Siberian Branch of Russian Academy of Science
BUDKER INSTITUTE OF NUCLEAR PHYSICS

T.V. Salikova, A.D. Oreshkov, Ying Wu, John M.J. Madey,

STATUS OF PORTING EPICS TO THE LYNXOS/PENTIUM PLATFORM.

Budker INP 98 20

Novosibirsk
1998

Status of porting EPICS to the LynxOS/Pentium platform.

T.V. Salikova, A.D. Oreshkov.
Budker Institute of Nuclear Physics SB RAS,
630090 Novosibirsk, Russia.

Ying Wu, John M.J. Madey.
Duke University, Free Electron Laser Laboratory, Durham, NC, USA.

Abstract

The Experimental Physics and Industrial Control System (EPICS) requires expensive hardware and software to provide the high performance and high reliability required for accelerator control. No more than several years ago, personal computers based on Intel x86 microprocessors lacked computing power to compete with workstations, such as SPARCstation. However, with the advent of the high performance and low cost Pentium (x86 platform) in the recent years, high performance PCs become an attractive alternative to the expensive workstations. The recent trends to search for low cost effective solutions in the research laboratories provides us enough incentives to port EPICS to a PC platform. LANL group has used PCs successfully in the EPICS environment in which a PC running WindowsNT (or Windows95) is employed as a host computer to provide an Operator Interface (OPI). A PC or a MVME167 mounted in a VME crate is used as the target computer to operate as the Input Output Controller (IOC). At present time, standard EPICS IOC uses VxWorks as real time operating system. The most recent version of VxWorks is provided by Wind River Systems as a part of a rather expensive TORNADO integrated environment tool kit for software cross development. Besides using PC hardware, further saving would come from replacing VxWorks with a less expensive real time OS, such as LynxOS. The overall cost of both IOC and OPI systems can be minimized by porting EPICS software to homogeneous environment of PC running LynxOS. This approach also simplifies the installation and development of EPICS based control systems. This will report the status of porting EPICS to the LynxOS/x86.

Introduction.

The Experimental Physics and Industrial Control System (EPICS) [1], initially co-developed by Los Alamos National Laboratory (LANL) and Argonne National Laboratory (ANL), later joined by a number of other laboratories, provides a set of software tools and a architecture framework for developing real time control systems for a variety of physics and industrial applications.

As sketched in Figure 1, EPICS consists of three basic components, the Operator Interface (OPI), input output controller (IOC), and local area network (LAN).
The EPICS OPI, running on the UNIX workstation, provides an interface to operate the control system via a set of OPI tools.
IOC consists of a VME/VXI based chassis with a Motorola 680x0 processor, various VME I/O modules, and VME interface modules with other I/O buses or chassis. The IOC software is built upon the realtime operating system (OS) VxWorks.
LAN is the local area network which provides the communication between IOCs and OPIs via the EPICS networking tool kit, Channel Access (CA).
By providing a framework for control system development, EPICS allows application developers to implement and develop customized control systems in a rapid manner. However, the standard hardware and software currently supported by EPICS has proven to be unaffordable for some laboratories with relatively low operation budget. Attempt has been made to find a cost-effective solution for implementing EPICS by using much less expensive hardware (such as PCs and PC control boards) and software (such as the realtime OS, LynxOS) [2].

The new generation personal computers powered by Intel Pentium chips (or their clones) are becoming ever powerful while remaining rather inexpensive compared with workstations with similar performance. With the improved reliability and availability of Unix-like operating systems for PCs, the PC hardware becomes an attractive choice for the low cost solution for computer control. In addition, a less expensive realtime system, LynxOS, is readily available for the several hardware platforms including x86 in both native and cross development environments.

LynxOS is a UNIX-compatible, POSIX-conforming, multiprocess, and multithreaded operating system designed for complex real-time applications that require fast, deterministic response. Like VxWorks, LynxOS supports all the necessary features for a realtime system, such as shared memory, memory locking, preemptive scheduling, priority inheritance, semaphores, pipes, events, messages, clocks, timers, and synchronous and asynchronous I/O.

Because of the bundled System V and BSD system-call interfaces and libraries shipped with LynxOS, LynxOS is highly compatible with UNIX. Porting of UNIX codes to LynxOS is relatively straight forward and painless. By porting EPICS codes to LynxOS/x86, powerful EPICS systems can be constructed with combined benefits of EPICS software and the low cost, high performance hardware and realtime OS.

