Data Logging Archiving Collection and Storage

Fermilab
Beams Division
Accelerator Controls Department

Kevin Cahill

Version 1.1
May 5, 2003

Introduction

Lack of Backups

The data loggers have never been backed up. The size of the data repositories coupled with the number of data loggers and the difficulty of locking down a database are problematic issues.

Historically, we have lost very little data logger data to disk crashes. Generally a failed disk on a VMS system yielded up its data file which was moved and rebuilt. Less than a half dozen times was a logger reinitialized empty of data. Recovering data files from failed disks on other operating systems seems less promising.

A proprietary method to backup or archive data from a single data logger is not desirable. A chosen implementation should be scalable to all the data loggers.

The data logger BACKUP has been implemented and is described at the end of this document..

Existing Data Logger Archive

An archive logger was released in 1994 to archive data with a minimum fifteen minute between sample resolution. It was moved to a Java node with data from January, 1996 and now has a minimum five minute between sample resolution.

Its disadvantages besides a paucity of data include the inability to distinguish archive data collected on clock event or by client loggers. Though the present slow archive has a minimum five minute sample resolution, it does not solve other shortcomings.

Big Archive Trials

Goals

To archive each periodically sampled device at its fastest rate.
To archive each device uniquely sampled on a Tevatron clock or software state transition.
To archive each uniquely client logged device.
To serve as a data source to rebuild a lost logger disk to a reasonable recent history.
To hold a deeper collection of data for devices sampled at a high rate.

Building the Jobs

The data logged devices and their collection rates are parsed to yield may jobs. Each job consists of a list of devices to collect from a specific logger on a specific collection event for the previous day. The Java data loggers allow retrieval based upon collection event whereas the VMS data loggers return all the data collected across all collection events, so a Java data logger is the preferred choice when available. Likewise, few VMS data loggers sample on hard Tevatron clock event, so a Java data logger is the preferred choice when available. Initially, 240 jobs define a daily collection, and more than 500,000 ACNET requests to data loggers collect 275,000,000 points.

Running the Jobs

Shortly after midnight the job lists are rebuilt to accommodate logging list changes. The jobs are then run with the number of concurrent jobs controlled by a compiled constant limit presently set at 10 and the number of writes queued to the daily database. The daily collection typically completes in less than 90 minutes. Jobs that fail are queued for retry no faster than every 30 minutes.

Excluding Loggers from Archiving

A compiled list identifies loggers to be excluded from archive collection. The SDA data loggers CBSDA and PBSDA are excluded since they can be regenerated from SDA data. The State logger is a duplicate software state transition logger on VMS and is excluded in favor of EventV, the Java software state transition logger. Snap1 and FTP1, the plot loggers, are excluded as having questionable value for archiving. Sets, the settings logger is excluded until it moves to a Java node and can identify to the archive its devices for retrieval.

A Database A Day

Initially, a single MySQL database table was created for each day's archive collection, but a single day's collection was already slightly larger than the maximum table size of 4 gigabytes.

Another trial involved four databases per day. One for the devices collected on periodic rate with names alphabetically from A through S, a second for periodic devices from T through Z, a third for devices collected on Tevatron clock event or software state transition, and a fourth for client logged devices. The total size of all the databases was about 4.1 Gigabytes for a single day's collection

Data Compression

The major component of saved data is the repeating 8 byte each timestamp and double precision floating point value of each collected point. Up to 25,000 points are stored in each blob. By applying data compression, the blob's size was reduced by 73% with little increase in total collection time (less than 90 minutes). Of course the cpu utilization increases during collection. Data retrieval rates are not appreciably slowed. Now a day's worth of data encompasses slightly more than one gigabyte.

Backing up the Archive

Each day's collection results in four new MySQL tables or eight new disk files (index and data) that may be copied to a backup device since they will not change after the collection is completed.

Deployment Choices

A single archive consumes about a gigabyte a day:

periodic "a-s"	270 MB
periodic "t-z"	432 MB
event	252 MB
client	125 MB

The number of days of data available on spinning media depend upon the disk capacity of the archive node. Client and event archives could be deployed on additional machines thereby increasing the effective archive capacity.

The BACKUP data logger

The data logger BACKUP has been implemented and deployed. It runs on DPE09. It begins collection shortly after midnight and finishes in about three hours.

Backup contains data since midnight April 8, 2003. Each night it runs 240 plus jobs utilizing 600,000 plus ACNET requests to collect data from 26,000 plus device elements spanning 300,000,000 plus points. It will be a few years before its current disk will require expansion or removal of older data.

The module DataLoggerTransfer in the scheduler class contains instructions for restoring a data logger whose disk was lost from the BACKUP data logger.