

UofT Machine Learning Computational Resources

2015

Main source of information:

<http://support.cs.toronto.edu/>

Computational resources:

- Your local machine
- Machine learning (ML) group machines
 - ▶ GPU servers: guppy
 - ▶ CPU servers: cluster
- Computer science lab (CSlab) machines
 - ▶ apps
 - ▶ comps

Purpose of machines:

- Your local machine: whatever you feel like doing locally
- ML group machines
 - ▶ guppy: GPU computation
 - ▶ cluster: CPU computation
- CSLab machines
 - ▶ apps: 32-bit Ubuntu for running applications (eMail, browser, editing)
 - ▶ comps: 64-bit Ubuntu for computing, software development, Matlab

Setting up your local machine

- CSlab VPN access
- Printer setup
- eMail
- Home directory
- Mounting other storage (gobi, etc.)

For additional information:

<http://support.cs.toronto.edu/>

ML group machines

GPU computation

- `deeplearning`, `krunch` and `guppy(x)` with $x \in \{1, \dots, 9, 11, 12\}$
- 4-8 GPUs per machine
- Home directory and other storage mounted
- Software installed in `/pkgs` and `/pkgs_local`

CPU computation

- `cluster(x)` with $x \in \{0, \dots, 12, 34, \dots 64\}$
- machines have different CPU, memory configuration
- Home directory and other storage mounted
- Software installed in `/pkgs` and `/pkgs_local`

Login via: `$ ssh guppy1`

For details check

<http://www.cs.toronto.edu/~pocai/mlcompres.html>

CSlab machines

Apps

- `apps(x)` with $x \in \{0, \dots, 3\}$
- 32-bit Ubuntu
- Only home directory mounted
- For installed software check `/var/lib/dpkg/alternatives/`

Comps

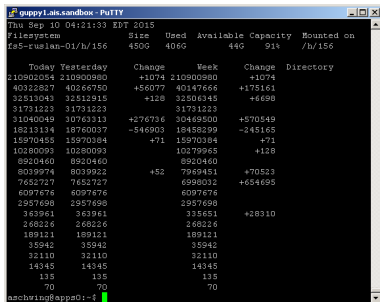
- `comps(x)` with $x \in \{0, \dots, 4\}$
- `compsbk(x)` with $x \in \{0, 2, 3, 4, 5\}$ (need booking)
- 64-bit Ubuntu
- GPUs on `compsbk(3-5)`
- Only home directory mounted
- For installed software check `/var/lib/dpkg/alternatives/`

For installed software and booking of machines also check
<http://support.cs.toronto.edu/>

Storage

Home directory

- Mounted on all machines
- Should be used for code and text
- Regularly backed up
- **Shared with your fellows**
- Check your usage via `$ hogs`



```
Thu Sep 10 04:21:33 EDT 2015
filesystem      Size  Used  Available Capacity  Mounted on
fs5-russian-01/h/156  450G  406G  44G      91%    /h/156

  Today Yesterday   Change   Week   Change Directory
210902054 210900980      +1074 210900980      +1074
40322827 40266750      +56077 40147666      +175161
32513043 32512915          +128 32506345      +6698
31731223 31731223
31040049 30763313      +276736 30469500      +570549
18213134 18760037      -546903 18458299      -245165
15970455 15970384          +71 15970384          +71
10280093 10280093
8920460 8920460
8039974 8039922          +52 7969451      +70523
7652727 7652727
6097676 6097676
2957698 2957698
363961 363961      335651      +28310
268226 268226
189121 189121
35942 35942
32110 32110      32110
14345 14345      14345
135 135
70 70
nachwing@ppp0:~$
```


Storage

File servers: gobi

- Mounted on ML group machines (guppy, cluster)
- Accessible on apps and personal machine via sshfs only
- Used for storing custom data
- Accessible, e.g., via `$ cd /ais/gobi3/u/<username>`

Local disks:

- Mounted on local machine only (?)
- Used for storing temporary custom data
- Accessible, e.g., via `$ cd /nobackup/<username>`
- Sometimes SSD hard-disks available, names can change: `$ cd /nobackup_c/<username>`

Ask Relu to get a file server directory right away. Local disk directories should be created on a need basis.

Storage

Datasets:

- Mounted on ML group machines (guppy, cluster)
- Used for storing datasets (everyone can contribute)
- Accessible via `$ cd /ais/gobi3/datasets`

Hint:

Use symbolic links from your home directory to data storage. Close to conference deadlines the file system might respond slowly.

Software and Libraries

ML group machines

- `$ /pkgs` and `$ /pkgs_local` directories
- contains libraries and software, e.g., CUDA, MPI, python, matlab, boost, caffe, etc.

CSlab machines

- `$ /var/lib/dpkg/alternatives/`

GPU Usage

- Lock GPU before usage to signal that you'd like to use it
- Better not use a GPU that is locked

Example:

```
$ python ~tang/bin/gpu_lock2.py
$ python ~tang/bin/gpu_lock2.py --free
$ python ~tang/bin/gpu_lock2.py --id
$ python ~tang/bin/gpu_lock2.py --ids
```

For useful commands: http://www.cs.toronto.edu/~nitish/gpu_lock.html

GPU Usage

Want to find an available GPU?

Repeat until convergence

- Login to a machine
- Check GPU usage
- Break if free GPU found

Sounds cumbersome? Use a shell script:

```
declare -a arr=("guppy2" "guppy3" "guppy4" "guppy5" "guppy6" "guppy7" "guppy8" "guppy9" "guppy11")
for i in "${arr[@]}"
do
    echo "$i"
    echo "-----"
    OUTPUT="$(mpiexec -n 1 -machine "$i" python ~tang/bin/gpu_lock2.py)"
    echo "$OUTPUT"
done
$
```

Note: GPU IDs 0 and 3 in `krunch` are interchanged

Interaction with servers

ssh (**very slow!**)

- `-X` or `-Y` option for opening GUI applications
- `screen/tmux` command

VNC

- Use less powerful machines for running the vnc server. E.g., `guppy5`, `guppy6`
- Configure using `~/ .vnc/xstartup`

Happy number crunching!