# "UPDATE ON NERSC PScheD EXPERIENCES, A CONTINUING SUCCESS STORY"

**Michael Welcome & Tina Butler**
**NERSC Systems Group**

## ABSTRACT

This paper will describe NERSC's PScheD experiences, including local configuration differences between UNICOS/mk 2.0.3 and 2.0.4; changes that were implemented with UNICOS/mk 2.0.4, and resulting utilization improvements. The paper will also describe plans for further improvements in scheduling and queuing simplifications.

## Introduction

This paper will be a follow up to the previously co-authored 1999 CUG "T3E Scheduling Update" paper. In addition, NERSC will addressed the continuing utilization success of PSCHED and NERSC's T3E.

## Background

NERSC is the National Energy Research Scientific Computing Center and is funded by the U.S. Department of Energy, Office of Science. It is located at Lawrence Berkeley National Laboratory in Berkeley California. NERSC has a 25-year history of providing high performance computing to the DOE community. It was founded at the Lawrence Livermore National Laboratory in 1974 as the Magnetic Fusion Energy Computer Center and moved to Berkeley in 1996. The center provides computational resources to DOE programs in the fields of Fusion Energy, High Energy and Nuclear Physics, Basic Energy Sciences, Biology and Environmental Research and Computational and Technology Research. NERSC currently serves approximately 2500 users from major universities and government laboratories across the country.

The NERSC T3E, named "mcurie", is a T3E 900 with 696 processing elements. In its current configuration, mcurie has 644 application PEs, each with 256 MB of memory. It has a 411 GB swap space and a 625 GB checkpoint file system, each composed of 5 disk partitions with each partition consisting of 5 or 6-way striped DA308 disk arrays. This configuration was designed for speed when checkpointing jobs and for swapping during PScheD rank switches. It has a 1.5 TB scratch file system and seven 27 GB home file systems under DMF control. The large file systems, including scratch and checkpoint are controlled by "remote mount" file servers.

A brief history of NERSC T3E hardware is listed in Table 1. Pierre was a second, smaller system in general use until it was merged with mcurie in October of 1998. The center also operates one J90 SE and three SV-1 systems, which provide traditional parallel vector computing cycles.

Table 1. History of T3E Hardware at NERSC

| System Name | Type | PEs | APP PEs | Date | Comment |
|---|---|---|---|---|---|
| Mcurie | T3E-600 | 136 | 128 | 9/96 | Initial System |
| Pierre | T3E-600 | 104 | 96 | 12/97 | Initial System |
| Mcurie | T3E-900 | 544 | 480-512 | 8/97 | Phase II |
| Pierre | T3E-900 | 152 | 128 | 6/98 | Upgrade |
| Mcurie | T3E-900 | 696 | 644 | 10/98 | Merge with Pierre |

NERSC T3E Workload Characterization

The NERSC T3E is used for application development, medium-sized capacity-based computing and large-scale "capability" problems. User codes include applications from chemistry, materials science, fusion research, geophysics, high-energy nuclear physics, biology, climate modeling, astrophysics and fluid dynamics. The data shown in Table 2 indicate that the current workload is diverse and dynamic. A large number of small, short-running development applications are run on the system; it supports a steady flow of medium-sized applications; and 9 percent of the total machine resources have been consumed by applications requiring more than 128 processing elements.

Table 2. Workload Characterization

| App Size (PEs) | % of All Apps | % of PE Hours |
|---|---|---|
| 2-16 | 55.8 | 6.5 |
| 17-64 | 38.1 | 55.7 |
| 65-128 | 4.8 | 28.7 |
| 129-512 | 1.3 | 9.1 |

| App Run Time | % of All Apps | % of PE Hours |
|---|---|---|
| 0-10 min | 55.7 | 1.1 |
| 10-30 min | 23.2 | 10.4 |
| 0.5-3.5 hr | 16.9 | 48.7 |
| 3.5-12.0 hr | 4.1 | 39.8 |

NERSC T3E Scheduling Goals

It is NERSC's goal to provide a system that will satisfy these three classes of users, as well as its DOE sponsors, who want to insure that this expensive resource is used efficiently. The site scheduling goals are:

1. Minimize idle time in the APP region.
2. Provide fast interactive response while managing the total interactive workload on the system.
3. Provide reasonable and even turnaround across all the batch queues, especially the large job queues.
4. Allow users (provide mechanism) a method of scheduling and charging jobs which run through the batch system based on queue "priority".

