

Web service data interface to accelerator control signal data

Tianmin Wan, Geyang Jiang, Liren Shen
Shanghai Synchrotron Radiation Facility, Shanghai, PR China



Introduction

During the running of accelerator, a large amount of signal was archived and monitored. People outside the control network may interested in different parts of the signal data, but our accelerator signal is transport on network by EPICS CA (Experimental Physics and industrial Control System Channel Access) protocol, one needs to install and configure some EPICS CA client to access these signal. In some case, this seems to be difficult and tedious. In our applications, a web server is configured to provide these signal, all of the data is provided in JSON (JavaScript Object Notation) format.

Structure

A c++ program monitor the signal in the IOC server, if the state of these signals has changed, then update the table data in the database. In order to improve the performance of the system, the MySQL memory table is used in database to improve the speed of read and write data, Django framework based web server is configured to provide data to internet. The Pv monitor, database and webserver is running on separate ubuntu server.

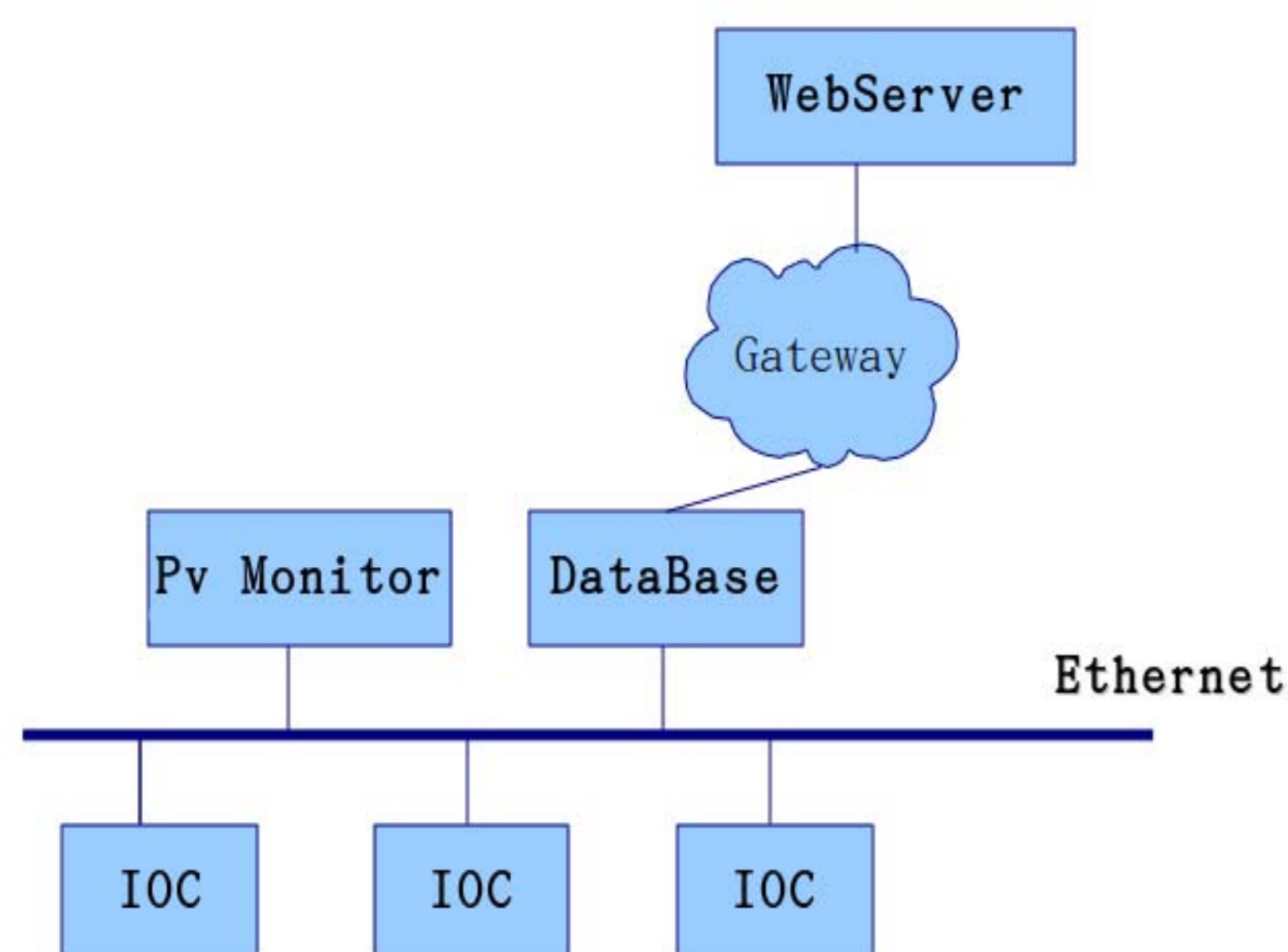


Figure 1. Structure of Data Interface application to Accelerator control data.

Data format

The data interface provide the value, timestamp, stat and severity field of the signal. these data field is provide in JSON format.

