MINOS Beam Monitoring Software

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

MINOS Week In The Woods, 2005

Outline

1. Tour of the Packages
   - BeamData
   - BeamDataUtil
   - BeamDataDbi
   - BeamDataNtuple
   - BeamDataMonitoring
2. Hardware and Shift Environment
   - Shift Computer Hardware
   - Process Details

BROOKHAVEN NATIONAL LABORATORY

Package Overview

- BeamData online code.
  - The Beam Data Process Server and monitor GUI.
  - The JAS3 application.
  - Also has an early analysis framework.
- BeamDataUtil common classes used by the next three packages.
  - BDProcessor micro-framework
  - SWIC calibrations
- BeamDataDbi misnamed database updater job modules and supporting classes.
- BeamDataNtuple ntuple definition classes and job modules to fill them.
- BeamDataMonitoring "online" shift histogram micro-framework.

BROOKHAVEN NATIONAL LABORATORY

The BeamData Package

Three major areas:

- `python/` holds the Beam Data Process Server `bdp-server`, its GUI monitor `bdp-gui` and various utilities.

The BeamData/python Sub-Package

Programs include:

- **bdp-server** the BeamData Process Server is the primary program in this package. Provides the glue between the Beam Division's XML-RPC server and our Rotorooter.
- **bdp-gui** communicates with bdp-server and displays its status.
- **start_pdb** shell script to start the bdp-server.
- **devices-ss-to-cfg** converts text dump of spread sheet holding beam devices into a Python config file.
- **poll-acnet** simple command line interface to ACNET device data.
- **inject-xmlrpc** generic command line to simple query/response XML-RPC servers.

BROOKHAVEN NATIONAL LABORATORY

The bdp-server

An XML-RPC to RotoTalk proxy.

1. Contacts our dedicated XML-RPC server in Beam Division (written by Charlie King) with a list of devices
2. Goes into listen mode - Beam Division server calls us back with each spill of data.
3. Pushes data through RotoTalk to `rotorooter`

Other features:

- Can run essentially anywhere, with ssh port forwarding even behind firewalls. Multiple instance can run on same host.
- Written in Python using the Twisted asynchronous networking framework. Also makes use of SWIG for a Python wrapper of our RotoTalk code.

