UNICOS Out of the Box
Setup Topics for Experienced UNIX Administrators

Lonnie A. Berkebile, Cray Research, Inc.

ABSTRACT: *UNIX system administrators who have workstation experience need some reassurance about how much of their experience is applicable in the UNICOS environment -- an overview of similarities and differences between UNICOS and "workstation" UNIX. This paper covers UNICOS file systems and I/O functionality, how they differ from other UNIX approaches, and how this affects system configuration decisions. It also discusses typical UNICOS distributed computing environments and how this affects network and communications configuration. Other important comparisons between UNICOS and "workstation" UNIX also will be noted.*

Introduction

Administering any system is a very challenging task. Moving from UNIX to UNICOS adds a level of difficulty for experienced UNIX administrators. In preparing this paper, several administrators were interviewed. During a beginning administration class, one administrator commented that he did not see a need for the user database (UDB), because the `/etc/passwd` file was sufficient. Another administrator was concerned that certain files were not available to a site's users, and another administrator identified menu tools as their issue.

There are very few changes that an administrator must make to move from UNIX to UNICOS; however, these differences make an already challenging task more difficult. There are changes that are differences in some basic user issues (for example, which shell is the default shell and which key press erases a character). Some reasons for purchasing a Cray Research system point to topics that need to be included in this paper. These topics include high-performance input/output, ability to more easily control user access to resources, ability to adjust scheduling priorities based on allowed usage, additional accounting information, advanced device management techniques as part of the standard code, and ability to manage batch work across the distributed environment. The topics discussed in this paper are authentication (for example, user database), file system and input/output issues (for example, logical device cache), resource management, and multilevel security (MLS).

Authentication

UNIX has provided a mechanism to authenticate users by matching information that is stored in a file against information that is entered by the user in response to a set of questions. The UNIX approach does not provide a way for administrators to control individual users to the same extent as the user database (UDB). A site may want to provide different levels of resource usage to each user. For example, one user might use unlabeled tapes while it might not be useful to allow all users to access unlabeled tapes. The priority (nice value) that one user's processes start at might need to be better than another user's. These examples can be set by the administrator for each individual user. System performance can be improved by adjusting the setting for users. The format of the user database (UDB) has changed considerably for UNICOS 9.0, but only a few data fields have been added.

UNIX and UNICOS both support controlling user access through password, password aging, valid groups, login shell, and directory. These fields can cause some confusion because UNICOS provides a set of menu-driven tools to support administration activities, and these tools have a set of default values. For example, will the default shell from `xadmin` provide the user with the same shell that they have been using. UNICOS also tracks the origin and time of the user's last interactive and batch logins. There is a series of permission bits (both UNICOS and site-controlled definitions). Fields exist for both batch and interactive controls of maximum processes, CPU, memory, and file size for each user. User-level limits for MLS security and fair-share scheduler activities exist. In UNICOS 9.0, support of limits for maximum shared segments, maximum shared memory, and per-process socket buffer limits can be set per user. The default MLS compartments can be set for the individual user. To control each user's use of file system space, the UDB controls the number of open files per-process and the per-process core file size.

Copyright © Cray Research Inc. All rights reserved.

Table 1. UNIX Versus UNICOS

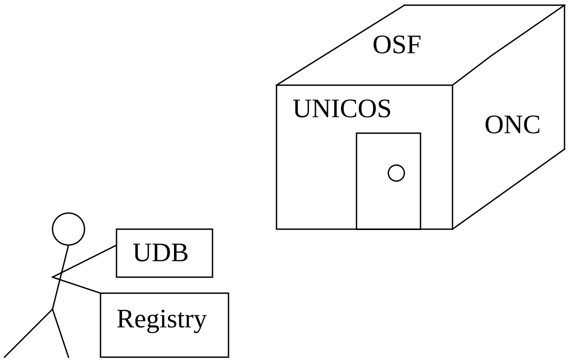
	UNIX /etc/passwd	UNICOS UDB
Password	X	X
Password aging	X	X
Login shell	X	X
Login directory	X	X
Fair-share scheduler usage and limits		X
Security levels and compartment limits		X
Security login level limits		X
Security login level and compartment minimums		X
User's process time, size, and file size limits		X
User permissions controlled by Op Sys		X
Permissions controlled by site		X

UNICOS supports both Open Network Computing (ONC) and Open Software Foundation (OSF) approaches to distributed computing. In the area of user authentication, this raises two issues for administrators who support Cray Research systems in a Distributed Computing Environment (DCE) cell. To use OSF's DCE service, users must be authenticated into the DCE cell. Authentication is accomplished by executing `login` and `dce_login`. This execution can be accomplished through either integrated login or as two separate steps for users that do not always need to access DCE services. Because there are two passwords, a user must ensure that they keep the passwords synchronized. During DCE configuration, administrators must determine which approach they want to use.

