Best Practice mini-guide "Chimera"

SGI Altix UV at PSNC

Michal Bialoskórski, ACC TASK

Maciej Szpindler, ICM

April 2013

Table of Contents

1. Introduction

This document present best practices for users of SGI Altix UV system "Chimera", installed at Poznan Supercomputer and Networking Center (PSNC) in Poland. Chimera is a compute cluster built on widely used x86_64 architecture with a unique hardware SMP technology which enables to use whole system memory in a single-image mode.

Figure 1. "Chimera" SGI Altix UV, source: Poznan Supercomputer and Networking Center

2. System Architecture and Configuration

The SGI Altix UV system is a shared memory machine built on the basis of widely recognised Intel x86_64 architecture. This section describes SGI UV installation located in PSNC [http://www.man.poznan.pl/], Poznan, Poland.

2.1. System configuration

"Chimera" is SGI Altix UV 1000 shared memory cluster, also referred to as multiprocessor distributed shared memory (DSM). It consists of 2048 Intel Xeon cores with 16 TB of memory controlled by a cache-coherent single image of Linux system. This means that all of memory is available to a single application and is shared by all of the processors in a system. Shared memory capability of the machine is a hardware extension to the commodity processor based nodes of the cluster.

Peak performance of Chimera is 21.8 TFlop/s.

Processor architecture

Chimera is powered by 256 Intel Xeon E7-8837 processors with a following parameters:

- eight-core
- Nehalem micro architecture
- clocked at 2.66 GHz
- 24 MB L3 cache
- 8 x 256 KB L2 cache

Operating system

System runs under control of Debian/GNU Linux 6.

Memory architecture

SGI Altix UV is cache-coherent non-uniform memory access (ccNUMA) architecture with global shared memory available as a single system image (SSI). System uses processor caches to reduce memory latency. Data in local or remote memory is stored in various processor caches throughout the system. Cache coherency mechanism keeps cached copies consistent.

Distributed shared memory (DSM) means that the memory is physically distributed between processor nodes within the system and also placed at various distances from the processors in a certain node. That results in memory access time is dependent on physical memory placement and is non-uniform across the node (NUMA).

Network infrastructure

On Chimera is active one 10Gigabit Ethernet port. It is shared with local connection for storage and public IP address of server.

SGI UV is using vendor specific interconnect called "NUMAlink" to provide global memory shared between the cluster nodes. NUMAlink 5 is capable of 15 GB/s of peak bandwidth through two 7.5 GB/s unidirectional links.

Additional hardware support for parallel execution is available for MPI based applications with MPI Offload Engine also being SGI technology.

Hardware accelerators

The SGI UV system has got hardware accelerator for MPI point-to-point and collective communication. The MPI acceleration is performed by the UV MPI Offload Engine (MOE) through message queues, synchronisation and multicast implemented in hardware. The MOE gives small MPI latency, and improve performance on some commonly used MPI collective communication operations.

All MPI platform installed on Chimera supports this hardware extension.

I/O subsystem

Chimera has got internal disks of capacity 500GB for scratch mounted to `/disks` directory and ram disks accessed from directories `/tmp` and `/dev/shm` of capacity 8TB and 3TB. Ram disks shares capacity with memory designed for programs.

2.2. File systems

On Chimera are available two real storage areas with disk and two pseudo file systems located in ram. Only the home directories are protected with a regular backup.

Table 1. File systems available on Chimera

mount point	file system	capacity	notice
`/home/users`	network, GPFS via NFS	51 TB	home directories

mount point	file system	capacity	notice
/data	local, ext4	523 GB	scratch directory
/tmp	ram disk, tmpfs	3 TB	scratch directory, all data will be lost while computer failure
/dev/shm	ram disk, tmpfs	8 TB	scratch, all data will be lost while computer failure

3. System Access

Application for an account

Users apply for access to the system with computational grant. Grant application form is available via PSNC's HPC Portal: http://hpc.man.poznan.pl. European researchers can apply for core hours using PRACE DECI calls.

