TRACKING ACCELERATOR SETTINGS*

Ted D'Ottavio[#], Wenge Fu, Daniel P. Ottavio, BNL, Upton, NY 11973, U.S.A

Abstract

Recording setting changes within an accelerator facility provides information that can be used to answer questions about when, why, and how changes were made to some accelerator system. This can be very useful during normal operations, but can also aid with security concerns and in detecting unusual software behavior. The Set History System (SHS) is a new client/server system developed at the Collider-Accelerator Department of Brookhaven National Laboratory to provide these capabilities. The SHS has been operational for over two years and currently stores about 100K settings per day into a commercial database management system. The SHS system consists of a server written in Java, client tools written in both Java and C++, and a web interface for querying the database of setting changes. The design of the SHS focuses on performance, portability, and a minimal impact on database resources. In this paper, we present an overview of the system design along with benchmark results showing the performance and reliability of the SHS over the last year.

BACKGROUND

Device settings lie at the heart of any control system. Most modern control systems contain infrastructure to save and restore settings and to log setting values at requested intervals. This infrastructure, however, often cannot answer some of the questions that arise when many people are responsible for the operations of a control system. Some of these questions are:

- A failure or some other event happened at a particular time. What changed at that time?
- A setting currently has an unexpected value. How (who, when, from where) did this happen?
- A particular setting is not archived or logged. How has its value changed over time?

The ability to answer these questions provides some obvious benefits to the people responsible for operating the control system. In addition, this ability also provides a way to satisfy security concerns regarding malicious behavior.

For all of the above reasons, the Controls group within the Collider-Accelerator Department at Brookhaven National Laborartory (BNL) began work in 2005 on a new system called the Set History System (SHS). The system was commissioned in 2006 and became fully operational in early 2007.

SYSTEM DESCRIPTION

The design of the Set History System was based on the following set of requirements:

- Data accumulated for each device setting should contain information about the value being set, the time of the setting, the application making the setting, the user of the application, and the machine where the application is running.
- The data should be available for viewing within a few minutes of the time of the set.
- The system should be able to handle one million sets per day without serious resource issues.
- The time to perform the set by the client should not be increased by more than 10%.
- A user interface should be available for answering the above questions with query response times of a few seconds or less.

The system constructed to satisfy these requirements is shown the diagram below (Figure 1).