File System and Input/Output

To obtain the best possible efficiency from UNICOS input/output (I/O), it is necessary to understand the I/O process. I/O begins in the user's program with the use of user buffers for reformatting, rearranging, and moving to/from user variables and arrays. Library buffers are not always required; when used, they increase the need for Heap space. After the data leaves the user's space, it is frequently moved to system buffer cache. System cache is part of the memory space taken by UNICOS, and it must be set for best I/O performance. If the size is too small, the system will not boot. If the size is too large, there will be wasted system time.

Password Administration



System cache is a typical UNIX approach. UNICOS has added some performance enhancements to this approach. UNICOS adds a possible second level of buffering before the data is moved to the peripheral device (for example, disk). This second level of buffering that is available to file systems is ldcache.

In addition to the variety of paths available to the I/O process, the user's approach to I/O makes a difference; for example, is the user using sequential or nonsequential (random, direct-access, and so on). The user's I/O might be synchronous, asynchronous, or asynchronous queued. Each approach makes a difference in how well the I/O can use logical device cache or system buffer cache.

System buffer cache is controlled through a set of system parameters. The number of allocated 512-word blocks is established by setting either the number `NBUF` or the reciprocal value for the percentage of memory `NBUF_FCTR`. For example, `NBUF_FCTR`=20 sets the percentage of memory to 5%=1/20. Although the default is 5%, values less than 2% of memory for central memory sizes greater than 64 Mwords work very well. For memory sizes greater than 512 Mwords, using less than 1% of memory for system cache is highly desirable. Additional parameters for controlling system buffer cache are `NHBUF_FCTR` (ratio of hash table entries; this value works very well when set to 2), `NBLK_FCTR` (maximum percent of system cache that can be requested in a single request), and `MAXRAH` (maximum read-aheads). `MAXRAH` is easy to determine if all I/O is either sequential or nonsequential. For sequential, the default value of 8 can provide good performance. For nonsequential, a value of 8 could mean that seven I/O buffers are moved by the system and never utilized.

I/O Architecture

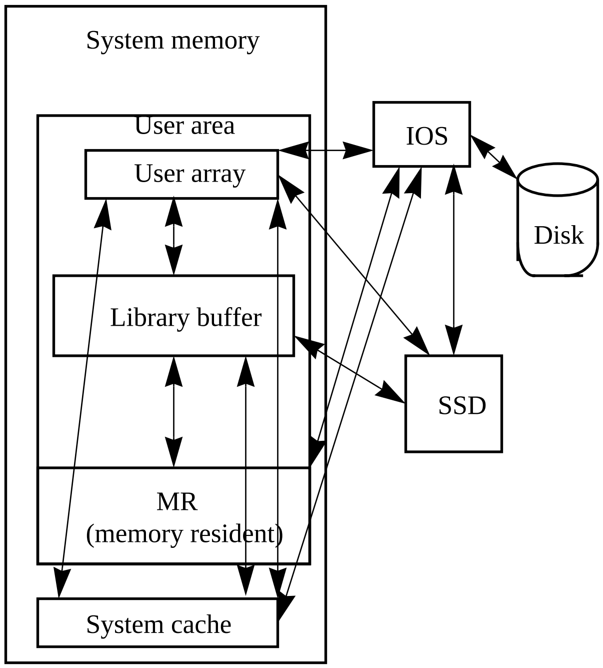
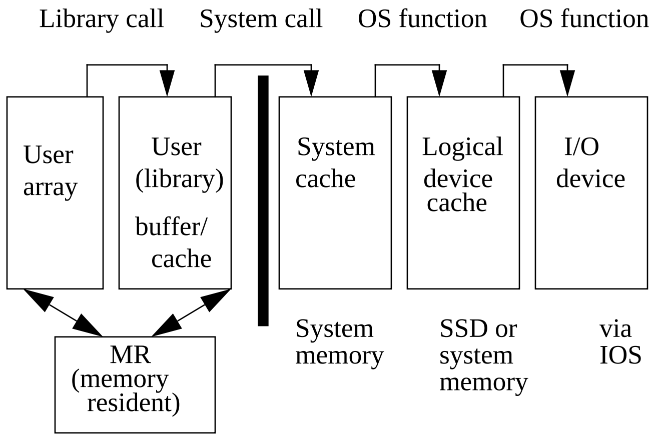


