# Control System Reliability at Jefferson Lab

Karen S. White, Hari Areti, Omar Garza
Thomas Jefferson National Accelerator Facility
12000 Jefferson Ave., Newport News, VA 23606
USA

## Abstract

At Thomas Jefferson National Accelerator Facility (Jefferson Lab), the availability of the control system is crucial to the operation of the accelerator for experimental programs. Jefferson Lab's control system, uses 68040 based microprocessors running VxWorks, Unix workstations, and a variety of VME, CAMAC. GPIB. and serial devices. The software consists of control system toolkit software, commercial packages, and over 200 custom and generic applications, some of which are highly complex. The challenge is to keep this highly diverse and still growing system, with over 162,000 control points, operating reliably, while managing changes and upgrades to both the hardware and software. Downtime attributable to the control system includes the time to troubleshoot and repair problems and the time to restore the machine to operation of the scheduled program. This paper describes the availability of the control system during the last year, the heaviest contributors to downtime and the response to problems. Strategies for improving the robustness of the control system are detailed and include changes in hardware, software, procedures and processes. The improvements range from the routine preventive hardware maintenance, to improving our ability to detect, predict and prevent problems. This paper also describes the software tools used to assist in control system troubleshooting, maintenance and failure recovery processes.

## 1 Introduction

The Continuous Electron Beam Accelerator at Jefferson Lab provides high current electron beams of up to 4 GeV energy to three experimental halls. The machine consists of two superconducting linear accelerators connected together with 9 arcs. An injector system provides polarized and unpolarized electrons from two different sources. RF and magnetic separation in the beam switch yard splits and transports beam to the halls. The range of deliverable currents to the halls spans <1 nanoAmp to 120 microAmps and requires a broad class of diagnostic equipment. The linacs contain 340 superconducting cavities and the accelerator has well over 2000 magnets. Additionally, the machine utilizes a variety of diagnostic devices for beam delivery.

The task of the control system is to provide control and monitoring for all these elements as well as various graphical user interfaces, archival of data, alarms and analysis tools for machine operators, engineers and physicists. The goal is to achieve 98% control system availability during accelerator operations in support of our goal of overall machine availability of 70%.

## 2 Control system

The Jefferson Lab Control System, which is built on the Experimental Physics and Industrial Control System (EPICS), is a distributed system using a client-server architecture, name based I/O and TCP/IP communication [1]. The system consists of two computing levels. The first level is composed of Unix workstations and X-terminals which execute a wide variety of system applications such as operator interfaces, archiving, backup and restore, alarms, and downtime logging. Also at this level, high level applications, for automated machine setup and control, are executed and displayed. The second level is composed of VME 68040 based single board computers running the VxWorks real-time operating system, EPICS system software and the corresponding device control applications. These computers are called Input/Output Controllers (IOCs) and connect to various types of hardware control modules. Some of the IOCs also communicate with embedded microprocessors for specialized device control. The computers at the two levels communicate over a segmented ethernet network.

The EPICS IOCs are connected to hardware modules through a variety of interfaces including CAMAC, VME, GPIB, Serial and some custom interfaces. The most common type of hardware interface used for control at Jefferson Lab is CAMAC. Over 100 CAMAC crates containing more than 1000 CAMAC modules communicate with the IOCs via a Serial Highway using Hytec Serial Highway Driver cards and L2 Serial Crate Controllers. More than 100 VME modules used for device control are located in the same VME enclosures which hold the IOCs. About a dozen GPIB devices are interfaced to EPICS IOCs using a second 68020 microprocessor in the same VME crate. Additionally, several different types of serial devices are connected directly to IOCs using the front panel serial ports. In order to support this complement of hardware, device support routines must be incorporated into EPICS, and, in some cases, custom VxWorks kernel must be built.

