Experience Using EPICS on PC Platforms

J.O. Hill, LANL P.O. Box 1663, Los Alamos, NM 87545 USA (LANL)
K.U. Kasemir, Universitat Osnabruck, Fachbereich Physik, D-49069 Osnabruck, Germany

Abstract

The Experimental Physics and Industrial Control System (EPICS) has been widely adopted in the accelerator community. Although EPICS is available on many platforms, the majority of implementations have used UNIX workstations as clients, and VME- or VXI-based processors for distributed input output controllers. Recently, a significant portion of EPICS has been ported to personal computer (PC) hardware platforms running Microsoft's operating systems, and also Wind River System's real time vxWorks operating system. This development should significantly reduce the cost of deploying EPICS systems, and the prospect of using EPICS together with the many high quality commercial components available for PC platforms is also encouraging. A hybrid system using both PC and traditional platforms is currently being implemented at LANL for LEDA, the low energy demonstration accelerator under construction as part of the Accelerator Production of Tritium (APT) project. To illustrate these developments we compare our recent experience deploying a PC-based EPICS system with experience deploying similar systems based on traditional (UNIX-hosted) EPICS hardware and software platforms.

1 Introduction

EPICS is a collaboratively developed process control and data acquisition software toolkit in use at over 70 sites world wide. The software is designed for general utility and has been successfully installed into a wide range of applications including particle accelerators, experimental physics detectors, astronomical observatories, municipal infrastructures, petroleum refineries, and manufacturing. A scalable, fault-tolerant system that follows the "standard model"[1] can be created with the toolkit. Compilers and filters are used to instantiate control algorithms in front-end computers from function block and state-machine formalism -based input. EPICS communication occurs within a software layer called Channel Access (CA) that follows the client server model and employs the internet protocols (Figure 1).

A mature set of client-side tools provide operator interface, alarm handling, archival tasks, backup, restore, state sequencing, and other capabilities. There is also an expanding library of hardware device drivers that have been written for use with EPICS. Recently we have seen a number of sites working on generic physics and control theory applications that will interface directly with EPICS. All of these components taken together form a toolkit that allows control system installation with a minimum of low level coding. Details can be obtained on the world-wide web [2] and from previous papers [3][4][5][6].

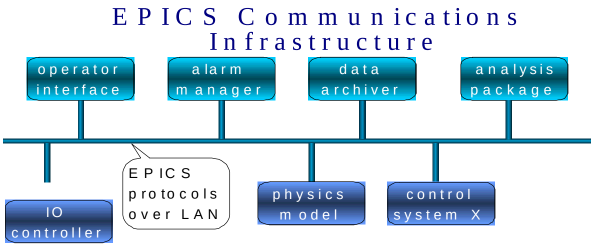

Figure 1

2 Economics

Over 80% of the desktop (client) computers in the world run Microsoft operating systems. The economics of scale results in a built-in cost advantage for Microsoft compatible hardware and software components. Frequently this results not only in lower cost components, but also in a wider selection of available hardware and software components.

Economics of scale also applies to programming and system administration personnel. With a growing number of individuals possessing Microsoft compatible skills, it may be easier to locate trained personnel for control system projects that use Microsoft compatible components. Conver-sely, personnel with control system related PC experience should find that their skills can be applied to the widest variety of future projects.

3 Potential uses of personal computers by EPICS

One can imagine three potential applications of PCs in EPICS-based control systems: i) as operator interfaces; ii) as input-output controllers; and iii) as a development environment. There are differing considerations in assess-ing the suitability of PCs in each of these roles. These are discussed below.

3.1 Operator console

Traditionally, EPICS Operator Interface (OPI) clients have been based upon the UNIX / X Window Systems. With the growing popularity of Microsoft Operating Systems, operators are frequently more familiar with that environment than with UNIX, and hardware and software costs are lower. Moreover, it might be more efficient for the OPI to be in close proximity to the defacto standard analysis, modelling, and office tools running on the PC. For all of the above reasons we are considering PCs as a potential new platform for the EPICS OPI.