Table 2. System Buffer Cache

NBUF	Number of 512-word blocks
NBUF_FCTR	{([1/number) * 100} % of total central memory
NHBUF_FCTR	Ratio NBUF to hash entries
NBLK_FCTR	Percent of NBUF allowed per request
MAXRAH	Maximum read-ahead

Logical device cache (ldcache) for current architectures means that the cache space is either in the SSD or central memory. When central memory is the only choice, use of ldcache is still encouraged, but it should be monitored even more closely. Improper use of ldcache adds system time and slows down the I/O; however, proper use of ldcache is a tremendous improvement in I/O performance. Use of ldcache should always be considered. Administrators must understand what types of I/O are going on within the normal job mix. Because assignment of ldcache is dynamic, administrators can modify ldcache assignments to meet the changing needs. After the number of headers and the amount of space for ldcache are determined at boot time, the actual allocation can be changed in the `ldcache` command executed when a process is about to start that might make very efficient use of ldcache. Ldcache has a positive impact on I/O when the data can be heavily reused. Data is moved along the paths identified below with the possibility of skipping some steps..

Input/Output Path



To define ldcache, specify the logical device, the number of buffers and the size of the buffers. By setting the number of buffers to 0, use of ldcache for the specified file system is terminated. Thus, the size of ldcache and number of buffers can be changed to meet the needs of specific processes. In UNICOS 9.0, it is possible during the build portion of the install menu to allocate the desired portion of the systems ldcache for the build process and then return this portion of ldcache to the pool through the use of menus. The size of buffers is used to partition a file system into equal size pieces. When a user accesses part of the space in one of the buffers, that partition is put in the ldcache space. An administrator can either sync (flush) the ldcache on a timed interval by setting `LDSYNCTM` or can manage the movement of buffers by using the `-h` and `-x` options to the `ldcache` command when adding this file system to the ldcached space. `LDSYNCTM` can cause a peak of system time while it moves a large ldcache block back to the disk. The `-x` option allows controlling when dirty blocks (blocks that have been written into) are moved back to disk based on being older than the specified minimum. Flushing will begin only when the oldest block has reached a maximum age. Similarly, the `-h` option lets administrators control flushing, based on the number of dirty buffers. A minimum number of remaining dirty blocks from possible rewrites and a maximum number to start the flushing process are set using the `-h` option.

To obtain maximum benefit from ldcache, administrators should know the I/O patterns of the workload. To minimize the amount of space, it is best to know when specific file systems will benefit from ldcache and the number of blocks needed to keep the data in ldcache so that it remains in ldcache long enough for maximum reuse. There are applications with scratch files that can be held in ldcache and obtain very fast I/O for the execution time..

Table 3. `ldcache` command options

`-l`	Logical device being ldcached
`-n`	Number of allocations (headers)
`-s`	Size of allocation in 512-word blocks
`-x`	Minimum age, maximum age
`-h`	Minimum number, maximum number

Users have the real impact on I/O and should be educated in what types of I/O are most efficient for their programs. By using the `assign` statement, it is possible to improve I/O on compiled programs. Table 4 summarizes the issues and trade-offs from a user's point of view.

Table 4. User Input/Output Issues and Trade-offs

I/O style	Advantages	Disadvantages
Formatted	Portable, human readable	Very slow
Unformated	Faster than formatted	Less portable
Blocked	Can skip blocks, ok for short records	Slow to do control words
Unblocked	Faster - no control words	Cannot skip around
Unbuffered I/O	One system call per transfer, no double or triple handling	Must be unblocked, Slow for short records (too many system calls)
User buffer size	Bigger okay - heavy use files	Use a lot of memory; Make sure Heap sizes are set properly
System cache	Apparent I/O speed	No user control
Synchronous	Simple, normal	Waste real time waiting
Asynchronous	Overlap real time I/O with CPU time	Potential data dependency, higher CPU/system overhead
Asynchronous queued	Multiple requests to one file asynchronously	Complicated, hard to debug, higher CPU/system overhead
Sequential	Simple - fast for most	Hard to skip around cannot change middle of file easily
Nonsequential	Can change file anywhere; Easy to skip around	Slow for sequential needs; higher CPU overhead

Because UNICOS supports several ways of sharing files between systems, administrators must determine which approach best meets their needs. The three ways are through the use of NFS (network file system), DFS (distributed file system) and SFS (shared file system). The level of administrative support varies between approaches, and it is possible to use combinations (for example, using DFS on files that are mounted through SFS). Each approach has a configuration tool to support the administrator in the file system activities. For more detailed information, see the CUG schedule, or see the *Software Training Catalog for Customers* for specific course information.

