July 14, 2012

User Assistance Best Practices

Kim Dillman/Purdue
Bill Brouwer/PSU

XSEDE
Extreme Science and Engineering Discovery Environment

NSF

0.0 Introduction

Take a look at XSEDE resources https://www.xsede.org/resources/overview and begin to assess the requirements of your software, in order to match with the appropriate hardware eg., at a minimum one should :

- Determine whether software requires a license or is open source
- Perform a search from https://www.xsede.org/software to test whether the desired software is installed already
- If installing, ensure adequate disk space exists on desired XSEDE resource **both** for required software and subsequent computations with your data
- Ascertain, document and deal with software dependencies
- Optimize the software for the ultimate architecture where possible
- Seek out existing benchmarks on XSEDE eg., NAMD https://www.xsede.org/wwwteragrid/archive/web/user-support/namd_benchmark.html
- Drill down further into the software/hardware solution for your problem...

1.0 High Performance Planning

Once a high level overview of software requirements is established, and possible candidate systems from XSEDE resources tentatively identified, one should ascertain :

- For parallel applications :
  - Scaling information eg., point of diminishing returns with increasing cores
  - Communication overhead, which will dictate choice of node interconnect technology
  - Memory model eg., shared or distributed
- For serial or almost serial (low communication) applications :
  - Ability to restart from checkpoints and suitability to high-throughput resources like Condor pools within XSEDE or the Open Science Grid https://www.opensciencegrid.org/bin/view
- For all applications :
  - Any specific hardware and/or OS requirements eg., Linux x86_64, CUDA/Nvidia GPU, Cray, SGI etc
  - Other potential issues posed by target system(s), described in detail within user guides https://www.xsede.org/web/guest/user-guides

2.0 Implementation Planning

With a good handle on the problem under study, a strategy for establishing required software and firmer ideas as to suitable XSEDE systems, some thought must be given to producing the actual workflow. Specifically, one needs to address the following questions:

- How long will jobs run (min/max/average)?
- How many total jobs?
- How much memory per processor or node do jobs need?
- How much disk space does each job require for both input and output (node local or system shared)?
- How much total disk space is needed to complete the project?
- Data archiving of output to another system/storage area?
- How long (time) does job output need to reside on the system where it was created (i.e. for further processing, input to other runs, etc.)?

3.0 Practical Aspects

After appropriate planning, when running one must keep in mind that there are many different job manager systems in use on the XSEDE resources eg., PBS Pro, Torque, SGE, LoadLeveler etc.

- Familiarize yourself both with job manager and with queue(s) on XSEDE resource eg., means for querying, submitting etc
- Each resource may require you to submit jobs to a queue in a different way eg.,
  - Submit directly to the correct execution queue that meets your job's specific requirements
  - Submit to a general queue which routes your job to the correct execution queue matching your job's requirements

Job manager software can implement widely varying policies that either LIMIT and/or PRIORITIZE jobs in the queue :

- Limits may be applied to jobs in a queue or across all queues, for example:
  - Number of jobs queued or running per user
  - Number of jobs queued or running per project allocation number
- Priorities can be based on a variety of attributes including but not limited to:
  - Wall time requested
  - Number of CPUs requested


Our reach will forever
exceed our grasp, but,
in stretching our horizon,
we forever improve our world.
XSEDE
Extreme Science and Engineering
Discovery Environment