BROOKHAVEN NATIONAL LABORATORY

The BeamData Package

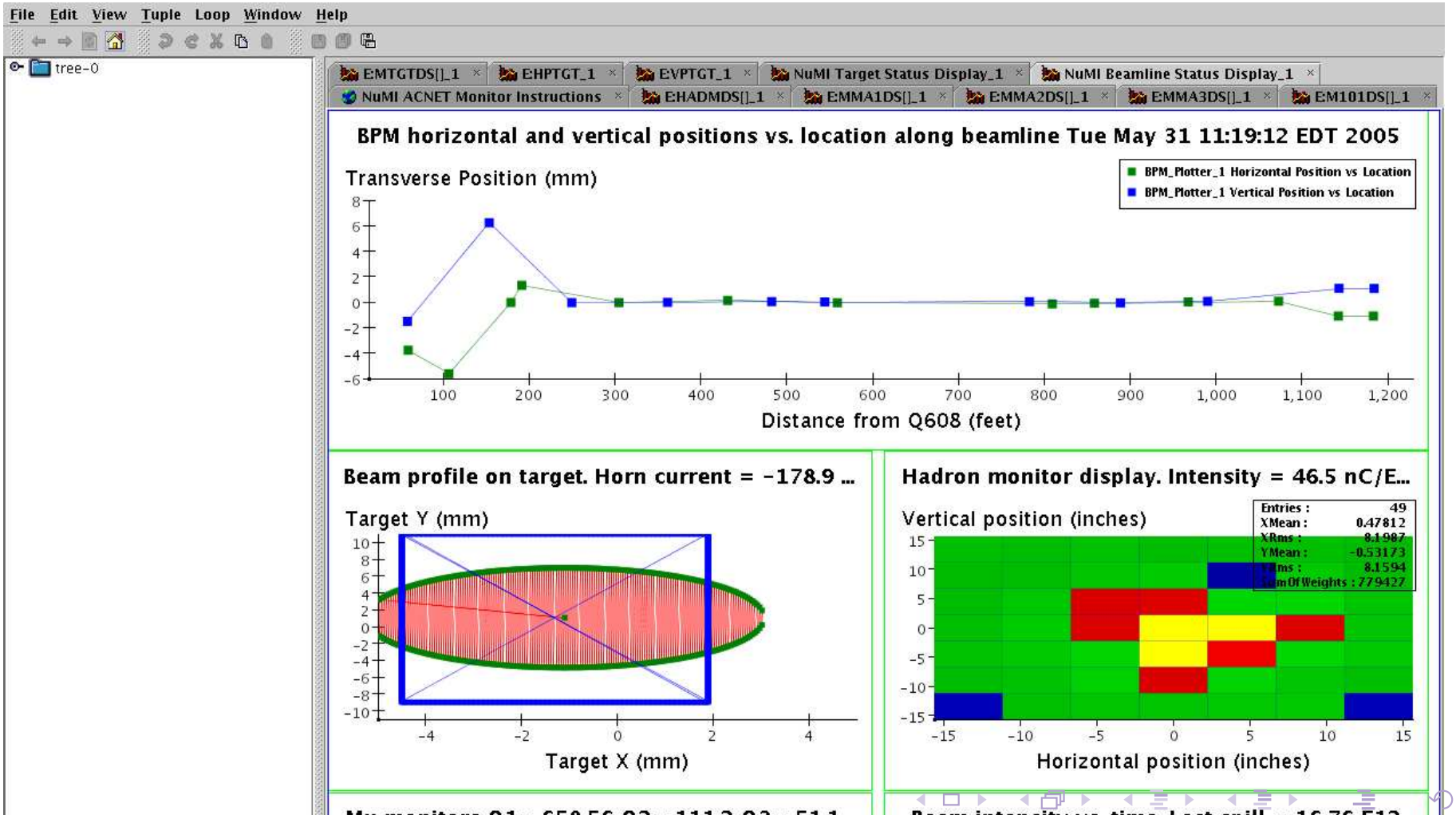
Three major areas:

- `python/` holds the Beam Data Process Server `bdp-server`, its GUI monitor `bdp-gui` and various utilities.
- `java/` holds the JAS3 application plugin.

The NuMIMon JAS3 Application Plugin

- Authored by Mary Bishai
- Provides promptly updated ("real time") plots and histograms
- Has proved indispensable, particularly in early running.
- Currently suffers from memory leaks, large CPU usage and instability.
- Will be "retooled" by BNL summer student into an alarm application - initiating a visual/audible alarm when data goes out of spec.

BROOKHAVEN NATIONAL LABORATORY


File Edit View Tuple Loop Window Help
tree-0
EMTGTDS[]_1
EHPTGT_1
EVPTGT_1
NuMI Target Status Display_1
NuMI Beamline Status Display_1
NuMI ACNET Monitor Instructions
EHADMDS[]_1
EMMA1DS[]_1
EMMA2DS[]_1
EMMA3DS[]_1
EM101DS[]_1
BPM horizontal and vertical positions vs. location along beamline Tue May 31 11:19:12 EDT 2005
Transverse Position (mm)
BPM_Plotter_1 Horizontal Position vs Location
BPM_Plotter_1 Vertical Position vs Location
Distance from Q608 (feet)
Beam profile on target. Horn current = -178.9 ...
Target Y (mm)
Target X (mm)
Hadron monitor display. Intensity = 46.5 nC/E...
Vertical position (inches)
Entries : 49
XMean : 0.47812
YMean : -0.53173
Horizontal position (inches)

The BeamData Package

Three major areas:

- `python/` holds the Beam Data Process Server `bdp-server`, its GUI monitor `bdp-gui` and various utilities.
- `java/` holds the JAS3 application plugin.
- `ana/` holds a preliminary, deprecated analysis framework.

