# **BEPCII EPICS DATA ARCHIVING AND RETRIEVING SYSTEM**

Zhao Zhuo, Liu Shu, Wang Chunhong, IHEP, CHINA.

### Abstract

IHEP has installed the BEPCII (Beijing Electron Positron Collider II) project that has been funded by the Chinese Government in order to further upgrade the BEPC which has operated since 1989. As a part of this updating project, control system is decided to adopt the Experimental Physics and Industrial Control System, EPICS. The Channel Archiver is an archiving toolset for EPICS. It can archive any EPICS channel value that is available via ChannelAccess (CA), the EPICS network protocol. The paper will introduce how to establish EPICS data archiving and retrieving system in BEPCII. It will also talk about how to configure and install Channel Archiver and Archive Viewer software framework based on HP Server and Disk Array platform.

## **INTRODUCTION**

BEPCII distributed control system is based on many EPICS IOCs which are connected though Ethernet and a lot of EPICS PVs are running on. Every EPICS PV means one control channel, we often need save these channels' value as history data in order to retrieve and analyze them in case of some problem. As BEPCII control system running stably, EPICS control subsystem such as Power Supply, Vaccum, Beam Measurement will produce plenty of new data. We should have the data be saved whenever we wanted search them. Channel Archiver is a very suitable and useful tool which can help store any EPICS channel's value and timestamp, furthermore, retrieve specified channel's value according to the timestamp.

# **ARCHIVER SERVER BUILDING**

We usually use "Archiver Server" refer to BEPCII EPICS Data Archiving and Retrieving System server. We choose HP server and disk array as hardware platform of Archiver Server and installed RedHat Linux operating system and Channel Archiver toolkits as software environment of Archiver Server.

# Hardware Platform

We configured the archiver server two same separated 146G SATA disks and one hardware RAID controller card in order to make RAID1 for the two disks as mirrored operating system drive. RAID1 means writes two copies of data simultaneously. If one of the mirrored drives suffers a mechanical failure or does not respond, the remaining drive will continue to function. A duplicated set of data will supply higher fault tolerance performance and higher system security and reliability.

We configured the DiskArray with 12 pieces of 750G SATA disks and one hardware RAID controller so that we can make RAID5 plus Hot Spare for all the drives, then at last the data storage capacity is about 7T. RAID5 plus Hot Spare can solve loss of the data because of a second drive

failure in the same RAID5 redundancy group before the administrator discovered the drive damage and rebuilding it. This reduces the mean time to recovery (MTTR) validly.

## Software Environment

We installed RedHat Enterprise Linux WS4 operating system on the two made RAID1 drives. We also installed the Third Party tools and EPICS software including:

- JRE1.5
- EPICS BaseR3.14.7
- EPICS Extensions, Channel Archiver,
- Archiver Viewer etc

## **ARCHIVER SERVER APPLICATION**

Archiver Server can help us to store and retrieve history data by channel archiver and archiver viewer. Three tools we used will be discussed in the following.

#### Archiver Engine

After setting up EPICS Channel Archiver, we can start Archiver Engine to store data. First, we configure ".xml" files, and then initiate the Channel Archiver Engine process. When the Engine is running, the value and timestamp of EPICS channels will be automatically stored.

For example, we make a new directory named "PS" for the sake of saving EPICS channel values of Power supply control subsystem. In the directory, we configure a new file named "engineconfigure.xml" as the XML format file. We list channels need to be saved in the file, and then we run the command "ArchiveEngine 4831 -p engineconfigure.xml index" and start the ArchiveEngine process. Here "-p" is configured as a TCP port, and "index" is a default archiving index file. When the ArchiveEngine is initiated, one binary format data file at most 30M such as 20081120-1 will be saved in the directory. We access website can the "http://server1:4831" by firefox browser to look up the running state of ArchiveEngine, on this web browser window shown in Figure 1, we can also check the configuration of ArchiveEngine or stop it.

