Scientific Computing Resources and Current Fair Usage

Quentin CAUDRON
Ellak SOMFAI
Stefan GROSSKINSKY

Centre for Complexity Science,
University of Warwick

Complexity DTC Annual Retreat, 2011

Documentation

Collected documentation links :

Computing section on `http://go.warwick.ac.uk/qcaudron`

The Cluster of Workstations

"The CoW is a system which manages computational work across the network of CSC-managed computers."
- CSC Wiki

- Collection of campus-wide desktop computers in a network
- Computers are for desktop-use, not dedicated compute nodes
- Allows a fair, noninvasive distribution of computational jobs
- Accessed through `Godzilla`, using a `Torque` system

Godzilla Access

From Windows :

- NX Client
- PuTTy

From Linux / Mac OS :

- `ssh username@godzilla.csc.warwick.ac.uk`
- `ssh godzilla.csc.warwick.ac.uk -l username`

`Godzilla` is an access point **only**, and should **never** be used for **any** computationally-intensive purposes.

So Why `Godzilla` ?

`Godzilla` allows access to the CoW, to submit jobs to the queue. It also allows `ssh` access to any other computer on the network.

name`.csc.warwick.ac.uk`

- paris
- rome
- lisbon
- madrid
- bern
- oslo
- prague
- watson

name`.complexity.warwick.ac.uk`

- palermo
- timbuktu
- valea

Current State

The CoW has been switched off in the medium term due to misuse leading to serious networkwide issues.

Desktops remain accessible and usable, but the ability to submit jobs to the queue for correct distribution on the CSC desktop network is unavailable.

Alternative methods for high performance computing remain available :

- Francesca and Minerva HPC systems
- Standard desktops

Francesca

`Francesca` is the *old* supercomputing cluster available for scientific computing use.

- 960 cores at 3 GHz
- 1.92 TB RAM
- Architecture : 240 nodes of 2 dual-core processors, 8 GB per node

`Francesca` has now been superceded by `Minerva` after reaching End of Life, but remains accessible, despite no longer being actively supported by CSC admins.

Minerva

`Minerva` is a more powerful, newly-introduced supercomputing cluster.

- 3096 compute cores at 2.66 GHz
- 6.2 TB RAM, plus 290 GB across two data processing nodes
- 12 NVIDIA Tesla M2050 GPU nodes, with 448 CUDA cores
- CPU Architecture : 258 nodes of 12 cores, 24 GB per node
- GPU Architecture : 6 nodes of 2 GPUs, 48 GB per node

Parallel Computing Resources

Both Francesca and Minerva are for use with parallelised codes. They offer no advantage to serial code. However, Francesca can be used for trivially-parallelisable averaging batch jobs.

Both of these supercomputers require manually-approved registration.

Both have an associated cost.

For non-communicating parallel applications such as averaging simulations, the CoW has a batch job mode.

Desktops

The desktop computers within Complexity are free-access.

From your laptops, they can be accessed through Godzilla if using Windows, or directly from Linux or Macs.

From here, you have access to your own files but are connected remotely to another computer. You may run jobs as you would locally.

However, this will lead to desktops becoming unresponsive and unusable. The user at the desktop will most likely reboot the computer, leading to the loss of any currently running simulations, both yours and theirs.

ssh

To access another computer :

- `ssh computername.csc.warwick.ac.uk`
- `ssh computername.complexity.warwick.ac.uk -X`

Then, you can run any code as normal.

- `gcc mycode.c -o myCompiledCode`
- `./myCompiledCode`

ssh

To access another computer :

- `ssh computername.csc.warwick.ac.uk`
- `ssh computername.complexity.warwick.ac.uk -X`

Then, you can run any code as normal.

- `gcc mycode.c -o myCompiledCode`
- `./myCompiledCode`

(but please, don't)

Checking the Computer's Load

The computer you have sshed into may have two or four cores. These cores may be more or less occupied by current jobs.

