Building highly available control system applications with Advanced Telecom Computing Architecture and open standards

Artem Kazakov*, Kazuro Furukawa

KEK, Oho-1-1, Tsukuba, Ibaraki, Japan

Abstract

Requirements for modern and future control systems for projects like ILC demand high availability for control system components. Recently telecom industry came up with a great open hardware specification – Advanced Telecom Computing Architecture (ATCA). This specification is aimed for better Reliability, Availability and Serviceability. Since its first market appearance in 2004 ATCA platform has shown tremendous growth and proved to be stable and well represented by a number of vendors. ATCA is an industry standard for highly available systems. On the other hand Service Availability Forum, a consortium of leading communications and computing companies, describes interaction between hardware and software. SAF defines a set of specifications such as Hardware Platform Interface, Application Interface Specification. SAF specifications provide extensive description of highly available systems, services and their interfaces. Originally aimed for telecom applications, these specifications can be used for accelerator controls software as well. This study describes benefits of using these specifications and their possible adoption to accelerator control systems. It is demonstrated how EPICS Redundant IOC was extended using Hardware Platform Interface specification, which made it possible to utilize benefits of the ATCA platform.

1. Introduction

Modern accelerator machines are complex and large scale structures. Projects like LHS and ILC consist of thousands of components that are spread over big distances in underground tunnels. Machines of that scale and complexity raise a set of challenges for all subsystems of the accelerator. And the accelerator control system is one of them. As an example of the current demands for the control system, let us see what are the essential requirements listed in International Linear Collider (ILC) Reference Design Report [1]. Among others there are the following requirements:

- Scalability. Big number of components, large distances, network bandwidth and real time accessibility throughout the site, and by remote access.
- High Availability (HA) for the whole system is aimed at 99%-99.9%. For a system built from 1200 shelves it roughly translates to 99.999% availability requirement for each shelf.
- Serviceability and manageability, standardization, quality assurance. Serviceability becomes an issue in relation to the scale of the machine (30 km long underground tunnel). A critical aspect of implementing a high availability control system will be the use of consistent ("best") work practices and a level of quality assurance process that is unprecedented in the accelerator controls environment. Commercial standards should be used wherever they can meet the requirements, for such things as hardware packaging and communication networks. The control system must specify standard interfaces between internal components and to all other systems.

Advanced Telecom Computing Architecture (ATCA) was suggested to fulfill these requirements.

* Corresponding author. e-mail: kazakov@gmail.com

1.1. ATCA

ATCA standard is defined by PCI Industrial Computer Manufactures Group. Availability of an ATCA crate is designed to be 99.999% [2]. Many components of the ATCA shelf are redundant power supplies, coolers, serial back-pane, etc. ATCA provides extensive manageability and serviceability capabilities, hot swap and high degree of hardware modularity. These and other features of the ATCA standard made it the platform of choice for the ILC control system.

ATCA is a good hardware platform, but it needs software that is capable to utilize all the features of the platform. If we are going to implement HA system, the software has to be designed with that idea in the first place. To face that problem and define the frameworks for HA applications, telecom companies formed the Service Availability Forum.

1.2. Service availability forum (SAF)

The Service Availability Forum™ [3] is a consortium of industry-leading communications and computing companies working together to develop and publish high availability and management software interface specifications.. Two main specifications by SA Forum are - Hardware Platform Interface (HPI) and Application Interface Specification (AIS). SAF specifications are primarily aimed at telecom industry, but they can be used for accelerator physics HA applications as well. SAF specifications represent current best practices in telecom industry and guidelines for building HA systems. Even if the accelerator society decides not to use them, it is beneficial to get acquainted with these specifications.

1.3. Hardware Platform Interface (HPI)

The HPI specification separates the hardware from management middleware and makes each independent of the other. The HPI concerns, primarily, the management of individual hardware components. The main purpose of the HPI is hardware control and monitoring. The HPI provides hot swap, the ability to replace hardware components within an operational system. Many of the mechanisms for providing service availability (such as managing standby components, failover and fault recovery) are provided by the AIS, rather than by the HPI. However, the HPI is self-contained and can operate independently of the AIS. The HPI is primarily used with ATCA, but HPI is a platform independent specification, therefore it does not require the usage of ATCA. There are HPI implementations that can run on common server-grade IBM PC compatible computers [5].

1.4. Application Interface Specification (AIS)

The SAF AIS standardizes the interface between SAF compliant High Availability (HA) middleware and service applications. The core element of the AIS is Application Management Framework (AMF). AMF provides service availability by coordinating redundant resources within a cluster to deliver a system with no single point of failure. Another important part is Software Management Framework (SMF) embodies standard mechanisms to deploy, configure and monitor the software used within the cluster. Besides AMF and SMF, the AIS defines a set of auxiliary services, that altogether provide standard API and infrastructure for building HA applications.