EPICS provides the basis for our control system in the form of the core code which executes and manages the device control applications which run on the IOCs, the communication layer software for name based I/O known as Channel Access, and the generic tools for graphical user interfaces, alarms, datalogging and backup and restore. For device control we have developed, using EPICS, 120 different applications. In many cases, similar hardware configurations are repeated throughout the machine, and the EPICS device control applications are replicated and loaded on multiple IOCs. A total of 850 instances of these applications are loaded on the 80 control system IOCs that run the accelerator.

On the Unix side, in addition to the standard EPICS tools mentioned above, a number of additional system and high level applications are utilized for machine operations. Some of these applications also use commercial software packages such as Matlab and ObjectStore. Other applications make use of freely distributed software such as the GNU tools, Tcl/Tk and the World Wide Web. In addition to the complexity inherent in this system of so many integrated pieces, is the problem of how to maintain stable control system operation while making frequent upgrades to the existing hardware and software and integrating completely new systems.

The standard operating schedule calls for machine operations 24 hours a day, 7 days a week with one 8 hour shift for maintenance every two weeks. Within this single shift, all hardware and software upgrades must be installed and tested. Additionally, we must provide the ability to return the control system to the previous operational state if testing during an upgrade reveals a problem.

The effect of a program error on the Unix side is usually that a single program or tool is unavailable until repair. Frequently, operations can continue in these cases. On the IOC side, the effect of an error or a hardware failure is far more damaging as the entire IOC is general rendered unavailable to operations until a repair is made. It is almost always the case that this type of problem stops machine operations and counts as accelerator downtime. For this reason, we have concentrated our initial efforts to provide software versioning and the ability to quickly roll back applications on the IOC software. This system is now in place and works very well. There have been many times when the rollback mechanism has been quickly and successfully invoked to return an IOC to operations when new software has introduced a bug. Now that this system is working well, we have turned our efforts in this area to the Unix systems.

## 3 Machine availability

Jefferson Lab's goal for total machine availability for Fiscal Year 1997 (FY97) was 70% uptime. Time is counted as uptime when the accelerator is delivering beam on a target in at least one of the three experimental halls. In order to support this goal, the Accelerator Control System had a goal of 98% uptime during scheduled machine operations. Unfortunately, and despite much hard work, we fell short of both goals. The actual machine availability for the year was 58.6% during 41.8 weeks of scheduled operations. Control system availability, at 94.3% contributed 402 hours of downtime, almost 262 hours over our goal. Clearly, this excessive downtime must be eliminated in order to support overall machine availability for physics experiments. Figure 1 shows control system downtime by month. The month of January is absent from the graph since this was scheduled downtime for machine maintenance and upgrades.

In order to improve the availability of the control system, we track causes of downtime, and focus our efforts on solving the problems which cause the most downtime. Downtime is tracked by an automated program called Downtime Logger and by operator input to this program. Using this information, we are able to allocate resources to reduce the most costly sources of downtime, and implement tools and procedures to address underlying causes of greater downtime such as troubleshooting time and change management. The data, which consists of 210 control system downtime entries this fiscal year, has been analyzed to determine the failures that were the single longest causes of downtime, and also the repeating causes which take only a short time to recover from, but which occur so often as to have a large cumulative effect on overall downtime. For FY97, the top five longest control system downtimes account for 32% of the total 402 hours of control system downtime and are shown in Table 1.

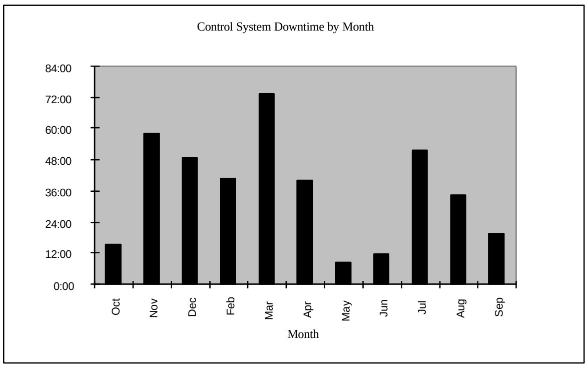

Figure 1. Control System Downtime by Month

Table 1. Top Five Downtime Events

| Downtime Cause | Time (Hours:Minutes) |
|---|---|
| CAMAC Power Supply | 39:24 |
| Fiber Optic U-Port | 25:34 |
| CAMAC Power Supply & L2 Controller | 22:14 |
| Disk Failure | 21:20 |
| Network Hub Failure | 21:01 |
| Total | 129:33 |