These are competing goals. It is easy to efficiently schedule a system by running a large number of small jobs. The PScheD load balancer has been a stable product for quite some time and will consolidate unused PEs such that small and medium-sized jobs can get reasonable throughput. Scheduling large jobs while trying to minimize idle PEs is difficult because small applications can easily block the launch of a large application. Interactive loads are unpredictable and can block production work, especially large jobs. We will discuss how NERSC attempted to achieve these goals prior to the release of 2.0.4 and how they have configured their system to satisfy these goals under 2.0.4.

## The NERSC T3E Batch System Configuration

At NERSC, the batch system consists of NQE, NQS and a collection of PERL scripts that dynamically control the NQS configuration.

The site uses NQE as a holding pen for incoming requests. The NQE scheduler has been modified so that different LWS limits apply to the production and debug pipe queues. Each user is allowed up to three requests in the NQS production queues at any point in time. In addition, each user can have at most one simultaneous request in the debug queues.

The production batch queues are configured based on MPP PE limits and are shown in Table 3. As of FY 2000, Grand Challenge (gc128 and gc256) queues have been merged/replaced with existing production queues with their time limits extended to 12 hours. In addition, NERSC has implemented, as part of the new FY "Priority Batch Scheduling".

Priority batch scheduling allows users to choose a priority level within a given queue and an associated charging rate. Premium priority allows a user job to move to the highest priority category within a queue, and thus have improved access to be scheduled for execution at an increased allocation cost. Low priority allows a user to choose to submit a job to run with very reduced access to execution scheduling, but at a much lower charging rate. This scheme was accomplished by modifications to NERSC's local NQE scheduler and the use of NQS intraqueue priorities. The NQE scheduler sets predetermined priority values and corresponding nice values for jobs according to an attribute specified by the user at job submittal time. Although the nice value has no real effect on execution scheduling with gang scheduling running, it was deemed the most practical way to propagate priority information into accounting records.

Table 3. NERSC Queue Structure

| Queue | PE Limits | Time Limits | Priority |
|---|---|---|---|
| pe512 | 512 | 4 hr | 45 |
| long256 | 256 | 12 hr | 41 |
| pe256 | 256 | 4 hr | 30 |
| long128 | 128 | 12 hr | 27 |
| pe128 | 128 | 4 hr | 25 |
| debug_medium | 128 | 10 min | 29 |
| debug_small | 32 | 30 min | 23 |
| pe64 | 64 | 4 hr | 20 |
| pe32 | 32 | 4 hr | 15 |
| pe16 | 16 | 4 hr | 10 |

Given this constraint, as well as the need to schedule 512 PE jobs, NERSC and local SGI staff developed a PERL script to dynamically control and alter the NQS configuration. The script is run periodically by cron. Upon execution, it reads a configuration file, then parses the output of the qstat, grmview, psview and ps to determine the current state of the system. Finally, it modifies the current queue configuration to match what has been specified by the configuration file. This configuration file consists of a collection of alternate queue configurations along with a schedule that specifies which configuration should be used at any time during the day. Each configuration specifies which queues should be on, which should be off, the queue, complex and global limits as well as other values. See Table 4 for the current queue schedule.

Table 4. Mcurie Queue Schedule

| Schedule | Configuration | Active Queues |
|---|---|---|
| 22:00 - 01:00 | Full Machine | pe512 (pe64, pe32 for backfill) |
| 01:00 - 07:30 | Batch Preferred | pe256, long256, long128, pe128 and smaller |
| 07:00 - 22:00 | Regular | pe128, long128 and smaller |

The UNICOS/MK checkpoint/restart facility is used to transition from one configuration to another. In order to minimize the time required to checkpoint the entire system, the NQS control script issues qmgr hold requests in parallel, with up to five checkpoints running simultaneously (matching the number of striped disk partitions in the checkpoint file system). NERSC can usually checkpoint the batch workload in 3-5 minutes. The state of the system and all actions taken by the NQS control script are logged to a timestamped log file for post-analysis in the event of scheduling problems.