The BeamData/ana Sub-Package

- Contains code to write "flat" TTrees from the raw BeamMon data stream.
- Includes a rather in depth, but deprecated analysis framework `libBeamDataAnaNt`
  - Spin over the "flat" files
  - Provide safe but still optimized access to the tree (MakeClass with brains).
  - Unpack Hadron/Profile monitors.
  - Generate and apply SWIC pedestals.
  - Plotting micro-framework.
- Contains Mark D's interface to Mary B's "beam summary" trees.
- Contains my inital "#PoT" plotting code (now improved and moved to `BeamDataUtil`)

BROOKHAVEN NATIONAL LABORATORY

The BeamDataUtil Package

Two main classes of classes:

The BeamDataUtil Package

Two main classes of classes:

1. Database Interface

Database Tables

Tables named like `BEAMMONXXXYYY` with coresponding classes like `BeamMonXxxYyy`.

Database Tables

Tables named like `BEAMMONXXXYYY` with coresponding classes like `BeamMonXxxYyy`.

BeamMonSpill Per-spill data.

Per-spill: BeamMonSpill table

One entry for each spill, but access optimized: 60 sec of spills written together.

- Earilest DAE and VME times
- 4 toroids, horn current
- BPM pos at target, BPM intensities
- PM pos target, widths
- Had/Mu total intensity
- Status word bit field:
  - Horn on, target in, target location (LE, psME, etc)?
  - Is spill used for pedestal? Number of BPM batches?
  - Is PM121/TGT in? What time source, earliest DAE or VME?

BROOKHAVEN NATIONAL LABORATORY

Access Per-spill Table

Accessing the per-spill table `BeamMonSpill` is done via `BDSpillAccessor`:

- Should be held long term to improve DB access performance.
- Returns closest `BeamMonSpill` to given `VldTimeStamp`.
- Uses the `DbiResultKey` mechanism to avoid unnecessary internal cache resets.
- DB hit takes few 100 msec and will happen once per a minute's worth of data.
- Can use `DbiResultPtr<BeamMonSpill>::NextQuery` to iterate through the spills.

Database Tables

Tables named like `BEAMMONXXXYYY` with coresponding classes like `BeamMonXxxYyy`.

BeamMonSpill Per-spill data.

BeamMonFileSummary Per-file data.

Per-file: BeamMonFileSummary table

One entry per file:

- Begin/end time
- Number of spills
- Number of protons
- TCLK trigger event and delay
- File name and size
- Root version, last mod time

Access only via `DbiResultPtr<BeamMonFileSummary>`.

Database Tables

Tables named like `BEAMMONXXXYYY` with coresponding classes like `BeamMonXxxYyy`.

BeamMonSpill Per-spill data.

BeamMonFileSummary Per-file data.

BeamMonSwicPeds SWIC channel pedestals

BeamMonSwicRel Had/Mu mon relative pixel

BeamMonSwicMask Arbitrary per SWIC channel weights

SWIC Pedestals: BeamMonSwicPeds table

Use all spills in a file which:

1. All 4 toroids existing and $< 0.1 \times 10^12$ protons
2. Any BPMs have total intensity < 5.0
3. SWIC VME and DAE times are compatible

There is one entry per file. Table is defined as:

device name	gives canonical ACNET SWIC device name
# samples	gives number of spills sampled
means	the mean pedestal for each 96 channel
sigma	the rms pedestal for each 96 channel

This allows combining of peds from different files to reach desired stats.

BeamMonSwicRel and BeamMonSwicMask

BeamMonSwicRel:

- Calibrates hadron/muon monitor pixels relative to central pixel
- Just holds device name and 96 constants

BeamMonSwicMask:

- Allows arbitrary per-channel weights
- Currently not used for anything
- Holds:
  - Device name
  - Begin and end time that the mask should be applied
  - Channel offset and weight

Using the BDSwicCalibrator

BDSwicCalibrator is a Singleton. SWIC like objects register with it on creation. To use:

1. Create any SWIC objects:
   - BDSwicDevice (base clase)
   - BDProfMon Profile Monitor
   - BDHadMuMon Hadron/Muon Monitor
2. Set their data

```
BDSwicDevice::SetData(const RawBeamData&);
```

3. Calibrate:

```
BDSwicCalibrator::Calibrate(const RawBeamMonHeaderBlock&,
                            const RawBeamMonBlock&);
```

The BeamDataUtil Package

Two main classes of classes:

1. Database Interface
2. Processor micro-framework

The BDProcessor Micro-Framework

- Encapsulate some useful calculation: `BDProcessor`
- Provides simple linear pipeline of processors: `BDPipeline`
- Has hooks for an as yet not realized notion of flagging gross in changes beam condition.
- Some useful base classes and implementations

`BDScalar`	maintains a running average of a device's data
`BDEarliest`	finds the best time to use for the spill
`BDTarget`	access target locations and project BPM and PM readings onto target.
`BDHadMuMon`	access calibraded Hadron/Muon monitor data. By-pixel unpacking.
`BDProfMon`	access calibraded Profile monitor data. By-foil unpacking.

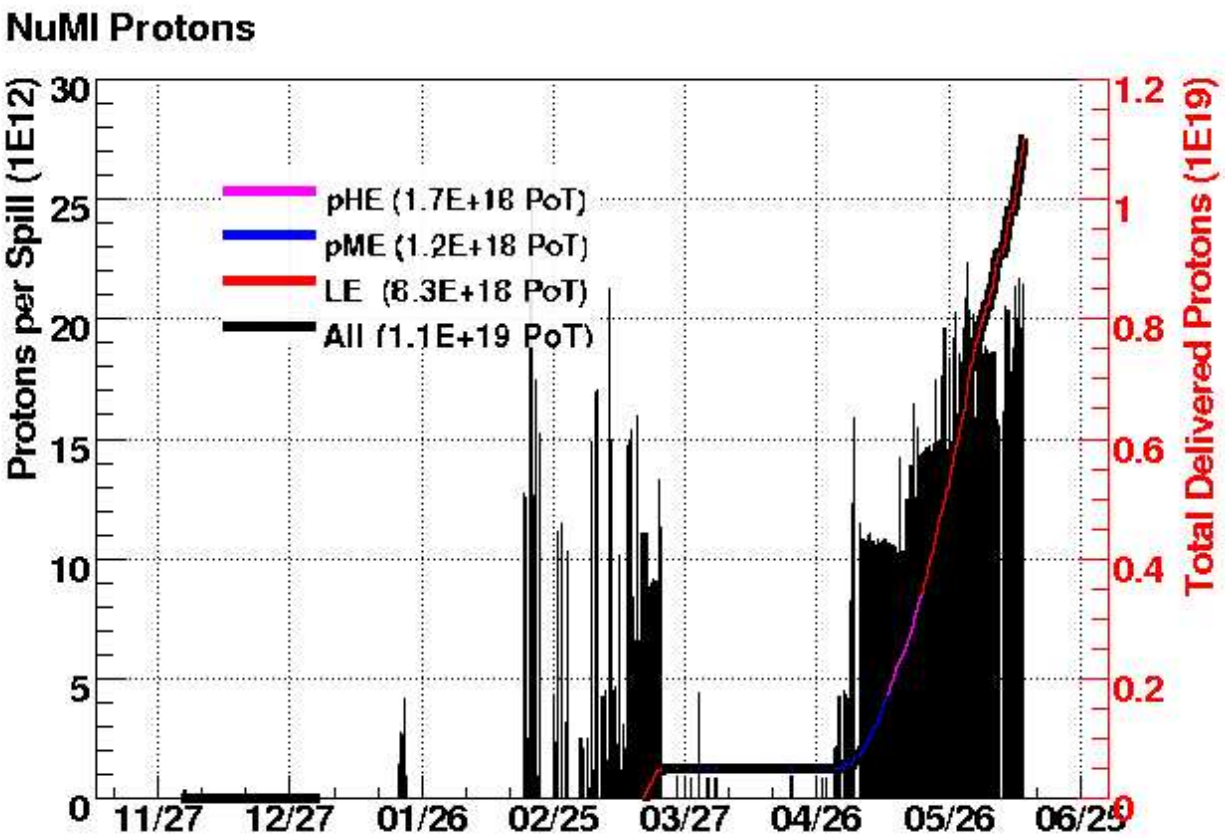
The BeamDataUtil Package

Two main classes of classes:

1. Database Interface
2. Processor micro-framework

Also holds the new #PoT plotter

The nPoT plotter



- Instantaneous & cumulative
- Per spill entries
- Trigger position as color

The BeamDataDbi Package

Misnamed package - should be BeamDataDbu as it updates DB tables with two modules:

BeamMonSwicPedsDbuModule calculates SWIC pedestals and fills BeamMonSwicPeds

BeamMonDbuModule fills:

- BeamMonSpill the per-spill table
- BeamMonFileSummary the per-file table

There is also:

- a macro to fill BeamMonSwicRel the calibration relative to central pixel for Had/Mu monitors using NuMI-B-1084.
- a currently unused BeamMonSwicMask that can provide arbitrary per channel weighting.