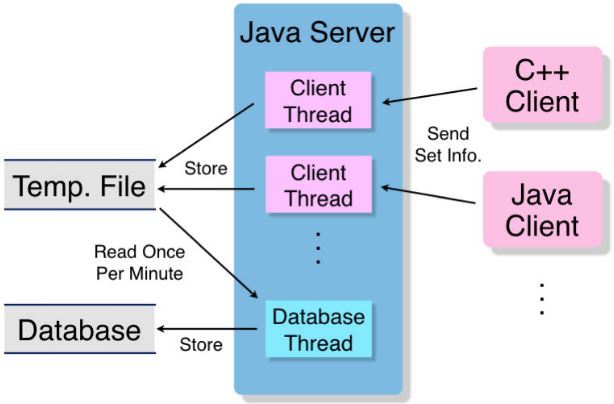


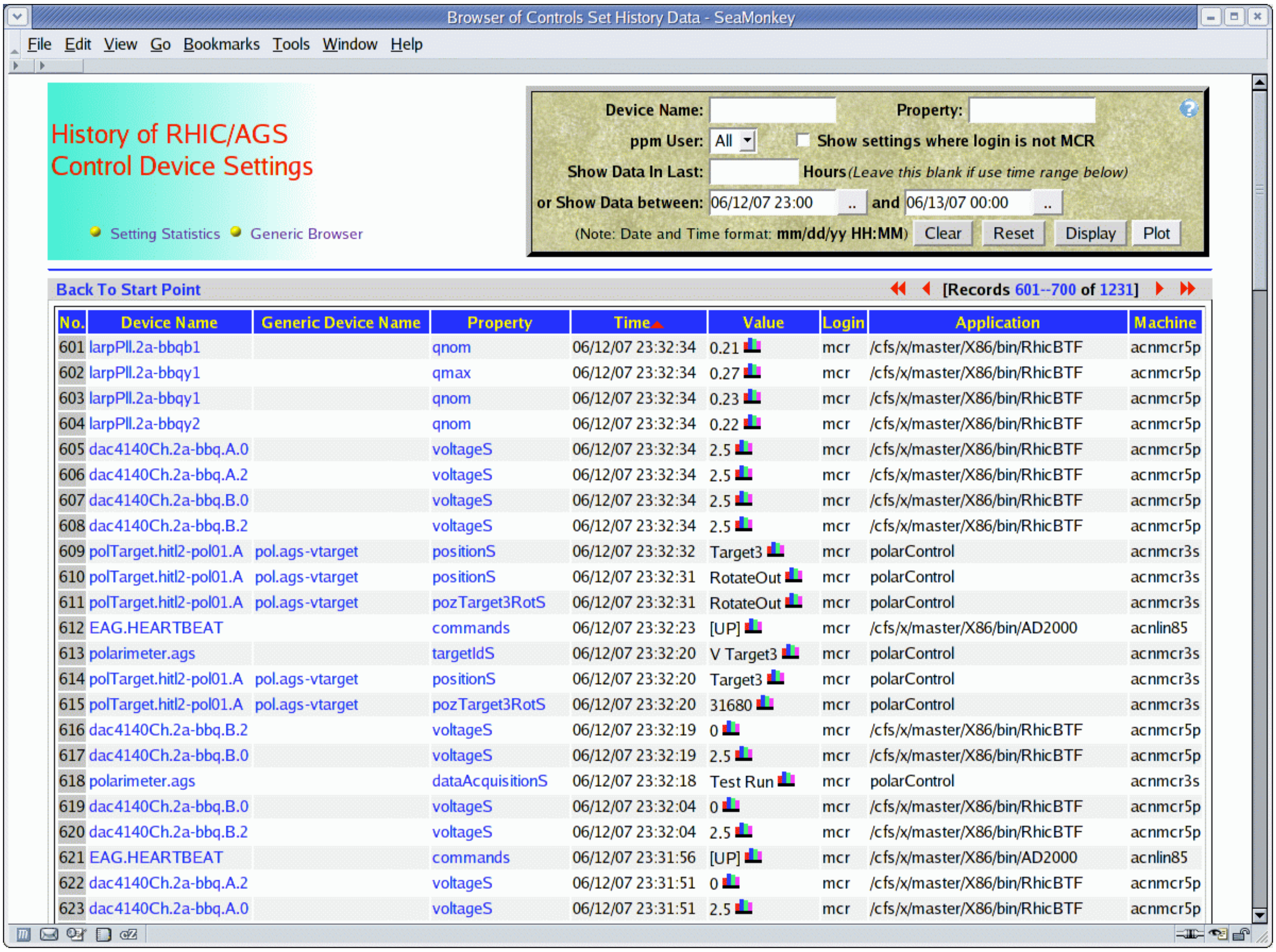
Figure 1: Diagram of the Set History System.

A client-server model is used. Communication takes place via sockets using a custom XML protocol. Client tools were constructed for both C++ and java application clients. In order to minimize the impact of the set storage on the performance of the actual set, the communication takes place in a separate thread within the client that is dedicated to sending the set information to the server.

The heart of the system is the Set History Server. Written in java, it uses a pool of client threads. Each thread is responsible for retrieving data from a client and storing it in a temporary file. A separate thread then reads the file once per minute and dumps the accumulated records into a commercial Sybase database management system. Here a batch insertion command was found to greatly reduce the impact on the database server.

Queries to the SHS database by users are made using an HTML browser. A custom web query page was developed for this purpose and is shown in Figure 2. Users can request records that match a particular device/property name or look for setting changes made at a particular time. Clicking on a bar chart link in the Value column causes the data for that device/property to be viewed in a stripchart display.