Adapting of Channel Access tools to LynxOS platform

While the local area network provides the physical media for IOCs and OPIs to communicate, Channel Access (CA) is a sophisticated EPICS software tool which supports network independent communication mechanism between the CA clients and servers [3]. The CA client software allows a CA client to search for an IOC for service via a unique process variable (PV) name, reads and writes the value of a process variable, and allows the CA sever to update the value or status of a process variable. A CA client is designed to communicate with an arbitrary number of CA servers running on IOCs. The CA server software handles the CA client search message, establishes connections with CA clients, and services the client requests. In addition, the CA software handles CA server and client crashes and reconnections in a graceful manner.
Our first step of porting EPICS to LynxOS/x86 is to port CA to the new platform so that a LynxOS/x86 station would be able to communicate with an operational IOC. Porting of CA requires changes to be made in the CA library [3] [4] function calls since most of system calls differ for LynxOS, VxWorks, and Solaris. Two LynxOS/x86 systems have been installed for the EPICS porting development. With this setup, we were able to perform a simple client/server connection test in the debug mode by running modified CA codes on the two computers. However, since we are still in the process of acquiring the necessary hardware, including a VME crate and embedded computer (MVME167), to complete the construction of a standard VME/VxWorks based testbed IOC system, testing of ported CA codes with an operational IOC is not possible at this time.
We hope that by the time the majority of CA codes have been ported to LynxOS, an operational IOC will be available for testing.

Porting OPI tools to LynxOS/x86.

Unlike the standard implementation of EPICS, which employs a non-realtime system (a version Unix such as Solaris, SunOS, or HP-UX) for code development and operator interface, and a realtime system (VxWorks) for realtime I/O control, the approach to port EPICS to LynxOS/x86 allows us to work in a unified OS environment of LynxOS for both the software development and I/O control operation. Using the System V and BSD system-call interfaces and libraries provided by LynxOS, as well as the graphical packages such as X11 libraries and Motif available for LynxOS, porting of OPI packages is relatively straight forward. After installation of all necessary GNU software recommended by EPICS collaboration, several UNIX OPI tools provided in EPICS R3.13.0beta11 have been ported and recompiled on LynxOS v2.5.
We start OPI porting process by working on a set of basic OPI tools, such as the Database Configuration tool (DCT) [5] which is used to create, modify, and maintain the EPICS record database files, the Display Manager (edd/dm) tool which allows the construction and operation of control windows, Alarm Handler (ALH) [6]which is used to detect and handle alarm states, and the Archiver (AR) and Archiver Retrieval Utility (ARR) [7] which provide acquisition, storage, manipulation, and display of the realtime data.
Porting of DCT has mostly been completed, while various degrees of progress have

been made with porting of other OPI tools such as edd/dm, ALH, and AR/ARR.
We hope that in the near future, the ported OPI tools will be able to be tested with the IOC tested station.

Porting of IOC

IOC part of EPICS software employs a large set of VxWorks specific system and library calls, porting of IOC codes is one of the most difficult tasks. To be able to run IOC software on a new platform, the core IOC software components, such as channel access, database, database access, scanners, monitors, and sequencer should be ported first [8]. Besides IOC core, there are a large number of record support routines to be ported as well. In addition, because most of the supported IO hardware by the existing EPICS implementation would not be available for the new platform with ISA and/or PCI bus, we expect great efforts will be devoted to developing new device/driver support routines. To speed up the process of achieving our goal of implementing an operational, low cost EPICS control system, we will limit ourselves initially to a minimal set of IOC software components (figure 2) required for operation. In particular, we will start our process by porting the IOC core software, and then port a subset of record support codes.
Further more, we plan to support only CAMAC IO interface in the beginning.

CAMAC interface is selected for the initial support because we have a large amount of existing CAMAC hardware which has to be utilized by the EPICS control systems developed for our accelerators. In addition, CAMAC equipment remains the standard hardware manufactured by the Budker Institute of Nuclear Physics (INP) for the accelerator controls. The first PC/CAMAC interface modules which we already have, would support two independent channels to be connected with two crate controllers.
In addition, up to eight secondary CAMAC crates can be linked up to the crate directly controlled by the PC IOC. Therefore, a PC is capable of supporting tree-structure of CAMAC crates with multiple ISA/CAMAC interface boards.
We have performed a set of measurements with the Pentium Pro 180MHz developing station with LynxOS v2.5, which yields the following results:
1. the average interrupt response is about 15 microseconds;
2. the average task response is about 40 microseconds;
3. the CAMAC NAF execution time is about 8 microseconds.
These measured data have met the system performance requirements set for our first accelerator system, an free electron laser (FEL) at INP.

