GLAST Large Area Telescope: Exploring the $\gamma$-ray Sky

Richard Dubois
Stanford Linear Accelerator Center
richard@slac.stanford.edu

http://www-glast.slac.stanford.edu/software
Über Outline

• Introduction to GLAST & C++ world

• Reconstruction events in a pair conversion telescope

• Astronomy analysis with GLAST

• Data Handling
Outline

• Introduction to GLAST

• The Instrument
  – Pair conversion telescope

• Code Development Environment

• Users: code installation, documentation

• Overview of C++ World: Gaudi, GEANT4 etc
GLAST measures the direction, energy and arrival time of celestial gamma rays

- LAT measures gamma-rays in the energy range $\sim 20$ MeV - $>300$ GeV
  - There is no telescope now covering this range!!

- GBM provides correlative observations of transient events in the energy range $\sim 20$ keV – 20 MeV

Launch: August 2007
Florida

Orbit: 565 km,
28.5° inclination

Lifetime: 5 years
(minimum; 10 yrs goal)
NASA - DoE Partnership on LAT

LAT is being built by an international team

Stanford University (SLAC & HEPL, Physics)
Goddard Space Flight Center
Naval Research Laboratory
University of California, Santa Cruz
University of Washington
Ohio State University
CEA/Saclay & IN2P3 (France)
INFN & ASI (Italy)
Hiroshima University, ISAS, RIKEN (Japan)
Royal Inst. of Technology & Stockholm Univ. (Sweden)

GBM is being built by US and Germany

MPE, Garching (Germany)
Marshall Space Flight Center

Spacecraft and integration - Spectrum Astro
GLAST science - the sky above 20 MeV

- Active Galactic Nuclei
- Unidentified sources
- Cosmic ray acceleration
- Solar flares
- Pulsars
- Gamma Ray Bursts
- Dark matter

<table>
<thead>
<tr>
<th>Energy</th>
<th>0.01 GeV</th>
<th>0.1 GeV</th>
<th>1 GeV</th>
<th>10 GeV</th>
<th>100 GeV</th>
<th>1 TeV</th>
</tr>
</thead>
<tbody>
<tr>
<td>R. Dubois</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>
γ detection – pair conversion telescope

Pair production is the dominant photon interaction in our energy range

GLAST Concept

- Low profile for wide f.o.v.
- Segmented anti-detector to minimize self-veto at high E.
- Finely segmented calorimeter for enhanced background rejection and shower leakage correction.
- High-efficiency, precise track detectors located close to the conversions foils to minimize multiple-scattering errors.
- Modular, redundant design.
- No consumables.
- Low power consumption (650 W)
GLAST Large Area Telescope (LAT)

Si Tracker Tower
- pitch = 228 µm
- 5.52 x 10^4 channels
- 12 layers × 3% X_0
  + 4 layers × 18% X_0
  + 2 layers

ACD
- Segmented scintillator tiles
- 0.9997 efficiency
  ⇒ minimize self-veto

Single Photon Angular Resolution
- 3.5° @ 100 MeV
- 0.15° @ 10 GeV

CsI Calorimeter
- Hodoscopic array
- 8.4 X_0 × 8 × 12 bars
- 2.0 × 2.7 × 33.6 cm
  ⇒ cosmic-ray rejection
  ⇒ shower leakage correction

Data acquisition

Good Energy Resolution
- ΔE/E ~ 10%; 100 MeV – 10 GeV
- ~ < 20%; 10 GeV – 300 GeV

3000 kg, 650 W (allocation)
- 1.8 m × 1.8 m × 1.0 m
- 20 MeV – 300 GeV

16 identical towers
300 Hz average downlink

R.Dubois
Cosmic Ray Muon for Two-Towers
GLAST MISSION ELEMENTS

TLM: S-band @ 1,2,4,8 kbps
Science data @ 40 Mbs (13 GB/day average)
CMD: S-band @ .25, 4 kbps

