CRI Question and Answer Panel

Julie Larson, Cray Research, Inc., Eagan, Minnesota

Panel Members

Tom Boyle	Customer Service - Technical Support
Steve Johnson	Engineering
Dave Judd	Software Division - Compilers
Janet Lebens	Software Division - Publications/UNICOS Coordination
Paul Rutherford	Software Division - Storage Systems
Dave Wallace	Software Division - Operating Systems

Introduction

Respondents to the CUG Site Survey produced by the Operations SIC are given the opportunity to comment on any hardware, software, or operations issues, problems, or concerns at their site. Several of these questions/comments are addressed formally during the CRI Q&A Panel discussion at the CUG meeting. In addition, the appropriate individuals at Cray Research have provided written responses in this paper.

If your comment or question has not been included, it is because CRI was unable to determine the exact nature of the concern. Site identification is kept confidential, so we were unable to get clarification from you. Please contact your local service representative who can forward your concern to Technical Support for a response.

1. Our "software problems" are typically not failures, but design points that are inconsistent or incompatible with most UNIX implementations. Usually these are subtle but very critical to our business. We are good at defining the problems! CRI usually dismisses them as "missing features". This is not good.

Cray Response: Cray makes a significant effort to adhere to a large set of industry standards (e.g., POSIX and X/OPEN), as directed by the majority of our customer base. It would be helpful for CUG to supply us with some information about which "standards" should constitute the "UNIX Standard." Some companies like SGI and IBM tend to consider their own versions of UNIX (e.g., AIX) as de-facto standards, while we have tended to follow the "official" standards. Does CUG see a need to change this policy on the horizon? If so, which are the important and appropriate standards?

When specific violations of standards (or other problems) are identified, we attempt to provide fixes as quickly as possible, consistent with the goals of (a) very high quality, and (b) customer desire and ability to upgrade their software. Also, please see the response to item number 8 below for more detail about timeliness of software fixes and the competing pressures for quick delivery and small delivery packages.

2. CRAY T3D C Compiler - poor performance.

Cray Response: The compiler development group maintains a number of different performance test suites whose codes have been chosen because of their importance to customer applications. With these codes, the compiler group can tune the performance of its development compilers so that these suites continue to execute as well or better than before. For the Cray PVP platforms, these performance test suites cover many of the important areas of compiler optimizations necessary for the best performance on these architectures, however not all of these test suites are well suited for measuring compiler performance on a Cray T3D platform.

There are performance test suites being used to tune compiler performance for the Cray T3D platform, but the number of codes that illustrate important customer applications for the Cray T3D are far fewer than what currently exists for the Cray PVP platforms. Although we continually work to build up our test base for use in measuring compiler performance, there is certainly room for improvement. If there are specific codes that exhibit poor performance, the compiler group would like to be able to obtain these codes. Having these codes available for inclusion in the performance test suites allows the compiler to be able to tune its optimizations against these codes and guarantees that these codes will not suffer performance regressions over time. SPRs that are opened to report performance degradations are also important methods for communicating what needs to be focused on for future compiler releases.

3. It is very difficult to determine if periods of poor performance are due to user errors or system weaknesses. If the latter, it is difficult to determine if those weaknesses can be corrected by system configuration parameters, nschedv() settings, or are inherent in the design of UNICOS, the schedulers or compilers/libraries.

Cray Response: When customers refer to "periods of poor response", they are normally referring to poor interactive response time. Less frequently to some form of system thrashing where there is activity, but no apparent progress.

There are actually two questions here. The first asks if one can diagnose the causes of performance problems. The answer is yes. There is a wealth of statistics, and tools to display those statistics in UNICOS. The commands sam(8), pddstat(8), sar(1), acctcom(1), and others can narrow down the source of undesirable system behavior to a user, a program, or system resource. In most cases, a qualified analyst can characterize these situations quickly, when given the relevant performance statistics.

The second question asks what options are available to modify system behavior in these circumstances.

UNICOS is designed to deliver maximum possible performance and capability to the user. This is a two-edged sword. On the one hand, a single user can easily make use of the entire machine, given appropriate permissions. On the other hand, a single user, given appropriate permissions, can dominate a system, interfering with other users.