Table 5. Sharing File Systems

NFS	ONC and ONC+
DFS	OSF/DCE
SFS	UNICOS proprietary

Resource Management

Resource management has many forms and most of them are supported as UNICOS (or asynchronous product) additions to UNIX. These additions are shown in Table 6:

As identified in the Authentication section, it is possible to provide restricted per-process and per-job resources on a specific user's interactive and batch workload. UNICOS adds two additional features for limiting individual users who are not part of standard UNIX. These features are fair-share scheduler and file system quotas. Each feature can be used to limit individual users or groups of users. The fair-share scheduler was developed to adjust scheduling priorities of all running processes on a regular interval, based on usage by a user or group of users. File system space is managed through the use of file system quotas. Through the use of file system quotas, a site has the tools to set quotas for users, groups, or accounts limiting the amount of space allowed for files and inodes. Another form of resource management is the Data Migration Facility (DMF), which allows for an oversubscription of disk space. The real disk space becomes a virtual disk space with files that are not actively being used moved off to secondary storage. An additional form of resource management is the addition of Network Queuing Environment (NQE) to Network Queuing System (NQS). With NQE users submit to a network load balancing queue that determines which machine meets the request's needs and can best service the request. This allows a level of load balancing among a diverse group of machines.

Table 6. Resource Management

Resource Manager	Manages
Unified Resource Manager (URM)	Jobs to be initiated based on resources available
User database	User's usage of resources
Fair-share scheduler	Scheduling priorities based on past resource usage
File system quotas	File system consumption
Data Migration Facility	Disk device usage
Network Queuing Environment	CPU usage in a distributed environment

URM acts as the UNICOS system gate-keeper by controlling the initiation of new sessions (also called jobs). The URM job scheduling daemon tries to ensure that system resources are neither under used nor overcommitted. Administrators define allowed level of use for system resources. By controlling the influx of jobs to the system, URM attempts to maintain system resource usage as close to a set of administrator predefined targets as possible.

After URM recommends a session for initiation, other portions of the system (such as the memory scheduler and the fair-share scheduler) take control of priorities, scheduling, resource allocation, and actual execution of the job. If the URM does not recommend initiation, the job is not added to the session table; therefore, does not execute. The root ID `cron` jobs and login sessions are exempt from URM restrictions.

Although URM does not control actual resource allocation and job execution, it controls the scheduling of secondary data segment (SDS) space with SDS preemption.

UNICOS process scheduling without fair-share scheduler is similar to UNIX process scheduling. The process nice value gets worse over a period of usage and is forgiven an amount on a regular basis. However, UNICOS bases the amount of forgiveness (improvement in value) on the nice value and forgives more of the usage for the better nice values. Thus, the worse nice values do not improve as much over a period of usage.

With the fair-share scheduler running, the amount of adjustment is based on the user's prior usage and the current nice value. This scheduler is devoted to assigning the machine's CPU resource to the most deserving processes. Scheduling fairly among users means that users who have the same proportion of shares should be able to consume the same amount of the CPU resource. Administrators can adjust the percentage of usage that is actually charged, the types of usage that are accumulated, and the percent of the usage to be tracked during a time interval. The types of usage that may be tracked are CPU time, system calls, block I/O, stream I/O and "clicks" of memory.

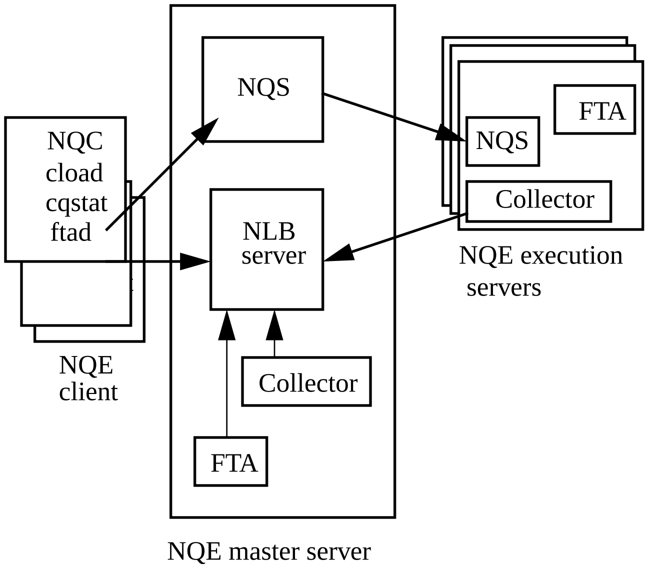
The Data Migration Facility (DMF) is designed to help manage online mass storage space effectively. The objective of data migration is to maintain a specified amount of free space available on a UNICOS file system by migration files offline when specified thresholds are exceeded. To efficiently use DMF, a site should be familiar with the rate at which files are created or grow and the length of time it takes to migrate the amounts of data that must be moved. It is possible to set the limits needed so that enough space is always available. The process of data migration must be started at a level so that the users who are creating new files will not outrun the speed of the media migrating the files; thus, always having free space. In a distributed environment, DMF allows for the migration of files from a set of systems to a common secondary storage device. This allows for easier access to migrated files.

