

# A WEB BASED REALTIME MONITOR ON EPICS DATA

L.F. Li<sup>#</sup>, C.H. Wang, IHEP, Beijing, China

## Abstract

Monitoring systems such as EDM and CSS are extremely important in EPICS system. Most of them are based on client/server(C/S). This paper designs and implements a web based realtime monitoring system on EPICS data. This system is based on browser and server(B/S). Through CAJ interface, It fetches EPICS data including beam energy, beam current, lifetime and luminosity and so on. Then all data is displayed in a real time chart in browser (IE or Firefox/Mozilla). The chart is refreshed every regular interval and can be zoomed and adjusted. Also, it provides data tips showing and full screen mode.

## INTRODUCTION

The BEPCII has successfully used EPICS to build up control system. It provides world-widely high energy experiments as well as synchrotron radiation experiments. Because there are many international collaborations, many scientists from outside the IHEP expects to view the status of the BEPCII running. A web based monitoring need to be developed.

In fact, there has been existing monitoring systems on EPICS data including JoiMint, WebCA and CAML.

JoiMint adopts "Java Web Start" to download Java application to local computers and runs it. In order to run Java application, Jre Plug-in or ActiveX is needed to install on client computers.

WebCA is a Web Channel Access client framework based on browser plug-in. It captures EPICS data directly to browsers and displays in text. CAML uses widgets to display EPICS data in graphics. WebCA and CAML can only run in browsers of Firefox/Mozilla.

Since Internet Explore (IE) is popular in China, many people would like to use IE. The EPICS interface to IE is planed, because Internet Explore (IE) uses ActiveX to run web applications while Firefox/Mozilla uses Plug-in. In order to display EPICS data both in IE and Firefox/Mozilla, a web based realtime monitoring system is developed to satisfy the needs of viewing the BEPCII status from all over the word. This system is based on B/S using Flex<sup>[1]</sup>. The Flex, developed by Adobe, is used to design web user interface, but it needs flash player. Normally, most of computers have been installed with flash player. So, the Flex can run both in IE and Firefox/Mozilla.

CAJ<sup>[2]</sup> is used as interface instead of JCA, for CAJ is 100% pure Java CA client library and more stable<sup>[3]</sup>. Through CAJ, Java application in web server can get EPICS data from CA gateways or IOC servers. Blazed<sup>[4]</sup> is used as the bridge between Flex and Java while Json is data format transferred from the web server to the browser.

<sup>#</sup>lilf@ihep.ac.cn

## DESIGN AND IMPLEMENTATION

The web based monitoring system is composed of applications in web server and in browser as shown in the figure 1. The parameters of the BEPCII such as beam energy, beam current, lifetime and luminosity runs on PC/Linux as a SoftIOc in control network. The web server runs on PC/Linux in campus network.

The web based monitoring system works as follows:

- The sampling thread connects CA gateways or IOC servers and keeps CA connections in the server's memory.
- Through CAJ interface, the sampling thread gets EPICS data from CA gateways or IOC servers. Then all the EPICS data is saved in the server's memory.
- When visitors view the web page of the monitoring system, the service thread in web server will return the initial EPICS data from the web server's memory to browsers.
- The browser parses EPICS data from the web server and displays it on the web page in patterns of charts and texts.
- By regular interval (10s,30s or 1minute), the browser sends a request to the web server, gets latest EPICS, and then refreshes the charts and texts.

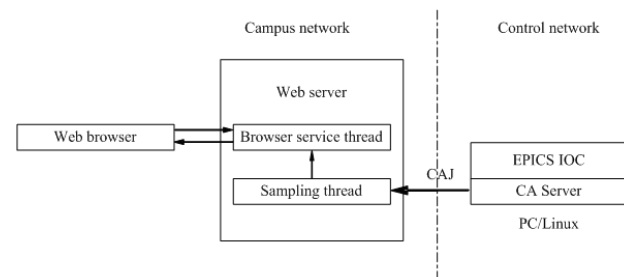


Figure 1: A web based monitoring system of the BEPCII.

## Java Applications in Web Server

Tomcat is used as the web server while log4j generates running logs. The Java applications in the web server mainly contain the sampling thread and the browser service thread.

