CONTROL SYSTEM ISSUES AND PLANNING FOR eRHIC *

K.A. Brown†, P. Chitnis, T. D'Ottavio, J. Jamilkowski,
J. Laster, J. Morris, S. Nemesure, C. Theisen
Collider-Accelerator Department, BNL, Upton, NY

Abstract

The next generation of high-energy nuclear physics experiments involves colliding high-energy electrons with ions, as well as colliding polarized electrons with polarized protons and polarized helions (Helium-3 nuclei). The eRHIC project proposes to add an electron accelerator to the RHIC complex. In this paper we discuss the controls systems issues for eRHIC, the technical challenges, and our vision of a control system ten years into the future. What we build over the next ten years will be what is used for the ten years following the start of operations. This presents opportunities to take advantage of changes in technologies but also many challenges in building reliable and stable controls and integrating those controls with existing RHIC systems. This also presents an opportunity to leverage on state of the art innovations and build collaborations both with industry and other institutions.

INTRODUCTION

The eRHIC project will convert the existing Relativistic Heavy Ion Collider (RHIC) at BNL into an electron-ion collider. But there is even more to it than that. Since RHIC is the only polarized proton collider in the world [1], eRHIC would also become a polarized electron, polarized proton collider, as well as a polarized electron, polarized He-3 collider. Figure 1 shows the layout of eRHIC along with some basic parameters. The project will retain one of the existing RHIC rings and add an electron accelerator into the existing RHIC tunnel. The luminosity requirements for eRHIC are ambitious and in order to reach the luminosity goals a very intense electron beam must be produced. Since the beam dynamics in the collisions of these intense beams will disrupt the electron bunches (e.g., beam-beam forces will strongly distort the electron beams) after a single collision, each electron bunch will only collide with the hadron beams once. With these requirements and to make producing such beams cost effective, the electrons will be accelerated and decelerated using an Energy Recovery LINAC (ERL) [2].

The physics of eRHIC aims to explore the quark gluon plasma (QGP), first discovered at RHIC [1]. eRHIC will allow precision imaging of the QGP, determining the spin, flavor, and spatial structure of the nucleon. It will also allow probing more deeply into the nature of the strong force and the properties of gluons, the particles that mediate the strong force.

* Work performed under Contract Number DE-AC02-98CH10886 with the auspices of the US Department of Energy.
† kbrown@bnl.gov

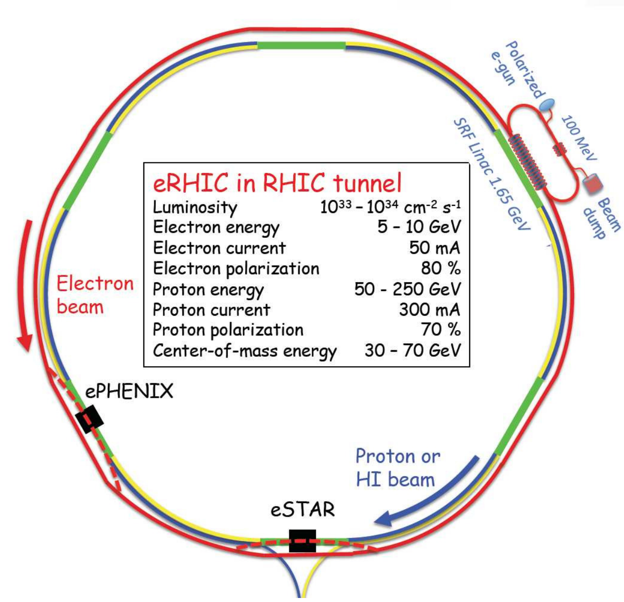

Figure 1: The eRHIC layout and basic parameters.

Keeping Pace with Physicists

eRHIC is mostly an idea. We know the physics goals and how those translate into requirements for the accelerator systems. We know that there are various solutions that will work, given the constraints of fitting an electron accelerator inside the existing RHIC tunnel. Much of what is being done at the moment is optimizing a design that meets these requirements within a reasonable cost. That design may change in some ways, but will basically be well enough defined for engineers to start thinking about how to build the various subsystems.

This leaves us, in Controls, with little to do, or so it would seem. We do know a few things that we can ponder while we await a final design. We know the basic components. We know that eRHIC will be composed of an ERL with a series of racetrack beamlines, perhaps one for each energy or perhaps each will be able to take multiple energy beams. We know there will be many corrector magnets and more or less how strong they need to be. We have a good idea how precise controls need to be. So we can consider what interfaces we think we will need for each system and we know we have to try for inexpensive interfaces that meet the technical requirements (including reliability and reproducibility, so we probably don't go with the cheapest possible approach).

eRHIC does pose some new challenges. The beam position monitors (BPMs) will have to measure the positions

ISBN 978-3-95450-139-7

of multiple beams in a single beam pipe (at least two, one accelerating and the other decelerating, but possibly more if there are multiple accelerating/decelerating beams in the same beam pipe.) Correcting the orbits of these beams is a problem still being studied. The machine protection system will have to dump the beams while keeping the beam currents balanced in the ERLs. The new systems will contain hundreds of camera systems along with thousands of BPMs and loss monitors.

