BEAM TRAIL TRACKING AT FERMILAB

Dennis J. Nicklaus, Linden Ralph Carmichael, Richard Neswold, Zongwei Yuan.
Fermilab, Batavia, IL 60510, USA

Abstract

We present a system for acquiring and sorting data from select devices depending on the destination of each particular beam pulse in the Fermilab accelerator chain. The 15 Hz beam that begins in the Fermilab ion source can be directed to a variety of additional accelerators, beam lines, beam dumps, and experiments. We have implemented a data acquisition system that senses the destination of each pulse and reads the appropriate beam intensity devices so that profiles of the beam can be stored and analysed for each type of beam trail. We envision utilizing this data long term to identify trends in the performance of the accelerators.

INTRODUCTION

The Fermilab particle beam starts in its ion source, and from there it can be directed to a wide variety of beamlines and experiments. This paper describes the software created to track the accelerator efficiencies (amount of beam transported without loss) along the various beamlines and for the designated destinations. We use the term trail to designate each of the potential uses and destinations.

Trails can have a wide range of complexity. The simplest trails consist of only one 15Hz Linac beam pulse, and might go to a local beam dump, or be used to study Linac performance. The more complicated pulses involve multiple 15Hz pulses from the Linac, and involve the Linac, Booster, Recycler, and Main Injector accelerators as shown in Fig. 1. Final destinations include various beam dumps along the way, a neutron therapy target, short and long baseline neutrino beamlines, the muon campus experiments, or a test beam area.

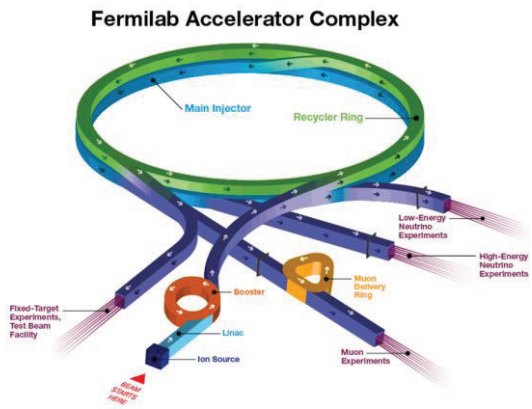

Figure 1: The Fermilab Accelerator Complex.

Collection of these more complex trails is also complicated by the fact that some of the simpler 15Hz pulse trails may occur while the more complicated trail is still underway in the downstream accelerators.

The process starts with the overall beam pulse planning controlled by Fermilab's Sequencer, which coordinates the production of timelines generated by the Time Line Generator (TLG) with other accelerator activities. At the implementation level, a timeline is a series of private bus events (TCLK). Various accelerator components are programmed to respond to these events to send the beam to the desired location. These systems have long been in routine use for accelerator operation.

For this beam trail tracking, we added or enhanced several software programs. To start the process, the TLG now encodes the trail number onto our MDAT data bus when it begins instructions for each beam pulse. The Beam Cycle Coordinator (BCC) decodes that trail number and makes it available to our Acnet control system, in addition to many digital status bits that indicate readiness of various beamlines, components, permits, and interlocks. These status indicators also show whether the accelerator chain was ready and beam pulse had the opportunity to follow the indicated trail, or whether it had to be aborted.

With the above groundwork laid, the main implementation for beam trail tracking begins. The tracking software has three main components:

- The Correlator, which collects data from Acnet-connected devices, groups together readings from the same pulse, and sorts them by trail number.
- The Acnet Formatter, which takes the data from the Correlator, aggregates and re-formats it and makes it available to Acnet.
- The Database Interface, which reads the data over Acnet and stores it into a relational database. Along the way, this also calculates various sums and means, and we have developed a user interface to help retrieve data of interest.

