APPLYING SOPHISTICATED ANALYTICS TO ACCELERATOR DATA AT BNL'S COLLIDER-ACCELERATOR COMPLEX: BRIDGING TO REPOSITORIES, TOOLS OF CHOICE, AND APPLICATIONS *

K.A. Brown†, P. Chitnis, T. D'Ottavio, J. Morris, S. Nemesure, S. Perez, D. Thomas
Collider-Accelerator Department., BNL, Upton, NY

Abstract

Analysis of accelerator data has traditionally been done using custom tools, either developed locally or at other laboratories. Much of the data analysis is done in real time when the data is being logged. However, sometimes users wish to apply improved algorithms, look for data correlations, or perform more sophisticated analysis. In recent years we have investigated the use of tools to bridge standard analysis systems, such as MATLAB®, R, or SciPy, to the controls data repositories. In this report we will discuss the tools used to extract data from the repositories, tools used to bridge the repositories to standard analysis systems, and directions we are considering for the future.

INTRODUCTION

Historically our data repositories have not been that large or complex. Over time these repositories have grown and the kinds of data have become more complex. Today it is not unusual to be working with multidimensional arrays or images, as well as simple scalar data. But more significantly, we store far more data than we analyze [1]. If we want to mine that data and look for particular patterns or do complex correlations, our legacy tools are inadequate.

The development of more sophisticated data collection and analysis systems started after 2000, once the Relativistic Heavy Ion Collider (RHIC), within the Collider-Accelerator Complex (C-AD) at BNL, became operational [2]. The new systems that were put in place standardized the data formats across all data sources and placed the data into well-defined filesystem hierarchies. General tools were built to access all data in these repositories.

The operation of RHIC and associated accelerator systems requires comprehensive data logging systems, collecting and storing measurements from many physical systems. Analysis of this data is a critical part of any troubleshooting process and involves the detection and study of composite behavior patterns. Since most of the tools in place to do this are custom built applications with limited functionality, bringing in other systems, such as MATLAB® or R, allows more sophisticated analytics to be applied to the data. However, the main issue isn't so much getting systems that can do sophisticated analytics, but easily linking such systems to the data that has been stored.

* Work performed under Contract Number DE-SC0012704 with the auspices of the US Department of Energy.
† kbrown@bnl.gov

Data Collection and Storage

One step towards making use of other analysis tools is to get the data already collected and stored to be available to any application. To do this we make use of HTTP protocol multithreaded data servers [3].

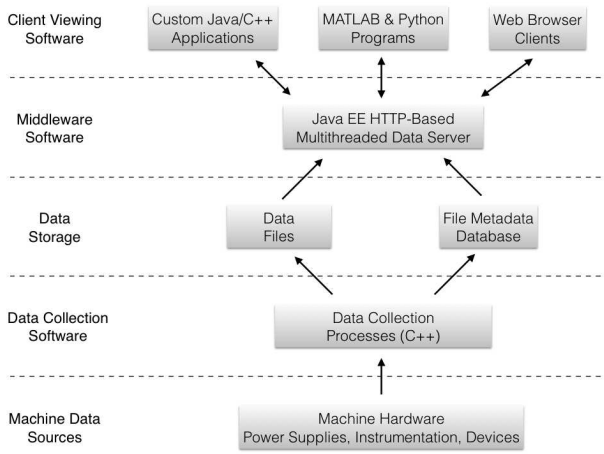

Figure 1: Three tier data layers for Accelerator Data.

Figure 1 shows how data gets collected and stored and then accessed using a data server. All the systems used at the *Machine Data Sources*, *Data Collection Software*, and *Data Storage* layers are legacy systems, for the most part, and serve the purpose of writing data into the filesystem. The *Data Server* provides a simple http-based interface to that data. We also have library level tools that provide this access and tools for obtaining live data.

Data Formats

The RHIC file storage and retrieval systems were designed to make use of the Self Describing Data Sets (SDDS) system developed at Argonne [4]. Using this file format a directory tree hierarchy was built, using a (standard relational) database to facilitate retrieval.

Today, the SDDS format remains the system used to store data, but that choice in format is now a constraint to new design choices. This is an issue in places where performance improvements need to be made. A lesson we seem to keep learning is that the opportunities of one era become the liabilities of a later era. Every custom design eventually becomes either a maintenance burden or a hurdle to progress [5].

ISBN 978-3-95450-148-9

MODERN TOOLS

The growth of analytics in industry opens opportunities for improving the mining and analysis of accelerator data. However, accelerator and scientific data can possess features that present challenges in using some of these tools. In the accelerator environment there is often significant support for in-situ processing, combining metadata and indexing with large-scale time series and multidimensional arrays. This is one of the challenges towards deploying data crawling techniques, for example, where complex correlations need to be performed.

