Distributed Computing with NQE 3.0

Victor Hazlewood, San Diego Supercomputer Center, P.O. Box 85608, San Diego, CA 92186

ABSTRACT: *The "Network as the Computer" concept has been a vision of the HPC community for several years now; however, no one software product has proven to be the standard for the "Network as the Computer" model. CraySoft NQE 3.0 was released in May 1996 as a product that could be used as a starting place for distributing jobs in a production heterogeneous computing environment. An evaluation of NQE 3.0 was initiated at SDSC. Features and SDSC requirements investigated include: platform availability, Web interface, scheduler configuration and flexibility, and DCE compatibility. The results of the evaluation are presented.*

1 Introduction

The San Diego Supercomputer Center (SDSC), a National Laboratory for Computational Science and Engineering, has traditionally provided big iron supercomputing resources including, Cray Vector processors, Cray MPPs, and Intel MPPs. Workstations and workstation-class servers have been used at SDSC for desktop windowing, visualization, web service, email services and other non-compute intensive services. With the ever increasing computational power of workstations and deskside systems such as the R10000 and UltraSPARC systems from SGI and Sun Microsystems, many compute intensive applications can now run effectively on these systems, either on single-cpu systems or in parallel. Providing clusters of these systems for running parallel applications or using spare cycles on otherwise idle desktop machines becomes a viable use of these increasingly powerful systems.

SDSC plans over the next several months to deploy over 30 R10000 Indigo systems and UltraSPARCs for desktop and classroom use (see Diagram 1). Complementary to these missions is the goal to make the spare cycles of these systems available to researchers as additional computational resources.

To deliver these capabilities, SDSC initiated a project to evaluate queueing software which could provide a suitable batch environment for delivery of these workstation cycles. One of the many packages available is called Network Queueing Environment (NQE) from CraySoft. NQE was evaluated in detail and its suitability in providing a heterogeneous batch queueing environment is presented in this paper. Evaluation criteria included platform availability, flexible batch environment, load balancing capabilities, fault tolerance, ease of installation into the SDSC workstation environment, security, DCE compatibility, support and cost.

2 NQE Overview

The Network Queueing Environment (NQE) allows a user to submit batch jobs which will run on an execution server which is a member of an NQE domain. Client software is available to allow batch job submission and to request status. Server software allows acceptance, initiation and return of the batch job to the user independently of an interactive terminal session. Additional software provides load balancing, scheduling and fault tolerance features discussed in more detail below.

NQE is composed of three basic components:

the NQE client,
NQE execution server,
and NQE master server.

NQE clients are composed of software which allow users to submit, monitor and control batch requests.

NQE execution servers contain the client software, plus an NQS server, a system statistic collector (NLB collector), and an NQE database Light Weight Server (LWS) to allow scheduling and initiation of batch requests, file transfers, collection of system statistics for the load balancer, and fault tolerance capabilities.

NQE master servers contain all of the software of an execution server, plus the Network Load Balancer (NLB), the NQE database (mSQL), and the NQE scheduler. These components allow job routing and initiation based on site-specific criteria configured by the NQE administrator and by information collected on each server in the NQE domain.

3 Platform Availability

SDSC currently has a broad range of workstation and supercomputer systems including (listed by operating system); Solaris 2.5, AIX 4.1, IRIX 6.2, Digital Unix (OSF/1 V3), UNICOS 9, and SunOS 4.1. NQE supports each of these operating systems. CraySoft plans to continue support for new releases of these operating systems. As a consequence of the acquisition of Cray Research (the parent of CraySoft) by Silicon Graphics Incorporated (SGI), Integration of NQE has been included into the release of IRIX 6.2 as an optional product called NQS 3.1.

4 Installation

NQE has evolved to a state where installation on single stand-alone systems and complex NFS workstation environments is flexible and well structured. The workstation environment at SDSC could be considered a complex NFS environment. A large Auspex file server serves home directories, public domain and third party software packages, and scratch areas for all workstation architectures. To simplify the software installation process at SDSC only one system of each architecture and operating system type has the software areas mounted read/write. This system is called the installation platform. The full NQE distribution must be loaded on the installation platform. NQE software for each subsequent server or client is loaded on the installation platform with the NQE *addnode* command.

