Dear Colleagues,

There are a number of important happenings, changes and new features at SHARCNET that I would like to bring to your attention. These topics are covered below under the following headings:

1. PI Account Renewal
2. System Enhancements
3. New Equipment
4. Operating Policies

My apologies in advance for a somewhat lengthy email. Item 1 requires the attention of Pis; items 2 and 3 are largely informational; item 4 describes the storage policy and some scheduling changes on two systems. As always, if you have any questions about these items or any other issues at SHARCNET please do not hesitate to contact me or SHARCNET staff.

## PI Account Renewal

SHARCNET requires that Pis renew their sponsored accounts at the end of August each year. This ensures that we can clean up student and visitor accounts that would otherwise become dormant etc. This will continue. We would ask, in particular, that for the people you sponsor (students, postdocs or visitors) who have left to take up a position elsewhere that you close their accounts unless they are continuing a collaboration with you that will lead to published work. Please also ensure that their profiles are updated if the account remains open.

We will now also be asking for annual renewal of PI accounts. The first renewal will take place Jan 16th, 2008. Thereafter, account renewal will normally take place on the anniversary of account creation (the minimum time between the first and second renewals will be a year).

For an account to be renewed it is required that your profile on the SHARCNET portal be up to date. This includes filling in new fields that request information about primary funding agencies and the high level discipline category in which you work, as well as making sure that your publications and research project descriptions are up to date.

It is vitally important that you update your profile to indicate both the use that your group has made of SHARCNET and to provide information so that we can report to funding agencies the support that we are providing to various disciplines. This process will also allow us to clean up dormant accounts. Note: in the event that an account is either suspended or removed, we will make every attempt to ensure that no data is lost.

The renewal procedure will be as follows: In the period leading up to the renewal date, the system will generate for each Site Leader a list of information about the data reported by each PI at that institution.

The Site Leader will review the data that you have entered and make a recommendation to SHARCNET about whether or not the account should be renewed. As an example, a PI whose group has used 3% of SN resources over the last year but who has reported no publications or research description would likely get a recommendation that the account not be renewed. A Site Leader may contact you to seek information and to encourage you to submit material to the portal. In order to ensure consistency, difficult cases will be coordinated globally across SHARCNET. If the update is acceptable, the account will be renewed and nothing will happen on Jan 16th, otherwise all accounts in the PI's group will be suspended.

It is in your best interests to make sure that your profile is updated as soon as possible. When your account is renewed you will be sent an email confirmation.

## System Enhancements/Features

Matlab: We have purchased a licence for 64 concurrent processes. This license allows users to run parallel Matlab codes on one of our clusters and is intended to permit users to solve problems that are significantly larger than could be addressed on a desktops. Users of this license are still required to have their own seat license for the Matlab engine and any toolboxes they require. Details describing the system on which this is installed and how to use it are available at: http://www.sharcnet.ca/Help/index.php?content=matlab

Certification: A user certification process has recently been instituted that classifies users into 3 levels. A user attains level 0 automatically on being granted an account; level I is granted after attendance at one of our fortnightly AccessGrid New-User seminars followed by successfully answering a small quiz. Each level permits access to a defined level of system resources (processor number and wallclock at present). The first level is intended to satisfy most users on an ongoing basis. For more demanding users/applications there is a second level that grants (at present) unrestricted access (within the general scheduling policies). Level II requires a consultation with a SHARCNET HPC consultant; we may also approach users whom we think should have level II certification. The intent of certification is to ensure that users receive the proper orientation and training in the use of SHARCNET at the level they need without being in any way burdensome or onerous. Existing users will be grandfathered to level I.

This process was described in the Summer, 2007 issue of SHARCBytes and is also available online at:

http://www.sharcnet.ca/Facilities/certification.php

Software email lists: A number of optional lists are accessible at https://www.sharcnet.ca/my/profile/mail that allow you to receive mail about updates to the various software packages that we support.

Global Serial queue: We are beginning to develop queues that can access resources from across many SHARCNET systems without a user having explicitly to find the appropriate system. The first instance of this is a serial queue that will find free cycles from a pool of systems and submit the job to a lightly loaded cluster. At present this is appropriate for serial jobs that have small data requirements – the job **must** be run out of /home. This queue can be accessed from any system by adding the argument "--global" to the standard "sqsub" command. In the future we hope to have an interface that will allow submission via the web portal and avoid having to explicitly log on to a cluster. See http://www.sharcnet.ca/Help/faq.php#Global-jobs for details on how to use the global serial queue.

Archiving data: SHARCNET has a large, 200TB file store for longer term storage of data than is available on /work. This is not directly mounted on any system but is accessible via a small set of "archive tools". These tools and their use are described at:

http://www.sharcnet.ca/Help/faq.php#Q:%20Archive%20tools or by typing "archive -h" on any system. Although archive is a large filesystem and is expandable in the future as needs dictate, we cannot guarantee that we will be able to be able to store user data indefinitely nor that there is absolute data integrity. For more details, please see the storage policy at

http://www.sharcnet.ca/Documents/policies/storage.html

Databases: If you have a need to create a small database or access a database on SHARCNET (mySQL or postgres) we now have some capability to support this. Please see http://www.sharcnet.ca/Help/faq.php#GBDB for details.