In this section we will briefly discuss a few potentially beneficial technologies, including some that are in use in accelerator environments.

HDF5

The Hierarchical Data Format (HDF) [6] was designed to be an architecture-independent library and file format for scientific data. Today it is extensively used in many areas of both science and industry.

HDF is not used at C-AD, but we are researching ways to take advantage of this technology. HDF5 is particularly appealing since it is specifically designed to help manage extremely large and complex data sets, implementing an API with C++ and Java interfaces.

The advantages of HDF5 are the versatility of the data model, the performance features that provide access time and space optimizations, and the tools for managing, manipulating, viewing, and analyzing data.

Distributed Databases

Distributed databases are Hadoop [7] related or inspired systems such as Cassandra [8] and MongoDB [9]. These systems allow data to be stored across multiple computers systems where data locality is spread over all systems in the database. The presentation of the data is independent of the actual locality of the data. The data can be spread out over multiple networks, can be joined and updated from multiple tables on different systems, and by distributing the data over multiple systems provides high performance, high data integrity access to large data sets.

We are not using distributed databases in our systems, but we are researching these technologies.

Many laboratories are moving towards distributed database systems using high-performance asynchronous messaging to build scalable high performance data streaming to repositories [10–14]. Parallel computing in the controls network is not common, but laboratories are finding they need to exploit parallelism to gain performance, such as at Spring8 and RHIC [15, 16]. Combining parallel computing technologies with distributed database technologies is a natural extension of both ideas. We are researching such systems, for example Apache Spark [17] bridges these two technologies to gain high performance in data mining and analysis [18].

MATLAB®

MATLAB® [19] is used in our systems largely for offline data processing. Although it is an environment for doing simulations, data analysis, data visualization, as well as providing an interface to other devices and instruments, we mostly make use of the analysis and visualization features in the controls system.

In the MATLAB® environment, scripts are a series of statements that operate on data in the workspace. They do not accept arguments. Functions accept input arguments and return output values. Internal variables are local to the function. Using scripts and functions we can build fairly well featured interfaces. However, MATLAB® is intended to be used from its own environment, so it is not ideal for building full featured applications.

What makes MATLAB® particularly powerful is its extensibility. There are many Toolboxes, as these extensions are called, from which to select. Toolboxes don't just provide greater capabilities, they also allow more complex tasks to be done more easily.

R

R is a free software project aimed at statistical computing and visualization [20]. In some respects much of what can be done in R can be done in MATLAB®. However, there are some use cases where R is more efficient. R also provides the capability to build scripts to simplify the access to the data through the server interface. We have used R only for a small number of off-line studies.

With R we can bring advanced statistical models into accelerator data analysis. Since R is an environment and a scripting language it is also a way of fast prototyping new methodologies in analysis.

There also exists an open source graphical environment for R, RStudio [21].

SciPy

Python [22] is quickly growing in popularity at C-AD. This is partly due to the ability to rapidly prototype new software, but also because Python is highly extensible with a very active user community. One very popular module distribution (a collection of python modules) is SciPy [23], an open source collection of tools for math, science, and engineering.

Much of what can be done by MATLAB® and in R can be done from SciPy. The distribution also includes the h5py package, a Python interface to HDF5. Many accelerator applications are being developed using Python and SciPy at C-AD. For access to the controls data, the HTTP interface data server is used, as well as control systems functions for accessing live data.

Since Python is meant to be a scripting language, as opposed to a scripting environment, full-featured applications can be built. In addition, the community puts significant focus on performance. This makes Python much more com-

ISBN 978-3-95450-148-9

petitive as a resource for building true controls applications, often replacing Java or C++ as the tool of choice.

EXAMPLES

MATLAB® Controls Data Viewer

The data viewer for the C-AD controls data is a custom built C++ application, called LogView. To offer improved offline analysis to the data we wrote a collection of MATLAB® functions and scripts. These scripts provide an interface to select and display data from the data server using simple menu selection tools. The key to making this tool useful is to package the data in such a way that end users can apply other MATLAB® functions to the data that was selected and displayed (Fig. 2).

The data servers are http protocol based servers. Requests are sent through http messages. The data returned is contained in an xml formatted message. In principle, one can send requests and get data through a normal web browser interface. To do this within the MATLAB® environment we can use the "urlread()" and "urlwrite()" functions. However, these are not the most convenient or efficient functions to use when the data returned is in xml format. Fortunately, there exists a useful function, called "xmlreadstring()", which returns a Document Object Model (DOM) node. Processing the data is simply a matter of working with the DOM package methods. These methods are well documented on the Oracle Java site [24].