The work flow of the sampling thread in the web server is illustrated in the figure 3. The sampling thread firstly creates CA connections and keeps all of them in the server's memory. Then it gets EPICS data from IOC servers or CA gateway and saves it in the server's memory (as an "application" of the web server). All procedures above executes every regular interval (10s,30s or 1minute). But there are also some differences between the first time and the other times. In the first time, the

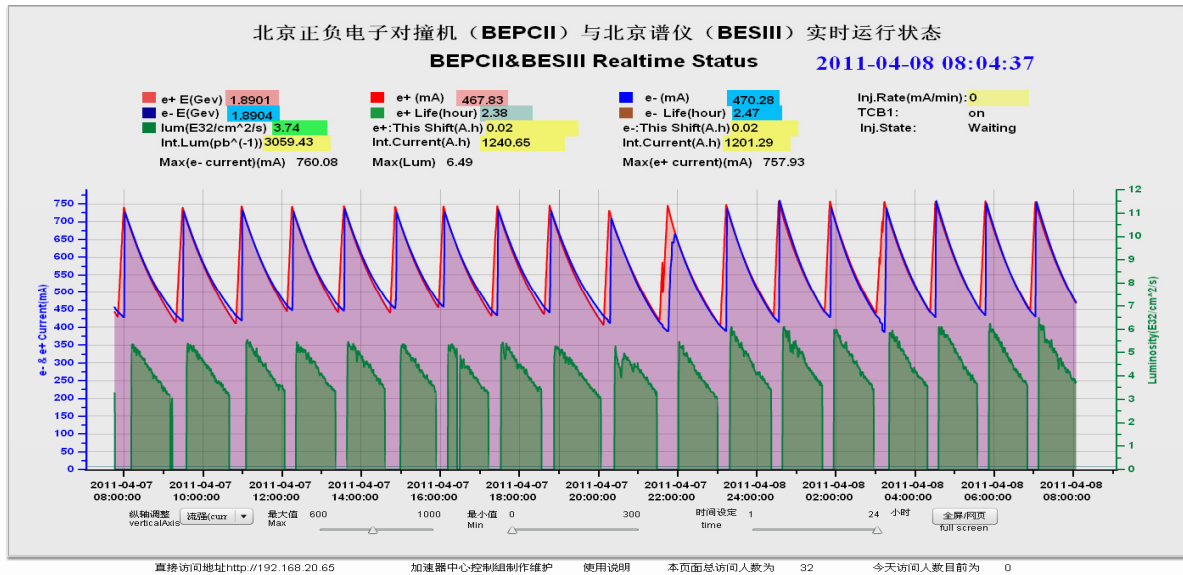


Figure 2: The web page of the status of the BEPCII running.

sampling thread creates CA connections. In the other times, it firstly gets CA connections from server's memory and checks their status. If CA is connected, the thread gets EPICS data directly. Otherwise, it recreates unconnected CA connections and keeps them in server's memory.

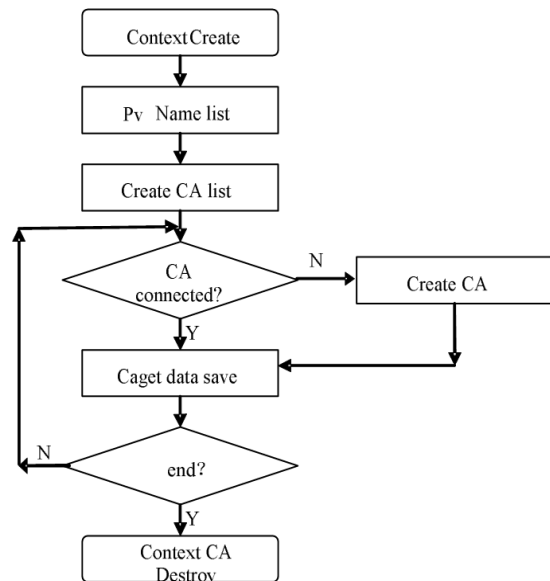


Figure 3: The flow of the sampling thread in web server.

As we know, CA connections take time and CA's UDP broadcasting times by times when recreating CA connections will cause the network traffic. To reduce the time consumed by making CA connections and the network traffic, CA connections should be kept in the server's memory so that the sampling interval will be minimized.

The browser service thread returns EPICS data in format of Json to browser from the server's memory. Then the Flex application in browser displays the data on the web page.

### Flex Applications in Drowser

The parameters of the BEPCII such as beam energy, beam current, machine mode and so on need to be view by the scientists all over the world. So, we use Flex to implement a web page of the status of the BEPCII running as shown in the figure 2.

This monitoring system adopts Flex and BlazeDS to design the web user interface. The BlazeDS is the bridge between Flex and Java.

When visitors view the web page, the browser sends a request to the web server and gets the initial data of 24 hours. Then, by regular interval (10s,30s or 1minute), the browser gets the latest EPICS data from the web server and refreshes the chart and texts.

The chart in the figure 2 can be zoomed, adjusted and provides full screen and data tips showing.

Flex applications in the browser mainly include modules as following:

### Data Initialization Oodule

This module gets data in format of Json from the web server and parses the data which is to initialize the chart.

### Data Tefreshing Oodule

By method of remote object, data refreshing module receives the latest data from the web server and refreshes the chart and texts.

### Timing Oodule

To fresh the data of the chart and texts, a timer is used to call the data refreshing module by every regular interval (10s,30s or 1minute).

