

UNICOS/mk Status and Update

Jim Harrell

Deputy Director, System Software Group
Cray Research, a division of Silicon Graphics, Inc.
655F Lone Oak Drive
Eagan, Minnesota 55121

ABSTRACT

The first part of this paper provides an update of the current plans for UNICOS/mk on the CRAY T3E system. Included are a discussion of progress towards completion of features related to UNICOS compatibility and the status of production system features in the field (e.g., political scheduling and checkpoint/restart). Development efforts are also discussed.

The second part of this paper discusses the status of performance and scalability from an operating system perspective. UNICOS/mk was designed to be a high performance distributed operating system. Measurements of the CRAY T3E operating system are presented and some comparisons are made with other relevant systems.

1 Introduction

The following subsections update the status and plans of the UNICOS/mk operating system. The following four topics are discussed:

- General direction
- Release plans
- Feature schedule
- Field status

2 General Direction

There are four general areas that make up the work currently being done for the UNICOS/mk project. All of these topics will be discussed in more detail in later subsections. The four areas are as follows:

- Finish UNICOS feature equivalence work
- Support
- Resilience
- Performance

The main goal of the project was to make the UNICOS/mk features and functionality equivalent to the content of the UNICOS system; that is, make UNICOS/mk look and work like UNICOS for a MPP platform. The majority of this work is almost done. This effort is shown in figure 1; the first three columns represent all the features currently available in the UNICOS/mk system, while the last column represents the work left to be done.

Figure 1



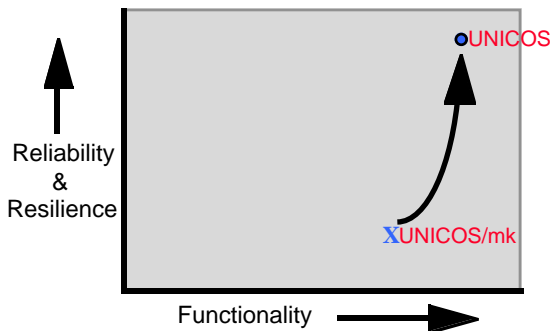
The most amount of time and effort lately has been in the area of support. Work is being done to fix system bugs and feature problems, and preparing systems to go to the field.

More time has been spent on resilience lately. Work has now been started to make UNICOS/mk more software and hardware resilient. The goal is to allow a machine

running user applications to continue when software/hardware errors occur that previously would have crashed an earlier version of the system.

Finally, performance is an area that needs improvement. The two areas we will be focusing on are scalability and general system performance. Figure 2 shows the general trend in effort and use of resources for the UNICOS/mk project. As the feature and functionality enhancement work levels off, work on reliability and resiliency increase.

Figure 2

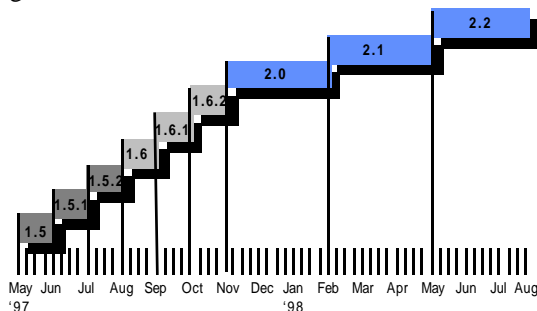


3 Release Plans

The latest UNICOS/mk system is the 1.5 version and was released on April 28, 1997. The 1.5 release is one of the UNICOS/mk releases that is being released on a 4-week schedule.

The monthly release cycle was planned to slow down at the 1.5 release, but because of feature slips and the need to get critical fixes to the field, this release schedule will continue until 4CY97 (as shown in figure 3). Note that support for a given release will also continue to be limited to the current release until the UNICOS/mk 2.0 release timeframe.

Figure 3



Starting with the UNICOS/mk 2.0 release, the plan is to increase the release cycle to a three month period. Releases after 2.0 could possibly be longer than three

months; the exact cycle time will be defined at a later date.

The weekly updates will continue to be produced as needed. This mechanism is a very good way to get critical fixes to the field in a timely manner. Customer feedback validates that it is an effective tool, and it will continue to be used indefinitely.

3.1 Recently Released Features

The following features were recently released for the UNICOS/mk system:

- Political scheduling
- Checkpoint/restart
- Pcache
- Remote mount

These features are discussed in more detail in the following paragraphs.

Political scheduling has been available since the UNICOS/mk 1.4.1 release and is being used at several sites. Initially, it had some problems, but they have been resolved since then. It is now functioning very well, and the documentation is being enhanced to describe the functionality in a more useful way.