NERSC T3E Scheduling Problems prior to UNICOS/MK 2.0.4

The only feature of PScheD that NERSC used in production prior to UNICOS/MK 2.0.4 was the load balancer. Extensive dedicated testing and a live exposure test convinced the site that the gang scheduling software was simply too unstable to run in a production environment. In order to control the interactive workload, NERSC configured the APP region into two subregions, Batch and Mixed, with the GRM attributes specified in
Table 5.

Table 5. GRM Configuration Prior to 2.0.4

| Name | PE Range | Min | Max | Service | Time In Effect |
|---|---|---|---|---|---|
| Batch | 0-511 | 2 | 512 | batch | always |
| Mixed | 512-644 | 2 | 64 | login | 06:00-18:00 M-F |
| Mixed | 512-644 | 2 | 64 | all | 18:00-23:00 M-F<br>03:00-06:00 M-F<br>03:00-23:00 S-S |
| Mixed | 512-644 | 2 | 64 | batch | 23:00-03:00 everyday |

The Batch region was used exclusively by NQS. The Mixed region ran interactive jobs during the weekdays, allowed batch and interactive in the evenings and on the weekend days, and ran batch-only during the midnight hours. The GRM configuration changes were controlled by a cron script and were coordinated with the NQS control script, which released more batch work when the Mixed region was allowed to run batch work.

Unfortunately, this system configuration required that the site run with two PScheD domains, one for each region. Since PScheD is unaware of the GRM PE attributes, running the load balancer in a single domain with mixed GRM attributes would cause regular and repeated migration failures as PScheD would attempt to migrate a job into a location that GRM would not allow. Similarly, GRM is unaware of the PScheD domains and would launch an application at the location in the torus with the best match based on the attributes of the application and the PEs. This would often result in an application being launched on a range of PEs that intersected the boundary of the two domains. Neither domain would claim responsibility for such an application because it was not contained entirely within the confines of either domain. Such applications would never get migrated to better locations unless moved manually using the "migrate" command.

The configuration philosophy was to run interactive and some small batch jobs (when allowed) in the Mixed region. The site attempted to control this by setting the "close_max" precedence value high enough so that small jobs would be more likely to launch into the Mixed region.

Even with this, the system would often get into a state where a large job was waiting to launch and the system had enough available PEs overall, but they were distributed throughout both the Batch and Mixed regions. PScheD could not migrate applications between domains so large jobs would stall in a GRM wait state while large numbers of PEs would sit idle.

NERSC wrote a PERL "torus packing" script to detect and possibly correct these situations. If an application intersected both PScheD domains the script would attempt to migrate the application into the

Mixed region and, if unsuccessful, into the Batch region. If the script detected a large batch job stuck in a GRM wait state, it would attempt to migrate small jobs from the Batch region into the Mixed region. This process was not always successful; depending on the existing job mix and how heavily the Mixed region was being used by interactive work.

Another problem the site experienced was job size "entropy". Immediately after checkpointing the system to change queue configurations, the Batch region would be empty. In this state it was easy for NQS and GRM to run large applications. Over time, the job mix would shift from running large applications to small applications. This would usually result when a non-optimally sized application would run in the larger queues. For example, a 108 PE application running in the pe128 queue would make room for NQS to schedule a small 20 PE application. After the 108 PE application completed only 108 or smaller applications would be selected. Generally a set of 64 PE, 32 PE and 8 PE applications would be launched and 128 PE applications would be starved.

To resolve this problem, NERSC wrote a "de-fragment" script that would stop the small queues, checkpoint a selected collection of smaller jobs and allows a larger job to start. This script was run manually, usually once or twice a day, to keep the large job throughput reasonable.

One of the NERSC systems group staff members, Brent Draney, took it upon himself to manually monitor and adjust the workload on the system by migrating jobs and running the de-fragment script. Brent, who was referred to as "B-sched" by the site staff, also had a cron job send him an electronic page when the system idle time exceeded a certain level. It was because of this constant manual intervention that the system utilization and large queue turnaround values were as high as they were during this time period.