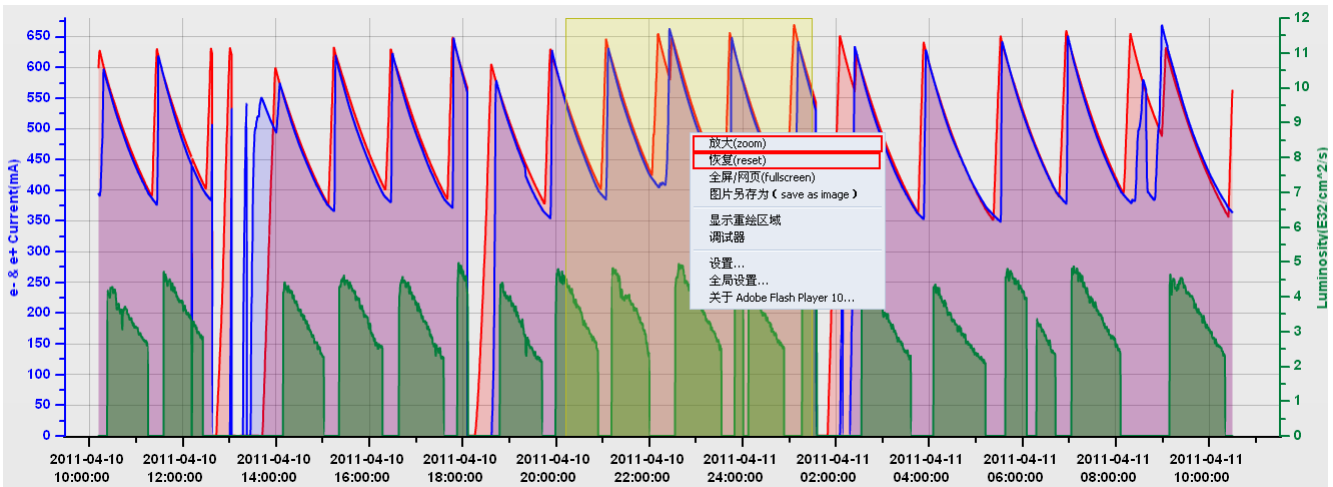


Figure 4: the selected area in the monitoring chart.

### Chart \ zooming Oodule

This module is mainly to zoom the chart to look up more detailed information. When a rectangle area in the chart is selected, the corresponding part of the chart is zoomed(see Fig. 4).

### Full Ucreen Oodule

The web page of the monitoring system can be changed from the web page mode to the full screen mode like movies in media player.

### Data Vips Uhowing Oodule

When the mouse is placed on the line of the chart, a tip will appears, showing the name, value and time of the EPICS data.

### Chart Cdjusting Oodule

By changing the minimum or maximum of axes, the vertical and horizontal axes can be adjusted so that the chart can move up or down, and the total time length of horizontal axes can be decreased.

## PERFORMACE TESTS

A web server of tomcat has been built and running almost 2 months without breaking down.

A simple version of the system without special effects ( chart zooming ,chart adjusting and full screen ) has been available on the web and you can visit the address <http://202.122.32.134:8080/RealTimeLineChart/>.

### Test Gnvironments

- HP Compaq dc7700 (Intel(R) Core(TM)2 CPU 6420 @ 2.13GHz 2G )
- Centos4 linux (synchronizing clock through ntp )
- Tomcat6.0 as web server
- Jdk1.6
- Jprofiler for memory test and thread test

- Apache ab for load test

### Performance

The test shows:

- The chart can run in many kinds of browsers(IE or Firefox/Mozilla)
- No EPICS data is lost(once data is lost, it is recorded in the server's log )
- Minimum sampling time=1s
- Concurrent visitors>100
- Web page response time <3s
- Successive working time of server >2 months
- Low load of web server

## CONCLUSIONS

This system takes BEPCII running parameters including energy, beam current, lifetime and luminosity as example and fetches these EPICS data through CAJ interface. These parameters can be displayed in IE in a real time curves which can be updated automatically. The web page of the monitoring system is designed by Flex and provides some flex functions including chart-zooming, chart-adjusting, data tips-showing and full screen. On the web page, it's easy for users to adjust the time span and zoom the chart. In order to improve the sampling performance, CA connections stays in the web memory. Since this system is thin based on browser/server, it is very easy to deploy and can be used in other projects.

## REFERENCES

- [1] <http://www.adobe.com/products/flex.html>
- [2] <http://epics-jca.sourceforge.net/caj/>
- [3] M Sekoranja " Native Java Implement of channel access for Epics", 10th ICALEPCS, Geneva, Oct 2005, PO2.089-5.
- [4] <http://opensource.adobe.com/wiki/display/blazeds/Blazeds>