Wiki

Child pages:

Bash tricks

Get disk usage

```
df -h
```

Will give you something like this:

```
Sys. de fichiers       Taille Utilisé Dispo Uti% Monté sur
/dev/sde1                110G    8,1G   97G   8% /
devtmpfs                 126G       0  126G   0% /dev
tmpfs                    126G       0  126G   0% /dev/shm
tmpfs                    126G    920K  126G   1% /run
tmpfs                    126G       0  126G   0% /sys/fs/cgroup
tmpfs                    126G    4,0K  126G   1% /tmp
/dev/mapper/ddf1_DATA2    15T    7,7G   14T   1% /home
```

Get CPU and RAM usage

```
htop
```

Will give you something like this:

htop.png

To get more information about htop see here

Start a Job without a queue

All process started from ssh are terminated when you close the ssh connection, even if you fork them (./app &). They are closed because when a process is closed the system send the SIGTERM signal to all its children, it's done to avoid zombies process on a machine. To keep your task alive when you disconnect from ssh, you should use screen (tutorial), it will block the SIGTERM signal. To start your application with screen:

```
screen # to start screen
./your_app   #to start your application or any other command
             #type ' Ctrl-A' d to leave screen with your application running in background
```

To reconnect to your previous session:

```
screen -ls # to list running sessions
screen -r 33287.pts-36.bender # to reconnect to 33287.pts-36.bender session
exit # to close your screen session
```

Start a mpi code:

Please note that the default mpi distribution on Bender /Flexo /Jakolass is a custom one compiled with intel compiler and libraries, to use the gcc one you should call mpi(cc/f90/run/...) with the full path /usr/lib64/openmpi/bin/mpi(cc/f90/run/...). To run a software compiled with mpi (mpic/cxx/fortran), uses mpirun with -np to set the number of mpi processes you want to start.

```
mpirun -np 32 /path_to_myapp/myapp
#to run myapp with 32 MPI processes
```

Start a openMP code:

When you run a software compiled with openMP library, you can tune the number of openMP threads your code will run. To do this you just have to set the OMP_NUM_THREADS environment variable.

```
export OMP_NUM_THREADS=32
/path_to_myapp/myapp
#to run myapp with 32 openMP threads
```

Abort a Job:

To stop a running job you can use the kill command in different ways

- You have just on Job running on the machine and you know its name

```
killall TheNameOfTheJob
```

- You have multiple jobs running, you have first to get the Process ID of the job

```
ps ax # will list the running processes
# You can filter, for example if you have started your job with mpi
ps ax | grep mpi
```

The first number you get is the Process ID

To kill it

```
kill -9 123456 # if 123456 is your Process ID
```

Files

htop.png	133 KB	06/11/2013	Alexis Jeandet

Building NetCDF with intel's openMPI

To build netCDF with openMPI based on intel compiler, first you have to folow this steps, then:

```
export CC=mpicc
export F77=mpif77
export F90=mpif90
export FC=mpif90
#export LIBS="-lirc"
export CFLAGS="-fPIC -DgFortran"
export CXXFLAGS="-fPIC -fno-second-underscore -DgFortran"
export FFLAGS="-fPIC -fno-second-underscore -DgFortran"
export F90FLAGS="-fPIC -fno-second-underscore -DgFortran"
export LDFLAGS="-fPIC"
./configure --prefix=/opt/netcdf --disable-shared --disable-dap --enable-f90
make -j 64  # or the number of cores you have
```

Then to install it, you must do it as super user:

```
sudo make install
```

Once installed you need to tell the linker the library path, to do so:

```
su root  #sudo -s on ubuntu
echo "/opt/netcdf/lib" > /etc/ld.so.conf.d/impinetcdf.conf
ldconfig  #Tell the linker to update cache
exit  #leave root session
```

Building openMPI with intel compiler

To build openMPI with intel compiler:

```
export CC=icc
export CXX=icpc
export F77=ifort
export F90=ifort
export FC=ifort
export LIBS="-lirc"
./configure
make -j 64  # or the number of cores you have
```

Then to install it, you must do it as super user:

```
su root   #sudo -s on ubuntu
make install
exit  #leave root session
```