To make the data selected for viewing useful it needs to have names the users understand. In MATLAB® all variables are arrays. MATLAB® uses data containers called cell arrays, or cells, to represent variables. Cells have the ability to hold complex representations of various data sets. But to have named variables, MATLAB® provides Structures, which are Cells where each field is named.

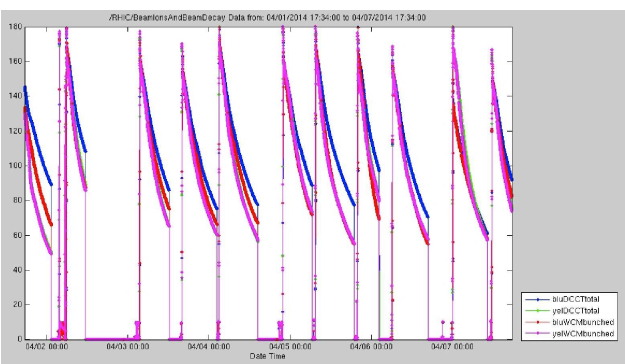

Figure 2: Data collected from the MATLAB® data viewer.

Analyzing Machine Protection Data

The Beam Permit System (BPS) of RHIC plays a key role in safeguarding against the faults occurring in the collider. Over the course of its 15 years of operation, RHIC has accumulated operational failure data. By integration of earlier reliability calculations with operational failure data using Bayesian analysis, the BPS reliability can be quantified using a two-parameter Weibull survival model, with unknown scale and shape parameters. As the joint posterior distribution (Fig. 3) for Weibull with both parameters unknown is analytically intractable, the Markov Chain Monte Carlo methodology with Metropolis-Hastings algorithm is used to obtain the inference [25].

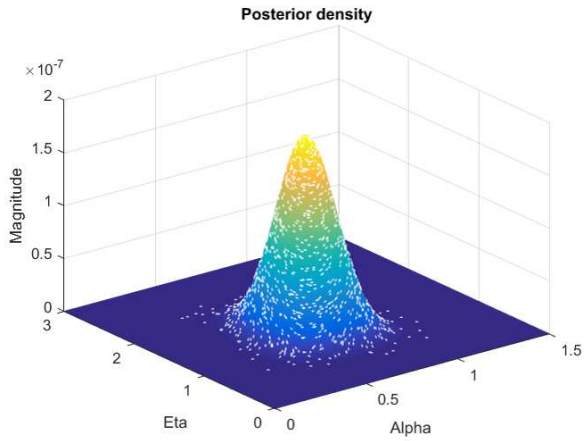

Figure 3: Using MATLAB® to apply advanced statistical techniques to quantify the reliability of the RHIC BPS. Shown here is the posterior density distribution from a Bayesian model of BPS failure.

Developing Better Quench Detector Tables

Due to the limitations of the electrical model used in the RHIC quench detector systems, many false quench events are generated that affect the RHIC availability. The nonlinear electrical behavior of the super conducting magnets was analyzed. The new models include eddy current components and parasitic capacitances, as opposed to simple inductance and lead resistance in the older model. Many data cleaning techniques were employed to reduce the noise in the observed data. Piecewise regression was used to examine the saturation effects in magnet inductance (Fig. 4). The goodness-of-fit of the models was assessed by comprehensive residual analysis. This analysis was done using both R and MATLAB® [26].

Neural Network and Markov Process Models

For this work we have been using NumPy [27] along with SciPy. For neural network analysis (reinforcement learning in the form of a Markov decision process) we use the optimize library for gradient descent to minimize cost functions; specifically the Broyden-Flectcher-Goldfarb-Shanno algorithm (BFGS). This is an interactive method for solving unconstrained non-linear optimization problems, where the Hessian doesn't need to be evaluated directly. In SciPy we can use Levenberg-Marquardt, for doing back-propagation.

Inspection of past data allows development of a model that will provide a prediction of future behavior. This can constantly be adapted as new data is acquired. If something

ISBN 978-3-95450-148-9

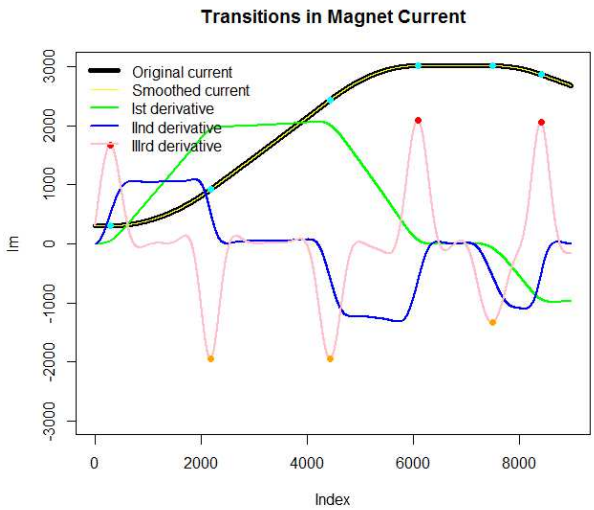

Figure 4: Segmenting data using R to apply different models for deriving superconducting magnet inductance.

