HIGHER AVAILABILITY WITH ATCA AND REDUNDANT IOC

○ A.Kazakov[1,2], K.Furukawa[1], M.Satoh[1], T.Suwada[1], S.Michizono[1]
1高エネルギー加速器研究機構(KEK)、2総合研究大学院大学(SOKENDAI)

Abstract

Reliability issues for modern accelerator machines become more and more important due to constantly growing and complexity of the machines. Redundant IOC for EPICS was introduced in 2006 and ported to Linux in 2007. Redundant IOC is aimed to provide redundancy and high availability for critical parts of an accelerator control system. The core of EPICS redundant IOC is Redundancy Monitoring Task (RMT). In this work Advanced Telecommunication Computing Architecture (ATCA) driver to RMT (and therefore to EPICS redundant IOC) is being developed. ATCA is a popular standard for new designs and installations of control systems. ATCA was suggested as a base architecture for International Linear Collider (ILC) control system. This paper describes the implementation of an EPICS redundant IOC (RIOC) in a PC-based environment with Linux or other Unix-like EPICS-supported OS's and benefits of using ATCA as platform for RIOC.

1. Introduction

An EPICS redundant IOC was originally developed at DESY. Two major fields of application were defined:

1. Redundancy for cryogenic plants. In this case, the failure may be caused by malfunctioning hardware such as power supplies or fans. An automatic fail-over mechanism should guarantee system stability. However, over the years it was sometimes necessary to manually switch between the main and backup processors due to maintenance work during the runtime period (which is usually one year or more). It might be useful for a software update. While the current commercial system used at DESY allows online updates of the database, EPICS does not allow addition or deletion of records and databases during operation.

2.Redundancy for controllers in the XFEL tunnel. Although the main origin of switch-over in the first case would be a manual action, it is expected to occur automatically in the XFEL tunnel. Due to high radiation, damage to the CPU and memory is highly possible. The software update is not very important because of more frequent maintenance days when this operation may be performed.

By the design draft one major goal was set: **Any redundant implementation must make the system more reliable than the nonredundant one. Precaution must be taken especially for the detection of errors that shall initiate the fail-over. This operation should only be activated if there is no doubt that maintaining the actual mastership definitely causes more damage to the controlled system than an automatic fail-over**. The fail-over time in any case was defined to be more than several seconds and less than 15 s. The final implementation could switch in less than 2 seconds.

Originally, the project was intended to support only vxWorks and the written code was very specific to it. However, later it was observed that support for other operating system is desirable. Here at KEK we use a software-IOC on Linux which functions as a "gateway" from an old control system to the EPICS-environment. In addition, for the ILC project ATCA-based systems under Linux control will be used. Redundant IOCs are highly desirable for this project. Thus, the redundant IOC has been ported to EPICS libCom/osi; this implies that the current implementation should work on any EPICS-supported OSs.

2. Hardware Architecture

The hardware architecture consists of two redundant IOCs controlling a remote I/O via shared media such as the Ethernet or MIL1553. The redundant pair shares two network connections for monitoring the state of health of their counterpart, where the private network connection is used to synchronize the backup to the primary and the global network is used to communicate data from the primary to any other network clients requiring the data.

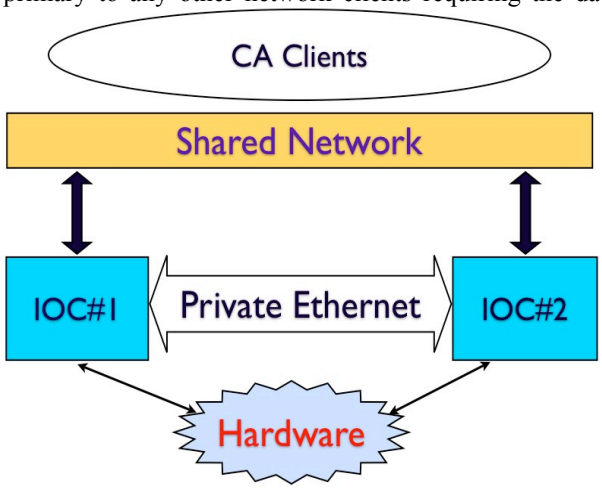

Figure 1: RIOC Hardware Architecture

3. Software Components

The current design contains three major parts: RMT (redundancy monitoring task), CCE (continuous control executive) and SNL (state notation language) executive.