Network Queuing Environment (NQE) is a framework for distributing work across a network of heterogeneous systems. NQE consists of Cray Network Queuing System (NQS), Network Queuing Client (NQC), Network Load Balancer (NLB), Network Job Status (NJS), and File Transfer Agent (FTA). NQS is compatible with other versions of NQS that are common to UNIX systems. The UNICOS version of NQS has some enhanced features.

Cray NQC provides a client interface to NQS that supports the submission and control of work without the need to run full NQS. This interface has minimal overhead and administrative cost. NQC is intended to run on all nodes where full batch execution and queuing capabilities of NQS are not required.

Cray NLB provides status and control of work scheduling within the batch complex. Sites can use the NLB to provide policy-based scheduling of batch work in the complex. Cray NJS provides refreshed status of work within the batch complex. Collectors periodically collect data about the current workload on the machine where they run. The collected data is sent to the NLB servers, which store the data and make it accessible to

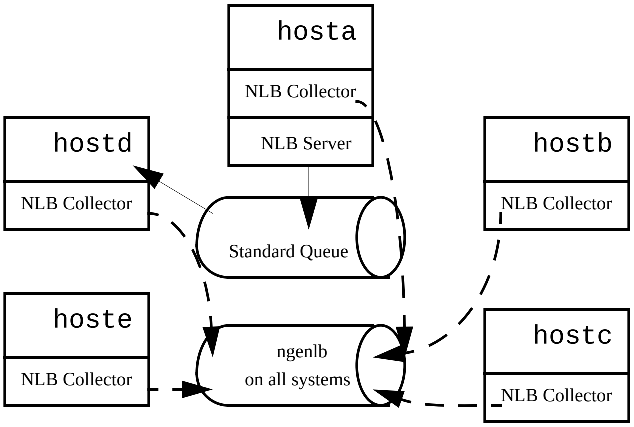
NQE Clients and Servers



client programs for monitoring machine load data and network job status.

FTA services requests for file transfers that have been issued by jobs and NQE.

NQS Configuration With Load Balancing



Multilevel Security (MLS)

The security features of UNICOS are designed to satisfy government (DoD Trusted Computer System Evaluation Criteria) and private industry computer security requirements. The secure UNICOS system supports both discretionary (need to know) and mandatory (authorization or clearance to access) access controls.

MLS provides the following mechanisms and assurances to protect both system integrity and sensitive information:

- Access control lists (ACLs)
- Labels - Levels and compartments
- Privilege assignment lists (PALs)
- Network access lists (NALs)
- Workstation access lists (WALs)
- IP Security Option (IPSO)

Access control lists are provided to give users the ability to control who has access to their files. Because UNIX has only three levels of control (owner, group, and other) it is frequently impossible to limit access to the set of users that a specific user needs. By using MLS access control lists, this can be accomplished. An ACL file is created that identifies users and groups and the level-of-access that is available to individuals satisfying the user and group identification. The MLS ACL is in addition to the UNIX security features. MLS ACLs are different than DCE ACLs. Although UNICOS allows setting DCE ACLs on objects in the DCE cells, it does not allow setting DCE ACLs on a UNICOS system’s objects.

When UNICOS is running in a secure state, all information has a label that represents the security level of an object and describes the sensitivity (that is, classification) of the data in the object. The sensitivity of information is designated by a hierarchical classification of security levels and a nonhierarchical set of compartments. Each object file in the system is labeled in its inode with a single security level and up to 63 compartments. The UDB provides the security level and compartments for the user at login. In addition, the UDB is divided into a public portion and a private portion. The private portion is protected and contains the security-related information.

