Introduction to Abel/Colossus and the queuing system

March 21, 2018

Sabry Razick

Research Infrastructure Services Group, USIT
Topics

• First 7 slides are about us and links
  • The Research Computing Services group
  • Abel/Colossus details, Getting an account & Logging in
  • Understanding resources
• Queuing system
• Running a simple job
The Research Computing Services
Seksjon for IT i Forskning

• The RCS group provides access to IT resources and high performance computing to researchers at UiO and to NOTUR users

• http://www.uio.no/english/services/it/research/

• Part of USIT

• Contact:
  • Abel : hpc-drift@usit.uio.no
  • TSD : tsd-drift@usit.uio.no
The Research Computing Services

• Operation of high performance computer clusters
• User support
• Data storage
• Secure data analysis and storage - TSD
• Portals
  • Lifeportal (https://lifeportal.uio.no/)
  • Lap Language (https://lap.clarino.uio.no/)
  • Geo (https://geoportal-dev.hpc.uio.no)
Abel/Colossus

• Computer cluster
  – Similar Computers connected by a local area network (LAN). Different than a Cloud or a Grid.
• Enables parallel computing
• Science presents multiple problems of parallel nature
  – Sequence database searches
  – Genome assembly and annotation
  – Simulations
Bunch of computers -> Cluster

- **Hardware**
  - Powerful computers (nodes)
  - High-speed connection between node
  - Access to a common file system

- **Software**
  - Operating system: 64 bit Centos 6.8 (Rocks Cluster Distribution based)
  - Identical mass installations.
  - Queuing system enables timely execution of many concurrent processes
Numbers

- Nodes - 700+ (Abel - 703), (Colossus - 68)
- Cores - 10000+ (Abel - 11,392), (Colossus - 1392)
- Total memory - 50 TiB+ (Abel 50), (Colossus 5)
- Total storage - 400 TiB using BeeGFS
- 96th most powerful in 2012, now 444th (June 2015)
Getting access

• If you are working or studying at UiO, you can have an Abel account directly from us.

• If you are Norwegian scientist (or need large resources), you can apply through NOTUR –
  • [https://www.sigma2.no/](https://www.sigma2.no/)

• Write to us for information:
  • hpc-drift@usit.uio.no / tsd-drift@usit.uio.no

• Read about getting access:
  • [http://www.uio.no/hpc/abel/help/access](http://www.uio.no/hpc/abel/help/access)
  • [https://www.uio.no/english/services/it/research/storage/sensitive-data/access/](https://www.uio.no/english/services/it/research/storage/sensitive-data/access/)
Connecting to Abel

- Linux
  - Redhat - RHEL
  - Ubuntu
- Windows - using Putty, Gitbash, WinSCP
  - https://git-for-windows.github.io/
- Mac OS
Connecting to Colossus

- Must come through TSD and need an account to use Colossus.
- Linux and Windows VMs (virtual machines)
- Thinlinc
  - https://login tl.tsd.usit.no/main/
- BLAST
  - https://view.tsd.usit.no/
Available software

• Available:  
  [http://www.uio.no/hpc/abel/help/software](http://www.uio.no/hpc/abel/help/software)

• Software organized as modules.
  – List all software (and version) organized in modules:
    • module avail
  – Load software from a module:
    • module load module_name
    • (e.g module load python/2.9.10)

• Install your own software
  – Separate lecture HPC 05.04.2017, 09:15 (Wednesday)
Using Abel

• When you log into Abel you are in one of the login nodes login0 - login3.

• Please **DO NOT** execute programs (jobs) directly on the login nodes.

• Jobs are submitted to Abel via the queuing system.

• The login nodes are just for logging in, copying files, editing, compiling, running short tests (no more than a couple of minutes), submitting jobs, checking job status, etc.