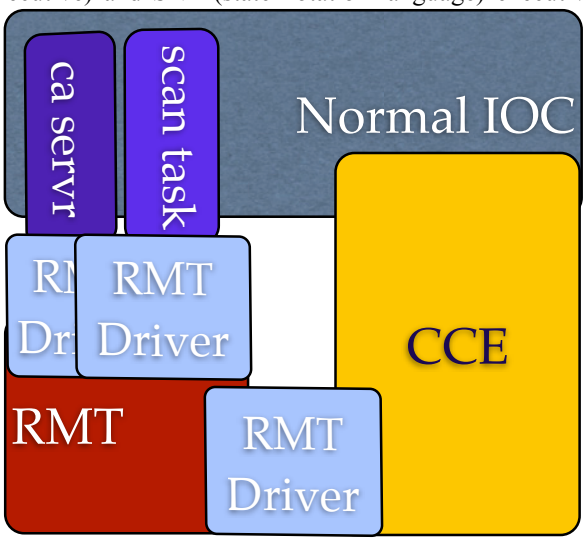

Figure 2: RIOC internals

The latter two are responsible for the synchronization of process variables (PVs) and internal states. The RMT is the core of the redundant system. It establishes and maintains the connection with the partner, controls other parts of the system and decides when to fail-over. All other components that have to be controlled by the RMT share the same software interface (which is defined in a header file rmtDrvIf.h). This interface defines the following functions:

1. start: Get access to the IO and start processing.
2. stop: Do not access the IO and stop processing.
3. testIO: Initiates a procedure to test access to the IO.
4. getStatus: Get status of the driver.
5. shutdown: This function is called before the IOC is rebooted. It terminates transient activities, deactivates interrupt sources and stops all driver tasks.
6. getUpdate: This routine tells the component to get an update from the redundant IOC. It is normally called by the RMT on the inactive IOC.
7. startUpdate: This routine tells the component to start updating data from the redundant IOC (monitoring). It will first read all fields (depending on the mode) from the redundant partner. It is normally called by the RMT on the inactive IOC.
8. stopUpdate: This routine tells the component to stop updating data from the redundant counterpart. This routine is normally called by the RMT on the inactive IOC.

The RMT can call these functions using an entry table that is transferred from the component to the RMT during initialization. The component calls the rmtRegisterDriver() function to register itself in the RMT and transfer the entry table.

CCE and the SNL executive are two such RMT-controlled components and they implement the RMT-driver interface mentioned above. Other components may be IO drivers, or any other piece of software. For example, the RSRV server in redundant IOC implementation is one of the RMT-controlled components and it implements the RMT-driver interface. Some of the interface functions may be unimplemented (set to NULL in the entry table) depending on the nature of the IO-driver and the tasks it performs.

4. Porting to libCom/OSI

Originally, all the developments were done only for vxWorks. The resulting code was not usable on any other OS. However, the people at KEK were interested in using a redundant IOC on Linux machines for a LINAC control system. Moreover, at DESY there was a demand for a redundant CA gateway. It appeared that the RMT has all the functionality needed for the solution. However, CA gateway runs on Linux (or other Unix-like OS), and the RMT was available only on vxWorks. Thus, it was decided to port the redundant IOC to Linux.

The ported OSI version of the redundant IOC was successfully used on vxWorks, Linux, Mac OS X, and Solaris. Tests showed that the system synchronisation speed limit was around ~5000 records/second for 2 Linux machines with 3GHz P4 1core, 2x 100Mbit Ethernet cards; functioning solely as a redundant IOC

5. AdvancedTCA

Advanced Telecommunications Computing Architecture (ATCA) is a new standard in telecom industry. Originally ATCA specification was designed with high availability in mind to be used in wide variety of applications requiring higher availability.

Every ATCA shelf consists of the following major parts[1]:

1. Subrack providing attachment points for the Backplane, as well as alignment, support, and mechanical engagement for the insertion and extraction of Front Boards and RTMs.
2. Backplane providing connector interfaces for power distribution and input/output connectivity between Front Boards.
3. Rear Transition Module (RTM) installed in the rear part of the shelf and mated to Front Boards.
4. Front Boards containing the desired electronic functions and the connectors required to interface with these functions.