There are also downsides. Newly-developed native WIN32 graphics codes for the PC will not be portable to other operating systems such as UNIX or VMS. JAVA is today the most frequently mentioned potential solution to this dilemma. However, the X Window System should not be dismissed as a portable graphics API. It is operating system and hardware architecture portable; the MOTIF GUI conventions are functionally compatible with Microsoft Windows GUI conventions; many existing graphics applications are already using this API; and many high quality / cost sensitive commercial X server components exist. Many of these components are seamlessly integrated with Microsoft Windows. Native X applications are not limited by the "lowest common denominator" of window system functionality provided by the JAVA virtual machine, but of course will never be downloadable from a web browser as are all JAVA applications.

3.2 Input output controller

Traditionally, EPICS Input-Output Controllers (IOCs) have been based upon the VME / VXI backplane bus standard (Figure 2). In contrast, PCs come with the widely-used ISA I/O bus, or with the newer PCI or Compact PCI buses (Figure 3).

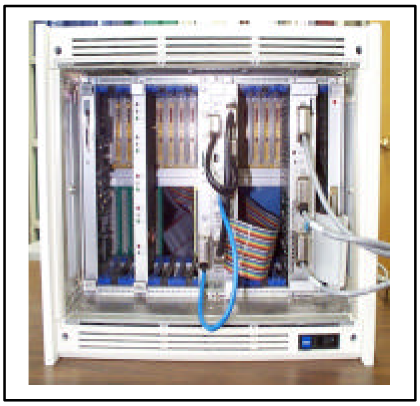

Figure 2 VME IOC

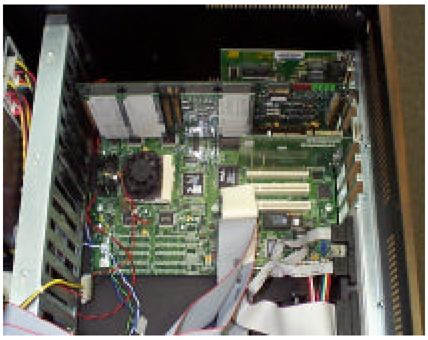

Figure 3 PC IOC

Today, the widest variety of industrial IO is available only in the ISA form factor, with many manufacturers still waiting to convert ISA bus designs to PCI and/or compact PCI standards. The ISA bus interrupt structure is primitive. There is no concept of an interrupt vector, and interrupt handlers must be daisy chained if there are more than 15 sources. Many of the 15 levels are reserved for essential peripherals such as the network interface card, the floppy disk, the hard disk controller, and the system clock. Both VME and PCI have a more flexible interrupt structure. Use of ISA and PCI cards will require new vxWorks device drivers, but improving vxWorks support for PCI may reduce the effort required. To address issues related to signal conditioning, wiring, and packaging in desktop PCs, repackaging into industrial, rack-mountable enclosures may be required. The reliability, even of repackaged desktop PCs, is as yet unproven in the physics experiment environment.

VME allows more slots than ISA or PCI (Table 1).

Table 1

	VME	PCI	Compact PCI
slots	21	4	8

Both the PCI bus and the ISA bus require card edge connectors that are more susceptible to wear than VME and Compact PCI. Unlike ISA and PCI, VME is a fully asynchronous, true multiprocessor bus. PCI offers better throughput than VME, but new versions of VME64 place VME64 in the same performance regime as PCI. Sensitive analogue instrumentation frequently requires the card-to-card noise immunity and analogue supply voltages present in VXI, but not available in VME, PCI, or ISA.

The vxWorks operating system used by EPICS IOCs requires a "board support package" (BSP) to interface the operating system with the host computer hardware. Traditionally, a new BSP must be purchased, modified, and maintained for each VME / VXI single board computer (SBC) used with EPICS. In the PC environment, the BSP interfaces directly with the PC basic input output system (PC-BIOS), and therefore only one BSP must be purchased, modified, and maintained for a wide range of PCs from different manufactures.