The BeamDataNtuple Package

- Developed completely by Mark Dierckxsens
- "Sue Style" format
- Syncronized to ND/FD snarls
- Two trees:

  NtpBDLite Contents of BeamMonSpill + SpillTimeND table. Grafts the data on to the ND/FD data during farm production.

  NtpBDFull Currently a place holder. To be used for fuller but still summarized ntuple. Will be based on Mary Bishai's interim "beam summary" trees.
- The "Lite" tree will be in production as of R1.17

BROOKHAVEN NATIONAL LABORATORY

The BeamDataMonitoring Package

- Provides "Online Monitoring" histograms for shifters
- Is distint from DAQ OM except shares same GUI "donated" from CDF
- Use standard JobCModules to produce histograms
- Fill HistMan folder "Monitoring"
- Can develop histograms independent of CDFMonitoringFwk and Dispatcher
- Uses special JobCModule to talk to CDFMonitoringFwk `Server` program.
- Uses built-in `loon` features to talk to Dispatcher

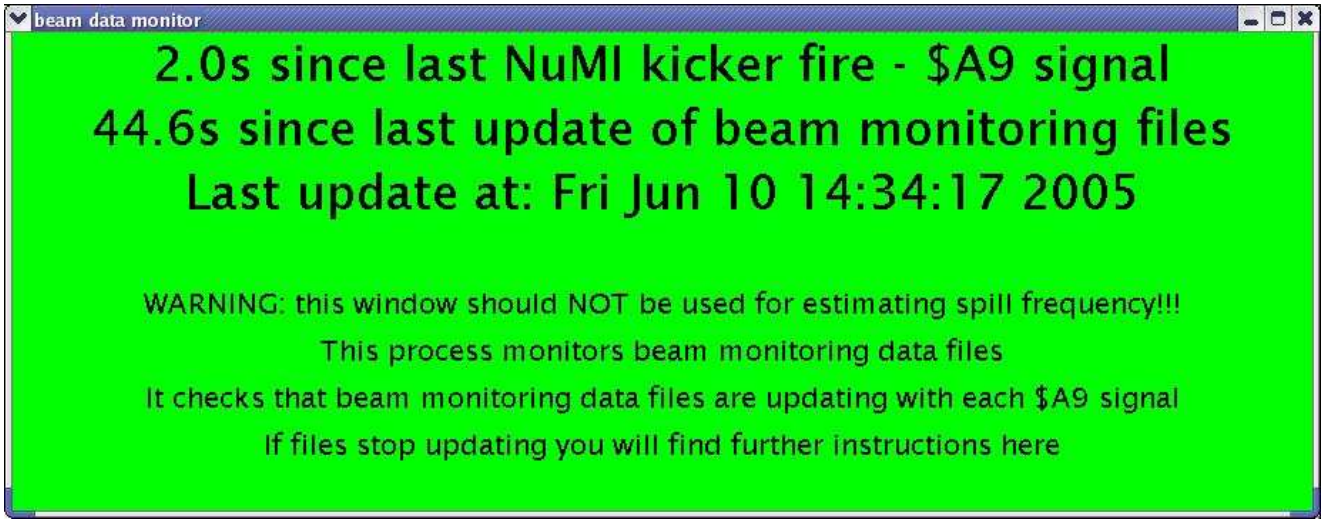
Tom Osiecki deserves a huge thanks for implementing many of the existing histograms and testing out the framework!

