UPPMAX System Rackham

1 Computer System

In Lab 2 and in the project we will be using the Rackham compute cluster at Uppmax. Rackham consists of 334 compute servers (nodes) where each compute server consists of two 10-core Xeon E5-2630 V4 processors running at 2.2 GHz (turbo 3.1 GHz). We provide 272 nodes with 128 GB memoryi (r33-r304) and 32 nodes with 256 GB (r1-r32). All nodes are interconnected with a 2:1 oversubscribed FDR (56 GB/s) Infiniband fabric. In total Rackham provides 6080 CPU cores in compute nodes.

2 Account on UPPMAX

To get access, you need to get a login account which you get by following these steps:

1. Sign up for a SNIC account at `https://supr.snic.se/`. Click "Register New Person", and then "Register Via SWAMID" to register with your existing UU login.
2. Login with your new SNIC account, and request membership in the course project. Click on "Projects" in the left menu bar, search for project "g2018006" in the search field, and then click request. I will check your course registration status, and approve the request.
3. Once approved, sign up for an Uppmax login account by clicking on "Accounts" in the left menu bar, and then following the instructions.

Once you have received an email confirming the creation of your Uppmax account, verify that you can login by issuing the following `ssh` command. If it doesn't work immediately, please wait a few minutes and try again since there might be a small setup delay.

`ssh -AX` *`your-new-username`*`@rackham.uppmax.uu.se`

As stated in the Uppmax account email, the first thing you should do is change your password which is done using the `passwd` command.

SLURM The Uppmax systems use the batch system SLURM to control the allocation of resources and submission of compute jobs. Although Uppmax has been quite generous with resources, we do have a limited quota of core hours which all students will share. Make sure to limit your runs both in time and in the number of cores you allocate, as to not prevent other students from working. Here, there is no strict enforcement preventing overuse; rather, a "freedom with responsibility" approach is used. Deliberate misuse will be detected and can result in suspension from the computer system.

For more information on SLURM and how to use the Uppmax clusters read the following useful links:

- `https://www.uppmax.uu.se/support/user-guides/guide--first-login-to-uppmax/`

- https://www.uppmax.uu.se/support/user-guides/slurm-user-guide/
- https://www.uppmax.uu.se/support/user-guides/rackham-user-guide/

3 Running programs

Once you have your account set up, you can login to the Rackham system using `ssh`:

```
$ ssh -AX your-user-name@rackham.uppmax.uu.se
```

On Uppmax, the `module` system is used to enable various packages. Using `module avail` you can see all the available modules. For this course, we are going to need `OpenMPI`, which in turn requires a recent version of the `GCC` compiler. Run the following command to enable both of them:

```
$ module load gcc openmpi
```

Now you can compile and run programs on the login node of Rackham. This is a noisy environment where the activity of other users affect the run time. To get exclusive access to a number of cores and to get more accurate measurements, we need to run the benchmark as a dedicated job on the compute nodes using the batch system SLURM. Specifically, a job is created using a batch script which is submitted using `sbatch`. Look at the script `hello.sh`;

```
#!/bin/bash -l
#SBATCH -A g2018006
#SBATCH -t 5:00
#SBATCH -p core -n 2

module load gcc openmpi
mpirun hello
```

it first contains a number of comment lines with configurations to `sbatch`, followed by the actual commands to run. `hello.sh` is configured for allocation of 2 cores for a maximum runtime of 5 minutes, and also specifies the course project ID. Submit the job by issuing

```
$ sbatch hello.sh
```

The output is now written to the file `slurm-<jobid>.out`.

Note: To see a list of your jobs in the queue, use the `squeue` command:

```
$ squeue -u <your-user-name>
```

To cancel a running or queued job, use the `scancel` command:

```
$ scancel <job-id>
```

If you want to run on more than 20 cores using several nodes then you need to change the partition in the script to node, e.g. running on two full nodes give `#SBATCH -p node -n 40`. If you request a large number of cores it can take long time (days) until you are allocated resources for your run. Also, when

running large jobs be sure that you have tested your program so that it is correct not waisting core hours reserved to the course. Note that the runtime is multiplied with the number of cores requested. Be sure to make your estimate of the maximum runtime with care, not too long if something goes wrong and not too short so that your run is not terminated early. Again, go through the user guides listed above to get more options.