Table 2 shows a cost/performance comparison between a VME 50 MHz MC68060 SBC-based IOC and a 166 MHz Pentium PC-based IOC. The test measured the execution delay for a single EPICS "calculation function block" executing a ramp function with one input, alarm checking, and no forward links. EPICS communications performance tests on these processors tend to be 10 Mb Ethernet bandwidth limited, and therefore no significant performance differences were identified.

Table 2

	50 MHz 68060 VME SBC	166 MHz Pentium PC
Cost of Card Cage and Processor	7k US $	3.5k US $
Function Blocks per Second	50 k	66.7 k
Cost per function block per second	14 US ¢	5 US ¢

In conclusion, we feel that PCs are a rational and cost effective IOC choice in certain, well-selected situations when interfacing with computer bus independent IO. For example PCs might be used when interfacing with GPIB, field buses, industrial LANs, industry packs, and any other IO that does not physically reside in the card cage of the PC. In this situation the differences between the various computer bus architectures, with the exception of cost, appear to be less significant. Another advantage of this type of IO is the potential to share device driver components between PC and VME-based systems. Industry pack IO is appealing in systems that interface to only a small number of signals of each I/O type because a higher effective packing density can be obtained.

3.3 Software development environment

Traditionally, almost all EPICS software development, including cross development for real-time systems, has occurred under UNIX. Recently, we have identified some advantages offered by the desktop PC software development environment. The advanced integrated development environments pioneered on PC platforms offer many tools to increase programmer productivity. Almost all of the traditional UNIX command line tools have a GUI- driven equivalent on the PC. The authors moved all of their software development from UNIX to the PC about 6 months ago. This was a positive experience, and currently there is (for them) no compelling reason to switch back. Another potential benefit is “programmer portability”. An increasing number of programmers are familiar with the PC software development environment, and this represents a pool of programming talent for control system applications developed on the PC.

Disadvantages include the facts that the integrated development environments all produce incompatible proprietary make file formats; and the Windows NT POSIX subsystem exists in isolation and therefore can’t be used together with many of the essential components of the Microsoft operating system environment such as the WIN32 subsystem and the windows socket library.

4 Personal computer port of EPICS

The port of EPICS to personal computers was a multiple lab initiative. The initial port of the CA client library was performed several years ago at LBL. Ports of the EPICS build system and the CA server were accomplished at LANL. Important structural changes to the cross development environment for EPICS function block databases occurred at ANL.

The EPICS build system has for several years been based on the GNU MAKE utility, the UNIX shell, and some UNIX pattern matching / replacement utilities {SED, AWK, GREP}. While public domain PC versions of all of these utilities can be located on the Internet, we consolidated instead onto a minimum set in order to simplify the installation of EPICS. We selected GNU MAKE and PERL for this role. During the port we moved most of the UNIX shell script fragments directly into GNU make. The remaining shell scripts used SED, AWK, and GREP and were easily converted to PERL. During the port it was also necessary to generalise the build system with respect to file extension names and shareable library creation. EPICS application development is now mostly identical on PCs and on UNIX.

5 Early experience with PCs

The RF Window Test is one of the first PC based EPICS installations implemented at LANL (Figure 4). In this system we have an interaction between many diverse systems. Low level control of the high power RF systems was implemented by a contractor using Allen Bradley programmable logic controllers (PLCs). The RF window efficiency and lifetime tests were implemented with National Instrument’s LabView. EPICS systems supply supervisory control, archiving, and alarm management. In the future, EPICS will also provide system integration and automation between the APT high level RF subsystem and other APT subsystems.

Currently the RF Window Test Stand is in a commissioning phase, with first experiments to begin during the 4^{th} quarter of 1997. For this system we are employing UNIX workstations as operator consoles, but are also evaluating JAVA, a Windows port of existing UNIX and X based software, and commercially available window systems compatibility layers as possible PC enabling software alternatives.