BROOKHAVEN NATIONAL LABORATORY

The Big Green Button

A meta-monitor:

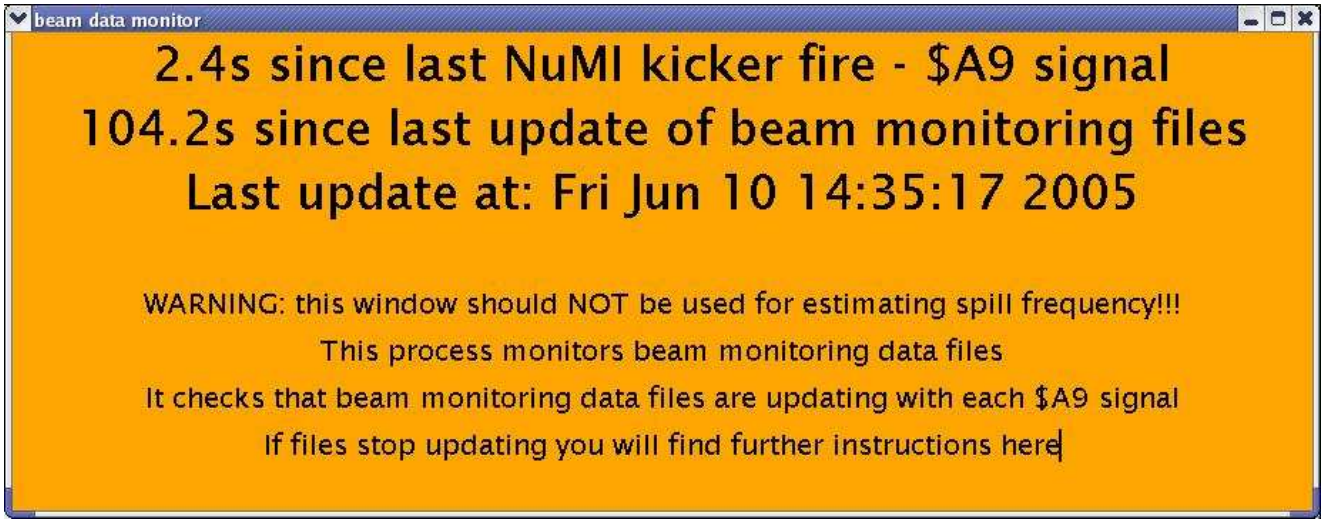
- Developed by Rustem Ospanov (big thanks!)
- Monitors the BeamData files and if files stop updating, checks ACNET
- If ACNET shows 0xA9 TCLK events still occuring goes from green...
- It has already proven crucial in catching problems quickly



The Big Orange Button

A meta-monitor:

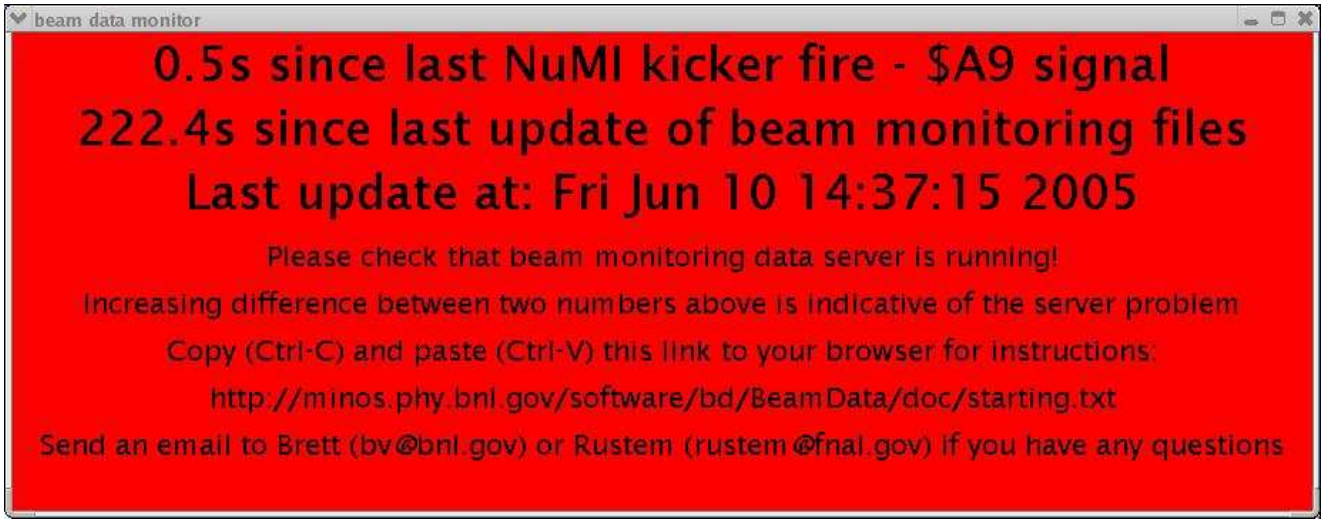
- Developed by Rustem Ospanov (big thanks!)
- Monitors the BeamData files and if files stop updating, checks ACNET
- If ACNET shows 0xA9 TCLK events still occuring to orange...
- It has already proven crucial in catching problems quickly