BACKGROUND

Main Injector Ramp Cycle

In order to understand some of the requirements of the trail tracking software, it is useful to have a basic understanding of our Main Injector ramp cycle. The Main Injector accelerates protons from 8 GeV to 120 GeV, taking a little over one second for this acceleration. While it is ramping up and down, multiple batches of protons from the Booster can be injected into the Recycler

ISBN 978-3-95450-148-9

and stacked into a more intense beam. When the Main Injector has finished the previous ramp cycle, it is injected with the new beam of 8 GeV protons from the Recycler. Overlapping the filling of the Recycler with the ramping of the Main Injector decreases the average full acceleration time for each beam pulse, and has enabled Fermilab to achieve record beam power levels.

Not every 120GeV trail uses the Recycler in this way. We can also inject directly from Booster to the Main Injector.

IMPLEMENTATION

Correlator

The initial data acquisition, the sorting by trail number, and the correlation into the appropriate 15Hz time slots is all done in the Correlator. Most of the data from the Linac and Booster is collected at 15Hz. Some of the other data is only collected upon a particular TCLK event, or on one of several different TCLK events, depending on the trail. We primarily collect beam intensity measurements from a variety of instrument types, although other readings such as the BCC status bits are also included. All data is collected through Fermilab's Acnet control system

Since the readings come from a variety of instruments connected to different front-ends, Acnet doesn't guarantee that the readings will always arrive in the same order, or even that a reading from one front-end will even arrive in the same 15Hz time slice as all the other readings from that same slice. The load on a particular node, or unforeseen network delays can all contribute to this. So in order to collect the maximal amount of data, with as few rejections for tardiness as possible, the Correlator maintains a sliding window of time slices it is currently collecting. When one slice is "full" (meaning all devices for that trail have arrived) or when that slice falls off the end of the sliding window, that 15Hz trail is collected for further processing and the time slice is closed. The Correlator follows the useful rule that collections can't go back in time, so when one slice is declared full, all previous unfilled slices in the window are also closed out for processing and any uncollected devices are noted. Thus the system is somewhat fault tolerant. If one response is dropped by the network, or if a particular front-end is temporarily unresponsive, data collection and processing can still proceed.

Another fault-tolerant adaptation that the Correlator performs relates to errors from devices. If one device replies with some error, then the Correlator will attempt to restart the collection process in hopes that the error was only temporary and has cleared. If a device is repeatedly in error, then the Correlator masks that device off the list of requested devices for all trails. Periodically, data collection is retried in hopes that the error condition has cleared and the device's reading can re-join the data set. Of course, faults in some crucial devices, such as the trail number, cannot be tolerated, and processing has to always wait on those errors to clear before proceeding.

Some of our most common trails will begin at the ion source, go through the Linac and Booster, then into the Recycler for beam stacking, then into the Main Injector for final acceleration, and then to another final destination, such as the long-baseline neutrino beamline. However, as noted above in the explanation of the Main Injector ramp cycle, one trail of beam may still be in the Main Injector while another trail begins assembling in the Recycler. Or while the Main Injector ramp is occurring, another trail through the Booster to the short baseline neutrino beamline may occur. The Correlator has to deal with these overlaps, making sure that the appropriate trails all stay sorted, and dealing with the fact that one Recycler and Main Injector fill will include multiple 15Hz pulses from the Linac-Booster chain. The software uses the term super-trail to describe any trail which includes multiple 15Hz batches.

When the Correlator knows it is finished with any trail or super-trail, they are sent off to the Acnet Formatter for further processing. The end of a super-trail is typically indicated by some TCLK event, which the Correlator has to monitor.

The devices to be read, acquisition frequencies or triggering events, devices needed per trail, and classifications of trails are all driven by a human-readable configuration file. Thus it is straight-forward to change the devices requested for any particular trail.

Acnet Formatter

The Acnet Formatter is much less complicated than the Correlator. As it receives trails from the Correlator, it builds them into a list of trails for each trail number. Periodically (typically on a 10 second interval), for each trail list, the Acnet Formatter reformats and packs the data into structures (arrays) that can be transmitted over Acnet to any requestor. A header is prepended onto the data structure and the structure is built into a documented format.

While performing these collection and reformatting tasks, the Acnet Formatter also creates several diagnostics for each trail, such as number of completely filled vs. unfilled trails, counts of the number of times a device is missing (uncollected) from a trail, which is device is most often missing, which device's reading most often returns with a timestamp outside of the sliding collection window, and the total number of device readings with such a bad arriving timestamp.

