The Transition from NQE to PBS

Jim Glidewell, Boeing Shared Services Group

Abstract

We have recently made the transition from NQE to PBSPro on our 256 CPU Origin 3800. In this paper, we will briefly describe our configuration, detail the reasons we chose PBSPro, and discuss the methods we used to make the transition a successful one for our user community.

Background

Our datacenter provides a shared engineering computing resource to The Boeing Company as a whole, with users throughout the United States, but with the bulk of our user community in the Puget Sound (Seattle) area. We are a long-time Cray shop, over 20 years, and have a mix of vendors including Cray, Sun, IBM, SGI, and STK. We tend to make heavy use of enhanced operating system features such as DMF, job and project accounting, and various scheduling features.

Our Origin 3800 is the first serious SGI high performance computing system in our datacenter. After running an eight-CPU Origin 2000 for a few years, our customers requested a significant improvement in non-vector compute equipment. We responded to this request by acquiring a 64 CPU, 64 gigabyte, Origin 3800. The success of this platform, coupled with a growing need for compute power, caused us to expand this system to 256 CPUs and 384 gigabytes of memory at the end of last year.

The workmix of jobs on this system can be broken into three general categories: a fairly small number of large parallel jobs, requiring 32 to 64 CPUs or more; a slightly larger number of moderate parallel jobs, requiring 4-16 CPUs; and finally a fairly large number of single CPU jobs, with varying memory requirements.

The Selection of PBSPro

While we had been reasonably happy with the scheduling functionality of NQE on the Origin 2000, we felt we needed to move to a new batch scheduling system with this new, much larger, system. We considered a number of batch scheduling options, including NQE, LSF, DQS, and PBS/PBSPro.

After significant deliberation, we chose PBS as our preferred batch scheduler. Reasons for this decision included: shared roots and design philosophy with NQS, the familiarity that many of our Aerodynamics had with PBS from experiences at NAS, and the availablilty of source code for both OpenPBS and the commercial PBSPro. In addition, PBS appeared to be fairly robust and flexible, and had an active user community. Finally, it was widely available on a number of platforms, including Cray.

This left us with the decision between OpenPBS (free) and PBSPro (commercial). We saw benefits to both options, but were swayed by a number of factors in PBSPro’s favor. PBSPro was under active development, and had a professional full time support staff.

And it offered the dynamic management of cpusets, which we were unwilling to commit to during initial implementation, but which we suspected that we might serious need in the future. We chose PBSPro 5.1 as our new batch scheduling system.

Initial Evaluation

To verify that PBSPro was a suitable product for our batch scheduling needs, we did an in-house evaluation of PBSPro. We found several minor bugs, many of which had already been corrected, along with a few design deficiencies which Veridian (the PBSPro vendor) agreed to address for us.

The one major design flaw with PBSPro for our site was the decision to always create a separate cpuset for every batch job, regardless of size, with the minimum cpuset created consisting on an entire node (on our Origin 3800, 4 CPUS.) We viewed this as a serious impediment to our ability to run a large number of single-cpu jobs, with the worst case being the case where all jobs were single-CPU, resulting in 75% of the CPUs being idle while the machine is fully subscribed. Clearly, this was not a good situation for a system with a significant number of single-CPU jobs.

Veridian agreed to address this issue, with the understanding that they would offer a solution to this issue within the following six months. Based on this timeline, we decided to go ahead with PBSPro and defer the implementation of PBS-managed cpusets until this feature was available.

Elements of a Successful Transition

To make the transition from NQE to PBS as painless as possible for our users, we focussed our efforts in four areas: transition planning, user documentation, transition tools, and user beta testing.

The transition plan detailed the timeline for moving from our existing batch subsystem to PBSPro. Details from the timeline are discussed below and in the appendix.

Our experience with PBSPro documentation indicated that the existing documents were far too large and unwieldy for our customers to absorb. We needed a relatively short document that would provide our users with the basic information they would need to move forward from NQE to PBS. To this end, we created a transition document entitled “PBS on the Origin 3800” (appendix 1) which provides the basic information a user would need to make the transition to PBS on our system.

In addition to this document, we provided a few command line tools to make the conversion as simple as possible. The primary conversion tool was provided directly by Veridian – the nqs2pbs command. This script takes an NQS job script as its input, and produces a job script with is runnable under both NQS and PBS, with various NQS options all mapped into their PBS counterparts. This tool worked fairly well, with a few caveats, particularly with regard to memory limits, that we addressed specifically in the transition document.

Since both PBS and NQE use commands such as “qsub” and “qstat”, we felt the need to somehow allow users to specify *which* “qsub” they are trying to use. Initially, we considered renaming the PBS commands, prefixing each with a “p” or some similar convention. This seemed awkward and problematic both during and after the transition, so we solicited suggestions from the PBS mailing list. Wendy Lin, of Purdue, suggested a cover script that would force the PATH and MANPATH to the desired batch system, a suggestion that we adopted. We provided 2 such scripts: “pbs” and “nqs”. These scripts merely made sure that the PATH and MANPATH were appropriate for the desired batch subsystem, then executed the specified command. Thus:

```
nqs qsub my_job_script  # executes the NQS version of qsub
pbs qsub my_job_script  # executes the PBS version of qsub
qsub my_job_script  # executes the default qsub
```

With the user transition document and transition tools in place, we were ready to begin the transition in earnest. It was time to let users at PBS…

The User Beta Test and Transition

The ability to run both NQE and PBS simultaneously on a single machine allowed us to make the transition for our users as gentle as possible.

We installed PBS as the “secondary” batch subsystem, with NQE still serving as the default. We announced the availability of PBS and suggested that users begin experimenting with it as soon as possible. Users could avail themselves of the nqs2pbs script, and use the “pbs” cover script to select PBS. We ran in this mode for a full month. Except for our Overflow users, who migrated immediately, we saw little movement toward PBS for production jobs, though there were clearly users experimenting with PBS.

The next step was to switch to PBS as the default batch subsystem. We expected this to be a fairly short interval, one to two weeks, with most of the user conversion having taken place during phase 1. This was not the case, however. Because of other demands on our user’s time, and the arrival of the Christmas holidays, this phase was stretched to just over a month.

Once PBS became the default, the number of jobs submitted to NQE dropped dramatically. It appears that our users had used the nqs2pbs script to convert their job scripts to run under either batch system, and were simply letting the default batch system accept their jobs.

Shortly after the change to default to PBS, we were in a position to relatively quickly move through the last two phases of the transition: the stopping of all NQE queues, followed a few weeks later with the removal of NQE entirely.

The transition was complete.

Conclusions

We had serious concerns about user acceptance of a new batch subsystem after many years of experience with NQS and NQE on both our Cray and SGI platforms. To ensure the success of our transition to PBS, we did a fair amount of up-front work to provide our users the tools, documentation, and a well thought-out transition plan, which would allow them to move to PBS with a minimum disruption to their workflow. The efforts we made up front paid off well, and our user community has accepted PBS and has made excellent use of it capabilities.

It appears that our choice of PBSPro has been a good one, and we have seen that PBSPro will allow us to maximize the resources of our Origin 3800.