This feature allows you to run gang scheduling on the application PEs or on both application and command PEs, and allows for load balancing. We believe this is a full-featured scheduling mechanism that includes a distributed version of the UNICOS fair-share scheduler functionality, and we recommend its use.

Be aware that to run multi-PE applications with gang scheduling where swapping is required, you will need swap space. It is highly recommended to use fast devices in this case (e.g., FCNs).

Checkpoint/restart development was to be implemented in three phases. The first phase of the work was released in the UNICOS/mk 1.5 release. This work allows an application or program to be checkpointed and restarted, but does not support all of the UNICOS checkpoint functionality. The primary missing piece of functionality in 1.5 is the lack of support for checkpointing open files. Code fragments in the documentation show how applications using open files can still be checkpointed with this 1.5 version by closing/opening them during the checkpoint/restart operation.

The second phase will be released in the UNICOS/mk 1.6 system. This phase is a beta version of the UNICOS-compatible checkpoint/restart. We are looking for sites who would be interested in verifying this version of the feature. Contact Jim Harrell if interested.

The third phase of the implementation will be available in the UNICOS/mk 2.0 release. It will fully compatible with the UNICOS version of checkpoint/restart and fully operational.

The pcache functionality has been available for some time now, but it has been slow to mature. We believe that the version available in the UNICOS/mk 1.5 release is the one that works as advertised.

Pcache is the feature that is the logical successor to ld-cache. It caches physical disk devices, and is associated with a disk server and uses memory on the node with the disk server to cache reads/writes. We have found that even with relatively small caches, performance improvements are substantial. We believe that by using multiple disk servers and pcaches, you can improve I/O performance substantially. Larger memory PEs allow for larger cache size, which gives a boost in the amount of data cached on the node.

Remote mount is available in the UNICOS/mk 1.5 release. It allows you to spread system's file system tree across multiple file servers, each residing on their own PE. Each file server can manage one or more file systems.

Since UNIX is a file-based operating system, it is no surprise to anyone that scalability of the file system would have a significant impact. Thus remote mount is a very important scalability feature. It provides a way to spread CPU utilization and memory buffer caching across multiple PEs. Good performance improvements have been noted in cases where I/O can be spread across multiple file servers that are remotely mounted.

3.2 *Brief Descriptions of Upcoming Feature Release*

The following subsections briefly describe the planned feature content for upcoming UNICOS/mk releases.

3.2.1 *UNICOS/mk 1.5 Release*

The UNICOS/mk 1.5 feature content is as follows:

- Shared-text support: Allows program text to be shared on PEs where multiple copies of the program are being run. This saves memory on serial or command PEs.
- Resource limits Phase II: Limits support for UIDs and jobs.
- (Asynchronous) Serverized DMF: Support for DMF.
- Remote mount: Remote mount is the ability to have multiple file servers each supporting one or more file systems. The file servers link by "mounting" the file systems.

- Shared File System (SFS) support: Support for the UNICOS Shared File System.
- SysV IPC Message Queues & Semaphores: UNIX System V IPC Message Queue and Semaphores.
- Political scheduling Phase II: Support for site exits allowing customers to implement specific scheduling strategies. Flexible configuration allowing the scheduling demon to configure itself based on the GRM configuration.
- Analysis of tolerance to failing compute PEs: The first phase of operating system resiliency to software and hardware failures. The system will continue to run if a compute PE or group or compute PEs panic or halt.
- Checkpoint/restart (Phase 1): The initial support for checkpoint/restart. This version of checkpoint requires some changes to applications in order to support regular open files.
- GigaRing network performance enhancement: The low-level device drivers have been modified to improve performance.
- pcache: Complete support for physical disk device caching. This is the replacement for the UNICOS ldcache feature.

3.2.2 *UNICOS/mk 1.5.1 Release*

The UNICOS/mk 1.5.1 feature content is as follows:

- T3E dump using 3rd party DMA: Improvements to dump performance by using I/O directly to MPN devices instead of to the SWS.
- T3E boot performance enhancements: A series of enhancements to boot performance.
- Resource limits enhancements: Job and process memory limit support.

3.2.3 *UNICOS/mk 1.5.2 Release*

The UNICOS/mk 1.5.2 feature content is as follows:

- Disk quotas: UNICOS disk quota support on a per file server basis.
- Console resiliency: Changes to the SWS to UNICOS/mk console driver protocol to allow the console to be reconnected in the case of an MPN or SWS reboot.
- Political scheduling and limits documentation enhancements: Enhanced documentation improving areas like suggested configurations.
- F-packet support in xdd driver: HiPPI disk support.

- Gen 5 Hippi disk support: Gen 5 HiPPI disk device support.