All the formatted trail data structures and the diagnostic values are made available as Acnet device readings.

Database Interface

The Database Interface periodically reads the packaged trail data from the Acnet Formatter and stores the readings in a Postgress database. It computes several averages and sums, for instance the sum of each intensity device in a trail over a day. These computed values are also written to database tables, so graphs and displays of more commonly used values can be created more quickly,

ISBN 978-3-95450-148-9

without having to retrieve all the raw data and compute everything at plotting time.

We have created a graphical interface into the database to enable users to create useful plots of the data. The Java program allows the user to apply multiple selection criteria, including date, device name, summed vs. raw data, and more. This and other Java classes enable us to create daily or weekly beam summaries and statistics.

Figure 2 shows a sample plot of the type that can be made with the database information using the Java interface program. Figure 2 shows the daily summed intensity for a variety of devices along the beam trail and accelerator chain.

PROGRAMMING

The Correlator and Acnet Formatter are implemented in Erlang in our Erlang-based ACSys front-end framework. The Acnet Formatter is one of the frameworks "device drivers", with custom code handling the receipt of trail messages from the correlator and the repackaging of that data into Acnet-accessible structures. The Correlator is implemented as a supporting process to that device driver, being started by the driver and reporting to it. Data collection is done using our standard Erlang-based data collection client, and this project was the heaviest test of that collection client software to date and suggested several refinements of it.

The Database Interface is written in Java and uses our standard Acnet libraries to read data from the beamtrail collection front end. A Java OAC performs the continuous job of reading the assembled trail data and exporting it to the database. Other standalone Java classes and programs provide a graphical user interface to the database data and can create plots with many options from the data.

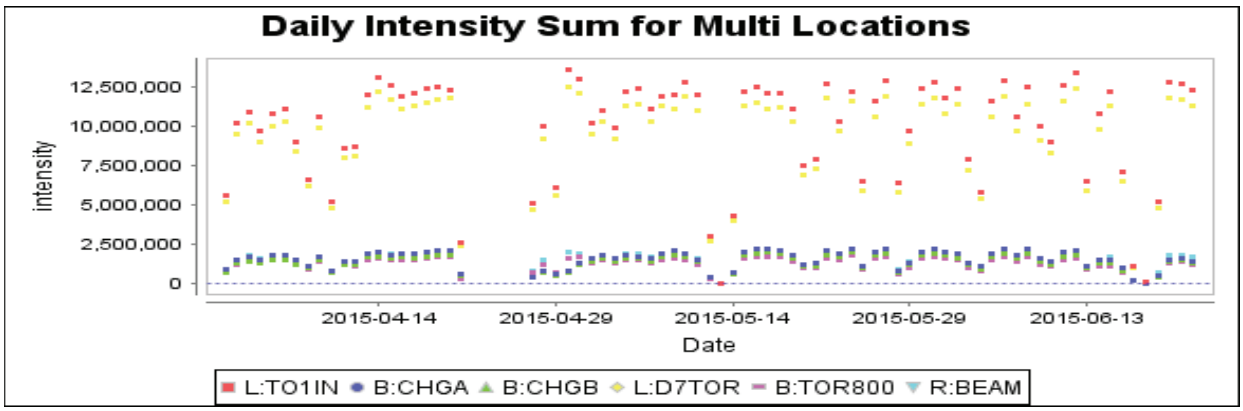

Figure 2: A sample plot showing summed intensities for a variety of device locations.

SUMMARY

We have implemented a set of programs that enhance Fermilab's ability to analyse accelerator performance. We collect beam intensity readings and sort them by the different trail each beam pulse follows through the accelerator and experimental target area chain. The software is able to handle temporary outages by individual reporting devices, and untangles overlapping beam cycles. With the database of information collected, we are able to produce summary plots and reports that provide information about short and long term performance of the accelerators.

ACKNOWLEDGEMENT

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

ISBN 978-3-95450-148-9