[http://people.math.gatech.edu/~llefton/4777/how\\_to\\_use\\_ssh\\_tunnels\\_for\\_file\\_transfer\\_to\\_balsa.txt](http://people.math.gatech.edu/~llefton/4777/how_to_use_ssh_tunnels_for_file_transfer_to_balsa.txt)

3. The first time you connect to balsa, you will be asked to generate an ssh key used for cluster specific commands. In this special setting, it is recommended that you choose an empty passphrase for convenience, but do not use this ssh key outside the cluster.

4. Once you are on balsa, you can ssh to any of the nodes. Currently we have:

compute-0-1, compute-0-6,

compute-0-10, compute-0-12: 24 Gb RAM, 8 cores (2 quad core CPUs), Intel Xeon E5520 (Nehalem) 2.26GHz

compute-0-11: 8 Gb RAM, 8 cores (4 dual core CPUs), AMD Opteron 2220, 2.8 GHz

compute-0-13: 16 Gb RAM, 8 cores (4 dual core CPUs), AMD Opteron 2220, 2.8 GHz

compute-0-18 – compute-0-20: 16 Gb RAM, 8 cores (2 quad core CPUs), Intel Xeon E5430, 2.66 GHz

compute-0-21 – compute-0-24: 16 Gb RAM, 8 cores (2 quad core CPUs), AMD Opteron 2354, 2.2 GHz

compute-0-14, 15, 16, 17: 2 Gb RAM, 2 cores (dual CPUs), AMD Opteron 246, 2.0 GHz

5. You can see the cluster status at <http://balsa.math.gatech.edu> but only from inside the math network. Use a SOCKS proxy (`ssh -D 2222 $USERNAME@ssh.math.gatech.edu`) and you can browse internally. At any given time some may be down. We probably know about it but don't hesitate to email [support@math.gatech.edu](mailto:support@math.gatech.edu) to verify.

6. Example, to run matlab code on compute-0-24 you could do the following:

```
ssh ssh.math.gatech.edu
ssh balsa.math.gatech.edu
ssh compute-0-24
matlab -nojvm -nodisplay < myprogram.m
```

The commands in myprogram.m should not require graphical interaction, but they may read input from other files. If you need to generate graphs or other output, you can have your code use the print capabilities of Matlab to put the output into files.

As you move into larger high performance computing environments, you will see that the computations are done in a distributed and remote fashion, and the graphical interactions are done after the intensive computing step in a post processing phase. Thus, it is a common practice to have your code dump the raw processed data to a file and then use Matlab (perhaps on your regular desktop or laptop) to read in the raw output data and perform any post processing to generate the graphs, etc.

7. You can start your jobs inside a screen. This allows them to live on, disconnected from your terminal such that you may be reconnected to the running job at any time.

```
ssh ssh.math.gatech.edu
ssh balsa.math.gatech.edu
ssh compute-0-24
screen -S my_compute_job
matlab -nojvm -nodisplay < myprogram.m
```

After starting the screen titled `my_compute_job`, you just get a new prompt and it appears that nothing has changed. However this is now a place that you can run jobs, detach from them, even logout, and later return to the same running job to check its status. To detach from a screen, press `<ctrl>+A, D`. To reattach make sure you are on the same node and type `screen -x my_compute_screen`. This application is a very useful utility and has many options, type `man screen` or read through this tutorial <http://www.kuro5hin.org/story/2004/3/9/16838/14935> for more details.

8. Useful `balsa` commands for running multiple jobs: Run a script/command named `myscript` on all available nodes simultaneously: `tentakel -g compute myscript`

9. We use the Rocks cluster distribution. There are many resources available from <http://www.rocksclusters.org/>.

10. Rocks uses `openmpi` as the default MPI library <http://www.open-mpi.org/>. There are many sample programs online. In particular, you can look at

[http://people.math.gatech.edu/~llefton/4777/mpi\\_examples](http://people.math.gatech.edu/~llefton/4777/mpi_examples)

or the ones already on `balsa` under `/opt/mpi-tests`.

Sample compile line is: `mpif90 -o pi3 pi3.f`

Sample run line: `mpirun -np 16 --machinefile $HOME/machines ../bin/mpi-ring`

Note that the file `machines` should contain one hostname on a line for each thread.

11. Torque is the job queueing system we use on `balsa`. See the guide on our internal page (math login required).

<https://www.math.gatech.edu/internal/support/faq/how-do-i-use-torque-job-queueing-system>

12. Compilers and libraries. We have both `gcc` and the portland group compilers. We also have Intel compilers but they haven't been installed yet.

13. Please let us know if you expect to be using a large amount of storage or if your application requires large datasets for input and/or output. We can work with you to use local scratch space on particular nodes.

14. If you are simply running code on the command line, please be considerate. We reserve the right to kill any jobs running on the cluster if necessary. You can use commands like `top` or `uptime` to see how loaded a node is. You can also run your command using the utility `nice` which allows you to run your programs at lower than interactive priority. For example, to run a job called `'program'`, you could start the job with: `nice -19 program &` The ampersand puts the job in the background.