* Work performed under the auspices of the U.S. Department of Energy. [#] dottavio@bnl.gov



Device Name: Property:

ppm User: All □ Show settings where login is not MCR

Show Data In Last: Hours *(Leave this blank if use time range below)*

or Show Data between: 06/12/07 23:00 and 06/13/07 00:00

(Note: Date and Time format: **mm/dd/yy HH:MM**)

[Records 601--700 of 1231]

No.	Device Name	Generic Device Name	Property	Time	Value	Login	Application	Machine
601	larpPll.2a-bbqb1		qnom	06/12/07 23:32:34	0.21	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
602	larpPll.2a-bbqy1		qmax	06/12/07 23:32:34	0.27	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
603	larpPll.2a-bbqy1		qnom	06/12/07 23:32:34	0.23	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
604	larpPll.2a-bbqy2		qnom	06/12/07 23:32:34	0.22	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
605	dac4140Ch.2a-bbq.A.0		voltageS	06/12/07 23:32:34	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
606	dac4140Ch.2a-bbq.A.2		voltageS	06/12/07 23:32:34	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
607	dac4140Ch.2a-bbq.B.0		voltageS	06/12/07 23:32:34	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
608	dac4140Ch.2a-bbq.B.2		voltageS	06/12/07 23:32:34	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
609	polTarget.hitl2-pol01.A	pol.ags-vtarget	positionS	06/12/07 23:32:32	Target3	mcr	polarControl	acnmcr3s
610	polTarget.hitl2-pol01.A	pol.ags-vtarget	positionS	06/12/07 23:32:31	RotateOut	mcr	polarControl	acnmcr3s
611	polTarget.hitl2-pol01.A	pol.ags-vtarget	pozTarget3RotS	06/12/07 23:32:31	RotateOut	mcr	polarControl	acnmcr3s
612	EAG.HEARTBEAT		commands	06/12/07 23:32:23	[UP]	mcr	/cfs/x/master/X86/bin/AD2000	acnlin85
613	polarimeter.ags		targetIdS	06/12/07 23:32:20	V Target3	mcr	polarControl	acnmcr3s
614	polTarget.hitl2-pol01.A	pol.ags-vtarget	positionS	06/12/07 23:32:20	Target3	mcr	polarControl	acnmcr3s
615	polTarget.hitl2-pol01.A	pol.ags-vtarget	pozTarget3RotS	06/12/07 23:32:20	31680	mcr	polarControl	acnmcr3s
616	dac4140Ch.2a-bbq.B.2		voltageS	06/12/07 23:32:19	0	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
617	dac4140Ch.2a-bbq.B.0		voltageS	06/12/07 23:32:19	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
618	polarimeter.ags		dataAcquisitionS	06/12/07 23:32:18	Test Run	mcr	polarControl	acnmcr3s
619	dac4140Ch.2a-bbq.B.0		voltageS	06/12/07 23:32:04	0	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
620	dac4140Ch.2a-bbq.B.2		voltageS	06/12/07 23:32:04	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
621	EAG.HEARTBEAT		commands	06/12/07 23:31:56	[UP]	mcr	/cfs/x/master/X86/bin/AD2000	acnlin85
622	dac4140Ch.2a-bbq.A.2		voltageS	06/12/07 23:31:51	0	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p
623	dac4140Ch.2a-bbq.A.0		voltageS	06/12/07 23:31:51	2.5	mcr	/cfs/x/master/X86/bin/RhicBTF	acnmcr5p

Figure 2: Custom web query page used to search the set history database to answer commonly asked questions.

RESULTS AND DISCUSSION

The Set History System went through a significant commissioning period during 2006. Performance was a particular concern because of the large number of setting records to be stored each day. Tests at the client level showed a minimal impact on the time to perform a set (a < 1% slowdown). We attribute this primarily to threaded design used to send information to the Set History Server.

