

The Fast History Conundrum

(Version 2)

Introduction

The History Subsystem

The History Subsystem is a file-driven system designed to record scalar data at fixed intervals. It deals with both SLC and EPICS data and stores most data in the whole control system. The update rate for most files is six minutes, with some files at three minutes. On a weekly basis, most data are “sparsified” to a two-hour interval. Physicists and operations ask that many important items NOT be sparsified.

The current “sparsification” mechanism is a simple sampler (just take the point closest to 2 hours apart). Clever algorithms could be added (min, max, avg).

This system was originally designed to hold a few items for a short while, and has grown to encompass all data forever.

The Correlation Plot Facility

Correlation plots offer a convenient way of sampling up to 160 variables at an interval of one second (or less!) for up to 512 samples (more recently 4096). These data are then typically exported to a MATLAB format and analyzed at a later time. In order to ensure “fresh” data, acquisition includes “poking” the micro computers to scan for and upload data. A history of micro “poking” exists and points only to PR02 as a basket case. The default Correlation Plot package enforces a one-second delay if some device is an analog input; special versions have subverted this feature.

“Glitch” Catching

A system is in place which records BPM values in the front end processors continuously. A trigger causes data recording to proceed for a short time, then freeze. These data can then be downloaded for analysis of a “trip”. Much work went into this solution, after other solutions had been thought through and rejected.

Fast-looping Button Macros

All button macros with the LOOP construct now have WAIT statements to prevent overly avid access. Ed has a program that scans button macros for the lack of a WAIT as a watchdog.

The Ideal Solution

A Fast History capability has been looked at from various perspectives over the past few years, and the requirements seem to boil down to:

- Sample data much faster, one Hz would be good.
- Be able to start and stop this sampling on demand.
- Make these data available to old (and new) on- and off-line tools.

And we would add:

- Ensure that the control system not creak to a halt.
- Ensure that the data do not overwhelm storage or bandwidth.
- Ensure ease of use for data specification and retrieval.

Current State of Affairs

BPM History Buffer

There is currently an interesting application which samples BPM data and writes them to disk - a kind of private BPM history data sampler. The application named BPMSAMPLER does not do this, due to the inherited SLC Beam code and private calibration requirements.

Fast Sampling and GPIB

Stephanie as recently added a general EPICS GPIB support. One application is in the RF stations, where sampling scopes are set up to trap “trips” and are read out to files via EPICS.

“Big” Correlation Plots

Greg has recently increased the maximum sample number from 512 to 4096 in correlation plots. This uncovered an interesting artifact that had a small impact with a low maximum but was devastating with a large number - one internal array was the sample size squared.

Portable Channel Access Capability

The PCAS has the capability of “rationalizing” access to individual jobs on individual micros. This recognizes normal and important clients, and can throttle the update rate according to the presence of important applications. At the moment this machinery is disabled. As noted below, the asynch update rate is a database parameter which can be set for each job in each micro. The PCAS mechanism and database-driven mechanisms are not exclusive of each other.

EPICS Archiving

The latest EPICS Archiver has some interesting capabilities on a per-channel basis:

- Sample data at a constant rate.
- Sample on change but at least at some rate.
- Sample on change but no faster than some rate.

LeeAnn and Bob Hall are currently looking at this archiver, investigating as well adding an Oracle database as the backing store.

Recurring Headaches

The current system is being constantly modified to handle “more and faster” in an ad-hoc manner with short-term vision. Files get too big and need to be split; signals get moved from one micro to another (PR02->PR00) or from one file to another and we are asked to provide “seamless” history. Ed is once again the person who researches, fixes, etc.

Resource Limitation

Sampling Capability of SLC History

The current SLC History system cannot support the fast history sampling requirements. Pushing all files to three minutes would even be a stretch. The limitation is simply the processing time for acquiring and storing all the data. An example of extra calculations is the RMS handling, where a bad RMS leads to a device being flagged as “bad”.

CPU

The MCC Alpha and the three EPICS gateways are all providing the necessary CPU cycles. Jingchen and Ed monitor the CPU usage and note daily occurrences of CPU-bound processes. Most problems are fleeting, but if the average rises, this will become more problematic.

Disk Storage

Ken and Terri have increased the disk capacity of the VMS cluster often. One of the requirements driving the rapid disk capacity expansion is the maintenance of uncompressed history data. The extensive data storage from correlation plot and buffered data acquisition comprises the other big expansion demand.

Recently a larger disk for the gateways has arrived, but its capacity is already planned for (CMLOG is a user).

Bandwidth

In general the networks are working fine. The VNC solution for controlling remote NT workstations and the NT-based television monitoring currently under scrutiny are examples of new applications that will stress the networks.

DBEX is not now a problem from either CPU or Direct I/O. Network bandwidth on SLCNET for PR02 IS a problem now.