We are going to be working from an existing controls system infrastructure that has been in place for over 15 years, albeit with a significant number of improvements that have been added as RHIC operations have evolved. However, eRHIC will not operate until 2024 or later.

There are many innovations that seem to continuously be coming to market. Users of our systems will want human interfaces that come to be intuitive with these innovations. It is hard to imagine a controls interface in the near future that does not allow for touch screen interaction or one that could not exist as an app on a touch pad device.

EVOLUTION OF RHIC CONTROLS

The RHIC Controls as they existed just after the construction and commissioning of RHIC are well described in a report by Barton [3]. The controls have evolved significantly over the past 12 years, particularly in various automation systems [4], feedback systems [5, 6, 7], and with all digital low-level RF systems [8, 9, 10, 11, 12]. As the controls have evolved they have grown from a system that was composed of roughly 0.5 million control points to what today consists of almost 1.4 million control points. The volume of data stored has increased in lock-step, from roughly 1.1 TB/year to over 100 TB/year. To manage the data volume and save on hardware costs, we now use data compression algorithms on all logged data (by far the largest fraction of the stored data) and only really store about 25 TB/year on disk.

In general, the architecture of the RHIC controls remains the same as described by Barton [3]. However, the system has grown in size, complexity, and speed [13]. The numbers of front end systems and the number of lines of computer code have each increased by a factor of three. The network bandwidth has been increased from 1 Gb/sec switched Ethernet to 20 Gb/sec. Network traffic has increased to a level that we now use stacked switch managed packet routing in high-density areas. Feedback systems introduced new demands on communication speed and performance. For the 10 Hz feedback system a new fast data distribution system was developed using commercial Gigabit technology [14].

We have had to keep pace with cyber security requirements, which were minimal at the time RHIC first began beam operations. We have multiple firewalls, sophisticated packet routing systems, and many controls networks are isolated. Much effort has gone into improving system monitoring and diagnostic tools, including automated system diagnostics. Data storage is always a challenging problem, requiring high reliability, fast recoverability from faults, and high fault tolerance. In addition, system users expect high availability of logged data with fast access times. This has led to a sophisticated data server system that provides the access users require [15]. This has naturally led to greater demands on data analysis and data mining tools, to where we have provided tools such as MATLAB® in addition to many custom analysis and data searching tools.

Front Ends

The RHIC controls system follows what can be described as a distributed object model architecture. Each front end consists of a server that contains specific accelerator device objects (ADOs), written in C++ [16, 17]. In implementation, a RHIC front end is a VME chassis with a VME processor board as the bus master, running on the VxWorks® real time operating system. Today's system model is basically the same, though we now also use ADO Managers that run from Linux servers. The ADO Manager allows for the same ADO interface model to be implemented for non-VME interfaces, such as communication to terminal servers(e.g., RS232/RS488/USB), direct Ethernet devices, or bus-bridging solutions (e.g., GPIB).

Middleware

The RHIC controls system follows the client-server communication model, with many layers of communication between the hardware and the application level software. Within the middleware layers of the systems are many possible paths to the hardware. Figure 2 shows a block diagram of the software system architecture. Depending on the high-level application requirements, there are many libraries and server interfaces that can be employed to access the hardware control points. One important innovation to the middleware layer is the development of Reflective servers, or what is basically a specialized proxy server to an ADO [18]. One of the main problems with a distributed object architecture is handling multiple client requests to a single ADO. For high demand front ends, inserting a proxy that abstracts to another server the device controls points greatly improves the performance of the front end. This allows the front end to perform its main function, controlling hardware, while still providing full access to the control points from many clients.