The first three entries in this table show many hours of downtime for simple replacements of standard hardware parts kept on hand. The time incurred here is the total time to diagnose and repair the problem and the time to restore the machine to the previous running condition. In two of these cases, time to determine the failed component was long. Each of these events also resulted in a crash of the Central Helium Liquefier (CHL) Plant due to loss of control, which added additional recovery time. In the case of the network failure, additional time was needed to obtain replacement parts. The disk failure problem was lengthy to repair because the failed disk was a root volume without a spare copy built. This problem was complicated by failures of the backup system in the preceding days making file recovery very difficult.

Grouping downtime events into categories, also helps to identify costly problems that are not so obvious. The data for the top five categories of downtime, as shown in Table 2, accounts for almost 77% of all control system downtime.

This data also reveals that the most significant category of control system downtime, IOC problems does not even show up at all in the top five most time consuming failures.

Table 2. Top Five Downtime Causes By Category

| Downtime Cause | Time (Hours:Minutes) |
|---|---|
| IOC | 110:28 |
| Network | 66:28 |
| CAMAC Serial Controller | 61:49 |
| CAMAC Power Supply | 48:30 |
| BPM Mux VME Card | 21:37 |
| Total | 308:52 |

This is because IOC failures do not usually take such a long time to recover from, but at times occur quite frequently. Of the 210 control system downtime entries logged in FY97, 113 were classified as IOC problems. One reason for such a large number here is that IOCs are the most visible part of the control system, and often, problems that really originate elsewhere are logged as IOC problems. Failures of the network, VME modules or VME crate power supplies are often first thought to be IOC problems. While we try to update or correct downtime entries when a different problem cause is found, this is difficult since machine operations are often restored long before the root cause of a problem is discovered.

Despite such reporting errors, IOCs still constitute a major source of control system downtime. Unlike the Unix programs, the effect of an error in a program running on an IOC is generally that the entire IOC and all applications running there fail to function when the bug is reached. We have discovered and fixed two such bugs during this reporting period which account for well over half of the reported IOC problems. These types of problems are generally difficult to isolate and debug due to the numerous applications running on each IOC and various system interactions.

The network also proved to be a significant source of control system downtime. This was due to a combination of actually hardware failures and configuration problems. These failures have proven time consuming to troubleshoot, and for that reason, we have added additional network monitoring tools and are training more people on this system.

The fact that crate power supplies contribute almost 50 hours of downtime can be traced, at least in part to the age of this equipment. Most of the CAMAC crates installed in the machine are now approaching ten years old. We are now systematically replacing and monitoring these power supplies in an attempt to reduce operational failures. The CAMAC Serial Crate Controllers have no corresponding reason to account for over 60 hours of downtime as most of these cards are less than four years old. We have found that controllers supplied by some manufacturers are significantly less reliable than others, and are replacing as many as possible with the better brand.

The VME Mux Controller cards were found to have poorly soldered connections, and since this problem has been fixed, this cause of downtime has been eliminated.

## 4 Downtime reduction

In order to reduce downtime attributable to the control system, we have examined the causes and identified the biggest time sinks, underlying difficulties and trends. We have established the following steps in an ongoing effort to improve troubleshooting and communication and minimize downtime.