*addnode [-m | -e | -c] [-d spool_dir] *
[-r target_root] hostname

For SDSC, this process prevents the unnecessary duplication of software on each workstation of the same architecture type (see Diagram 2). The only disadvantage of this method is that NQE contains "root access" setuid programs (17 of them). This requires special effort to resolve or requires setuid privileges for each NFS file system that contains NQE software. At SDSC, we elected to copy the setuid programs to /var/nqe/bin on each local disk and create symbolic links to this area on the installation platform.

With these changes in place security policies were not violated. The most important security aspect was to prevent the requirement of setuid on the NFS mounts of client workstation systems.

The complete list of local file requirements known at this time for this environment are:

17 setuid programs in /var/nqe/bin
spool area (/var/nqe/spool)
/etc/nqeinfo symbolic link
update to /etc/services, /etc/inetd.conf,
/etc/init.d/nqe and symbolic link to /etc/rc2.d/S96nqe
/etc/fta.nppa for File Transfer Authority passwords

5 Web Interface

The Web interface with NQE 3.0 contains mostly cosmetic changes to the cgi-bin scripts available under NQE 2.0. Items addressed by the Web interface include:

job submission
job status
job control

The cgi-bin scripts are provided on the ftp site ftp://ftp.cray.com/pub/nqe. In conjunction with an NCSA httpd or Apache Web server these scripts provide all that is needed to allow job submission, status and control through a Web interface. The NQE Web interface requires the Web server to run as user "root." For security reasons, I recommend running an instance of the Web server solely for the NQE Web interface to limit exposure to possible security threats.

6 Load Balancer

One of the key requirements of using a product like NQE for a cluster of workstations is the ability to load-balance applications across the systems in the cluster. The load balancing component of NQE, called NLB, is comprised of several elements. The statistic collector runs on each execution server in the NQE domain and this program sends information concerning idle time, free memory, temporary disk space, etc. to the master server. The master server stores this statistical information in a database. At job submission, if the user selects a pipe or batch queue controlled by the NLB, the destination queue is selected by the computation of a policy. Policies are configurable and contain constraints and sorting functions for selection of a target execution server. The following is an example policy:

policy: SCALAR
constraint: [NLB_HARDWARE = "sun4m"] | |
[NLB_HARDWARE == "IP25"] ||
[NLB_HARDWARE = "alpha"]
sort: NLB_A_IDLE

The above policy selects only those architectures that match *sun4m*, *IP25* and *alpha* which are return values for the NLB_HARDWARE attribute sent by a collector from a SPARCstation 4, Power Challenge R10000, and a DEC AlphaServer, respectively. For each system that satisfies the constraints, the NLB computes the average CPU idle time as specified by the *sort* statement and chooses the system with the highest NLB_A_IDLE value.

Policies are added or changed by modifying the *$NQE_SPOOL/nlbdir/policies* file and running the NQE command *$NQE_BIN/nlbconfig -pol* to reread the updated policy file. In addition, the *nlbpolicy* command can be used to test policies. For the above example policy, the following would add the new policy:

 # vi $NQE_SPOOL/nlbdir/policies

 # $NQE_BIN/nlbconfig -pol
host: policy file read successful
 # $NQE_BIN/nlbpolicy -p SCALAR

NLB installation and configuration is well documented in the on-line documentation reader "cdoc" which comes with NQE.

7 The NQE Database (mSQL)

Another important feature of the NQE package is the NQE database. Known as mSQL. This subsystem, which runs on the master server, can provide additional scheduling services not available through the NLB. These services include:

Network-wide limits. If the NQE administrator wants to enforce NQE domain user limits, the mSQL can provide this.

Rerun of jobs network-wide. If a server fails, jobs running on that server can be rerouted and run on another server.

Site-specific scheduling. Site specific issues, such as application licensing issues can be used as criteria for submitting and running batch requests.

Diagram 3 gives an overview of the mSQL components in an NQE domain.