Application Level Interfaces

The general focus is on building generic application interfaces, thus allowing non-programmers to build custom interfaces to various subsystems. Any ADO parameter can be accessed directly, either from the command line in a terminal window, or through the Parameter Editing Tool (i.e., pet). Data collection tools, the General Purpose Monitor or through Logging Request files, allow users to build data collection for display or logging of ADO parameters. More sophisticated interfaces can be built using a synoptic display builder application, again for use by non-programmers, that can include multiple graphics objects

ISBN 978-3-95450-139-7

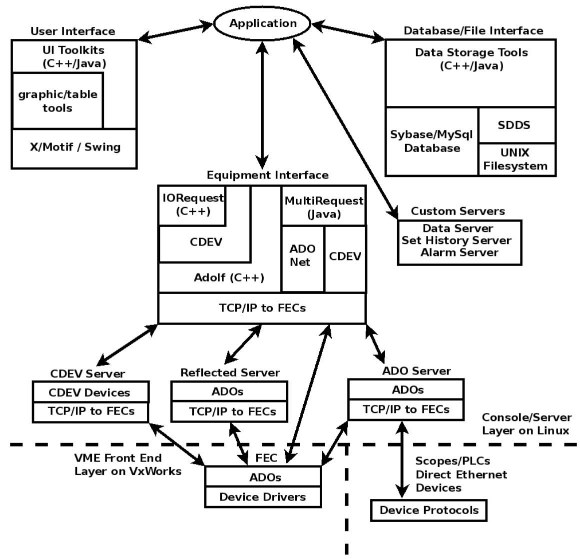

Figure 2: The RHIC Controls system software architecture.

within a single display.

Standard services include custom applications, such as alarms servers, elogs, databases support, data servers, and various system monitor, archive and set history servers.

Much recent effort has gone into building more applications and lower level interfaces with Java instead of C++. The goal is to move to a more platform-independent code base that will allow our existing interfaces to evolve with changes in technology.

The Rest of It

RHIC uses event links for global timing [19], a beam synchronization link [20] for more fine grained timing synchronization with the beam frequency, a real time data link that provides direct real time access to global data [21], such as time stamps, and beam permit and quench links, that form the nervous system for the machine protection systems [22, 23, 24].

Management of the controls' systems include network architecture management, Linux and VxWorks® system administration, hardware support and maintenance, operations support, call-in support, documentation, and user training.

ERHIC CONSIDERATIONS

We remain locked in a model in which people control accelerators from Control Rooms. But at RHIC, this is more procedural than a necessity. Quite often people control RHIC from their offices or even from home. We do not restrict where someone has to be to access the controls system.

Surely, when we come to operating eRHIC, the human-machine interface will be much different. What is used will strongly be connected to the intuition people develop through the use of commercial systems, such as smart phones or automobiles.

Data mining systems that use sophisticated search algorithms will become more important to our ability to process the huge volumes of data that future systems will collect. Even today, using Markov chain and eigenvalue search based algorithms, such as the Google PageRank [25], require sophisticated database and computational systems. Certainly when we come to designing the eRHIC systems a major consideration will be the architecture of the data collection, storage, and analysis systems.

Front Ends

eRHIC will add around 300 new VME systems, over 80 miles of copper based cable, and about 10 miles of fiber optic cable. There may be as many as 2850 new power supplies. For machine protection there may be as many as 10,000 coil temperature monitors, 720 water flow monitors, and new quench detectors for superconducting interaction region magnets. For instrumentation, depending on the final design, there will be 700 to 2100 BPMs, 240 to 730 synchrotron radiation cameras, 100 to 200 imaging cameras, 800 to 2400 loss monitors, and dozens of other systems. The plan for the RF controls is to follow the current approach of using processors attached to large gate arrays.

We are seeing more and more commercial devices with sophisticated front-end interfaces. Many power supply companies now provide direct Ethernet connections with their own controls and measurement interfaces. This is a place where working closely with our industrial partners is extremely important.

The current wave of inexpensive ARM based development systems [26], such as BeagleBoards [27], Arduino boards [28], and the Raspberry Pi [29], provide what could be very powerful and inexpensive interfaces to other systems. In addition, ARM systems are forming a network of Internet enabled devices, or an Internet of Things [26]. Future accelerator systems will surely consist of an Internet of things.

It is hard to imagine a control system that doesn't have some custom interfaces. To this end the current wave in the use of processors attached to large gate arrays seems a logical path to the future. The main problem with these systems is they require significant engineering and development, and so while the hardware may not be expensive, the engineering time can be. So we see the Open Hardware Repository [30] as a key ingredient towards developing high performance systems that are cost effective. We support this community and hope to see it grow.

Middleware

For RHIC, the key areas of improvement to the accelerator performance were in the development of feedback systems. A highly successful system that utilized an optics model of the accelerator is the RHIC 10 Hz feedback system [14]. As such systems improve, the need to have a more sophisticated optics model will grow, likely moving to particle tracking codes, as opposed to matrix solvers. Online models are imperfect and computationally demanding. Future online models will be more powerful but will need high performance computing systems.

We expect to see more generic middleware layers that rely on communication protocols such as http, that lead to the building of more enterprise (Java EE) style servers along with a wider variety of thin clients (web interfaces, mobile device apps as well as custom UIs)

We also expect to use distributed data storage architectures (like hadoop) that allow for cheaper, more scalable data storage and also allow for faster and more sophisticated data processing at the source.

