

The Practice of High Performance Computing

Gregory G. Howes
Department of Physics and Astronomy
University of Iowa

PHYS 5905: Numerical Simulation of Plasmas
Department of Physics and Astronomy
University of Iowa
Spring 2019



Outline

- Remote Computing at National Supercomputing Centers
- Data Management
- Code Management
- Scientific Validation and Benchmarking
- Managing a Computational Research Program
- Applying for a Computing Allocation

National Supercomputing Centers

- Most Supercomputing Centers require running in **Batch mode** using PBS (Portable Batch System) and a scheduler (like **Argon**)
 - Usually your job will wait around in a queue for a while (sometimes days, or even weeks) before running
 - It is important to get used to this method of remote computing
- To best take advantage of the computer, **learn the queue rules** for that computer:
 - Often, if you modify how you run slightly (i.e., number of processors or total wallclock time), you can achieve much higher **throughput** of runs
 - Many people pay no attention, and consequently have poor throughput

Queue Rules (Scheduling Policy)

The way you run jobs will differ dramatically depending on the queue rules

#1 Jaguar Cray XT5 ORNL (DOE)

#3 Kraken Cray XT5 NICS (NSF)

Priority/Limits Based on Job Size

Bin	Cores		XT5 Partition	
	Min	Max	Maximum Wall-Time (Hours)	Aging Boost (Days)
1	120,000		24.0	15
2	80,004	119,999	24.0	10
3	40,008	80,003	24.0	5
4	5,004	40,007	12.0	0
5	2,004	5,003	6.0	0
6	1	2,003	2.0	0

Queue	Min Size	Max Size	Max Wall Clock Limit
small	0	512	24:00:00
*longsmall	0	256	60:00:00
medium	513	8192	24:00:00
large	8193	49536	24:00:00
capability	32769	99072	24:00:00

Outline

- Remote Computing at National Supercomputing Centers
- **Data Management**
- Code Management
- Scientific Validation and Benchmarking
- Managing a Computational Research Program
- Applying for a Computing Allocation

Data Management

- Standard Procedure for Supercomputing Centers:
 - **HOME** directory is for source code and small files
 - **SCRATCH** directory is where your code will produce output
 - Usually this directory has **faster access** to the compute nodes
 - This directory is **not backed up**
 - **ARCHIVAL STORAGE** is where you will store your large data sets
- Transferring Files to/from Offsite:
 - Moving TB of data is a very slow process
 - You can use secure parallel file copy facilities such as **bbcp**
 - Globus transfer service (www.globus.org) is a new research data management service
 - You'll often want to work with support staff to figure out the best way
- Compressed, Portable, Self-Describing data formats are highly recommended
 - **NetCDF** (Network Common Data Form)
 - **HDF** (Hierarchical Data Format) also has parallel I/O capability
 - You can link your code using these widely used, standardized libraries
- **National Science Foundation now requires all funding proposals to include an explicit Data Management Plan**

Outline

- Remote Computing at National Supercomputing Centers
- Data Management
- Code Management
- Scientific Validation and Benchmarking
- Managing a Computational Research Program
- Applying for a Computing Allocation

Code Management

If you are developing a code that may be widely used, here is some advice:

- Use the **standard language**, not specialized extensions of a specific compiler
- **Comments, Comments, Comments!**
- Software tools for team development
 - **Subversion** <http://subversion.apache.org/>
 - **GitHub** <https://github.com>
 - Other older packages: **CVS** (Concurrent Versions System), RCS, PRCS
- Packaging code so that it is easy to transport and compile
 - tar archives are useful to allow unpack the entire directory structure
 - Makefiles make compiling easy
 - HYDRO gives an example of how to do this
- Porting
 - If many people use your code, porting to different computers is important
- Documentation
 - Also important is to write up clear documentation on how to use code
- Making a Code Publicly Available
 - **Sourceforge** (<http://sourceforge.net/>) will host open source software that is to be shared with the community

Outline

- Remote Computing at National Supercomputing Centers
- Data Management
- Code Management
- **Scientific Validation and Benchmarking**
- Managing a Computational Research Program
- Applying for a Computing Allocation

Scientific Validation and Benchmarking

- Scientific codes need to be validated before most will accept the results
 - Publish results of validation tests in peer-reviewed scientific journals (i.e. [Journal of Computational Physics](#), etc.)
 - Often you can find **standard test problems** in the literature
- When packaging up code for distribution:
 - Include input files for the suite of test runs that you used for validation
 - A benchmark run is helpful
 - The benchmark is a relatively short run that test the full capability of the code and will only yield the correct answer if the code works correctly
 - This is good to verify proper compilation and installation and can be used to compare code performance on different computers

Outline

- Remote Computing at National Supercomputing Centers
- Data Management
- Code Management
- Scientific Validation and Benchmarking
- **Managing a Computational Research Program**
- Applying for a Computing Allocation

Managing a Computational Research Program

- Organization of simulation runs is fundamentally important
 - You need to **keep good records** of runs performed in the past
 - What were the parameters, where is the data, etc.
 - I keep a notebook with all of my large-scale runs logged into it.
- Think hard about the research questions you want to answer
 - What are the critical runs that will enable you to answer those questions?
- Allocation Management:
 - It is **always better to use up your allocation** before the end of the award period than to have some left over
 - Often when this happens, they will give you a little more if you need it

Outline

- Remote Computing at National Supercomputing Centers
- Data Management
- Code Management
- Scientific Validation and Benchmarking
- Managing a Computational Research Program
- **Applying for a Computing Allocation**

Applying for a Computing Allocation

- National Supercomputing Resources:
 - **NSF** XSEDE (Extreme Science and Engineering Discovery Environment)
<https://www.xsede.org/>
 - **DOE** Office of Advanced Scientific Computing Research
<http://science.energy.gov/ascr/>
 - **NASA** Advanced Supercomputing Division
<http://www.nas.nasa.gov/>
 - **NIH** Center for Information Technology
<http://www.cit.nih.gov/science.html>
 -
- Computing Proposal:
 - Describe the scientific problem you want to solve
 - Describe your code (algorithm, parallelization strategy, etc.)
 - Provide parallel performance results (strong and/or weak scaling)

Applying through NSF XSEDE

Startup Allocation

- The application procedure for a startup allocation is rather simple
- Cannot apply as a graduate student, but can as a postdoc
 - But you can apply with your advisor as PI and then your advisor can set you up with an account
- Can apply for up to 200,000 SUs (valid for 1 year)
NOTE: 8760 hours/year, thus 200,000 cpu-h is like 23 cores for a year!
- **Application requires only:**
 - **Estimate of computing time needed**
 - **Short abstract of computational project**
 - **CV for the PI**
- Review of your proposal will be returned within 1 week

Research Allocation

- Requires a **10-15 page proposal** (depending on size of request)
- Requires supporting **code performance and scaling information**
- Reviewed quarterly with multiple reviewers

Local Resources at the University of Iowa



Research Services of ITS

- Supports computational research at the University
 - Ben Rogers, Director of Research Services, ben-rogers@uiowa.edu
- Argon
 - Shared resource funded by University and many faculty research groups
 - Open to all University researchers (using [all.q](#) or [UI](#))
 - Documentation

<https://wiki.uiowa.edu/display/hpcdocs/Cluster+Systems+Documentation>