Survey: You may recall that a survey was run in the Spring. The full results are up on the website at:

http://www.sharcnet.ca/Survey/results_summary.php Of particular interest are some of the comments and suggestions at: http://www.sharcnet.ca/Survey/results_comments.php this page includes some initial responses. Several of the suggestions have been implemented and we are looking at following up on many of the others.

AccessGrid: We now have AccessGrid rooms running at 12 of the 16 SHARCNET partners, with the other four expected to be in operation by early Spring (see

1. http://www.sharcnet.ca/Facilities/accessgrid.php). The facilities have been used for the introductory seminars, training, graduate supervision, graduate courses, numerous SHARCNET management and technical meetings and have also allowed us to participate in the Coast-to-Coast seminar series that runs between ACEnet and WestGrid. If you are interested in using the system or joining these activities please contact Barb Autterson (barb@sharcnet.ca) for a booking and information.

## New Equipment

A number of small systems have recently been brought on line or will shortly come on line. If you are interested in using any of these please contact SHARCNET staff.

Grid lab. This is a part of the SHARCNET2 grant and is an installation that will allow, in a lab setting, experimentation on and investigation of various aspects of grid computing. The network infrastructure connects four small clusters and includes equipment to simulate noise and latency as would be present in wide area networks.

SGI Altix: This system has been contributed by Hermann Eberl at Guelph. It is a 64 processor shared memory (NUMA) system with 128GB of RAM. Many thanks are due to Hermann for this valuable contribution.

FPGA system: This SGI hardware was donated to SHARCNET. We have purchased a software environment and compiler that will allow users to experiment with development on FPGAs (Field Programmable Gate Arrays). For some algorithms the potential speed gains can be one or two orders of magnitude.

SVA: The Scalable Visualization Array is a part of the SHARCNET2 CFI purchase. As its name suggests, it is a scalable visualization system

2. • essentially it is the equivalent of an HPC cluster but for visualization. It will allow visualization of very large datasets. Software is being installed that will allow for the turnkey display of common data formats (for example, Fluent, hdf5). The system also allows for the efficient remote display of rendered data anywhere across SHARCNET.

## Operating Policies

## *Disk use policies*

A draft policy governing disc usage is on the SHARCNET website at: http://www.sharcnet.ca/Documents/policies/storage.html As usage of the systems increases we need to start implementing these policies. Some users will have already received automated email messages if their disc usage exceeds the limits described in the document. Although we do not plan in the short term to delete data, we may have to move it if resources become scarce. I would remind you again of the availability of archive and would encourage you to use it. Please also review your disc usage on mySHARCNET and see if there is old, but still needed, data that can be moved along the /scratch -> /work -> /archive hierarchy. We would also encourage you to think about the ultimate fate of your data: will you be migrating it to your own private storage (a 500GB disc is now <~$150) or can some of it – perhaps that which can trivially be regenerated – be deleted?

If you have particularly demanding data needs please contact us. These needs and solutions could range from temporary very large storage on our file systems to “contributing” additional storage to be attached to SHARCNET (see

http://www.sharcnet.ca/Documents/policies/contribution.html). Note also that the dedicated resources programme that runs every six months can allocate very large system resources including disc space.

In general the storage policy is constructed to ensure effective use of the computational resources. Thus /scratch must maintain significant free space to ensure that running code can write output to disc and not wait – or worse – crash because of lack of space. The time limit for old files on this filesystems has been set at 2 weeks to ensure that data is quickly migrated off to /work as soon as it has been written. (For these purposes the time stamp is that of modification time: hence a file that is updated over a period of time does not become old until 2 weeks after the last update.) There is no disadvantage to having data on /work. It is just as fast and can be used just as effectively for storing, for example, initial conditions data or data for postprocessing. In principle you could write to /work directly from your code, but we will allow /work to become fuller and so there may be a risk that it could not accommodate a large series of outputs that might happen over a weekend when system monitoring is less intensive. The restriction on /scratch is not meant to be onerous. At worst it would mean that one had to move data from scratch to work every two weeks. If you require more than the 200GB storage on /work that is the present default, you should email help@sharcnet.ca justifying your need.

On a more abstract level, it is probably worth noting again what SHARCNET is trying to provide. In almost all cases our systems are constructed of components that are identical to those on your desktop. What we provide is scale: more processors than you might reasonably be expected to have in your lab and faster, directly connected, filesystems. This allows you to do new things that are simply not possible otherwise and it is this kind of use that we want to encourage and that will get priority. In many respects we provide a transient resource and, in a shared environment, will have difficulty ensuring effective use unless all users help to ensure that the storage closest to the computers is reserved for very new files.

## *Scheduling policies*

Scheduling the huge range of user jobs that we encounter remains perhaps the key challenge at SHARCNET. For the systems running serial jobs (whale in particular) the task is relatively straightforward: queued jobs go to the next processor that becomes free and there is obviously no need to ensure any coordination with other processors. (I would note in passing that queue ordering is dynamic such that if you submit a single job 30 seconds after another user submits 20,000 your job will be scheduled ahead of her/his queued jobs provided that your fairshare priority has not been hit by too much recent use.)