changes in all the processes that influence the data being analyzed, then the prediction will deviate. Pattern recognition of past deviations could be used to determine off normal events (Fig. 5).

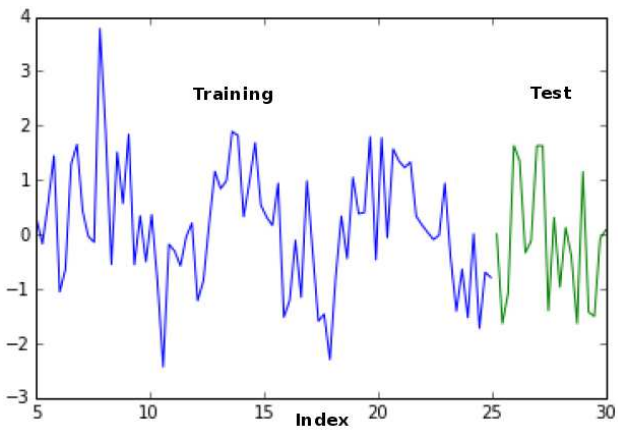

Figure 5: An example of predictive analysis using a neural network derived model of past data.

FUTURE EFFORTS

The ability to efficiently mine and crawl through data to find aberrations and complex correlations remains a future effort. The architecture of the underlying data repository systems needs to be evaluated and methods to mine and crawl the data need to be developed. The classic approaches to data mining of defining pattern associations, path and sequencing analysis, and classification of patterns can be adapted and we could leverage into using existing tools from open source projects or industry (e.g., rapidminer [28], WEKA [29], R [20], Orange [30], KNIME [31], NLTK [32]). These include predictive analytics, machine learning, Bayesian and Markov models, informatics, and natural language processing.

REFERENCES

[1] K. A. Brown et al., ICALEPCS 2013, MOMIB03, http://jacow.org

[2] J. Morris, T. D'Ottavio, ICALEPCS 2001, TUAT002, http://jacow.org

[3] T. D'Ottavio et al., ICALEPCS 2011, MOMAU002, http://jacow.org

[4] M. Borland, PAC 1995, WAE11, http://jacow.org

[5] J. Skelly, J. Morris, ICALEPCS 2001, WC1P05, http://jacow.org

[6] https://www.hdfgroup.org/

[7] https://hadoop.apache.org/

[8] http://cassandra.apache.org/

[9] https://www.mongodb.org/

[10] M. Kago, A. Yamashita, ICALEPCS 2013, TUMIB06, http://jacow.org

[11] S. Marsching, ICALEPCS 2013, MOPPC099, http://jacow.org

[12] N. Kikuzawa et al., ICALEPCS 2013, TUPPC017, http://jacow.org

[13] K. Fuchsberger et al., ICALEPCS 2013, TUPPC026, http://jacow.org

[14] T. C. Shen et al., ICALEPCS 2013, WECOBA06, http://jacow.org

[15] M. Ishii et al., ICALEPCS 2013, MOPPC128, http://jacow.org

[16] B. Frak et al., ICALEPCS 2013, MOPPC157, http://jacow.org

[17] https://spark.apache.org/

[18] N. Malitsky, these proceedings, ICALEPCS 2015, WEPGF058, http://jacow.org

[19] http://www.mathworks.com/products/matlab/

[20] https://www.r-project.org/

[21] https://www.rstudio.com/

[22] https://www.python.org/

[23] http://www.scipy.org/

[24] http://docs.oracle.com/javase/6/docs/api/org/w3c/dom/package-summary.html

[25] P. Chitnis et al., these proceedings, ICALEPCS 2015, MOD3I01, http://jacow.org

[26] P. Chitnis et al., these proceedings, ICALEPCS 2015, MOM310, http://jacow.org

[27] http://www.numpy.org/

[28] https://rapidminer.com/

[29] Mark Hall et al., SIGKDD Explorations, Vol. 11, Issue 1 http://sourceforge.net/projects/weka/

[30] http://orange.biolab.si/

[31] https://www.knime.org/

[32] http://www.nltk.org/

ISBN 978-3-95450-148-9