- `top`
- `w`

These commands allow you to check how busy a computer is. Please find one that isn't being overwhelmed by other jobs before putting your own on it.

Checking the Computer's Load

```
top - 10:20:41 up 11 days, 22:20, 11 users,  load average: 0.02, 0.03, 0.03
Tasks: 181 total,   3 running, 177 sleeping,   1 stopped,   0 zombie
Cpu(s): 15.3%us,  1.5%sy,  0.0%ni, 83.2%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3990556k total,  3925524k used,    65032k free,   142228k buffers
Swap:  2088956k total,   454520k used,  1634436k free,  2406572k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6239 phrfat    20   0 1525m 671m  19m R   33 17.2 643:39.76 firefox
20085 phrfat    20   0 95200 9.9m 2688 R    1  0.3  67:41.20 npviewer.bin
 1763 root      20   0 24396  728  628 S    0  0.0   3:51.27 hald-addon-stor
17437 phrfat    20   0  8672 1160  788 R    0  0.0   0:00.06 top
    1 root      20   0 12408  364  364 S    0  0.0   0:06.42 init
    2 root      20   0     0    0    0 S    0  0.0   0:00.11 kthreadd
    3 root      RT   0     0    0    0 S    0  0.0   0:00.25 migration/0
    4 root      20   0     0    0    0 S    0  0.0   0:11.39 ksoftirqd/0
```

```
phrfat@oslo:~> w
 10:14:05 up 8 days, 19:51,  5 users,  load average: 1.75, 1.65, 1.31
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
phrfat   pts/1     10:14    0.00s  0.45s  0.00s w
phreax   pts/6     Mon15   42:19m  0.00s  5.00s kdeinit4: kded4 [kdeinit]
```

nice

The Linux `nice` command allows you to set any task's priority as you launch it.

- `nice -n 18 myCompiledCode`
- `renice -n 18 -p PID`

Any job run on another computer **must** be run in this way. The `nice` value of 18 specifies a low priority which will not reduce a computer to a sputtering mess due to overloading.

If you forget to run the job using `nice`, you can use `top` for find out the Process ID and then run `renice` to correct the value.

finger

If your desktop is unresponsive, you can find out which user is responsible for the computationally-intensive jobs using `top`. The `finger` command allows you to look up the owner of the username from found from `top`.

```
phrfat@watson:~> finger phrfat
Login: phrfat                           Name: Quentin Caudron
Directory: /home/phrfat                 Shell: /bin/bash
```

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

We want to find out what the current load is on this computer.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

The `load average` over the last 1, 5 and 15 minutes is 1.0.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

75% of the processor is idle.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

The remaining load is not due to userspace loads.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

The system is also not responsible for the 25% load.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

We see that the load is due to `niced` processes.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

In more detail, we can look at what processes are running.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

User `phrfat` is running `top` under *normal* priority.

Case Study

```
phrfat@oslo:~> top
top - 11:20:28 up 8 days, 20:58,  6 users,  load average: 1.00, 1.00, 1.00
Tasks: 224 total,   2 running, 222 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.1%sy, 24.9%ni, 74.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4057796k total,  3934108k used,   123688k free,   144488k buffers
Swap:  2088956k total,     8296k used,  2080660k free,  2833352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6199 phrhat    38  18 21588 4164 1404 R  100  0.1   5523:50 exec
   41 root      20   0     0    0    0 S    0  0.0   1:48.98 kondemand/0
30387 phrfat    20   0  8788 1276  852 R    0  0.0   0:00.03 top
```

User `phrhat` is running their own code, `exec`, using `nice` value 18.

Case Study

Summary

For serial codes :

- Find a computer with a low load, using `top`
- Run your job using `nice -n 18`
- Please do not abuse the desktops - distribute fairly

For repeat jobs or averaging :

- Submit to Francesca's queue

For computationally-demanding parallel jobs :

- Submit to Minerva's queue