For parallel systems the challenge is much greater. We want to encourage users to take advantage of the significant capability that is offered and that means we will support, and in many case give priority to, large parallel jobs. This means that when a user submits, for example, a 64-way job and there are no free processors we have to make a decision about how to obtain 64 processors. One way is to wait until jobs finish and reserve processors as they become free. If done in the most naive fashion this leads to processors that are idle – potentially for a very long time as other jobs finish. The best way to avoid wasted cycles in this approach is to backfill the idle cycles with other, shorter, jobs. This will only work if we know the runtime of jobs and it thus requires users to state the wallclock runtime when the job is submitted. If a job exceeds this, it is killed (usually some grace period is allowed). A risk is that users overstate the time needed to ensure that their jobs finish. It is, of course, fairly easy to detect and deal with this sort of antisocial behaviour! Further, gross overestimates reduce the probability that your job will find a long enough backfill slot. There are a number of other issues with this approach: it requires a system with a fairly large throughput of jobs so that timeslots can be reserved in a way that ensures that large jobs can be scheduled and start in a reasonable amount of time. It also may lead to fragmentation of jobs across processors unless contiguous processors are reserved which would further limit the scheduler's flexibility.

We will be investigating reservation and backfill for a trial period on narwhal to see if we can improve job scheduling. Since this is our "Utility Parallel System" which runs a large mix of modestly parallel jobs it seems like the most appropriate target. Starting immediately, we will require all user jobs on any system to state an estimated runtime as an argument to sqsub as follows: "sqsub -W hh:mm ..." (see "man sqsub" or "sqsub --man" for more details).

Another way to achieve throughput for large parallel jobs is simply to pre-empt smaller or lower priority jobs. The key question then is what to do with the pre-empted jobs? They could simply be killed or suspended. If suspended, what does one do with them? Ideally it would be good if they could checkpoint and then be restarted elsewhere. In practice there is as yet no established automatic checkpoint/restart for parallel jobs. (We are about to investigate a product that may provide a solution in the future.) If the system suspends a job, the job must subsequently be allowed some time to reach a user-defined state at which it can checkpoint on its own so that it can then stop or be killed. Only if the pre-empting job is short does suspension work at present. In this case the suspended job simply resumes after a short period of time (this is how the test queues work). On a system such as requin, which is designed to support a population of large, long running jobs, it can become very difficult to ensure fair throughput for a range of user jobs. Furthermore, the machine can become very fragmented as processors (or groups of processors) become free at often widely spread locations across the machine. For tightly coupled applications this is not optimal.

In order to try to address this issue, we plan to try queue flushing on requin every week. What this means is that all jobs will be killed on requin every week at a particular time – probably Monday Noon. The machine will then be repopulated with jobs from the queue. This will have two benefits. It will mean, in the context of long running jobs, that it will be easier to maintain fair access to the machine across user jobs (it will act like a very coarse grained timneslicing) and it will be much easier to allocate contiguous processors to every parallel job; at a minimum there will be a weekly "defragmentation". A "SIGTERM" (signal) will be sent ahead of the "SIGKILL" (signal) and codes that are able to catch this can use it to force a checkpoint. Please contact the staff for help if you are interested in making use of this feature.

The choice of the queue-flushing timescale is, of course, a free parameter. It cannot be too long, as then nothing changes, or too short that nothing can practically be accomplished in the interval between flushes. Most supercomputer sites in the US and Europe operate in this way. Some have flush times as short as 6 hours most are of order one or two days. In order to try to minimize disruption we will start with a week.

The corollary of this approach is that users must either complete their jobs within a week of wallclock or they MUST CHECKPOINT. This is, of course, good practice anyway and becomes more and more critical as larger and larger applications are run and systems constructed. For most applications it should be relatively straightforward to write the system state in a way that allows a restart, and for many commercial applications this is the case. For applications for which it is difficult to checkpoint or for which it is not obvious how, please contact SHARCNET staff for assistance. The bottom line is that it is dangerous in any case to assume that any system of this size will be up for more than a week.

These scheduler changes will come into effect on Monday December 3rd and we will reassess the success of these two models in the New Year. With luck, they will lead to improvements in throughput for your jobs. Details are available at:

http://www.sharcnet.ca/Facilities/new_scheduling_directives.php

It is clear that the proposed scheduling changes and enforcement of disc space limits will require, in some cases, more attention by users to their jobs. Whilst it is naturally convenient to "fire and forget" when submitting jobs, this convenience has to be balanced with the need to share the resources with others. I hope that the modest additional work required to monitor your jobs will lead overall to a better user experience and, indeed, appreciation of the very capable systems at your disposal.

My apologies for the very long email. For those of you who got this far – thank you. I hope that the features and explanations are useful and I look forward to any comments that you may have.

Hugh Couchman

Scientific Director, SHARCNET

couchman@sharcnet.ca