- *Work towards a uniform installed base of hardware and software wherever possible.* This policy helps by reducing the potential sources of software failures by standardizing, and reduces the number of different types of hardware that must be understood and maintained for spares.
- *Minimize the number of changes made at one time.* This policy speeds the troubleshooting process because fewer changes means fewer items to work through to determine the cause of problems in a previously operational system.
- *Phase in changes after testing in a limited area.* This policy reduces recovery time in cases where new software or hardware must be removed from the system to restore operability. It is significantly faster to rollback software on 3 IOCs than 80.
- *Document installation, testing and recovery procedures for new features.* This policy speeds repair time by providing a written procedure for how to remove a new feature. This usually enables any on-site or on-call software personnel to restore a system rather than requiring a particular individual be reached. This policy is currently only utilized for software, but we believe there is a corresponding benefit for hardware as the effect is to ensure the work and testing is well thought out and documented, also improving communication between groups. Providing these testplans alone has reduced the instances of software rollbacks because quality has improved through this preplanning effort.
- *Schedule Control System testing time along with Machine Recovery.* Experience has shown us that the earlier a problem is discovered after machine maintenance, the lower the operational impact. By extensively testing control system upgrades during machine recovery time, we have often avoided machine downtime since problems could be resolved before scheduled beam delivery. One form this policy has taken is the use of standardized checks of all IOCs and CAMAC modules during machine lockup. This has allowed problematic IOCs to be restored before instead of during beam operations.
- *Maintain critical spares.* Having on hand tested spare parts, organized in an accessible location speeds repair time once a hardware problem is identified.
- *Preventive Maintenance.* Power supplies on VME and CAMAC crates are now routinely checked and

repaired as needed. Filters are periodically cleaned and fans replaced in an effort to prevent operational failures.

- *Provide on-site support during machine recovery.* The groups responsible for control system hardware and software now provide on-site support during machine recovery after a maintenance period. This eliminates the time required to call in a person for troubleshooting and repairs.
- *Improve and add tools to assist with troubleshooting.* Many problems we have encountered this year have proven difficult to troubleshoot. We have developed several new software tools to assist in this area. One program monitors the response of CAMAC modules and reports errors. This has proved a valuable diagnostic tool for failures of individual CAMAC modules, and allows bad modules to be quickly identified and swapped, virtually eliminating troubleshooting time. We plan to further enhance this program to detect problems with the CAMAC serial highway. A second program checks indicators of IOC problems, such as suspended tasks, corrupted stacks and low memory, and reports any problem areas. This allows for a quick check of all IOCs and is often used as a first step to troubleshoot individual IOC problems. We also plan to add more extensive checks to this program. A third program, called IOC Core Dump, is used to capture valuable information about IOC problems. The program stores status information from the IOC and the contents of the IOC memory. Once this program has completed running, the IOC can be rebooted and returned to operation, and the data can be analyzed off-line. This method was used to find and fix the two software bugs that were responsible for a great deal of IOC downtime.
- *Improve software detection and reporting of failures.* In the cases of the frequently failing BPM Mux Controllers, the software was modified to identify the bad controller, raise an operator alarm and continue running, avoiding the bad card. This was a huge improvement over the original program behavior which disabled the entire IOC. We also added alarms to warn operators of problems such as low memory on an IOC to give time for a controlled reboot instead of surprise crash.
- *Identify and focus effort on fixing the most time consuming problems*. This approach has already been discussed in the data analysis. This is a necessity since available resources are not enough to address every problem at this time.

## 5 Conclusions

Control system downtime has exceed our target goal by almost a factor of three this year. We have also observed that problems temporarily increase after machine downtimes due to changes and upgrades, and have focused additional effort during these times. Since these steps were first outlined in April 1997, the trend for control system downtime has improved. Dividing the year into two six month periods, before and after these steps, shows that downtime during the second period is significantly lower. October 1996 - March 1997 accounted for 236 hours of downtime, and April 1997 - September 1997 had 166 hours. The first six month period really contains only five months January was a scheduled maintenance period. Even the smaller number of hours for the second half of the year still exceeds our goal of 2% control system downtime, and we will continue our efforts to make improvements.

## Acknowledgment

Thanks to Tom Hassler who provided data about overall machine availability and Steve Suhring who provided detailed data for this paper.

## References

[1] K. S. White, H. Shoaee, W. A. Watson, M. Wise, "The Migration of the CEBAF Accelerator Control System from TACL to EPICS", CEBAF Control System Review (1994).