It is worth pointing out that a typical PC based IOC would be a more powerful and capable system than a VME based IOC due to its faster processor and a large system memory. Currently a standard IOC system employs an embedded MVME167 board with a 33 MHz (or 66) MC680x0 processor and a limited amount of memory up to 16MB. However, with a fraction of the cost of a VME based IOC, typical PC based IOC system would be equipped with a much faster CPU, typically 133 MHz to 230 MHz, and a much large system memory pool, 32 MB to 128 MB. A large memory pool allows a much large EPICS database to be used for the realtime control and a much improved system clock speed allows the IOC to handle more EPICS records and/or to improve the realtime response. It is envisioned that a single PC IOC will be adequate to handle the need for a small size accelerator such as our FEL with two to three thousand control records. Additional PC IOC may be required for fast data acquisition.

Until now, we have been working mostly on porting the EPICS channel access package and OPI tools, with various degrees of progress. However, our next priority is to take on the porting of IOC software. We hope that significant progress can be made in the near future.

This material is based upon work supported by U.S. Civilian Research and Development Foundation under Award No RE1 182.

References

1. L.R. Dalesio, M.R. Kraimer, A.J. Kozubal « EPICS Architecture» Proc. ICALEPCS, Berlin, Germany, 1993, pp 179-184.
2. A.D. Oreshkov, T.V. Salikova, J.M.J Madey, Ying Wu «Project for development low cost and high performce version of EPICS» NOBUGS97, Argonne, 1997.
3. J. O. Hill «EPICS R3.12 Channel Access Reference Manual» LANL, 1995
4. M.R. Kraimer, J.O. Hill «EPICS Channel Access Security Design» LANL, 1994
5. F. Lenkszus «Database Configuratin Tool user's guide» ANL, 1991
6. M.R. Kraimer, B.K. Cha, M. Anderson, J. Anderson «Alarm Handler User's Guide» ANL, 1997
7. R. Cole «Archiving Reference Manual» ANL, 1993