Edit (wie Co Bookmask Tods Help        • • • • • • • • • • • • • • •		Archive Engine - Mozilla Firefox	
Red Hat New Rok   Sport   Shop   Products   Training     Archive Evaluation   Skinov   Skinov   Skinov   Skinov     Verdion   2 ki. j. buik Jul 12 2006, 22 25:16   Skinov   Skinov   Skinov     Observation   Skinov   Skinov   Skinov   Skinov     Archive Infex   Off1/12009 152:855 [41997000   Skinov   Skinov     Archive Infex   Index   Skinov   Skinov     Channels   5477   Skinov   Skinov     Verdio   3665   Skinov   Skinov   Skinov     Verdio Verdio   0.162 sec   Skinov   Skinov   Skinov     Ordio Verdio Verdio   No   Skinov   Skinov   Skinov   Skinov     Verdio Ve			
Archive Engine   Version 2.8.1, built Jul 12 2006, 22 25:16   Description 2.9.1, built Jul 12 2006, 22 25:16   Description EPICS Channel Archiver Engine   Ostritte 05/11/2009 (95 28:55, 142979000)   Archive Index index   Channels 547   Connected 535   Arg. Pres. Delay 4.262 sec   Write Duration 0.162 sec   Next Wreit ime 07/02/0000 (93:35:00.00000000)   Currently writing No   Write Period 30.0 sec   File Size Linit 30 MB			🖌 🛈 Go 🚺
Verdom 2.8.1, built Juil 2006; 22.25:16   Description EPIC:S Channel Archiver Engine   Started 05/11/2006; 15.28:55.14:297000   Archive Index 547   Connected 535   Arg. Pros. Debay 4.452 sec   Write Count 3563   Orked Vertices 07.01/2000 (97.33:00.0000000   Next write into 01.162 sec   Next write into 07.01/2000 (97.33:00.0000000   Write Period 30.0 sec   File Size Linit 30 MB	Inc. 🗋 Red Hat Netz	ork CSuppor: CShop CProducts CTraining	
Image: Im	chive En	gine	
Version28.1. bail. J 12 2006, 22.25:16DescriptionEPIC:S Channel Archiver EngineStarted05/11/2009 15.28:55.142979000Archive IndexindexChannels547Connected533Arg. Pres. Delay4.362 areWrite Count0.162 areNext write inter07/002009 09:3350.00000000Currently writin30.0 arecWrite Period30.0 arecFile Size Limit30 MB		•	
DescriptionIPICS Channel Archiver EngineOs/11/2009 15:28:55.14:2979000Archive IndexArchive IndexChannelsChannelsConnectolS33Arg. Pres. DelayParcile IndexS665Write DurationOrXourcentiverOrXourcentiverNo.Write PeriodS0.0. secFile Size LimitS00 MB	Archive	Engine Info	
Skried 05/11/2009 15-28:55 142979000   Archive Index index   Channels 5-47   Connected 5-33   Arg. Prec. Delay 4-262 sec   Write Cward 5-665   Next wrieitime 07/01/2000 09-33:00.0000000   Ourrently writing No   Write Period 30.0 sec   Flie Size Limit 30 MB	Version 2.3	:1, built Jul 12 2006, 22:25:16	
Archive Index index   Channelse 5.47   Connected 5.33   Arg. Pres. Delay 4.262 are   Write Count 3663   Write Daration 07.012000 09:33:00.0000000   Currently writing 070.012000 09:33:00.0000000   Write Period 30.0.0 arc   Flie Size Limit 30 MB	Description EF	ICS Channel Archiver Engine	
Channels     547       Connected     533       Arg. Proc. Delay     4.262 sec       Write Count     5665       Write Duration     0.162 sec       Next write ime     070/00000 093-300.00000000       Currently write     No       Write Period     30.0 sec       File Size Limit     30 MB	Started 06	/11/2009 15:28:55.142979000	
Connected     533       Asg. Proc. Delay     4.262 sec       Write Count     5665       Write Daration     0.162 sec       Oxtor write itime     070/002009 09.33-00.00000000       Currently write Period     No       Write Period     30.0 sec       File Size Limit     30 MB	rchive Index	index	
Avg. Prec. Delay     4.262 sec       Write Count     5665       Write Duration     0.162 sec       Next write time     07/0/2009 09-33:00.00000000       Currently writin     No       Write Period     3.0.0 sec       File Size Limit     30 MB	Channels	547	
Write Count     3665       Write Duration     0.162 sec       Next write time     07/0/2009 09:33:00.00000000       Currently writing     No       Write Period     30.0 sec       File Stac Limit     30 MB	Connected	533	
Write Duration 0.162 sec   Next write time 07/20/2009 09:33:30.00000000   Currently writing No   Write Period 30.0 sec   File Stee Limit 30 MB	g. Proc. Delay	4.262 sec	
Next write time     07/20/2009 09:33:30.00000000       Currently writing     No       Write Period     33:0.0000       File Star Limit     30 MB	Write Count	5665	
Currently writing No   Write Period 30.0 sec   File Size Limit 30 MB	rite Duration	0.162 sec	
Write Period     30.0 sec       File Size Limit     30 MB	ext write time 07	/30/2009 09:33:00.000000000	
File Size Limit 30 MB	rrently writing	No	
	Write Period	30.0 sec	
Get Threshold 20.0 are	ile Size Limit	30 MB	
	ct Threshold	20.0 see	
Disconn. on disable No	conn. on disable	No	