6 Conclusions

The core components of EPICS are now running on PCs. System programmers have made the transition from UNIX to Windows NT, and have been able to work effectively on both systems. While there are still many VME / VXI advantages, PCs must be considered because of their economics of scale. When a greater variety of PCI and especially compact PCI IO cards are available, the economic balance may shift even further in the direction of PC- based systems. The first LANL PC I/O controllers are interfacing with computer bus independent I/O, and this appears to have been a reasonable choice.

References

[1] B. Kuiper: ‘Issues in Accelerator Controls’, Proc. ICALEPCS, Tsukuba, Japan, 1991, pp 602-611.

[2] W. McDowell et al.: ‘EPICS Home Page’, “http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/WWWPages/EpicsFrames.html”

[3] L. Dalesio et al.: ‘The Experimental Physics and Industrial Control System Architecture: Past, Present, and Future’, Proc. ICALEPCS, Berlin, Germany, 1993, pp 179-184.

[4] L. Dalesio et al: ‘The Los Alamos Accelerator Control System Database: A Generic Instrumentation Interface’, Proc. ICALEPCS, Vancouver, Canada, 1989, pp 405-407.

[5] J. Hill: ‘Channel Access: A Software Bus for the LAACS’, Proc. ICALEPCS, Vancouver, Canada, 1989, pp 352-355.

[6] J. Hill: ‘EPICS Communication Loss Management’, Proc. ICALEPCS, Berlin, Germany, 1993, pp 218-220.

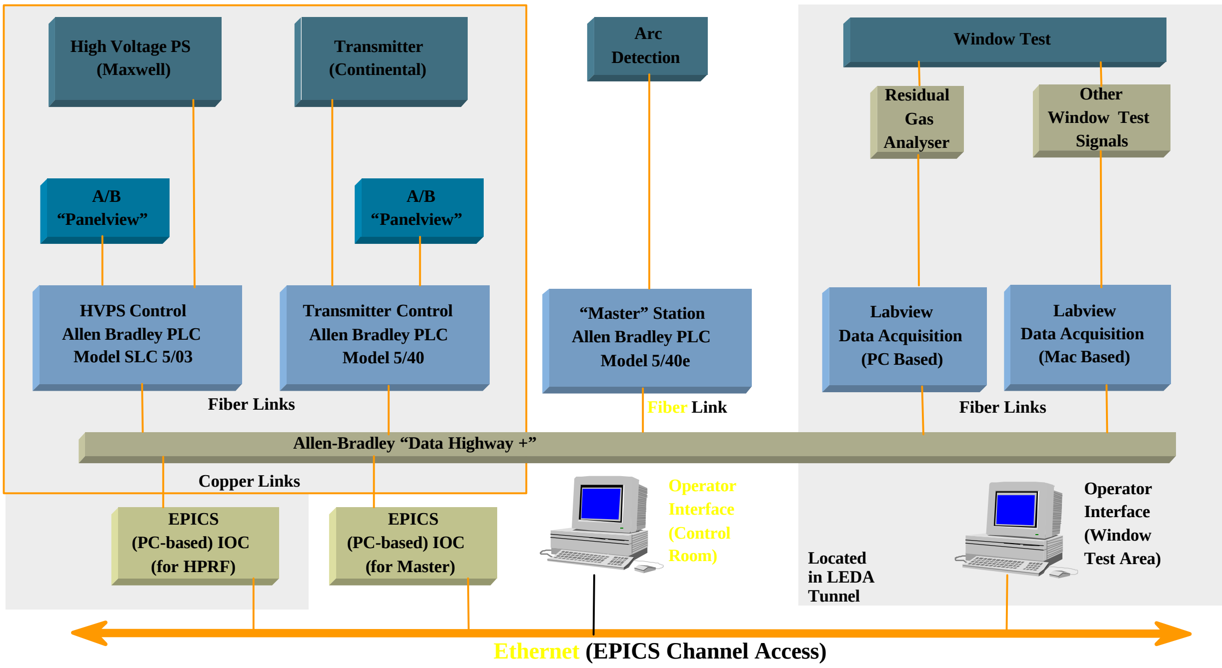

Figure 4