SAM processing time

The SAM modules require about 800ms to prepare data. This limits reasonable analog status data acquisition to one hertz. Since the SAMs are used by at least three micro jobs, they are often over stimulated.

SAM modules have two hardware one-shots that are a cause of problems when one reads them rapidly. The hardware group has modified some of the critical SAMs in PR02 and it appears that they have a Rev. for the SAM with the changes.

We recently had a problem with a SAM in PR04 that needs to be read twice to create two devices with different alarm points to either trip the beam or request that LFB Amplifiers be turned off. Michael Harms replaced the SAM in question with one of the Rev'd. ones, which fixed the problem. Presumably, as SAM's are repaired, they receive these modifications. Unfortunately, there are lots of SAMs in existence and it may take some time before all have been upgraded.

Possibly Relevant Subsystems and Ideas

SAM Driver

Ken Underwood and Nancy have discussed providing a SAM driver on the micro, so that each micro job can ask for data as often as it wants. The SAM driver would deliver the latest data and ensure that the SAM module is not over triggered. This is not a quick fix, due to both Feedback's usage and the various Rev. levels of the SAMs.

Analog Status update times

Currently there is a correlation plot application running off a button macro in the front of the control room. This reads ASTS,PR02,DATA,CH117, the luminosity, but serves to trigger the analog status update from PR02 on a one second basis. We could simply tell the analog status job to update data at this rate. In that case, the correlation plot would have to NOT ask for a specific update.

We could add an additional function code to the Analog Job, which says "Update only if not done within time interval", where the time interval could be database driven for each micro.

If the update had been done within the time interval for that micro, the micro job would not gather/process data, but immediately return "OK" to caller. Correlation plot could either always use this feature or have a "hidden" toggle to switch between this type of request and the current check function code, which would remain in the Analog micro job.

We are also ready to switch over to using luminosity data from PR00 rather than PR02 and we will be proposing to move Stan Ecklund's Near IR thermocouples to PR00 during the downtime.

Route ASTS requests through PCAS

The PCAS has a throttling mechanism based on micro and job. If all requests went through this spigot, we could ensure maximum data rate with no contention. The drawback is that this is a small-bore spigot. The NLCTA might be a nice test-bed.

Dynamic EPICS-based Archiving

This would provide an excellent, flexible way of collecting fast data. It could use the recently commissioned command server (Cmdsrv) to process requests. It would require extensive Unix disk storage (or Oracle storage).

Standard History Data Protocol

If we have more than one history data storage format or machine, the history viewing and analysis interface(s) need access to all, and some cleverness to choose the most dense data. Currently the Epics Collaboration is working on a standard Channel Access protocol for historical data. This mechanism, or some thing similar, would be needed to support multiple history storage systems.

This same protocol would allow the use of new tools with our old history data.

An Attempt to Categorize

SAM Driver

[Ken Underwood]

- **Upside:** Stops the hammering on SAMs
- **Downside:** Micro-level work (Analog, Magnet, Feedback)

Analog Status Update Times

[Nancy, Ken U., and Stephanie]

- **Upside:** Remove necessity of CRR SCP in control room.
- **Downside:** CRR changes
Micro changes
Management difficult? (Nancy thinks not)

Use of PCAS

[Stephanie]

- **Upside:** Rationalize access at the Alpha level
Undo “micro poke” kludges
Management structure already in place
- **Downside:** Applications would have to use Epics notation
PCAS would become a choke-point

Epics based Archiver

Leeann and Hamid know the status.

- **Upside:** On-demand fast archiving
- **Downside:** Still at the evaluation stage
Requires History Protocol work
Requires planning for storage

Standard History Protocol

[Hamid, Bob, and Stephanie]

- **Upside:** We will do this (or have it done) anyway
Allows use of foreign display/analysis tools
Necessary infrastructure for EPICS
- **Downside:** Requires lots of work retrofitting data access and display tools.

A Recommendation

My feeling is that we should pursue both short- and long-term solutions:

Short Term

1. Move all possible signals to PR00. (Downtime activity).
2. Work with the Epics community on the Standard History Protocol. We need this anyway.
3. Release Greg's 4096 sample CRR facility. He has removed the "sample squared" array, so the impact is now much less than before that improvement.
4. Investigate modifying the ASTS automatic update time for important micros (PR02, PR00, for example). This should follow the implementation of the SAM driver.

Medium Term

1. Do the SAM driver. This will remove the source of many problems. Maybe even the one that results in IEEE mode?
2. Improve the "sparsification" algorithm.
3. Upgrade to TCP/IP for PRxx micros. (Waiting for iRMX with TCP/IP?)

Long Term

Keep working on the EPICS archiver. This will be important. It will provide the dynamic, fast archiving we need, with the data available to the tools we know and love.

Uncertain

The PCAS throttling is a good idea, but the downside may outweigh the upside. It is something to keep in the back pocket if things bog down.