NQE UPDATE

Janet Clegg and *Daryl Coulthart*, Cray Research, Inc.

ABSTRACT: *NQE 3.0 provided a new architecture allowing greater flexibility in workload management. This new architecture is the basis for cluster-wide scheduling, parallel execution, and cluster-wide rerun of jobs. This paper will explain the NQE product strategy, execution, and plans for the expansion of a tool for a single system to a tool that can also be used for multiple Cray systems, multiple IRIX systems, multiple SGI heterogeneous systems, and a network of systems from multiple vendors.*

1 NQE Overview

The Network Queueing Environment (NQE) is a workload management product that schedules, monitors, and controls the execution of jobs in complex heterogeneous environments. NQE provides a consistent user interface across multiple heterogeneous UNIX systems. Users can submit their jobs to the NQE batch complex, specify the resources needed for the job, and not worry about where the jobs will run. Based on the resources needed and the resources available in the NQE batch complex, NQE will select an appropriate server, route the job request to that server, schedule and initiate the job request, and return STDOUT and STDERR files to the user.

NQE is the follow-on product to NQS which was originally provided to improve system performance by managing the workload in order to control resource bottlenecks and minimize swapping. NQE expands this functionality by providing the ability to balance the workload across one or more systems.

2 NQE Workload Management

The NQE clients submit UNIX user shell scripts from UNIX systems to the NQE servers. The commands and displays are identical across all systems, so that users see a uniform environment and interface regardless of their workstation type. Motif based X-Window client programs display the status of jobs and the system load on each NQE server.

NQE contains a job scheduling program to route and schedule batch jobs and interactive commands to an appropriate server in the NQE batch complex. Scheduling and selection is based on resources requested by the user and policies formulated by the site. These are policies and schedules implemented in the Network Load Balancer (NLB) and the NQE scheduler, which provide status and control of work scheduling, either balancing the network load and routing jobs to the best available server, or honoring requests for specific servers and resources.

Job requests may be submitted to the NQE scheduler which in turn will schedule the job for execution according to the NQE scheduler policy. Job requests can still be submitted directly to NQS servers or routed using NLB policies.

NQE's batch component, the Network Queueing System (NQS) provides fault tolerance and a rich set of administrative controls for high-end UNIX servers.

Job dependency gives users the ability to specify criteria which must be met before a job request will initiate. The most common application is to run a set of jobs in sequence.

NQE provides complex-wide status and control of all jobs though the NQE GUI. All jobs owned by a user are visible regardless of what system they are on.

The File Transfer Agent (FTA) provides unattended file transfer across a network using the `ftp` protocol.

NQE uses FTA or `rcp` to return job output. Output from jobs is returned to the location requested by the user or to the current working directory from which the job request was made.

In addition, both an interactive utility (`ftua`) and a command line interface (`rft`) are available to initiate file transfers on NQE servers. The command line interface is particularly attractive when needed in batch job requests. Network Peer-to-Peer Authorization (NPPA) solves the problem of needing to supply passwords in job scripts or in .netrc files.

The user interface provides the ability to submit, display, status, signal, and batch jobs, using a command line interface or a GUI. The NQE GUI provides a consolidated user interface for NQE user functions.

3 NQE Architecture

NQE consists of clients and servers. The Master Server is the NQE server with the consolidated information about workload and system status in the Network Queueing Environment.

The NQE Clients provide all the necessary commands for users to submit jobs and monitor the status of their work.

The Master Server matches the incoming workload and priorities to the system load of NQE servers in the batch complex, and schedules jobs for the most appropriate NQE server. The selected NQE server may be the Master Server or an Execution Server. Jobs may arrive from other NQE Servers as well as from Clients.

There may be only one NQE Master Server in an NQE batch complex. The NQE Master Server runs the Network Load Balancer (NLB), the NQE SQL database, and the NQE scheduler. The NLB may be duplicated on other NQE servers, which may serve as a backup to the NLB if it becomes unavailable. The Master Server contains both client and server code, and may also function as a Client or an Execution Server as needed.