The Big Red Button

A meta-monitor:

- Developed by Rustem Ospanov (big thanks!)
- Monitors the BeamData files and if files stop updating, checks ACNET
- If ACNET shows 0xA9 TCLK events still occuring and finally to red.
- It has already proven crucial in catching problems quickly



1 Tour of the Packages

2 Hardware and Shift Environment
- Shift Computer Hardware
- Process Details

BeamMonitoring Computer Hardware

- Currently have `minos-acnet` a UP P4 2.4GHz 2GB RAM.
- Too overloaded to run all that is needed
- On order (thanks Liz!) DP P4 3.2 GHz 4 GB RAM
- Will become `minos-acnet`. Old `minos-acnet` will become `minos-beamdata`.
- New `minos-acnet` run JAS3 alarms, BeamDataMonitoring, nPoT plot, "Big Green Button" and whatever other cool things come up.
- New `minos-beamdata` will run `bdp-server` and the DBU job only
- May remove `minos-beamdata` from control room

The BeamData Process: bdp-server and DBU

Server:

- Start the server - click on button or do

```
shell$ start_bdp
```

- Stop the server - explicitly `kill` or

```
shell$ shutdown_bdp
```

- Check if dead

```
shell$ ps auxww | egrep 'bdp|roto'
```

- This needs some cleaning up....

DBU:

- Runs from cron every 10 minutes.
- Lock file mechanism assures only one instance
- Uses symlinks to find input (see next)

BROOKHAVEN NATIONAL LABORATORY

File System

On `minos-acnet` (soon to be `minos-beamdata`):

- Data goes to `/data/bdpdcp/data-files`. File closed every 8 hours.
- Rotorooter maintains symlink for Dispatcher `/data/bdpdcp/data-files/currentfile`
- When file closed symlink made in `/data/bdpdcp/archiver/data-to-archive` `/data/bdpdcp/dbu/data-to-dbu`
- When file is archived/dbu'ed `/data/bdpdcp/archiver/data-archived` `/data/bdpdcp/dbu/data-dbued`
- Logs go to `/home/minos/run/bdp/bdp-server.log`. Are rolled every so offten.

Currently still have all data on disk - 13GB.

Summary

- Beam Monitoring data is being taken (13 GB so far)
- Per-spill, per-file and SWIC calibration database tables are defined and are being filled automatically 3× daily
- The basic offline analysis framework exists
- Standard ntuples are (all-but) being produced (next frozen release)
- Shift histogram "online monitoring" framework exists with many plots implemented
- A BeamData process monitor has been written to give shifters early warnings of problems.

A big thanks for constributions from Mark, Mary, Rustem and Tom!

BROOKHAVEN NATIONAL LABORATORY

Still To Do

Mostly done except for setting up the new system:

- Wait for new `minos-acnet` to be delivered and set up (John Urish)
- Install JAS3, BeamDataMonitoring, B.G.B.
- "Set up" minos-beamdata (mostly just a name change)
- Set up a backup bdp-server on one of `minosXX` (different building, network, power) to cover the few times the primary server has problems.

There is also a need for a centralized warning system to which BeamMonitoring data monitors will attach. I'm working with Sergei to identify possible systems and people to work on this. If this interests you, let me know.

BROOKHAVEN NATIONAL LABORATORY