MERCED CLUSTER BASICS

Multi-Environment Research Computer for Exploration and Discovery
A Centerpiece for Computational Science at UC Merced

Sarvani Chadalapaka
HPC Administrator
University of California Merced, Office of Information Technology
schadalapaka@ucmerced.edu
it.ucmerced.edu

Computing Cluster

- The computer clustering approach usually connects a number of readily available computing nodes (e.g. personal computers used as servers) via a fast network. The activities of the computing nodes are orchestrated by "clustering middleware", a software layer that sits atop the nodes and allows the users to treat the cluster as by and large one cohesive computing unit, e.g. via a single system image concept.

- Computer clustering relies on a centralized management approach which makes the nodes available as orchestrated shared servers.

Physical Setup of Merced Cluster

- Merced Cluster is located in SE-2 040
- Head node(merced) + 72 CPU compute nodes(mrcd01 – mrcd72) +
  4 GPU compute nodes(mrcdg01-mrcdg04) +
  Cluster Storage Unit (clusterstorage)
- The GPU nodes contain NVIDIA GPUs and Intel Xeon Phi GPUs
- Each node has 128GB of RAM
- The nodes are interconnected with Infiniband network

Infiniband Architecture

- InfiniBand was introduced in 2000 as a way to tie memory and processors of multiple servers together so tightly that communications among them would be as if they were on the same printed circuit board. To do this, InfiniBand is architecturally sacrilegious, combining the bottom four layers of the OSI (Open Systems Interconnection) networking stack -- the physical, data link, network and transport layers -- into a single architecture.
- InfiniBand is a flat fabric, topologically speaking, meaning each node has a direct connection to all the others. InfiniBand's special sauce is RDMA (Remote Direct Memory Access), which allows the network card to write and read data on a server, eliminating the need for the server processor to conduct this work itself.

Infiniband vs 10GigE

- 10GigE has 5-6 times the latency of InfiniBand
- InfiniBand has 3.7x the throughput of 10GigE
- Beyond 1-8 nodes, many times InfiniBand provides much better performance than 10GigE and the performance difference grows rapidly as the number of nodes increases
- Putting HPC message passing traffic and storage traffic on a single TCP network may not provide enough data throughput for either.
- Many HPC applications are IOPS driven and need a low-latency network for best performance.
- There are a number of examples that show 10GigE has limited scalability for HPC applications and InfiniBand proves to be a better performance, price/ performance, and power solution than 10GigE

Scheduler

- Scheduling is the method by which work specified by some means is assigned to resources that complete the work. The work may be virtual computation elements such as threads, processes or data flows, which are in turn scheduled onto hardware resources such as processors, network links or expansion cards.
- A scheduler is what carries out the scheduling activity. Schedulers are often implemented so they keep all computer resources busy (as in load balancing), allow multiple users to share system resources effectively, or to achieve a target quality of service.

Sun Grid Engine

- Grid Engine is typically used on a HPC cluster and is responsible for accepting, scheduling, dispatching, and managing the remote and distributed execution of large numbers of standalone, parallel or interactive user jobs. It also manages and schedules the allocation of distributed resources such as processors, memory, disk space, and software licenses.

WHY SGE?

- Multiple advanced scheduling algorithms allow powerful policy-based resource allocation
- Cluster queues
- Job and scheduler fault tolerance - Grid Engine continues to operate as long as there is one or more hosts available
- Job arrays and job tasks
- Resource reservation
- XML status reporting (*qstat* and *qhost*), and the *xml-qstat* web interface
- Parallel jobs (MPI, OpenMP), and scalable parallel job startup
- Usage accounting
- Accounting and Reporting COnsole (ARCO)
- GUI Installer and SGE Inspect
- ……

Queuing

SGE includes both a scheduler for allocating resources (CPUs!) to computational jobs and a queueing mechanism. Each queue is associated with a number of *slots*: one computational process runs in each slot; each compute node in the HPC cluster provides one or more slots.

Each queue utilizes same resources but has different “rules” for job execution

MERCED Cluster Queues & Resource Limitations

	# Nodes		Limits	
	20-Core Nodes	24-Core Nodes	Wall Clock	Max Cores Per User
std.q	30	38	24 hours	800
fast.q	36	40	4 hours	200
long.q	28	18	14 days	200
gpu.q	4	0	14 days	Unlimited

PRIORITIES

- Currently, priority on the queues is setup such that each project is given equal priority (each project is assigned to a PI) and each member in that project (or PI group) has equal priority.

Job Submission Script

#! /bin/bash → “The hashbang line”

#$ -S /bin/bash →specifies the interpreting shell for the job

#$ -q fast.q →Defines a list of cluster queues which may be used to execute this job

#$ -cwd → working_dir

#$ -N testprojectile → Name of the job

#$ -j y → Specifies whether or not the standard error stream of the job is merged into the standard output stream.

#$ -o test2.qlog → The path used for the standard output stream of the job.

#$ -pe smp 1 → Parallel programming environment (PE) to instantiate.

whoami

How do I actually submit the job?

qsub sample.sub

Simple Job Submission Exercise

- Run projectile.exe
- Step-1: Copy projectile.exe file from /home/ICGE_Job_Submission_Exercise path to respective home paths
- Step-2: Copy sample_projectile.sub script from /home path to respective home paths
- Modify the script to run projectile.exe (Hint: Just need to add one line of code at the end to execute projectile.exe)
- Enter the command to submit job
- Check the status of the job using “qstat” command

Ganglia URL

- http://merced.ucmerced.edu/ganglia/