AdvancedTCA provides for extensive management capabilities, which may be used by the overall System Manager. The Shelf Management system does the following:

- Monitors, controls, and assures proper operation of

AdvancedTCA Boards and other Shelf components
• Watches over the basic health of the system, reports anomalies, and takes corrective action when needed
• Retrieves inventory information and sensor readings as well as receive event reports and failure notifications from Boards and other Intelligent FRUs
• Performs basic recovery operations such as power cycle or reset of managed entities
• Provides low-level hardware management services to manage the power, cooling, and interconnect resources of a Shelf

One of the key components of ATCA Shelf Managements system is Shelf Manager board (preferably redundant). The Shelf Manager does the following:
• Watches over managed devices, reporting anomalous conditions to the System Manager and taking whatever corrective actions it can to prevent system failure
• Handles hot-swap events from removable devices, indicating their entry into the Shelf and detecting their shutdown or removal
• Negotiates power budgets with Boards and other FRUs so that the Shelf operates within those capacities.
• Initiates changes in fan levels when event messages show that temperatures are outside of prescribed bounds.

The management is realized via Intelligent Platform Management Bus (IPMB), which is a part of Intelligent Platform Management Interface (IMPI) architecture.

6. Advanced TCA and RIOC

ATCA platform uses redundancy internally for management system connections, shelf managers, backplane interconnections, network subsystem. Therefore it is a good basis for critical applications in accelerator control systems. ATCA was chosen as a standard for future ILC control system. But just running EPICS IOC's on the ATCA-standard CPU board does not make it redundant, even if there is another identical board. To solve this problem Redundant IOC should be utilized.

Ported to OSI library version of RIOC was successfully tested to run in ATCA-standard shelf. This configuration provides EPICS redundancy, which is essential for critical tasks and based on highly reliable ATCA standard.

7. ATCA driver for RMT and RIOC

Running RIOC in ATCA shelf is great in the sense of delivering better availability, but it does not utilize any of ATCA management features available.
A lot of management and health information can be retrieved via Shelf Manager board. This information can be acquired using Hardware Platform Interface. Hardware Platform Interface specification separates the hardware from management middleware and makes each independent of the other. Therefore, an application developed using HPI should be independent from particular hardware realization.

Using OpenHPI library an RMT-driver for ATCA is going to be implemented. Via this driver RMT can get information about health status of the partner board and the shelf itself. This information can give an ability to predict the failure and initiate a failover process before actual hardware starts to fail. For example if for some reason the temperature on the master board starts to raise, RMT could switch over to a slave board when the master board is still running. Therefore the failover happens in more stable and controlled environment. Because the hardware is not failing channel access connections could be gracefully closed and clients would initiate the reconnect procedure immediately after switchover to slave IOC. In a matter of a second all the connections would be restored and total control is regained. In case of typical scenario when the connections are not closed gracefully it takes 30 seconds for clients to reconnect to slave IOC.

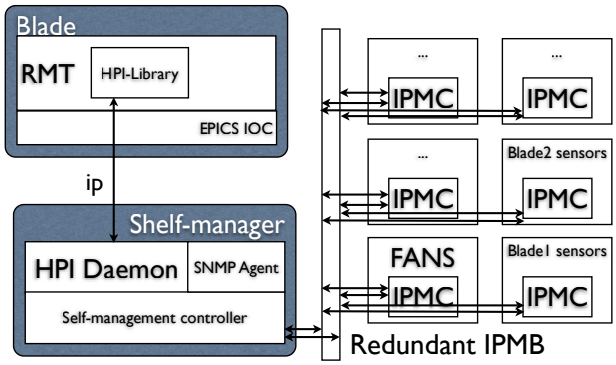

Figure 3: RIOC and ATCA driver

Summary

Porting redundant IOC to Unix-environment allowed a much wider application of this system. It was shown that the RMT functioning as the core of the redundant system can be utilized to add redundancy to other software. And it can be used to build highly reliable redundant systems based on ATCA-standard. Development of RMT driver to support ATCA-management system will deliver even better stability and performance.

[1] AdvancedTCA, PCIMG 3.0 Short Form Specification, January 2003.