The actual question is seldom one of whether there is an "inherent" problem in the system, but one of users competing for resources, and the resulting contention causing inefficiency. Commands such as nschedv(8) allow the system administrator to assign priorities to various competing users of system resources and allow them to be consumed in a more orderly fashion.

The job of the administrator is always made easier by users who run jobs that run as efficiently as possible. Even the best system administration and tuning will not cause highly inefficient programs to become efficient. A system running inefficient programs will always be an inefficient system.

4. There are many bugs in the CRAY T3D debugger; not very useful!

Cray Response: We did have some stability problems with our initial release of TotalView on the CRAY T3D. Most of the known problems were fixed in the CrayTools 1.3 release (TotalView 1.1.0.2 and later). Because of this we have recommended that all Cray T3D customers upgrade to the CrayTools 1.3 or later release. CrayTools 1.3 was released as part of the 1.2 Programming Environments at the end of 1994.

We have had good luck working directly with some CRAY T3D sites on TotalView stability and usability issues. This effort included on-site visits and follow up discussions to identify problems which are now fixed. These sites are also helping to identify and prioritize the usability and stability issues. Cray is very appreciative of the efforts from these sites and we would like to recognize their contribution to the TotalView product.

If sites are having problems with the current TotalView, we need to hear about it. Please submit SPRs; We have a good history of providing TotalView fixes to SPRs. If you have comments and questions that don't seem appropriate as an SPR, we also encourage you to contact your local service organization, who can then forward your comments or put you in touch with Technical Support and Software Development for TotalView.

We are continuing to work on stability in 1995. In March of 1995 we will release CrayTools 1.3.1 (TotalView 1.2); this will provide additional fixes, support for cut and paste of text within TotalView which should make it easier to report problems, and xhelp changes to improve navigation through the on-line documentation. In addition to stability, we will be working on TotalView usability improvements and scalable debugging for MPP programs throughout 1995.

5. Operators are taught how to use xsam, systop, sysmon, syscall, where swap, SDS and checkpoint images are written, how memory packer, swapper, fair shares and nice-ticks influence the service provided. None of this is documented for the operations staff.

Cray Response: The comments in items 5 and 6 are closely related. The response below is directed toward both comments.

6. Not aware of any "operator" documentation.

Cray Response: In responding to these two comments, we'll assume what the customers are asking for is UNICOS-specific operator documentation (as distinct from OWS-E documentation.) This is a request we've heard before, but one we've run into issues with in trying to resolve.

The first issue we've run into has been coming up with a good definition of what information an operator needs. In investigating this topic, we've found that the role of an operator varies greatly from site to site, and even within a single site.

The second issue is the possible overlap of information. Do we document information in both an administrator's guide and an operator's guide if it is information both groups of people will want to know? In the past we've been reluctant to do this because of both support and proliferation of information issues.

With these two issues in mind, we see two possible solutions for providing UNICOS-specific operator documentation.

1. If there is general consensus on the importance of separate operator documentation, we could provide the documentation at a future date if the operations SIC would agree to define for us the scope of the necessary information.

2. You can use the journaling feature of CrayDoc to create hyperlinks to different sections or subsections of documentation, then have the operator step through this prescribed path to use the information. The journal files are something you can create yourself, and therefore customize both for your site and for each operator. We could provide some example operator journal files to help you get a start on this.

This solution avoids the redundancy of information problem discussed earlier. However, to work well, it will require our documentation to be more modular, a long-term goal of ours that hasn't been completely implemented yet.

Two related things before we close the topic. First, we do have a Support System Administrator's Guide (SG-3078) which covers what we think operators will use of our support system software on the OWS.

Second, there is an operator's training workbook which covers some basic UNICOS operator information.

7. Poor Integration testing of 8.0, MLS, CRL, and DMF delayed our UNICOS 8.0 install for 6 months. Cray is pushing more for the sites to do, while not providing the proper tools.

Cray Response: Throughout the development for UNICOS 8.0, where the emphasis was on obtaining a U.S. DoD B1 security rating, we put most or our resources into testing the Trusted UNICOS configuration. Since this time, and because of the number of SPRs related to the Priv-TFM configuration of both DMF and CRL, we have instituted testing procedures which ensure that all configurations are examined in a equitable manner.