UNICOS MLS supports a privilege mechanism to enforce the principle of least privilege. Various administrative roles are defined with the system’s required administrative tasks assigned to those roles. Privileges are assigned to a specific task and not to a user. The assigned privilege attributes can vary, depending on the active category of the user. The privileges are in a privilege assignment list (PAL), which is assigned to an executable file and resides in a file system data block. The PAL is composed of PAL category records that identify the administrator’s active category, the privilege set to be granted the particular process, and privilege text that can be interpreted and cause some action.

Network security is every administrator’s responsibility. UNICOS MLS provides mechanisms not found in other UNIX systems and supports the IP Security Option (IPSO), which is not supported by most UNIX systems. The UNICOS specific features are network access list (NAL) and workstation access list (WAL).

To provide trusted communications, the UNICOS system requires that hosts and networks connected to the evaluated system be defined in a NAL and thereby authorized by a security administrator. The NAL describes the security privileges associated with each remote host. This information is used to exercise the mandatory access controls on the network interfaces. Each entry in the NAL consists of the following information about the remote host:

- Minimum and maximum security label
- Send and receive message authorization modes
- Security class
- Security option (IPSO) support information

Each incoming and outgoing Internet Protocol (IP) datagram has a security label associated with it. This label can be implicit (as defined in the NAL) or explicit (as when you use IP security options). The datagram label is used to enforce the restrictions imposed by the NAL and the network-interface label range and to ensure that data is delivered only to applications with proper clearance.

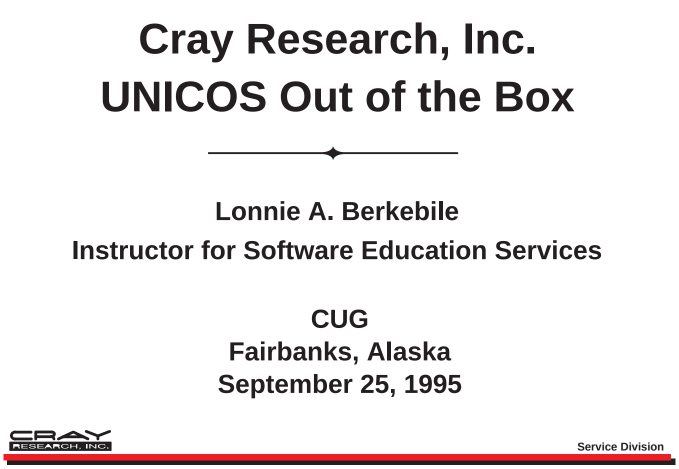
At the application level, the UNICOS trusted network services provide protection by requiring positive identification and authentication for all network transactions (except those that provide public information). In addition, the workstation access list (WAL) optionally controls application access, based on user ID and/or group ID and the host from which access is desired. A WAL entry consists of a list of user/group pairs (wildcards can be used) and services for which those users and groups are allowed access. The login user ID and primary group ID are used to perform WAL permission checks.

Conclusion

The differences that administrators of a new UNICOS system should recognize are as follows:

- Primarily standard UNIX - menu tool replaces former tools
- Few required differences - User database (UDB) and system buffer cache
- Several enhancements - ldcache, sharing file systems, resource management tools, and security.

This paper began with the comment that there was very little that an administrator had to do moving from UNIX to UNICOS. This statement was correct in that only the user database (UDB) and the system buffer cache are required. The other features identified are things that Cray Research hopes you find make it possible to easily do your job and help you accomplish more work through your Cray Research system than possible with other computer systems.