Login to Chimera

- User access is available via SSH: **ssh my_username@chimera.man.poznan.pl**
- Access with gsissh: **gsissh my_username@chimera.man.poznan.pl -p 2222**

Further information and services

Users can access storage via sftp protocol. To access data utilities like WinSCP, FileZilla, sftp, scp, sshfs can be used.

4. User environment and programming

Chimera is a single image system running under control of a standard Linux distribution. Job scheduling is handled by SLURM batch system.

Users access directly to machine's system, there is no access node servers (like on clusters).

4.1. Generic x86 environment

On Chimera common programming tools are installed: Intel Composer XE 2011, Intel Vtune Amplifier XE 2011 and Intel Inspector XE 2011.

Compilers:

- GNU compilers 4.4
- Intel Compilers

Libraries:

- Intel MPI 4.0 u3
- Intel MKL 2011
- Intel Threading Building Block

Further reading

For more information on using GNU and Intel compilers please refer to the PRACE Generic x86 Best Practice Guide [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].

4.2. System specific environment

Default shell for users is **bash**.

All preferred utilities (compilers, MPI, etc..) are accessible on command line. All are placed in `PATH` variable.

In default `.bashrc` file contents user's specific settings of environment should be made at beginning of `.bashrc` file. All settings made at the end of this file will be activated only in log-in shell and will be inactive in job scripts.

4.2.1. Batch system

Queue system on Chimera is governed by SLURM [http://www.schedmd.com/slurmdocs/].

Resources

Queues in SLURM queue system are called partitions.

Table 2. SLURM partitions available for users.

name	time limit
long	72h
standard	24h
quick	4h

The default partition is `standard`. Chimera is single image system so there is only one node in partitions: `chimera`. Processors in job resources have to be reserved as tasks. To get actual information on partitions and time limits command **sinfo** can be used.

Submitting jobs to queue system

Recommended way to run jobs on Chimera is batch mode in contrast to interactive mode. Jobs are submitted with **sbatch** command. As last argument user have to add script name, all arguments of sbatch commnad can be placed at beginning of script file with `#SBATCH` prefix. Schema of script for MPI job is presented below:

```
#!/bin/bash
#SBATCH -J MY_JOB_NAME
#SBATCH --partition=QUEUE
#SBATCH --get-user-env
#SBATCH --ntasks=NUMBER_OF_PROCESSORS
#SBATCH --mail-type=end
#SBATCH --mail-user=HERE_PUT_YOUR_email
#SBATCH --time=01:00:00


mpiexec my-program.exe options...
```

Before run this script user has to modify fields `MY_JOB_NAME`, `QUEUE`, `NUMBER_OF_PROCESSORS`, `HERE_PUT_YOUR_email`, time and program name with options. In case the program other than MPI parallel paradigm there is no need to use `mpiexec`.

By default all environment variables are propagated to jobs. To avoid propagating variables use option: `--export=NONE`. To export specific environment variables to the batch job (program) please use: `--export=VARIABLE` option for example: `--export=SCRATCH=/data/myscartch`.

Controlling jobs

To see all jobs in slurm queue please use **squeue**.

To see details of specific job with id *JOBID* please use **scontrol show job** *JOBID*

Job cancellation can be done with command: **scancel** *JOBID*

4.2.2. System specific compilers and libraries

There is no need to use SGI specific MPI implementation MPT. Support of hardware acceleration of MPI is included in default on Chimera IntelMPI.

Portability and compatibility with other x86 systems

System is full compatible with other Linux operating systems with x86_64 architecture, especially with Debian GNU/Linux 6.

4.3. Known issues

Intel Compiler Suite location

Complete Intel Compiler Suite is installed in `/opt/intel` directory.

5. Performance Analysis

Available generic x86 performance analysis tools on Chimera includes:

- VTune Performance Analyzer
- Intel Trace Analyzer

Further reading

For more information on using VTune Performance Analyzer please refer to the PRACE Generic x86 Best Practice Guide [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].

There is no system specific performance analysis tools available on Chimera computer.