NQE Execution Servers collect information about system load and job status and send this information to the NLB on the Master Server (as well as to any backup Master Servers). Execution Servers use NQS, the Network Queueing System, to initiate job requests and FTA for file transfers. Execution Servers also contain client commands and may function as a Client as needed.

4 NQE Product Strategy

NQE provides workload management for Cray UNICOS, UNICOS/mk and several UNIX workstation systems including IRIX. The focus for NQE is to provide workload management on computation server systems, provide user interfaces for connectivity from desktop systems to computation server systems and support clusters in a hetrogeneous environment.

NQE provides consistent base capabilities for all supported platforms and provides value added value on UNICOS and IRIX by supporting unique operating system features. An important theme in NQE is to provide open interfaces to allow installations to customize NQE for their unique requirements.

The NQE 3.0 release introduced a new scheduling architecture that uses a central job database. The central database holds jobs until they are ready to execute. This feature provides complex-wide job rerun because, at the time of execution, a copy of the job is forwarded to the execution server while the original remains in the central NQE database.

The central database and scheduler are layered on top of the NQS and the NQE 2.0 structure. This maintains the NQS and the NQE 2.0 environment while providing significant new functionality. The new architecture will be the basis for parallel job scheduling and parallel job management. Over time, all components of NQS and current architecture will be replaced. As new functionality is implemented, the NQS presence will subside.

Beginning with the NQE 3.0 release, the dependency on the NQS architecture is being reduced. For example, a new scheduler provides flexible, site-modifiable scheduling without using the scheduling portion of NQS. Today, NQS is still required to initiate the job, but that functionality will also be replaced in a future NQE release. However, NQE will always retain the ability to inter-operate with public domain, protocol-based NQS systems.

5 NQE Releases in 1996

In the interest of responding to customers' needs more quickly, the NQE project is delivering smaller, more frequent releases. Three versions have been released in 1996, with one more planned for December 1996. The following is a list of NQE releases since the start of the product:

NQE 1.0 released December 1993.
NQE 1.1 released June 1994.
NQE 1.2 released December 1994.
NQE 2.0 released May 1995.
NQE 3.0 released April 1996.
NQE 3.0.1 released June 1996.
NQE 3.1 released September 1996.
NQE 3.2 is planned to release December 1996.

5.1 NQE 3.0 Features

The NQE 3.0 release has the following new features:

- Integrated GUI for building and submitting job scripts, monitoring job status, controlling jobs, and monitoring NQE system status
- Display of FTA file transfer information in the GUI status displays
- GUI interface to cron, for launching predefined jobs at specific dates and times
- Ability to modify the scheduling of NQE job requests with Tcl (Tool Command Language)
- DCE/DFS integration, to provide NQE users with transparent access to their DFS files, while running NQE job scripts
- batch complex rerun, to automatically rerun jobs on another server if the execution server goes down
- Extensible collector to gather and display additional system information easily
- Interactive Load Balancing, to execute interactive UNIX commands on servers based on scheduling policies defined in the NLB
- Byte level recovery of FTA file transfers, so that file transfers are restarted from where they left off, in the event of network or system failures
- Support of new versions of IRIX, Solaris, and AIX UNIX operating systems

5.1.1 NQE Client/Server Model

NQE 3.0 provides a significant improvement to workload scheduling for NQE customers. This new architecture allows greater flexibility in workload management and forms the basis for parallel execution, complex-wide scheduling, and

complex-wide rerun of jobs. The model is centered around a client/server SQL database and a central scheduler. This new NQE scheduler, available in 3.0, is written in the popular Tcl scripting language. This new feature allows sites to create a customized scheduler or to rework NQE's existing scheduler. For example, sites can elect to schedule based on NQE 2.0 style NLB policies, in which case the 3.0 Tcl scheduler calls the NLB policy module. But a site may instead choose to look at all jobs in the batch complex and pick out a specific one to run first. With NQE 3.0 scheduling, sites now have the ability to enhance scheduling according to their individual needs.