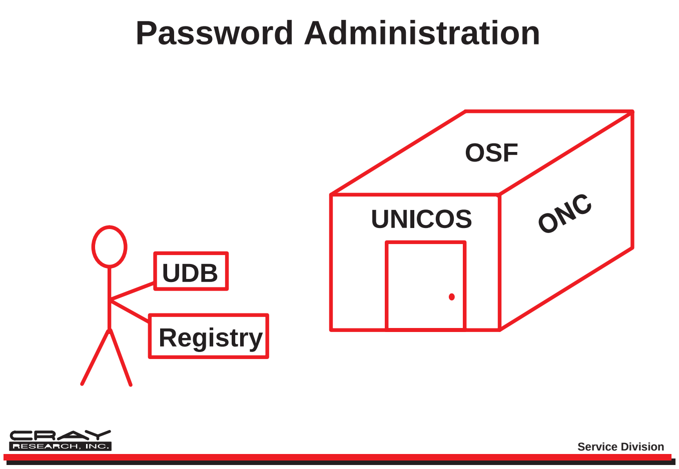


Cray Research, Inc.
UNICOS Out of the Box

Lonnie A. Berkebile
Instructor for Software Education Services

CUG
Fairbanks, Alaska
September 25, 1995

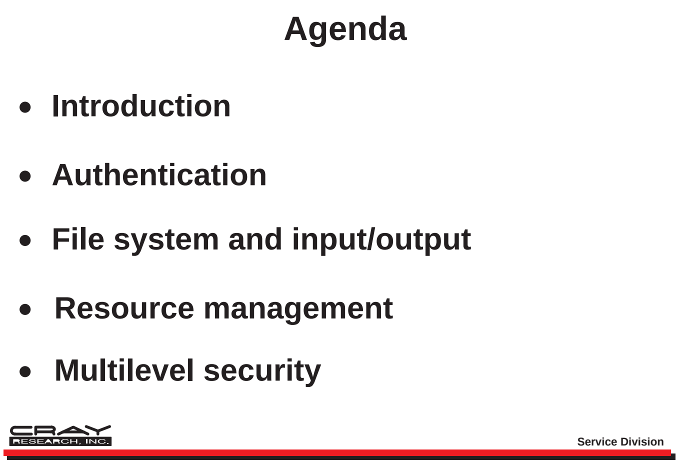
Service Division



Password Administration

OSF
UNICOS
ONC
UDB
Registry

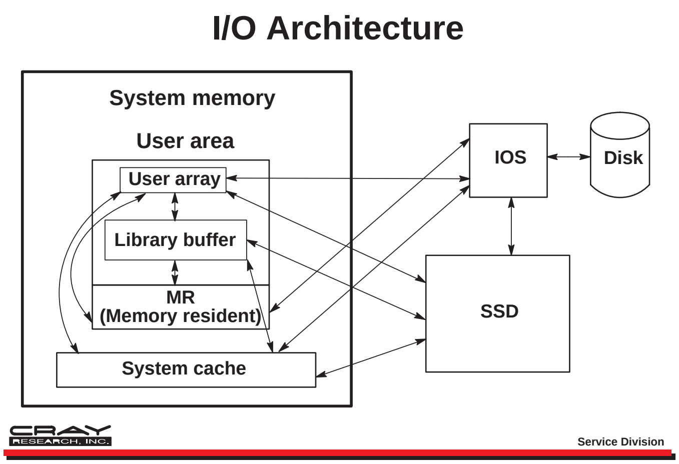
Service Division



Agenda

- Introduction
- Authentication
- File system and input/output
- Resource management
- Multilevel security

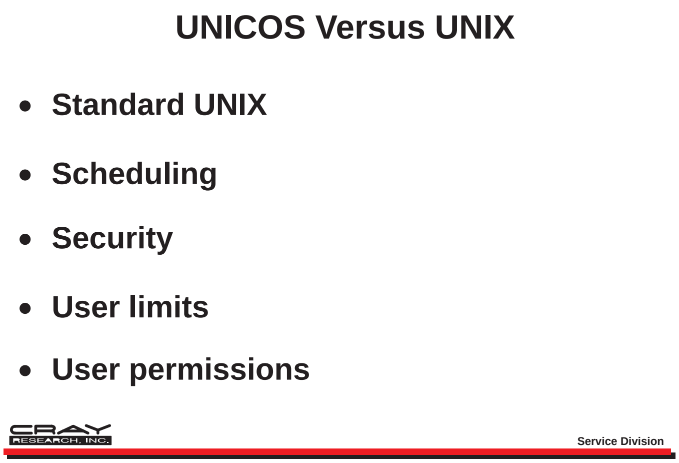
Service Division



I/O Architecture

System memory
User area
User array
Library buffer
MR
(Memory resident)
System cache
IOS
Disk
SSD