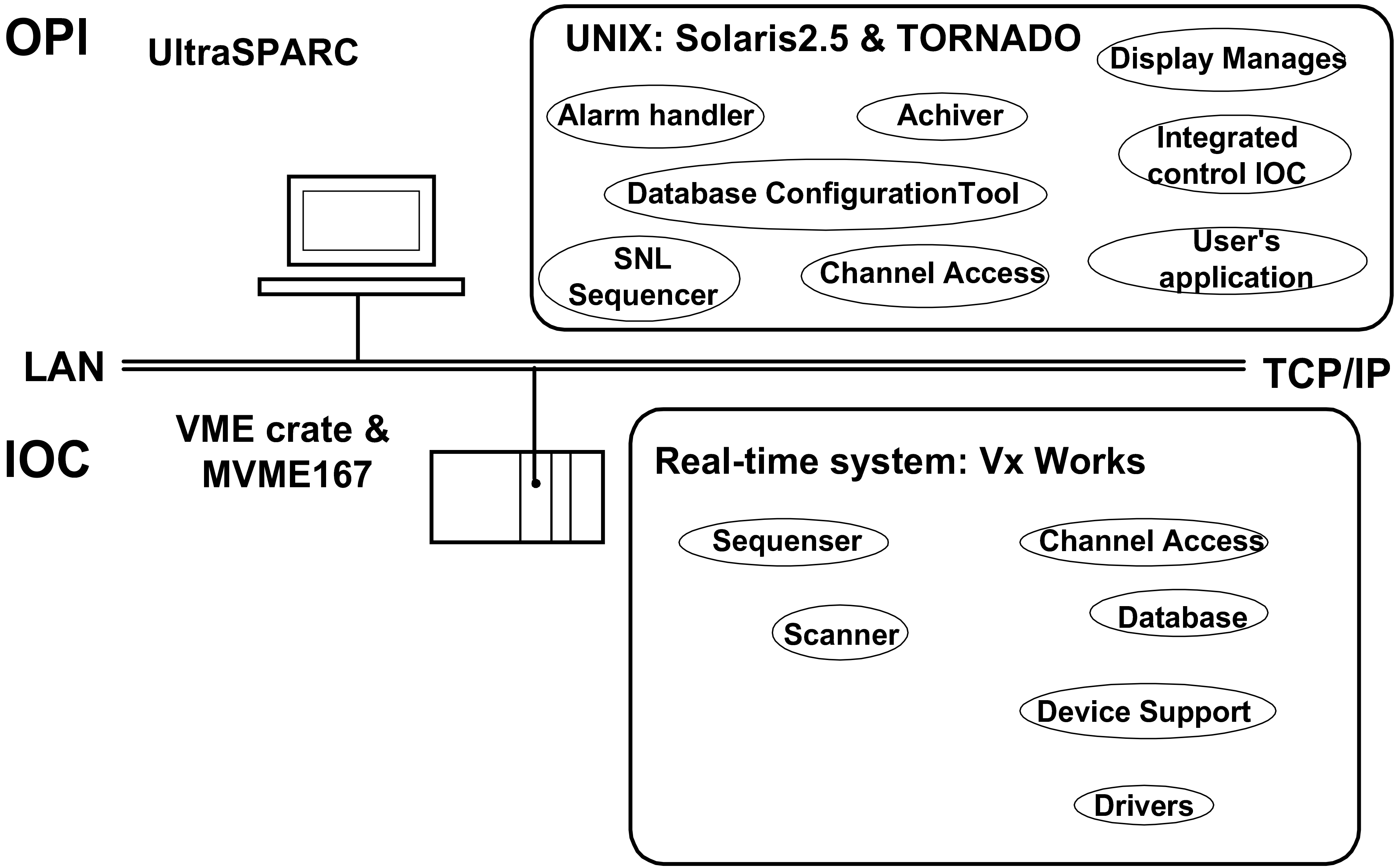

Figure 1. Structure of EPICS.

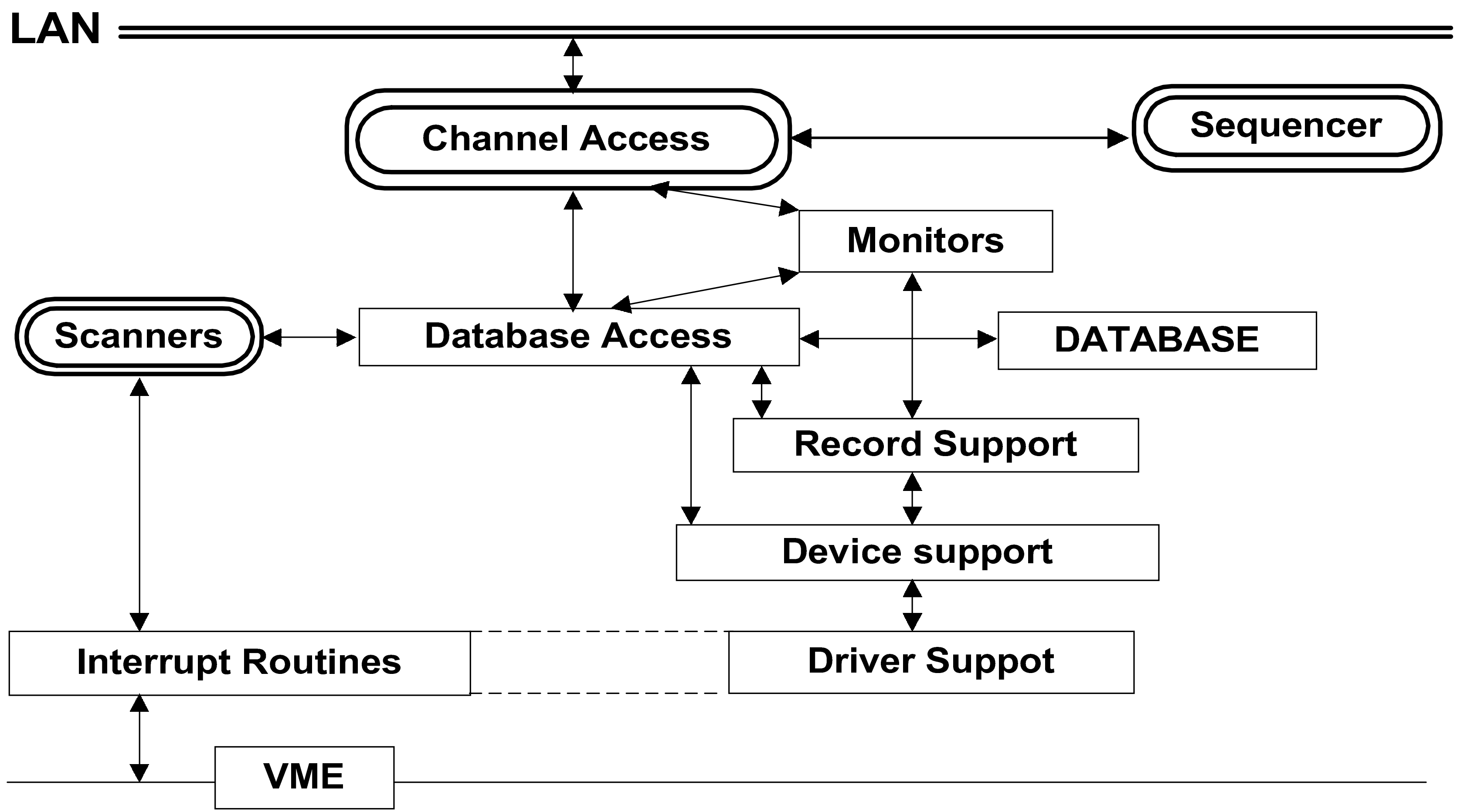

Figure 2. IOC components.