5.1.2 DCE/DFS Integration

NQE 3.0 provides the facility to perform DCE authentication at job initiation for HP-UX, OSF/1, and Solaris. If so configured, DCE authentication will be attempted when a password is supplied with the `cqsub` command or the `nqe` GUI. If password validation is not in effect at the NQE server accepting the job, the password will only be used for DCE authentication and the usual NQS validation (such as file validation) will also take place.

If DCE authentication is successful, NQE supports access of DFS files within the user's DCE cell. This includes support of job scripts, job I/O, return of job output, and $HOME directories in DFS file space.

DCE authentication is not required for status, signal, and delete operations.

5.1.3 NQE GUI

The `nqe` command invokes a new graphical user interface (GUI) that lets users do the following:

- Use the Submit window to open and edit a job script; to save changes made to a job script; to submit a request to NQE; to view, segment, delete, or reset the NQE GUI log; and to set or unset password. The Submit window will set and save job-related options.
- Use a launching capability to submit a job request periodically, at specific or repeating intervals.
- Use a Status window to verify the status of requests and FTA file transfers. Use the Status window also to delete a request, to send a specified signal to a request, to get a detailed status of a request, and to set or unset password.
- Obtain context-sensitive help by gliding the mouse cursor over a menu or field name in a window. A brief description of the menu or field will appear at the bottom of the display.
- Use the Load window to view a continually, updated display of the system load for NQE servers in the complex. This data may be grouped by host or by type of data. Users may also view data about a specific NQE server.
- Use the Config window to set specific user preferences and see how variables are currently set.

5.1.4 Complex-Wide Job Rerun

If a job has been assigned to an NQE server and that server has been down for a specified length of time, NQE 3.0 provides support for rerunning that job request on a different NQE server in the batch complex.

5.1.5 Interactive Load Balancing

The new ilb feature executes commands on an NQE server chosen by the NLB. To use this, enter the ilb command followed by the command to execute. The NLB is queried to determine the machine to execute the command. The command is executed and I/O is connected to the terminal session or to an optional pipe to another command.

5.1.6 FTA Enhancements

FTA has been enhanced in NQE 3.0 to provide byte-level recovery of interrupted file transfers, to display additional data from the NLB, and to display the status of FTA file transfers in the `nqe` GUI.

5.1.7 Extensible Collector

It is now possible to store dynamic, site-specific information in the NLB database. This information is periodically sent to the NLB by each collector process, along with the data that is normally stored and updated in the NLB. Once the customized data is in the NLB, it can be used in policies or displays just as any other NLB data. NQE collects and stores the customized data, but the site defines, generates, and updates it.

The new `ccollect -C` file name option and the new NQE_CUSTOM_FILE_LIST variable provide the support for this feature.

5.1.8 Application Program Interfaces

NQE provides a language callable interface to `cqsub`, `cqdel`, and `cqstatl`. Sites can use these interfaces to provide customized commands for their users.

5.1.9 Supported OS levels

NQE 3.0 supports the following workstation OS levels

- AIX 4.1.3
- IRIX 5.3
- IRIX 6.1
- HP-UX 9.0
- OSF/1 V3.0
- Solaris 2.4
- SunOS 4.1.3

5.2 NQE 3.0.1 Features

NQE 3.0.1 is a limited feature release primarily to add support of IRIX 6.2.

NQE 3.0.1 supports the project nqme and array services feature of the POWER CHALLENGEarray product.

5.2.1 Project Name

The IRIX project name is handled in a manner similar to UNICOS account names. NQE sets an appropriate project id,

either from the `/etc/project` file or from the -A option of the `cqsub` and `qsub` commands. The project name is displayed by the `qstat -f` and `cqstatl -f` commands.