This configuration achieved the goal of managing the interactive workload on the system but did not really achieve the other site goals of minimizing idle time and queue turnaround without extensive automated and manual intervention.

NERSC T3E Scheduling Configuration after the UNICOS/mk 2.0.4 Upgrade

NERSC upgraded to UNICOS/mk 2.0.4 in March 1999. The PScheD software was put through a series of tests in non-production dedicated timeslots and proved to be robust and stable. Within two weeks of the upgrade mcurie was re-configured to use a single uniform 644 PE APP region with one PScheD domain. The GRM configuration was set with app_max=1 and abs_app_max=2. This allowed at most one non-prime application on a PE with at most two applications per PE overall (at least one must be prime). PScheD has been configured to run the load balancer, the gang scheduler and the prime job feature of the resource manager.

The site also began running a beta version of the 2.0.5 GRM service limits. NERSC requested this feature in a design SPR. It allows the site to limit the total interactive workload to a configurable value - 132 PEs during the day and 4 PEs during the midnight hours (22:00-03:00). GRM will not launch an interactive application if doing so would cause the total interactive load on the system to exceed this value. The blocked applications are held in the GRM wait queue. In order to provide a responsive interactive system for code development, NERSC runs all interactive sessions as "prime" jobs between the hours of 05:30 and 22:00.

The NQS configurations were modified to over-subscribe the system with a global MPP PE limit of 832. In addition, the NQS scripts and configuration files were modified to allow jobs in selected queues to run with prime status. Each configuration specifies which queues are allowed to run prime jobs and the total amount of prime batch work allowed at any point in time. NERSC uses this to prime jobs in the large queues. Since prime jobs not only preempt non-prime work but also have GRM launching priority, this is an effective mechanism to keep large jobs running on the system, reversing the effects of job size entropy.

The torus packing scripts manual migrations and "B-sched" is no longer in operation. The system is much simpler to manage and utilization has improved substantially. To everyone's delight, during a formal acceptance of the PScheD software, the system ran for two weeks with an average utilization of greater than 92% based on sar user plus system time in the APP region. In multiple 24-hour periods, the site has averaged 95% utilization.

NERSC has found that if the system has sufficient work in the NQS queues, they can easily achieve greater than 90% sustained utilization with this configuration. The site also feels they can improve upon this by enhancing the NQS control script.

PScheD Continued Success at NERSC

NERSC users are given an allocation based on PE connect hours. As part of the accounting system, a database of all MPP accounting records is kept. The graph in Figure 1 shows the PE connect time in hours per day broken down into application size. The data has been smoothed with a seven-day moving average. No effort was made to adjust for downtime or system dedicated time. Using this data, Table 6 shows the average utilization by connect time before the 2.0.4 upgrade, since the upgrade, and since the system was re-configured to its current state.

Figure 1.

