Technische Universität München
Lehrstuhl für Informatik V
P. Neumann
B. Uekermann

SS 2016

MAC-Cluster

In the following, the steps to execute and compile the program on the cluster are briefly described. For a more detailed description of the cluster, available libraries/modules etc., check out the webpage on the MAC-cluster at www.mac.tum.de.

1 Environment and Login

The MAC-cluster is a heterogeneous environment in the sense that it comprises various types of processors (AMD Bulldozer systems and various kinds of Intel architectures). You can log in to the MAC-cluster system using either the AMD or Intel login nodes

- AMD: mac-login-amd.tum-mac.cos.lrz.de
- Intel: mac-login-intel.tum-mac.cos.lrz.de

and your login at the LRZ.

2 Compiling

Various modules, that is libraries and tools, are available on the cluster. Type `module avail` for a complete list. The command `module list` shows the currently activated modules in your account. To load/unload a module, type `module load/unload MODULENAME`.

For compiling, you probably want to modify your Makefile: choose the compiler wrapper `mpiCC` in the makefile.

3 Execution

There are basically two methods to run your parallel program on the MAC-cluster: interactively or via batch jobs.

- **Interactive jobs**: you can use the command `salloc` to allocate resources for your parallel execution:

  ```
  salloc -N SOME_NUMBER -p SOME_PARTITION
  ```

  Example: `salloc -N 1 -p bdz` allocates one node (64 cores) of the bulldozer partition. Using `salloc` opens a new shell and gives you exclusive access to the respective resources and you can immediately run your program in the usual way using `mpirun -np ...`. Make sure to free the resources immediately at the end of your session so that other cluster users obtain access to them (just close your session typing Ctrl+D)!

- **Batch jobs**: write a batch job file and submit it to the queuing system of the MAC-cluster. Detailed information on the respective job files is given on the login screen after logging in to the cluster. An exemplary job (let's call it `myjob.job`) file is given in the following:

  ```
  #!/bin/bash
  #SBATCH -o /home/hpc/pr63so/MYLOGIN/MYJOB.out
  #SBATCH -D /home/hpc/pr63so/MYLOGIN/
  #SBATCH -J MYJOB
  #SBATCH --get-user-env
  #SBATCH --partition=bdz
  #SBATCH --ntasks=512 # 8 nodes * 64 processes per node = 512
  #SBATCH --cpus-per-task=1 # one thread per process
  #SBATCH --mail-type=end
  #SBATCH --mail-user=MYEMAIL@MYEMAIL.COM
  #SBATCH --export=NONE
  #SBATCH --time=02:00:00
  source /etc/profile.d/modules.sh

  # -ppn = processes per node
  # -n   = total number of processes
  mpiexec.hydra -genv OMP_NUM_THREADS 1 -ppn 64 -n 512 /home/hpc/pr63so/MYLOGIN/ns
  /home/hpc/pr63so/MYLOGIN/conf512proc.xml
  ```

From the directory which contains the job file, submit this job by typing

sbatch myjob.job.

You can check the status of your job by typing `squeue`. This will show you the current queue of the cluster.

Note: to have a single process per core on the snb partition, specify `--cpus-per-task=2`.

Note: the file space is limited on the MAC-cluster. The cluster is actually built for code development and testing, but not for production runs! Therefore: if you want to write larger amounts of data, please use the /scratch directory of your account. There's more space available here; however, data is automatically deleted after two weeks. Data within your home directory is backuped and safe.