8 *csuspend*

The *csuspend* command provides the ability to enable and disable batch work. This feature allows idle cycles to be used effectively and prevent the batch work from interfering with interactive use of a desktop system. *csuspend* is a Korn shell script which uses *sar* to determine tty activity. The NQE administrator can set the threshold level of tty activity and, once met, the script suspend all batch work and prevents the acceptance of incoming batch requests. Batch activity is resumed once the tty activity falls below the threshold level. Being implemented as a script this features gives added flexibility to the NQE administrator to suspend jobs based on local criteria. This option proves to be invaluable for using idle cycles without interrupting interactive use of a desktop system.

9 Other features

Other features of possible interest to the reader, but not discussed in detail in this paper include the file transfer agent (FTA), and using the Distributed Computing Environment (DCE) within NQE. Both of these features are described in NQE documentation.

NQE claims DCE compatibility and SDSC plans to continue research into the use of DCE within NQE for future projects. Goals for DCE at SDSC include; no plaintext passwords on the network, DCE authenticated fta, and single sign-on capability.

10 Conclusion

SDSC plans to continue investigating ways to complement the supercomputer resources available to researchers by making effective use of workstation resources. This includes taking advantage of idle cycles on powerful desktop and server workstations with a batch environment.

It has been determined that NQE 3.0 is a viable choice for providing a batch environment for Unix workstation systems. The combination of compatibility with NQS, multiple-platform availability, fault tolerance for batch jobs and file transfers, ease and flexibility of installation, load balancing capabilities, and configurable scheduling provide a feature-rich batch environment for workstations and workstation clusters.

11 Acknowledgments

This work was funded in part by the National Science Foundation Cooperative Agreement ASC-8902825.