3.2.4 UNICOS/mk 1.6 Release

The UNICOS/mk 1.6 feature content is as follows:

- Build UNICOS/mk kernel with optimization: Support for UNICOS/mk archives built using compiler optimization. The optimization improves system call performance. The performance improvements range from 3 to 35%, average improvement is about 18%.
- Enable MLS MAC: Support for the multilevel security mandatory access control.
- DCE DFS client: Support for DCE Distributed File System client.
- DCE Core Services: Distributed Computing Environment Core Services support.
- Checkpoint/restart (Phase II): Beta version of the UNICOS-compatible checkpoint/restart.

3.2.5 UNICOS/mk 1.6.1 Release

The UNICOS/mk 1.6.1 feature content is as follows:

- Error reporting for T3E chip options: Report generation and formatting for CRAY T3E chip options.

3.2.6 UNICOS/mk 2.0 release

The UNICOS/mk 2.0 feature content is as follows:

- DMF 2.5 under UNICOS/mk: Upgrade of DMF to the 2.5 level.
- Port Cray ReelLibrarian (CRL) to UNICOS/mk: Support for UNICOS Cray ReelLibrarian (CRL).
- Complete checkpoint/restart: Complete support for UNICOS-compatible checkpoint/restart.
- TCP/IP over GigaRing (Host-to-Host): Support for TCP/IP between GigaRing-connected hosts.

3.2.7 UNICOS/mk 2.1 Release

The UNICOS/mk 2.1 feature content is as follows

- Guest (UUU): Support for multiple operating system environments on the same machine. This is similar to UNICOS Under UNICOS, except that the systems will run in different partitions of PEs, not on the same CPUs and memory.
- IBM 3590 Escon device support: Support for IBM 3590 tape devices.
- AMPEX DST 310 device support: Support for AMPEX DST 310 tape devices.

4 System Status

It would be easy to focus on only the good aspects of the UNICOS/mk status, but the rest of this paper will focus on the two areas that are major issues:

- Stability
- Resiliency

In addition, several features that affect performance are discussed.

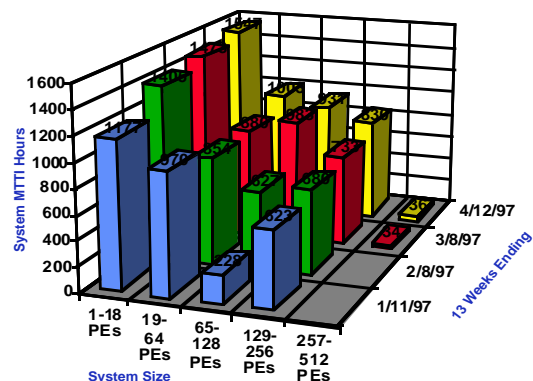
4.1 CRAY T3E MTTI History

The graph in figure 4 is taken from data that we have on system reliability in the field. The numbers themselves may be off by some percentage in either direction, but it is the best data available at this time.

The numbers show for a given size machine, the 13 week rolling system MTTI average. Obviously, the smaller systems have good MTTI, but it goes down as the size of the system increases. Thus, the largest systems present the biggest challenge in this area. History shows that improvement on the large machines has a good effect on the smaller systems as well.

It should be no surprise that the larger system MTTIs are lower. If this graph could be extended back in time, you would see a ramp up of MTTI as each succeeding larger machine was introduced. Work on improving the large system MTTIs will continue.

Figure 4



4.2 Resiliency

The UNICOS/mk 1.5 release contained a feature called analysis of tolerance of failing compute PEs. This feature is the first part of the software/hardware resiliency effort. What it does is isolate a PE that has failed because of a hardware/software error and allows the system to continue running.

We have been successful in isolating many types of software errors and some of the hardware errors. So far, proving that we can isolate any particular error will take more time. The feature tries to clean up an application/program running on the PEs that have failed. There

may be some cases where this needs further work. We plan on extending this feature, as it is the centerpiece of the system resiliency work.

4.3 Performance

The following areas of performance are of importance:

Application performance

Remote mount and file system assistant (FSA)

Pcache

Single disk performance

Application performance was discussed by Kent Koeninger at the San Jose CUG; see his paper for more information on this topic.

Remote mount and FSA were discussed by Mark DuChene and Brian Gaffey at the San Jose CUG; see their paper on their tutorial for more information on this topic. This topic is also discussed briefly in the following paragraphs.

The pcache performance issues are described in more detail in the following paragraphs. For the single disk performance, the differences in Cray products is discussed using fiber disks as an example in the following paragraphs.