Application Level Interfaces

What tools do we need to write controls applications for touch pad devices? Such an interface has popular appeal. But there are many other interfaces. Will people want to speak to the equipment?

We expect to see user interfaces that use a wider variety of inputs (mouse, touch, voice), and that are available on a much wider variety of devices including laptops, tablets, phones, and watches.

The main thing that is clear here is that access to the controls will not be from just one kind of system. We need to plan on using multiple high-level interfaces as well as low level interfaces from many different operating systems.

The Rest of It

The timing, beam synchronization, and real time data links remain good ideas that will be used for eRHIC, although we plan for these systems to become more sophisticated.

Linux has become the industrial operating system for our business and it seems this will remain true into the future. A challenge to system administration is managing the many flavors and hardware platforms that need to be supported. Better tools to perform system administration are dearly needed, especially when we consider keeping all the various flavors of Linux updated and keeping our software working properly with these updates.

PROJECTING TO ERHIC

Between 2002 and 2013 the number of front end systems for RHIC increased by a factor of 3. The number of control points grew to 1.4 million, the number of lines of computer code grew to 3.76 million, the number of system servers grew to 111, the amount of data stored per year grew to 100 TB, and the maximum bandwidth was increased from 1 Gb/sec to 20 Gb/sec.

If we use the growth over that eleven years to project to what we might expect by 2024, we may find we end up with close to 1000 front end systems, over 3 million control points, up to 10 million lines of code, and up to 500 system servers. The amount of data stored per year could grow to as much as 2 PB. Even with data compression we will need a significant amount of storage space. And what maximum bandwidth might we expect? Current technologies all break the Tb/sec barrier, with test systems getting up to 14 Tb/sec [31, 32].

So, what do we expect an eRHIC control system to look like? We definitely see a greater need for more automation and better tools for building automation into the controls. Feedback systems will continue to grow and make operations more stable and reproducible. Dependencies on models at all levels will increase. There will be a greater use of portable/mobile interfaces. And, we expect to see a greater amount of adaptation and learning being built into the controls infrastructure. Finally, we see a need for more collaboration and industrial partnerships. The complexity of the systems that we use will grow, but our local resources will remain limited.

REFERENCES

[1] RHIC Science web site, http://www.bnl.gov/rhic/
[2] V.N. Litvinenko, et al., arXiv:1109.2819 [physics.acc-ph].
[3] D.S. Barton, et al., Nuclear Instruments and Methods in Physics Research Section A, Vol. 499, 2-3, 2003
[4] T. D'Ottavio, et al., Proc. of ICALEPCS 2007, ROPB01
[5] M. Minty, et al., Proc. of PAC 2011, WEOBN1
[6] M. Minty, et al., Proc. of IPAC 2011, MOPO022
[7] R.J. Michnoff, et al., Proc. of PAC 2011, MOP268
[8] T. Hayes, et al., Proc. of PAC 2011, MOP282
[9] T. Hayes, K. Smith, Proc. of PAC 2011, MOP283
[10] T. Hayes, et al., Proc. of PAC 2011, MOP284
[11] F. Severino, et al., Proc. of PAC 2011, MOP296
[12] F. Severino, et al., these proceedings
[13] K.A. Brown, et al., Proc. of ICALEPCS'11, MOBAUST04
[14] R. Hulsart, et al., Proc. of PAC 2011, MOP267
[15] T. D'Ottavio, et al., Proc. of ICALEPCS 2011, MOMAU002
[16] L. Hoff, J. Skelly, Nuclear Instruments and Methods in Physics Research Section A, Vol. 352, 1-2, 1994
[17] L. Hoff, J. Skelly, Proc. of ICALEPCS 1999, MC1P31
[18] B. Frak, et al., these proceedings
[19] B. Oerter, Proc. of ICALEPCS 1999, MC1P06
[20] H. Hartmann, et al., Proc. of PAC 1999, MOP19
[21] H. Hartmann, et al., Proc. of PAC 1997, 6P024
[22] K.A. Brown, et al., these proceedings.
[23] P. Chitnis, et al., these proceedings.
[24] P. Chitnis, T. Robertazzi, K.A. Brown, these proceedings.
[25] http://en.wikipedia.org/wiki/PageRank
[26] http://www.arm.com/index.php
[27] http://beagleboard.org/
[28] http://arduino.cc/
[29] http://www.raspberrypi.org/
[30] http://www.ohwr.org/
[31] http://phys.org/news/2013-08-intel-event-discuss-tbits-mxc.html
[32] http://arstechnica.com/gadgets/2006/10/7878/

ISBN 978-3-95450-139-7