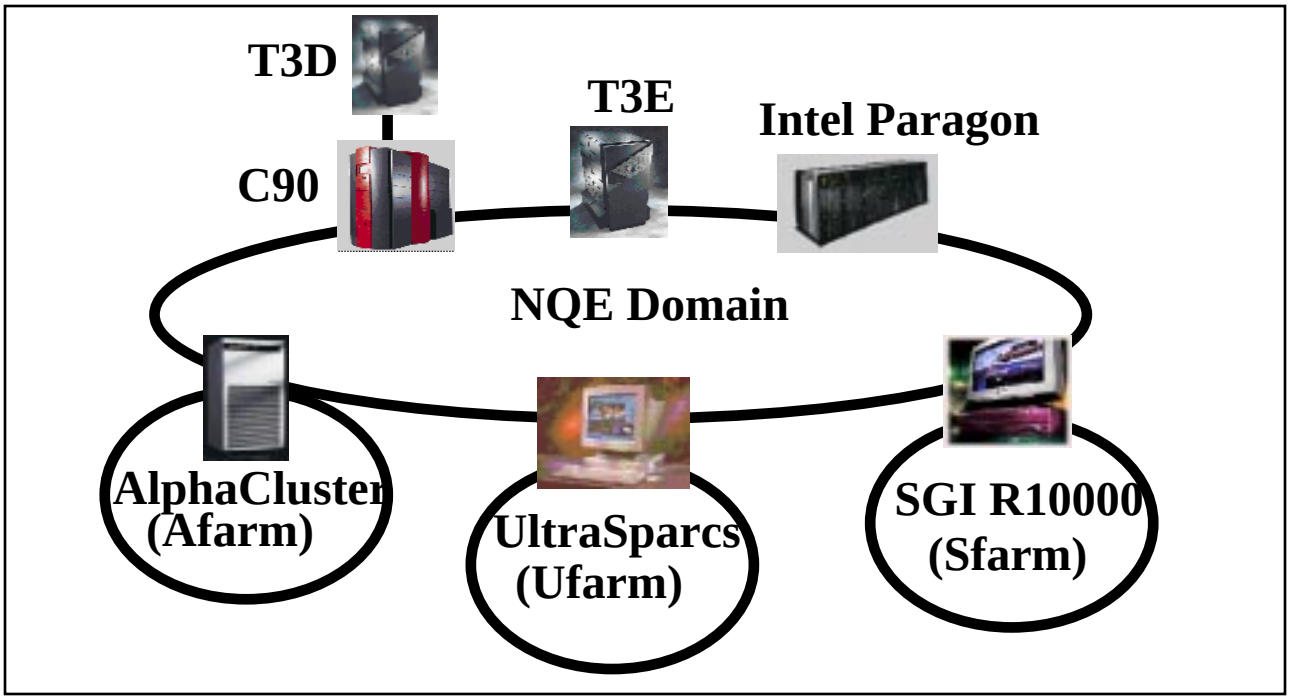

Diagram 1: NQE Domain Goal

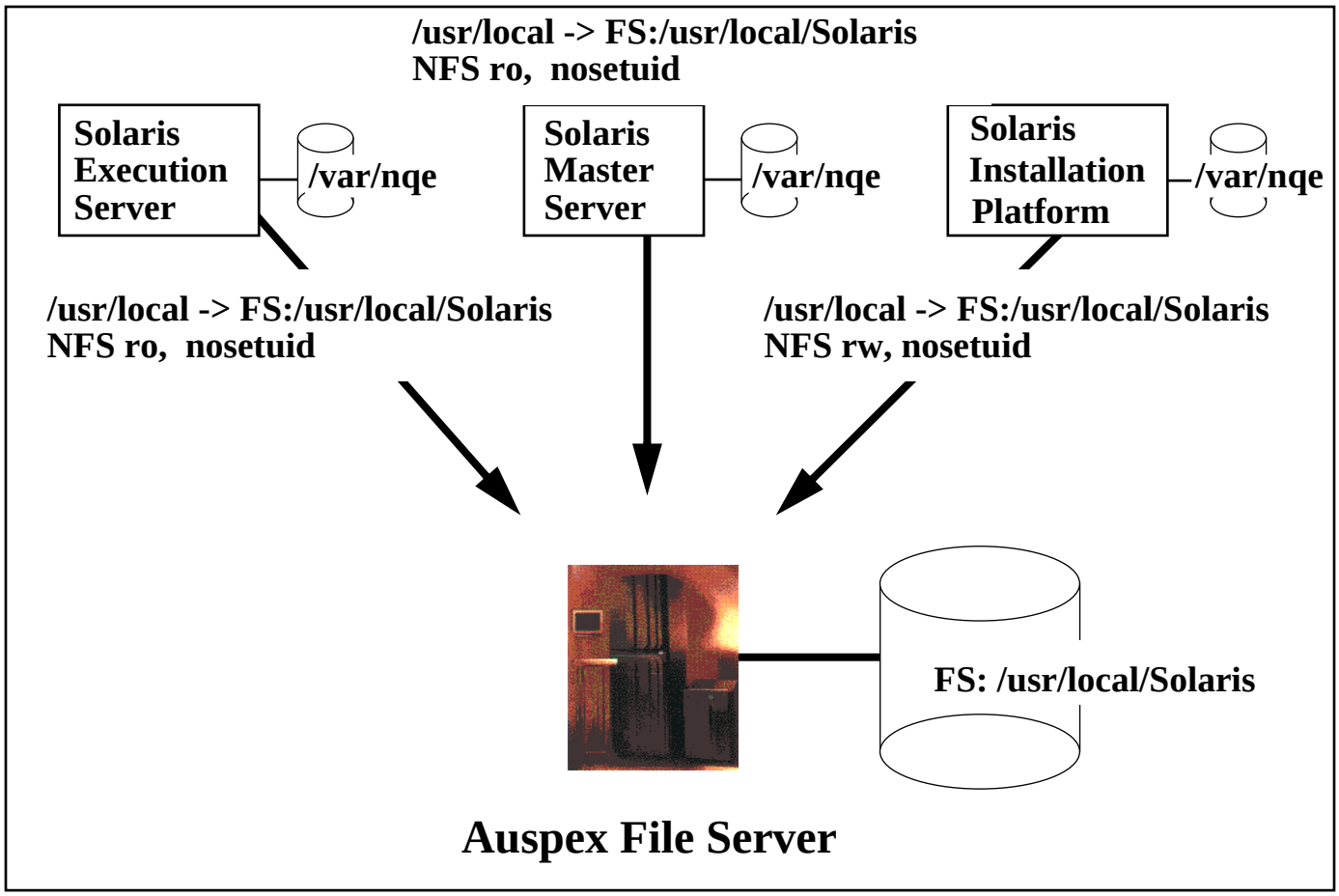

Diagram 2: NFS environment