5.2.2 Array Session

Using the project ID, NQE acquires a global array session handle (ASH) prior to initiating a job. The global ASH value is accessible outside of the job for use with the array services user commands. The global ASH is displayed in the job log.

5.2.3 Supported OS levels

NQE 3.0.1 supports the following workstation OS levels

- AIX 4.1.3
- IRIX 5.3
- IRIX 6.1
- IRIX 6.2 (new)
- HP-UX 9.0
- OSF/1 V3.0
- Solaris 2.4
- SunOS 4.1.3

5.3 NQE 3.1 Features

NQE 3.1 is a release primarily to support all platforms simultaneously. With this release, NQE 3.0 features are available for UNICOS and UNICOS/mk. In addition, new utilities have been added to configure NQE and to maintain multiple versions.

5.3.1 Version Maintenance Utility

A new NQE version maintenance utility is provided with the `nqemaint`(8) command. This utility is used to switch between versions of NQE. It handles the setup of all the symbolic links that NQE requires to operate normally.

5.3.2 Configuration Utility

A new NQE configuration utility is provided, which is invoked by using the new `nqeconfig`(8) command. The NQE configuration utility is used to modify the NQE configuration values specified in the `nqeinfo` file, as well as to specify additional configuration variables not included in the default configuration.

5.3.3 DCE/DFS Support

With NQE 3.1, support for DCE authentication and DFS file space is available for AIX, HP-UX, OSF/1, Solaris, and UNICOS.

New DCE authentication will be obtained before restarting a checkpointed job on UNICOS.

On UNICOS only, job requests with DCE credentials can have their credentials automatically refreshed. This is provided for sites with batch jobs that run longer than a user's credential expiration period. The refresh rate may be set using the NQE_DCE_REFRESH variable in the nqeinfo file.

5.3.4 Asynchronous, All-Platform Release

Previous to NQE 3.1, NQE released synchronously with UNICOS. Due to differences in release requirements, NQE released at different times for UNICOS than for other platforms.

With NQE 3.1, NQE is packaged separately from UNICOS, and all platforms are supported with a single NQE package. You should now be able to upgrade all of your NQE servers and clients with the same NQE release at the same time.

As a result, all customers, including Cray customers, must now order NQE as a separate package. This is true even for PVP customers who are only running the NQE subset, formerly known by the separate product names NQS and FTA.

5.3.5 NQS and FTA Become the NQE Subset

Beginning with UNICOS 9.2, NQS and FTA will no longer be included with UNICOS releases, nor will they be included with UNICOS/mk releases. These two products are now known as the NQE subset.

Starting with UNICOS 9.0, customers who previously used only NQS and/or FTA should now order the NQE subset in order to get updates for NQS and FTA.

The NQE subset will continue to be licensed as part of UNICOS and available with no additional charge. No FLEXlm license is necessary when running the NQE subset on a UNICOS system.

5.3.6 NQX Becomes NQE/UNICOS

Starting with NQE 3.1, UNICOS customers who previously ordered the advanced UNICOS Network Queueing Extensions (NQX) should now order NQE for UNICOS. NQE provides the functionality previously offered by the combinati8on of NQS, FTA, and NQX products. The Flexlm license for NQX will be replaced with a Flexlm license for NQE.