2. Building HA control system application based on EPICS, HPI and ATCA

ATCA is a rich hardware platform, perfect for building HA applications. It is possible to run EPICS IOC on ATCA CPU board. Despite the fact that ATCA provides great extent of redundancy internally, running “plain” EPICS IOC on ATCA does not make this system redundant or HA. Even if there are two identical CPU boards present and both of them are running the same EPICS IOC. As a first step to HA, EPICS Redundant IOC has to be utilized.

2.1. EPICS Redundant IOC

An EPICS redundant IOC (RIOC) was developed at DESY [4] as an extension to EPICS IOC to provide redundancy for critical applications such as:

1. Redundancy for cryogenic plants. An automatic fail-over mechanism should guarantee system stability.
2. Redundancy for controllers in the tunnel with the high radiation exposure.

The hardware architecture of an EPICS RIOC consists of two redundant IOCs controlling a remote I/O via shared media such as the Ethernet. 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.

On the software side an EPICS RIOC consists from four major parts Redundancy Monitoring Task (RMT), Continuous Control Executive (CCE), State Notation Language Executive (SNLE) and IOC-part (which is same as non-redundant IOC). RMT – is a key component of that system. RMT is responsible for monitoring all other parts of the EPICS RIOC, checking connectivity and making decisions regarding fail-over. Other parts of the RIOC are controlled by RMT and are called "RMT drivers" accordingly. Any other software, that needs to be redundant, has to implement RMT-driver API interface. Both standalone and IOC-related software, such as device drivers, can be made redundant using RMT API. Each RMT driver implements its own logic for checking whether it is "OK" or "not OK" depending on the particular driver, implementation and hardware the driver interfaces to. Exactly this approach was taken to make ATCA-aware extension for RIOC - it was implemented as an RMT driver.

CCE and SNLE are RMT-drivers that synchronize IOC-database and SLE programs between peers. Those parts are not relevant to the topic of this paper.

2.2. EPICS RIOC on ATCA

In case of using ATCA as hardware platform, the two IOCs are running on two CPU boards located in the same ATCA shelf. Even though it is beneficial to use ATCA to run EPICS RIOC in a sense of using reliable hardware, it does not differ much from using two separate PCs. ATCA provides extensive monitoring and management capabilities, which are not utilized by "plain" EPICS RIOC. HPI driver for EPICS RIOC was developed in order to make EPICS RIOC aware of ATCA hardware.

We chose HPI library to develop an extension for EPICS RIOC (in a form of RMT driver). That extension allowed including the status of the ATCA hardware into the fail-over decision process. Due to high level of abstraction and hardware independence of HPI it is possible to monitor any available set of sensor on the system without modification to the source code. The configuration is done by means of editing plain text file.

2.3. Benefits of using EPICS RIOC with ATCA support

The ability to monitor the hardware of the system allowed us to improve the reliability of EPICS RIOC. For example, if a CPU temperature starts to rise, there is some limited time before it will crash. And if properly monitored, we can initiate the fail-over process before the actual hardware failure happens. For the EPICS RIOC applications it gives us two major benefits:

- Fail-over happens while the system is still working, so actual transition happens in a stable and controlled environment.
- Reconnect time for Channel Access clients is drastically reduced, because connections can be gracefully closed. Normally in case of "hard" failure of a master RIOC it takes up to 30 seconds for CAC to reconnect to the slave RIOC (30 seconds is a default EPICS connection time-out, actual time-out may be changed by user). Even though the slave EPICS RIOC notices the problem instantly and within 2 seconds takes over.

The usage of HPI allowed us to avoid any hardware specific programming in the first place, but also made our solution portable to platforms other than ATCA. For example an open source implementation of HPI - OpenHPI - it can run on top of Linux 2.6 /dev/sysfs. And if the hardware allows provide access to the sensors. Therefore, without any source-code modification our "hardware-aware" EPICS RIOC can be used on common server-grade computers.

3. Summary

ATCA platform is being adopted by accelerator community as a platform for new high performance and highly available applications. ATCA brings new capabilities to the accelerator control system applications along with the new challenges to the software developers. New software needs to be designed in order to fully realize the ATCA potential. Telecom industry is offering a set of software specifications that can be utilized in accelerator physics. As it was demonstrated in this work, usage of standard software framework simplifies development and brings real benefits such as portability and modularity. Using HPI library EPICS redundant IOC was extended to support ATCA hardware.

References

[1] International Linear Collider Reference Design Report, Vol. 3, ILC Global Design Effort and World Wide Study, August 2007

[2] AdvancedTCA, PCIMG 3.0 Short Form Specification

[3] http://www.saforum.org/ and SAForum training materials

[4] REDUNDANCY FOR EPICS IOCS M. Clausen, G. Liu, B. Schoeneburg, Proceedings of ICALEPCS07, Knoxville, Tennessee, USA

[5] http://openhpi.org/

[6] USING EPICS REDUNDANT IOC IN UNIX ENVIRONMENT, A. Kazakov, K. Furukawa, M. Clausen, G. Liu, Proceedings of ICALEPCS07, Knoxville, Tennessee, USA.