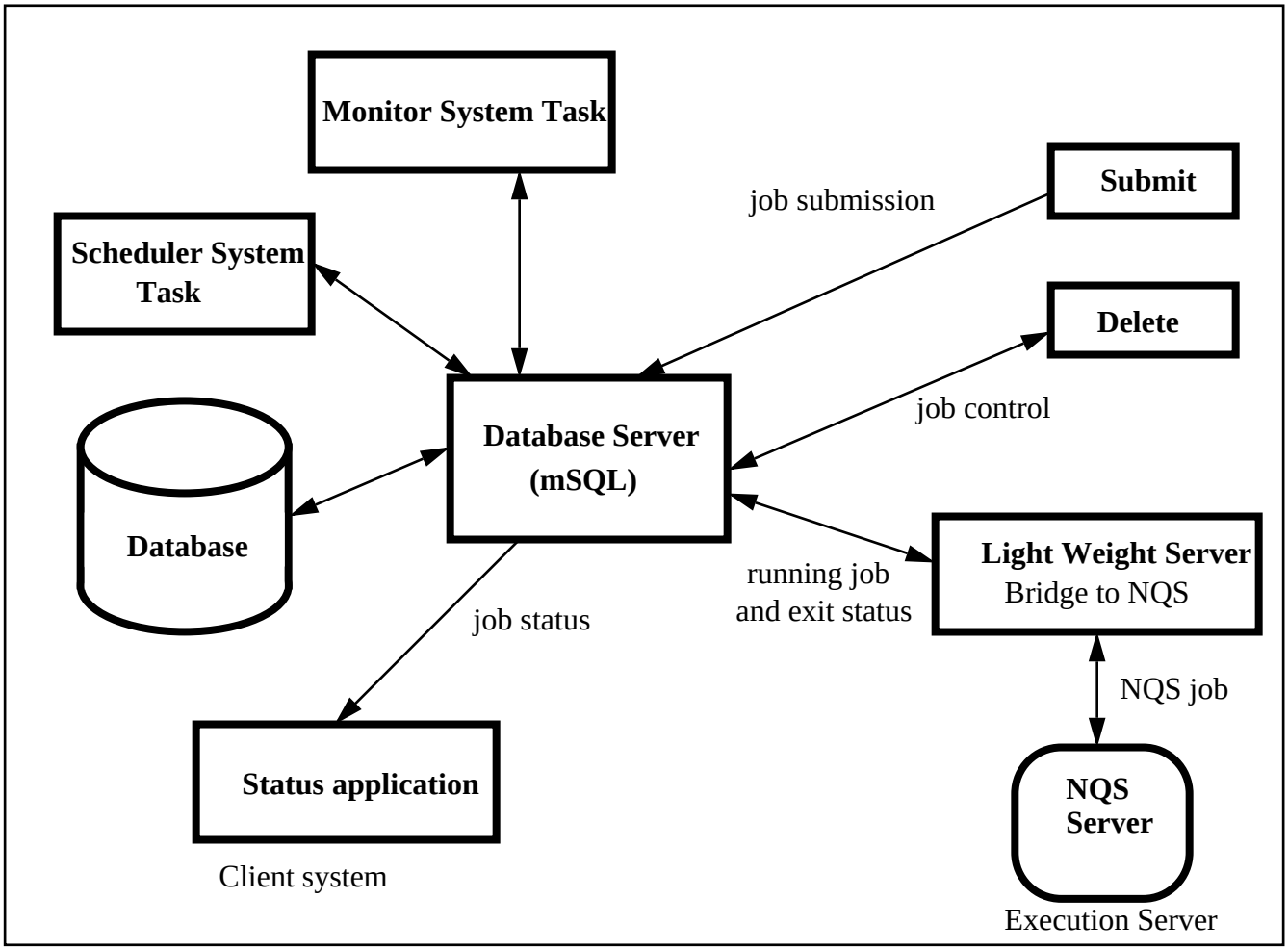

Diagram 3: Overview of mSQL components*

*Diagram taken from CraySoft NQE Administrators Guide