Without more specific information regarding which UNICOS 8.0 level and what the problems were, it is difficult to address this question more specifically. Similar concerns were raised in the CUG site survey last fall. The following response was provided to CUG members during the Fall 1994 conference in Tours and explains in detail what these problems are and what we have done to address them.

> *October 1994 Cray Response*: The priority in the area of security for UNICOS 8.0 was on obtaining a U.S. DoD B1 security rating. This meant that many of Cray's resources were directed toward this particular security configuration at the expense of other Multi-Level Security (MLS) configurations. Many code changes were made throughout the UNICOS MLS product in support of Trusted UNICOS, the product targeted to receive this rating.
>
> The testing efforts too were focused on the same priority, the B1 rating. Although some minimal testing occurred on MLS (non-Trusted UNICOS configurations), it is true that most of the testing efforts (including feature, installation, configuration, and regression testing) were focused on the B1 evaluation itself.
>
> These priorities led to numerous MLS related issues after UNICOS 8.0 was released. The majority of these problems were related to configuring and installing a UNICOS MLS system. There are five general categories in which the problems encountered can be grouped.
>
> - documentation
> - installation and configuration
> - software bugs
> - 7.0-style trusted facility management (TFMgmt)
> - administrative expectations for Trusted UNICOS
>
> Although most of the MLS related documentation and installation/configuration issues were minor, they were numerous and dispersed. Thus, these issues were a major annoyance to the customers who experienced them.
>
> So far, there have been very few software bugs that have shown up in UNICOS 8.0 MLS systems, and most of these were minor in nature. There have been, however, a few urgent or critical problems. When customers have encountered urgent or critical problems, Software Development has worked directly with the affected sites to resolve the problems and to provide fixes as quickly as possible.
>
> Of the installation/configuration, documentation, and software bug issues, the more critical problems were addressed in Cray Research Service Bulletin (CRSB) articles and/or via the Software Field Notice (SFN) mechanism. Also, fixes for all are included in the UNICOS 8.0.3 release. Finally, additional MLS testing was applied to the UNICOS 8.0.3 release.
>
> The fourth problem area is related to the 7.0-style TFMgmt mechanism. This mechanism is supported in UNICOS MLS 8.0, along with two new mechanisms. The shortcomings of this particular mechanism were documented via a past CRSB article and in the 8.0 user documentation. In short, if levels/compartments (security labels) are used, many system utilities do not work correctly when the 7.0-style TFMgmt mechanism is enabled. Also, this mechanism does not support a true separation of operator / administrative roles. Although these issues are documented, they are often misunderstood. Some customers migrating from a previous UNICOS MLS system to 8.0 enabled security labeling and thus encountered these 7.0-style TFMgmt problems for the first time. The problems related to the 7.0-style TFMgmt mechanism are basic design flaws. For this reason, Cray expects to retire this mechanism in the future. An SFN documents this issue and suggests a course of action to resolve it for 8.0. Also, the administrative documentation is being clarified.
>
> The last general problem area is related to Trusted UNICOS itself. Its configuration requires a Privilege Assignment List (PAL)-based TFMgmt mechanism. Although this mechanism works correctly, it is quite different from an administrative viewpoint from the other two supported mechanisms. It takes some getting used to. The documentation didn't make this explicitly clear, and some customers were thus surprised. The documentation is being enhanced to make this clear.
>
> In summary, most of the UNICOS MLS and Trusted UNICOS issues have either been addressed or are in the process of being addressed. Cray's expectations are that the MLS problems encountered in the initial 8.0 releases have been addressed, and that customers planning to move to a UNICOS MLS 8.0.3 configuration should experience a smooth transition.

End of extract from 1994 Cray Response.

8. In general, Cray has very well trained people, both hardware and software. The software fixes are generally slower in coming unless on a “fast track”. This is an unacceptable, unstable alternative.

Cray Response: The timeliness of fixes is dependent on several key items:

- Severity of the problem
- Our ability to provide a high-quality, regression-tested fix
- Customers’ desire or ability to accept and install software upgrades

This final item is a function of both frequency and size of the release vehicle. Our experience to date with the majority of our customers indicates that they are unwilling or unable to take our upgrades and revisions as quickly as we can generate them.