• For interactive execution use qlogin.
Login node

Compute nodes

Qlogin

Batch script

Compute nodes
We do NOT run jobs here.
Using Colossus

- You may run jobs in your VM if you wish
- For heavy jobs use Colossus (procedure same as Abel)
- There is no qlogin on TSD
Queue management - SLURM

- Simple Linux Utility for Resource Management (workload manager)
- Allocates exclusive and/or non-exclusive access to resources (computer nodes) to users for some duration of time
- Provides a framework for starting, executing, and monitoring work on a set of allocated nodes.
- Managing a queue of pending work.
Fair resource allocation

- When you request resources, SLURM will consider number of things before granting it
  - Does your project has enough CPU hours to pay for this. It will consider total allocated and reserved (running jobs) when doing this.
  - Is your account using more than the allowed resources.
  - Can/should the cluster provide you with resources (resource combination)
  - Depending on the current load. how long others need to wait if you job starts.
SLURM
Computing on Abel

• Submit a job to the queuing system
  – Software that executes jobs on available resources on the cluster (and much more)

• Communicate with the queuing system using a shell (or job) script

• Retrieve results (or errors) when the job is done

• Read tutorial: http://www.uio.no/hpc/abel/help/user-guide
Running a job on a laptop compared to submitting to a queue

- Double click on an icon, give parameters or upload data - wait
- Terminal
  - ./myprg input1.txt out_put.file
- Inspect results
Interactive login (Qlogin)
Abel only
Running a job on the cluster - 1

- Login to Abel from your laptop
- Request to occupy some resources from SLURM
- Wait until SLURM grants you the resources
- Execute the job as it was in your laptop
Job script
Abel and Colossus
Running a job on the cluster - 2

- Login to Abel from your laptop
- Create a job script, with parameters and include the program to run
- Hand it over to the workload manager
- The workload manager will handle the job queue, monitor the progress and let you know the outcome.
- Inspect results
Resources
- Supermicro X9DRT compute node
- Dual Intel E5-2670 (Sandy Bridge) based running at 2.6 GHz (2 sockets)
- 16 physical compute cores.
- Each node have 64 GiBytes of Samsung DDR3 memory
tasks?

- A piece of work to be done
- The computing resource needed for that
- A normal compute node on abel has two processors which can do 8 things each
- So a compute node can do 16 things at once
#SBATCH --ntasks=8
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
8 \times 1 = 8
*All tasks will share memory*

#SBATCH --ntasks=2
#SBATCH --cpus-per-task=8
8 \times 2 = 16

#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
2 \times 4 = 8
Calculating CPU hours

- Use one task for one hour = 1 cpu hour
- If you use one entire compute node for one our
  - 16 X 1 = 16 CPU hours
- For more precise value - next slide
Calculating CPU hours

KMEM=4580.2007628294
(/cluster/var/accounting/PE_factor)

PE= NumCPUs

if(MinMemoryCPU>KMEM){
    PE=PE*(MinMemoryCPU/KMEM)
}

PE_hours = $PE * TimeLimit / 3600
#SBATCH --nodes=1
#SBATCH --time=01:00:00
#SBATCH --ntasks-per-node=4
#SBATCH --mem-per-cpu=15G

(4 X 1) + 12 = 16

*All memory occupied,

KMEM=4580.2007628294
PE = 4

#(15 * 1024)>KMEM so
PE = 4 * ((15 * 1024) / KMEM) = 13.41
PE_hours = 13.41 * (1 * 60 * 60) / 3600 = 13.41

**Use the command `cost` to check account balance.**
Project/Account

• Each user belongs to one or more project on Abel
• Colossus uses use the same value as their TSD project
• Each project has set of resources
• Learn about your project(s):
  – Use: projects
Arrayrun

- Parallel jobs - executing many instances of the same executable at the same time.
- Many input datasets
- Simulations with different input parameters.
- Possible to split a large input file into chunks and parallelize your job.
MPI

- Message Passing Interface
- MPI is a language-independent communications protocol used for programming parallel computers.
- We support OpenMPI
  - module load openmpi
- jobs specifying more than one node automatically get
  - #SBATCH --constraint=ib
- More on this Thursday
Thank you.

http://www.uio.no/english/services/it/research/hpc/abel/

hpc-drift@usit.uio.no