5.3.7 Combined documentation for all components

The NQE documentation set was revised and reprinted for NQE 3.1. Documentation that was previously provided in the *Network Queueing System (NQS) User's Guide*, publication SG-2105, the *UNICOS NQS and NQE Administrator's Guide*, publication SG2305, and the *FTA User and Administrator Manual*, publication SG-2144, has been incorporated in the NQE 3.1 documentation set. The documentation set includes the following manuals:

- *Introducing NQE*, publication IN-2153 3.1
- *NQE Administration*, publication SG-2150 3.1
- *NQE User's Guide*, publication SG-2148 3.1
- *NQE Release Overview and Installation Bulletin*, publication RO-5237 3.1

PostScript files of these manuals are available from the following World Wide Web (WWW) URL:

`ftp://ftp.cray.com/pub/nqe/nqe31`

5.3.8 DynaWeb documentation

The NQE 3.1 documentation is provided on-line through the Cray DynaWeb server. The Cray DynaWeb server software and

the documentation is included with the release on a CD-ROM. The Cray DynaWeb server allows users to access information by using a WWW browser, such as Netscape. for more information about the Cray DynaWeb server implementation, see the “Cray software documentation to be available on the WWW” article in the June 1996 *Service Bulletin*.

Craydoc is no longer included in the NQE release.

5.3.9 Supported OS Levels

NQE 3.1 supports the following workstation OS levels

- AIX 4.2 (new)
- HP-UX 9.0
- IRIX 6.2
- OSF/1 V3.0
- Solaris 2.4
- SunOS 4.1.3
- UNICOS 9.0, 9.1, 9.2 (new)
- UNICOS/mk 1.2.4, 1.3 (new)

5.4 NQE 3.2 Plans

NQE 3.2 is a limited feature release primarily intended to provide new hardware and OS level support. The planned release date is December 1996.

5.4.1 T90 recovery feature

If a job is terminated through receipt of a SIGRPE or SIGUME signal, both of which indicate hardware problems, NQE now re-queues the job, rather than deleting it, if the job is re-runnable or if the job is re-startable and has a restart file. By default, each NQS job is both re-runnable and re-startable. These defaults can be changed with the `-nr` and `-nc` options on the `qsub` and `cqsub` commands or with the `-r n` and `-c n` options on the `qalter` command.

5.4.2 T3D rollin/rollout

Beginning with the MAX 1.3.0.3 release, NQE will support checkpoint of a PVP job that is running processes on an attached T3D machine. NQS will invoke the `mpprollout` command when checkpointing a PVP job if the PE limit for the job is greater than zero.

5.4.3 Origin 2000 support

NQE 3.2 will support IRIX 6.4 on an Origin2000 platform.

5.4.4 Checkpoint/Restart for IRIX 6.4

NQE will support checkpoint and restart on the Origin2000 platform with IRIX 6.4. Functionality will be very similar to that available on UNICOS systems.

5.4.5 Simpler Flex license

Beginning with the 3.2 release, NQE will no longer enforce restrictions on the number of users who can simultaneously run jobs. NQE will only check for the “nqe_nl” license. The “nqe_fl” and “fta_nl licenses will no longer be required.

5.4.6 Supported OS levels

- AIX 4.2
- HP-UX 9.0
- IRIX 6.2
- IRIX 6.3, 6.4 (new)
- OSF 4.0 (new)
- Solaris 2.5 (new)
- SunOS 4.1.4
- UNICOS 9.0, 9.1, 9.2
- UNICOS/mk 1.2.4, 1.3

6 NQE Future Directions

The mission for the NQE project is to provide the tools that allow sites to manage their workload among one or more computation servers. As high-performance computing evolves and needs change, NQE is evolving to meet these needs.

The future direction for NQE is to continue evolving workload management for clusters, array systems, and heterogeneous networked computers. Resiliency, fault tolerance, and performance continue to grow in importance as the number of nodes increases. Scaling is necessary to distribute the workload among the increasing number of nodes. NQE will evolve to meet these needs and continue to provide features unique to UNICOS, UNICOS/mk and IRIX. Themes for future releases include:

- support T3E unicos/mk features, including checkpoint/restart and political scheduling
- support values added IRIX features
- enhance NQE resiliency
- improve NQE performance
- improve NQE scalability
- provide additional scheduling capabilities
- provide complete DCE integration
- continue simultaneous release for all platforms
- continue small and frequent releases