As a consequence of this nearly-uniform hesitancy to upgrade at the pace which we can provide reliable, tested fixes, we have been led to a policy of spending more of our resources on doing preventive maintenance on future systems (e.g., 9.0) than on a frenzy of upgrades (which may be dominated in size by “major” and “minor” fixes) that are not being picked up by our customers. We will continue to provide our fastest service for the most important problems, but we are making a conscious choice to include majors and minors only in Major releases. We believe that both customers and Cray will achieve higher satisfaction via this process. Your input into this set of priorities will be very important to us as we move forward through 1995 and beyond. For example, it may be useful to provide us with feedback which suggests that different areas of Software could benefit from different sets of priorities.

Severity of problems, and our goals for responsiveness, are shown below.

SPR Severity	Guideline	Deliverable
Critical (OS)	1 week 8 weeks	Workaround, plan to fix or individual fix Released package (update or revision)
Critical (other)	2 weeks	Released package (update, revision or major) or individual fix (for products without updates)
Urgent	8 weeks	Released package (update, revision or major)
Major	30 weeks	Future revision or next major release
Minor	60 weeks	Next major release
Design	4 weeks	Initial response

Cray is not perfect in meeting these goals, but we do strive to get fixes available as quickly as possible. As a sample data point, the TLC Group (chosen because the author of this text is in TLC) provided this profile of responsiveness (i.e., % of time responsiveness goals were met) over the last 365 days. (Other groups had different profiles, some better, some worse.)

Product Name	% of all Severities
CF77	82
CF90	100
CF90_S	76
CRAYLIBS	73
CRAYTOOLS	88
DPE	100
EMULATOR	100
NETWORK_VM-3	83
UNICOS	91
UNICOS-MAX	96
All Releases	87

9. The Programming Environment (compiler, etc.) for the CRAY T3D seems to be unstable. We continue to have problems with features not having been tested, with compartments causing features to “not work”!

Cray Response: We do not understand and have been unable to determine what is meant by *features not having been tested with compartments causing features to “not work”*. The T3D compilers are tested with a test suite that consists of almost 700,000 lines of Fortran code. These tests include the FCVS Fortran validation suite as well as a very large number of tests created specifically to test CRAFT features. We regularly add tests for all of the SPRs that we receive as well. We are always trying to expand our test base and would be happy to add applications that customers have a history of discovering compiler problems with. We would also be willing to create additional tests ourselves for specific weak areas in the compiler if whoever submitted this comment could identify them for us (just contact your local service representative and ask that your concerns be forwarded to the appropriate individuals in Technical Support.)

10. Problems continue with register parity errors. They are now hitting user jobs rather than the system as in previous periods. The ability to

identify hardware problems, e.g. double bit errors, is still a cause for concern. Having to swap boards to track failure is unacceptable.

Cray Response: This is a two part question, first, the issue relating to register parity errors.

CRI has implemented several improvements in the hardware and software to reduce the incidence of register parity errors (RPE) in C-90 systems. The effect of these improvements has been profound at all C-90 sites. Prior to implementation of these improvements CRI would observe one or more reported RPEs on a daily basis. Today, we observe one or less reported RPEs on a weekly basis. Where at one time, RPEs accounted for half of all field CPU reported interrupts, today it is well under 10%. Briefly, the improvements included new IC packaging, improved system grounding and significant resiliency improvements in software. The resiliency improvements are designed to cope with RPEs in such a manner that if the RPE is taken in a user job, the system will not interrupt. Although we would like to eliminate RPEs entirely, we believe that keeping the system from taking an interrupt is a significant improvement.

Part two: We believe this questions refers specifically to memory related interrupts on C-90 systems.

The architecture of the C-90 memory, which provides for the high bandwidth of C-90 memory, also makes diagnosing an intermittent failure quite difficult. Within the past six months, CRI has developed and implemented a new tool called SMON, which stands for System Monitor, that we believe has greatly improved our ability to diagnose any hardware failure. When a hardware failure does occur, SMON is triggered. SMON then, in effect, takes a mini-dump of all key registers and areas of memory which contain information to aid CRI service personnel in determining which FRU is failing. SMON has frequently proven beneficial, with the result being less interrupts, over the past few months. CRI continues to enhance SMON as we learn more about failure modes These enhancements are made available to sites on a regular basis.