Remote Mount is an important feature for scaling file system performance. Figure 5 shows that when it is possible to direct I/O to a number of file servers, scaling is excellent. With the six file servers that were tested, no roll off in performance was noted.

Figure 5

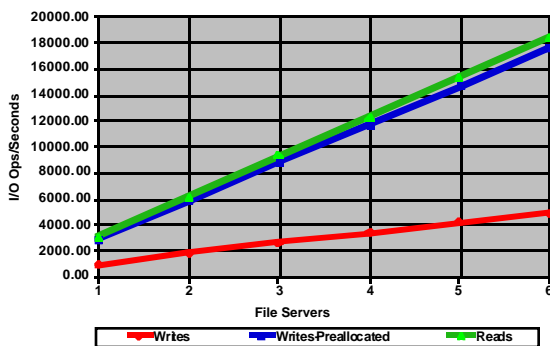


Figure 6 illustrates the difference between using raw I/O and using FSA. What is shown is that the preallocated FSA writes and FSA reads scale quite well using increasingly larger numbers of files and PEs, while raw I/O does not scale as well. So, using FSA is an advantage for multi-PE applications because I/O is processed locally. Note that un-preallocated writes currently perform terribly as shown on the bottom line of figure 5. This situation will be improved in a future release.

Figure 6

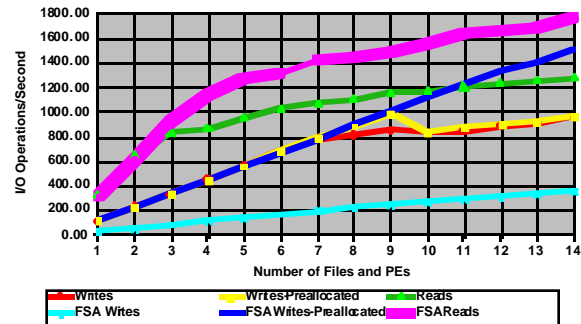
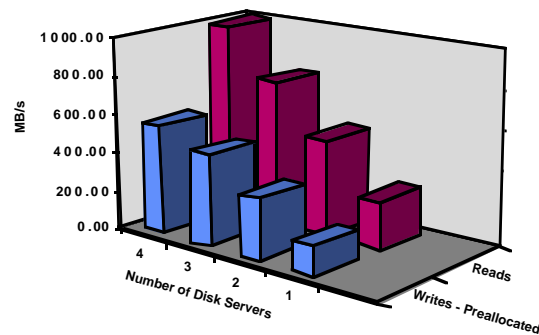


Figure 7 shows pcache performance using four different disk servers and therefore, four different disk caches. The cache size is 20 MB per PE. This example is something of a best case, given that the file fits into the cache. The two items to note is that performance is good for single cache, but scales well up to the four caches tested.

Figure 7

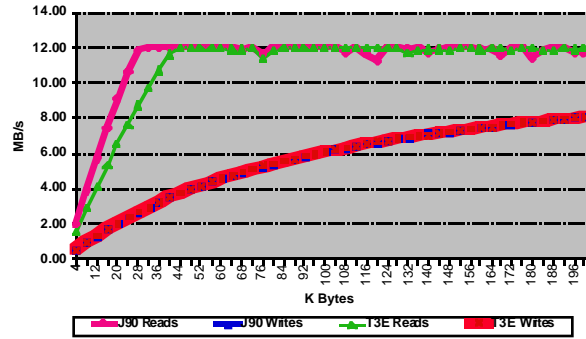


Normally, Cray does not compare performance issues between two architectures, especially when they are as different as the CRAY J90se and the CRAY T3E are. However, it is sometimes useful to compare system activities to determine where improvements can be made. What is shown in figure 8 is that both system are capable of getting maximum performance out of DD308s for both reads and writes. There is a difference in the rate at which the maximum read performance is achieved. This difference comes well before the 48 Kilo-byte point; 48 KB is the buffer size used by the libraries. In order to see this difference, a program would have to do its own I/O buffering.

The difference between these two machines is that for small buffer sizes, the CRAY T3E does not get read performance that the CRAY J90se does. We know that part of this will be regained with the use of the opti-

mized compiler. Also, there are some planned performance changes that will regain part of the difference. Some it may be due to memory speed and interrupt structure. It is important to remember that the CRAY T3E is a distributed system. However, this comparison shows that for general use, the differences are not obvious.

Figure 8



5 Conclusion

Much work has been done to both port and improve the UNICOS/mk system on the CRAY T3E platform. Much of the feature/functionality work is done or near completion. This will allow us to refocus our efforts and resources on enhancing the resiliency, scalability, and performance of the CRAY T3E system such that it will be a very useful tool for our customers.