Once installed you need to tell the linker the library path, to do so:

```
su root  #sudo -s on ubuntu
echo "/usr/local/lib" > /etc/ld.so.conf.d/iopenmpi.conf
ldconfig  #Tell the linker to update cache
exit  #leave root session
```

Computers

---

bender.jpg
Bender

Hardware

- Motherboard SuperMicro H8QG6-F
- Chassis SuperMicro superchassis 748TQ-R1400BTOUR
- 32 cores, 4x AMD Opteron(tm) Processor 6128 (8 cores each) @2GHz.
  - L1 per core cache Instructions 64kB Data 64kB.
  - L2 per core 512kB.
  - L3 2x 6MB shared.
- 128GB, 16x8GB DDR3 @1333MHz ref BLS4CP8G3D1609DS1S00BEU.
- 7TB, 2x 3TB + 1x 1TB + 2x 256GB SSD (reserved for the OS).
  - /dev/sdd Seagate Barracuda 3TB ref "ST3000DM001", Part 1 mounted as /DATA1.
  - /dev/sde Seagate Constellation 3TB ref "ST3000NM0033", Part 1 mounted as /DATA2.
  - /dev/sdb OCZ Vertex 256GB ref "OCZ-VERTEX2 3.5"; Part 1 mounted as /.
- Geforce GTX260 for GPGPU or visualization.

Software

- Fedora Linux 17 64 bits
- gcc 4.7.2
- openMPI
  - local openMPI 1.5.4
  - ICC openMPI 1.6.5
- openMP 3.0
- icc & MKL 13.1.2.183 Build 20130514
- IDL 8.2.3

---

bender.jpg
Flexo

Hardware

- Motherboard SuperMicro H8QG6-F
- Chassis SuperMicro superchassis 748TQ-R1400BTOUR
- 64 cores, 4x AMD Opteron(tm) Processor 6274 (16 cores each) @2.2GHz.
- 256GB, 32x8GB DDR3 @1333MHz.
- 15TB + 128GB SSD (reserved for the OS).
  - /dev/sde Samsung Evo SSD 120GB ref "Samsung SSD 840 EVO 120GB" Part 1 mounted as /.
  - /dev/mapper/ddf1_DATA2 4x Hitachi Deskstar 4TB in Raid 0 mounted as /home.
- Geforce GTX760 with 4GB of RAM for GPGPU.

Software

- Fedora Linux 19 64 bits
- gcc 4.8.2
- openMPI 1.6.5(default based on icc)
- openMP
- icc 13.1.2 & MKL
- Matlab 2013a and all general purpose libraries
- CUDA5.5

---

bender.jpg
Jakolass

Hardware

- Motherboard SuperMicro H8QG6-F
- Chassis SuperMicro superchassis 748TQ-R1400BTOUR
- 64 cores, 4x AMD Opteron(tm) Processor 6274 (16 cores each) @2.2GHz.
- 256GB, 32x8GB DDR3 @1333MHz.
- 15TB + 128GB SSD (reserved for the OS).

Software

- Fedora Linux 19 64 bits
- gcc 4.8.2
- openMPI 1.7.3(default based on icc)
- openMP
- icc 13.1.2 & MKL
- Matlab 2013a

Files

bender.jpg	525 KB	18/03/2014	Alexis Jeandet

Purge Linux Cache Memory

Sometime the linux kernell uses a lot of cache memory, it can prevent you from allocating memory in your programs. To solve this issue, one solution is to periodically force the kernell to flush its caches buffers.

```
mkdir -p /opt/scripts
echo '#!/bin/bash' >  /opt/scripts/clearcachemem.sh
echo 'sync; echo 3 > /proc/sys/vm/drop_caches' >>  /opt/scripts/clearcachemem.sh
chmod +x /opt/scripts/clearcachemem.sh # make it executable
```

Then edit cron config with:

```
sudo crontab -e
# enter insert mode with "i"
# paste "0 0 * * * /opt/scripts/clearcachemem.sh"
# exit with "echap" key then type ":wq" then "return" key
```

System Backups

Clonezilla images of compute machines are currently stored in :

```
\\ibis\Mymac\EquipeInformatique\Images-CloneZilla\
```