11. We would like to have well defined documentation for UNICOS installation or upgrading.

Cray Response: There is a set of four UNICOS installation documents. These documents and their catalog descriptions are as follows:

SG-2112 UNICOS Installation Guide

> Describes how system analysts and support people can install and do basic configuration of the latest release of UNICOS, using the Cray Research menu-driven, installation tool. Upgrade installations from older versions of the UNICOS operating system and revisions to them, as well as initial installations are discussed. Appendixes contain lists of files that the installation tool affects and files that have been changed, renamed, or moved from older versions of UNICOS to the latest version.
>
> This document is revised and reprinted whenever significant enhancements are made to the installation tool or the installation process.

SR-2412 UNICOS Installation Tools Menus and Help Files Reference Manual

> Describes the interface to the UNICOS Installation and Configuration Menu System, how to navigate through the system, and shows all of the menu screens and help files.

SQ-2411 UNICOS Installation Menu System Reference Card

> Describes the UNICOS installation and configuration menu system interface. This document briefly summarizes the input and command keys, shortcuts, possible customizations, and menu information areas of the menu tool.

SR-3090 UNICOS Installation Menu and Configuration Tool Reference Manual

> Documents the UNICOS Installation and Configuration Menu System for system analysts, software developers, and others who need to know how the menu system works.

We have also recognized a customer requirement for improvements to release-specific information regarding release content and known problems, as well as feedback on current experiences with a new release. A key 1995 initiative for the Technical Support group in Customer Service is to improve software installation support. This effort is focused on improving the release-specific information available to customers who are doing their own UNICOS installations or upgrades.

We have implemented a prototype for an on-line document that contains known problem information similar to that currently found in the standard UNICOS errata. Because this document will be provided online, rather than as a hard copy created at the time of release, this “living errata” will be more accurate and up-to-date for each customer whenever they schedule the upgrade on their Cray Systems. Current plans are to introduce the online errata for UNICOS 9.0, and possibly UNICOS 8.0.4. You will be able to access this information directly using CRInform.

Additional changes are being evaluated and planned as we gather customer input regarding specific information requirements for UNICOS installation and upgrade support.

12. Very, very disappointed with the late delivery of the new machine (3 weeks late) and it is coming in a provisional configuration. The final configuration is expected a month and one half after the expected delivery date.

To start, all of the CRI groups involved in the design and implementation of the J916 project are also disappointed in missing the goal for first product shipment. In looking back over the 30+ months that we have worked on the J916 project, there is one area where we were certainly much more optimistic than we should have been. The area is called physical design and is the process where the logical design of the ASICS is turned into a physical design. Both CRI and IBM (chip manufacturer) under-estimated the scope of the task. CRI set new upper limits for Gate Array and Standard Cell designs, and when you are the pioneer, you don’t know all the answers ahead of time.

Although CRI initially thought we would need to ship limited configurations for the first few systems because of material availability issues, CRI has worked through these issues and plans to ship all systems per the configuration ordered.

13. We are becoming increasingly concerned with network security issues. Cray will need to address Kerberos in a manner suitable to a heterogeneous environment.

Cray Response: As of Unicos 9.0 Cray will offer two flavors of Kerberos: Kerberos Version 4 and DCE Kerberos.

Kerberos V4 has been integrated into several products, including the following:

- telnet, rlogin
- ftp, rcp
- NFS
- AUTH_KERB RPC

The Kerberized versions of telnet, rlogin, ftp, and rcp provide secure, encrypted interactive sessions and file transfers over untrusted networks.

Kerberized NFS provides secure access to remote file systems by authenticating the NFS requests that traverse the network. However, it does not encrypt the data in the NFS request, nor the data which NFS returns.

AUTH_KERB is a new flavor of Remote Procedure Call (RPC) which provides secure communications between client-server programs over untrusted networks. Programmers must add library calls to create the necessary authentication structures, and to process the authentication. Users of kerberized RPC applications must have a kerberos ticket granting ticket to access a kerberized RPC server.