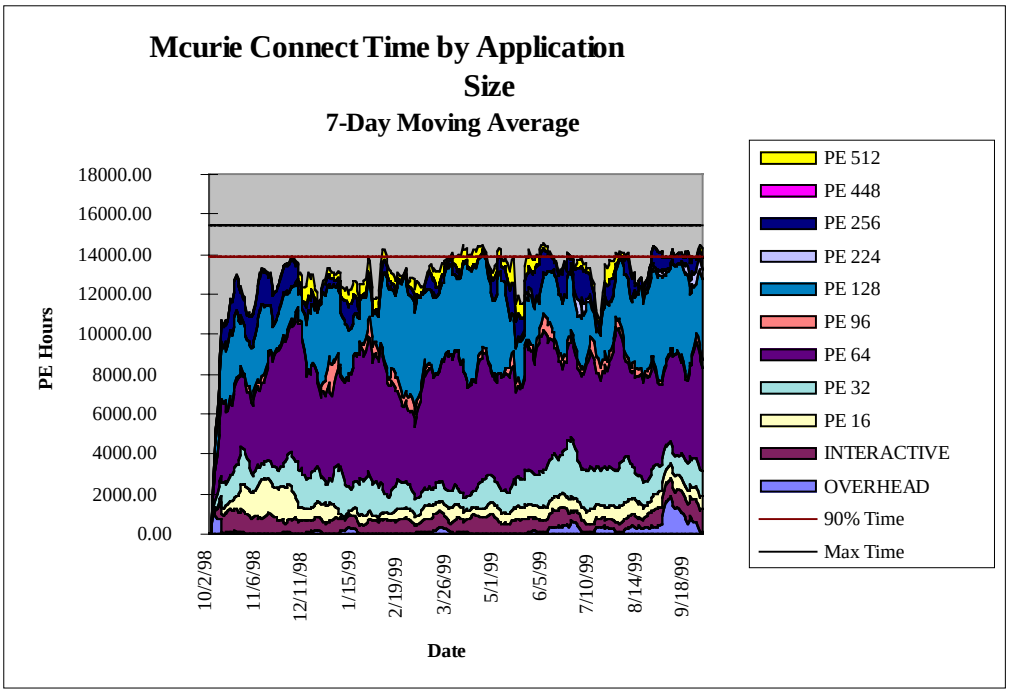


Table 6. Average System Utilization

| Dates | Utilization | Comments |
|---|---|---|
| 10/01/98 - 03/04/99 | 79.4% | Prior to 2.0.4 |
| 03/05/99 - 03/24/99 | 85.6% | Since 2.0.4 |
| 03/25/99 - 05/08/99 | 90.2% | Current Configuration |
| 05/09/99 – 09/30/99 | 87.3 | Allocation Problems |

After the system was upgraded and phased into its current configuration the system utilization increased to greater than 90 percent, approximately a 10 percent increase from the 2.0.3 configuration. NERSC users were able to spend their quarterly allocations more quickly than before. This resulted in excessive system idle time as users ran out of allocation. NERSC was plagued with allocation shortfalls for the remainder of the fiscal year, which resulted in a disappointing overall utilization of 87.3 percent since early May. With sufficient workload volume and distribution, NERSC has observed sustained connect time utilization of between 90 and 92 percent.

Using the ability to prime large batch jobs NERSC has been able to reduce the queue wait times for the large queues. The graph in Figure 2 shows the monthly average queue wait times since October 1998. The wait times are now consistent across the queues and are under 16 hours, except for the long128 queue, which has a 12-hour time limit and a run limit of one. The pe16 queue is a recent addition for which historical data is not available.

Clearly, NERSC's goals of minimizing idle time, managing the interactive workload and providing even queue turnaround are being met.

## Further Progress From Last Paper

The improved scheduling characteristics provided by PScheD and GRM in UNICOS/mk 2.0.4 have prompted NERSC to pursue several further goals in scheduling their T3E.

Improved turnaround for jobs using large (>64) numbers of PEs made it feasible to develop large debug queues that run in prime time, using the prime job feature to preempt lower-priority work where necessary. NERSC feels that this has encouraged and enabled more users to scale their applications to greater numbers of PEs, and thus increase the proportion of T3E resources used for capability computing.

## FY 2000 Future Plans

The current release of PScheD has demonstrated much greater robustness and reliability. As a result of these improvements, NERSC is planning more extensive tests of gang scheduling to see if further efficiencies can be attained with NERSC's dynamic workload. Setting the GRM attribute "app_max=2" and using NQS and interactive service limits to control the total volume of work on the system should allow the load balancer to align applications in a more optimal distribution on the torus. As part of the development of a cross-system utilization benchmark, a preliminary test of gang scheduling has been run with an artificial job mix and a radically simplified queue structure. Under these test conditions, PScheD continued to demonstrate excellent stability. The job mix and queue structure of the test make it very difficult to extrapolate performance to what might be expected with NERSC’s full set of queues, priming, and alternate configurations.

The success of the debug_medium queue, allowing 128 PE debug jobs to run in prime time, has encouraged the on-going development of a full-machine debugging capability for prime time. This queue, and an associated schedule time slot, will allow users to debug jobs up to 512 PEs during working hours.

This work was supported by the Director, Office of Advanced Scientific Computing Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract number DE-AC03-76SF00098.