Figure 1: The Web Browser Window of Running ArchiveEngine for Storing BEPCII EPICS data.

# Archiver Export

The stored data file is a binary file. We need to export it into a text file that we can read. We can do this by ArchiveExport tool. For example, in the directory including the binary file, we run the command "ArchiveExport -start "2008/11/20 10:00:00" -end "2008/11/20 13:00:00" -o OutFile -f decimal -p 4 index rps:ISet", and a text file named "OutFile" will be produced. And then we can use text view tool such as "less" to read it.

# ArchiveViewer

Another retrieving data method is to use graphical user interface tool ArchiveViewer. ArchiveViewer is a Java Archive Client tool. We can use it to browse the available data, generate plots, and export data to spreadsheets from any computer on the network by accessing archiver data server. It is an open source client application and is written in Java for processing archived EPICS data. The user can run it on many operating systems such as Linux, Windows or Solaris on which JRE1.4 or later is installed. It can produce different kinds of visual time plots output including lines, scatters and steps. Here, we need to install Java Running Environment (JRE1.5), Apache2.2 and ArchiveViewer software packet. And then we need to configure "serverconfig.xml" in the apache cgi-bin path. In the file, we list all index files that are created by ArchiveEngine so that the requests from Archiver Viewer clients know where it can get the data on the Apache server. We can run the command "Archvieviewer -u http://server1/cgi-bin/archiver/ArchiveDataServer.cgi" and get the data via ArchiverViewer clients over the Ethernet. The interactive GUI is shown in Figure 2.



Figure 2: The ArchiveViwer window of BEPCII EPICS data retrieving.

# CONCLUSION

Archiver server has stored the data about 2T for BEPCII control system since it was initiated in Nov 2006. It is very helpful and useful for BEPCII commissioning physicist and hardware engineers to analyze problems and solve device fails etc. It has been a very important and necessary tool to retrieve history data and diagnose system for BEPCII operator.

I would like to thank the EPICS community in Control Group for their support. In particular my colleague Liu Shu who ever did a lot of work in debugging and installation, and Prof. Wang ChunHong who arranged the whole history data archiving and retrieving system framework and suggested to use Archive Viewer client graph tool for the first time.

### REFERENCES

- [1] Bob Dalesio et al: Channel Archiver Manual for R3.14.4 and higher. March 27, 2006.
- [2] Sergei Chevtsov: EPICS ArchiveViewer Version 1.0.4. Oak Ridge National Laboratory. April 7, 2005.
- [3] EPICS Web Page, http://www.aps.anl.gov/epics/.
- [4] Marty Kraimer et al: IOC Application Developer's Guide R3.14.4.