A number of tests were also done to determine the performance of the server. We were able to see significant performance gains within the server when the software was changed from creating a new thread for each client transaction to reusing a thread from a rotating thread pool. With this software in place, the server was able to collect one million sets per day with a < 1% CPU utilization rate on a 1 GHz Intel X86 single CPU machine running Redhat Linux. Further stress tests showed that 10 clients sending 40 million sets per day to the server would raise the CPU usage to about 20%.

More serious performance problems occurred at the database server level. Attempting to insert records, one at a time, at the one million records per day rate, caused our Sybase database server machine (a dual-CPU 1 GHz Sun SPARC running Solaris 8) to show CPU utilization rates of 50-60%. We were able to reduce this usage by a factor of ten, however, by using an alternative record insertion method available on Sybase servers called bulk copy or bcp. With this method, record information is accumulated in a file and then sent to the server using the Sybase bcp system call. The Set History Server supports this method of database insertion. Further tests showed that database insertion using bcp could achieve rates of 10 million sets per day at a CPU utilization rate of about 25%. A secondary benefit of this method is that the intermediate file buffers data and keeps the Set History System working through temporary problems with the database.

The Set History System was gradually made operational in the latter half of 2006 through the rebuild and release of individual applications and servers. Initially all applications and servers participated in the system and insertion rates grew to about the one million sets per day figure that had been predicted. However, it was quickly noted that a few programs dominated the insertion of records into the setting database and, further, that these programs were setting the same values repetitively. Further investigation revealed either programming errors or situations for which there was no interest in storing these values. These cases were either fixed or eliminated from the Set History System by changing selected clients to not participate in the automatic storage of settings for some or all of their settings. Subsequently, other cases were removed from the system since they were found to be "polluting" the set history database without any obvious benefit. After a few months, the system settled down to storing a few hundred thousand sets per day. A chart showing the number of sets per day stored by Set History System over a 6 month time period in the first half of 2007 is shown in Figure 3.

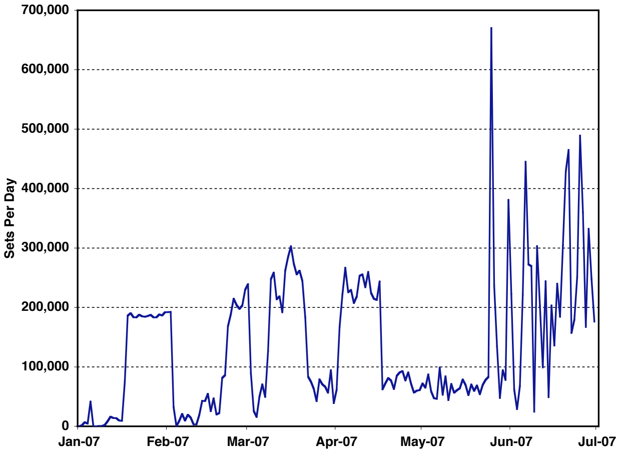


Figure 3: Number of sets collected each day in 2007.

The database query tool shown in Figure 2 is the normal way that information is retrieved from the set history database. Web logs indicate that this page is accessed for information several times per week and anecdotal evidence suggests that a significant question is answered about once a week that couldn't have been answered without this system in place.

Reliability of the Set History System has been excellent, with no significant downtime periods since the system became operational almost one year ago. Note that 100% reliability was never an important goal when designing this system because of the low probability that any particular setting record will be needed to answer commonly asked questions.

CONCLUSIONS

The Set History System is a new and important system that has recently been added to the diagnostic tools available to the Operations, Controls and Physics personnel responsible for operating the particle accelerator complex at BNL. It has been able to store important setting information at a rate of several hundred thousand sets per day with good performance and reliability and with no noticeable impact on application performance.

The most significant resource impact of this system has been on our database system. The set history database has quickly become our biggest database in terms of number of records stored and we have decided to make a few hardware changes to accommodate the increased load and storage requirements.

Judging by the number of queries to the system and by the response of its users, the SHS is answering important operational questions that would have been difficult or impossible to answer beforehand.