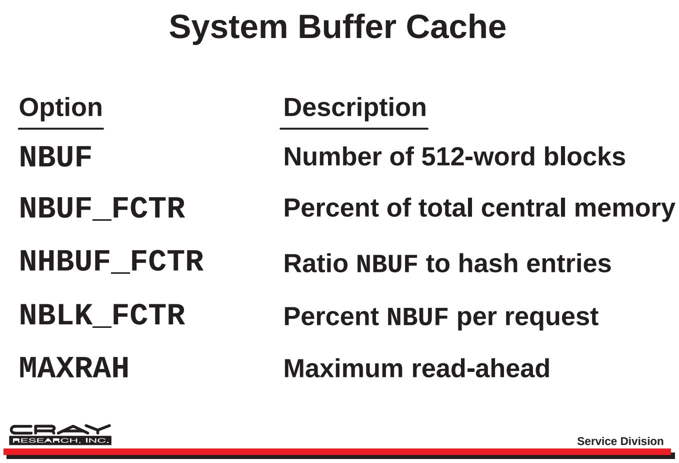
Service Division



UNICOS Versus UNIX

- Standard UNIX
- Scheduling
- Security
- User limits
- User permissions

Service Division

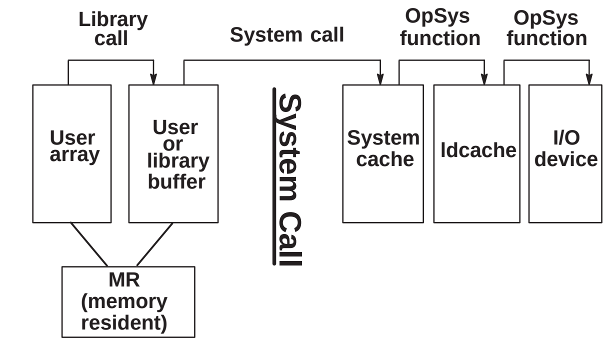


System Buffer Cache

Option	Description
`NBUF`	Number of 512-word blocks
`NBUF_FCTR`	Percent of total central memory
`NHBUF_FCTR`	Ratio `NBUF` to hash entries
`NBLK_FCTR`	Percent `NBUF` per request
`MAXRAH`	Maximum read-ahead

Service Division

Input/Output Path



Service Division

`ldcache` Command Options

Option	Description
`-l`	Logical device being ldcached
`-n`	Number of allocations (headers)
`-s`	Size of allocation in 512-word blocks
`-x`	Minimum age, maximum age
`-h`	Minimum number, maximum number

Service Division

User Input/Output Issues and Tradeoffs

- Formatted *versus* unformatted
- Blocked *versus* unblocked
- Unbuffered *versus* user buffer *versus* system cache
- Synchronous *versus* asynchronous *versus* asynchronous queued
- Sequential *versus* nonsequential

Service Division

Shared File Systems

Type	Origin
Network file system (NFS)	ONC and ONC+
Distributed file system (DFS)	OSF/DCE
Shared file system (SFS)	UNICOS proprietary

Service Division

Resource Management

- Unified resource manager

  Job initiation
- User database

  User resource usage and limits
- Fair-share scheduler

  CPU scheduling priorities

Service Division

Resource Management (continued)

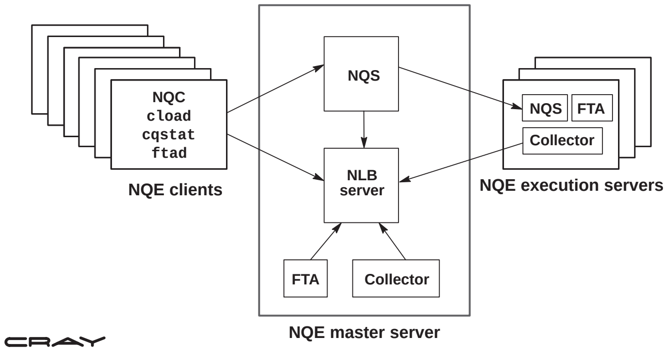
- Data Migration Facility

  Disk device usage
- Network Queuing Environment

  CPU used distributed environment

Service Division

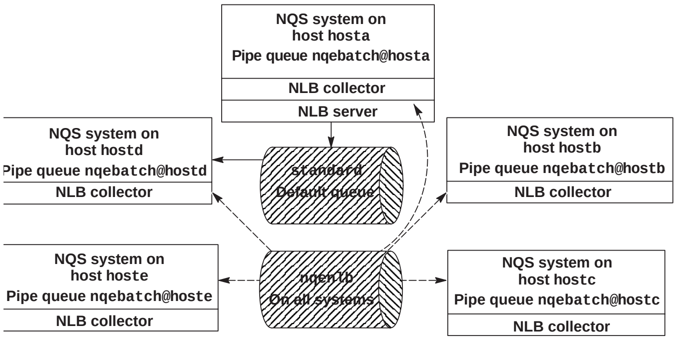
NQE Clients and Servers



Multilevel Security (MLS)

- Access control lists (ACLs)
- Labels – Levels and compartments
- Privilege assignment lists (PALs)
- Network access lists (NALs)
- Workstation access lists (WALs)
- IP Security Option (IPSO)

NQS Configuration With Load Balancing



Conclusions

- Primarily standard UNIX
  - Use UNICOS menu tool versus former ways
- Few required differences
  - User database
  - System buffer cache
- Several enhancements
  - `ldcache` for input/output
  - Sharing file systems
  - Resource management tools
  - Security