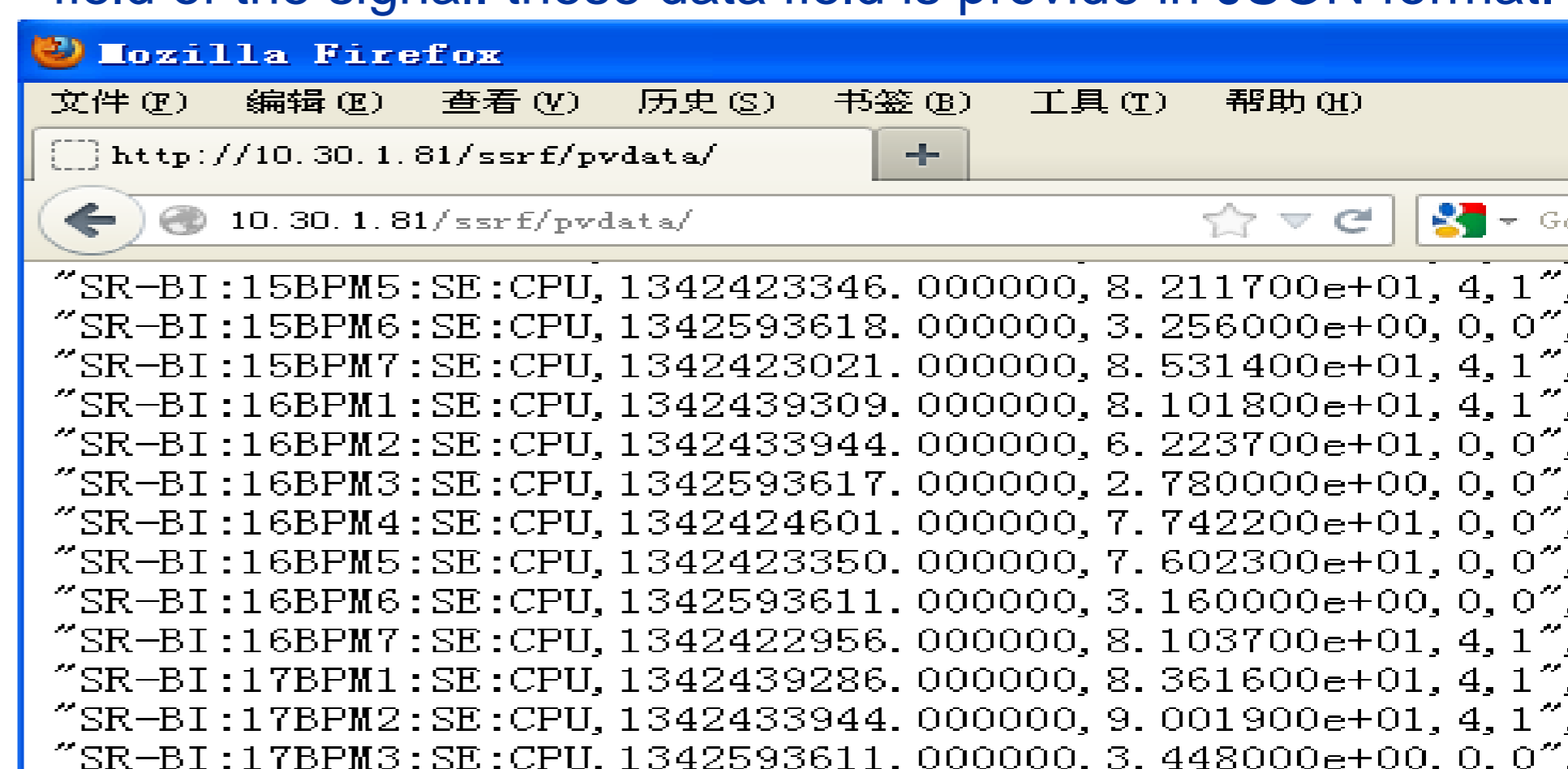


Figure 2. data format of data interface

Performance test

During these test, we use the curl command to fetch data from this data interface, we test for fetch all the data in data Interface (about 9000 signal) and fetch data for some pattern. The test result list follow.

Table 1. The performance test of the data interface

Count	time cost of fetch some pattern (second)	time cost of fetch all data (second)
1	0.067	0.171
10	0.692	1.742
100	6.698	17.081
1000	67.274	170.835

Areas for improvement

It was found that the Pv monitor program takes up 5% cpu on server, and MySQL takes up about 5% cpu too. With the increase number of signals, these program may occupy more cpu, consider two aspect to improve the data interface:

- use threads in the Pv monitor
- use non-relational database such as Hypertable.

Acknowledgments

In our application, we use the caObject2 class which supply the EPICS CA operation, this class saves a lot of time for us.

References

- 1 <http://www.aps.anl.gov/epics/base/R3-14/12-docs/CAref.html>
- 2 <https://docs.djangoproject.com/en/1.4/>