Mission Operations Center (MOC)
GLAST Science Support Center
LAT Instrument Operations Center
HEASARC

GRB Coordinates Network
Internet 2
Schedules

White Sands
TDRSS SN S & Ku

R.Dubois
GLAST

LAT sim/recon

ScienceTools

Infrastructure

System Tests

Simulation

Architects

Analysis Tools

Pipeline

ACD

Likelihood

Code Distribution

Release Manager

CAL

Pulsars

System Tests

Data Server

TKR

GRBs

Issues Tracker

Documentation

GEANT4

Obs Sim

SLAC Linux environment

SLAC Windows Environment

Recon

Event Interpretation

User Interface

Release Manager

List the rest

Sundry Utilities

Calibrations

ACD

I/O

Release Manager

CAL

I/O

TKR

Architecture

Flight Int Support

R.Dubois

~25 FTEs
Software Development Approach

- Enable distributed development via cvs repository @ SLAC
- Extensive use of electronic communications
  - Web conferencing (VRVS), Instant Messaging (icq)
- CMT tool permits equal development on Windows and Linux
  - ‘requirements’ file generates MS Project or gnu Makefiles from single source
  - Superior development environment on Windows; compute cycles on linux
- documentation and coding reviews enforce coding rules
- “Continuous integration”
  - Eliminate surprises for incoming code releases
  - Build code when packages are tagged; alert owners to failures in build or running of unit tests. Results tracked in database.
  - Developing comprehensive system tests in multiple source configurations. Track results in database; web viewable.

R.Dubois
Follow on lead from SLD, BABAR, but …

- work with Tech Writer
  - skilled at extracting information from us wackos
  - worries about layout, organization
  - can write good
  - we’re struggling with apparent conflict of web navigation vs “printed book”. Pursuing the former.
Code Distribution

- Tied in to Release Manager builds database
- Provide self-contained scripts to run executables sans CMT

Java WebStart app
MRvcmt – gui for code development

- Run apps
- Tabbed output buffers
- cvs operations
- Clean, config, make, debug
- Package tree

Fox/Ruby app
GLAST plugin
GlastRelease config

Event control

Fox/Ruby/C++ app

Graphics tree

Graphics metadata:
HepRep

3D controls

Multiple views
Issues Tracker; CCB; wiki

• **User JIRA web issues tracker**
  – Commercial product but affordable
  – Handles bugs, features, improvements
  – Full user/group management
  – “roadmaps” for version evolution/project management

• **Change Control Board**
  – Code used in pipeline – sim/recon; executive scripts; pipeline itself
  – Require documentation of all changes – preferably backed up by JIRA issues
  – Demonstration that fixes work; system tests on sim/recon
  – Using wiki tool to record actions
  – 4-person board – adjudicated by email so far

• **Wiki**
  – Commercial product (Atlassian – same parent as JIRA)
  – Simple web editing independent of user OS
  – Space management; same groups and users as JIRA
Performing builds for Science Tools also

Past release
Release in progress
Future release

Display created from database query

Build status
Unit test status

R. Dubois

Alex
More Code Builds

Multiple packages being tracked

Web tag collector

All builds done in batch
- windows
- linux

R. Dubois
System Tests

Comparison of current to previous release.

Matt
# Sim/Recon Toolkit

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
<th>Provider</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACD, CAL, TKR</td>
<td>Data reconstruction</td>
<td>LAT</td>
<td>90% done</td>
</tr>
<tr>
<td>Recon</td>
<td></td>
<td></td>
<td>In use</td>
</tr>
<tr>
<td>ACD, CAL, TKR</td>
<td>Instrument sim</td>
<td>LAT</td>
<td>95% done</td>
</tr>
<tr>
<td>Sim</td>
<td></td>
<td></td>
<td>In use</td>
</tr>
<tr>
<td>GEANT4</td>
<td>Particle transport sim</td>
<td>G4 worldwide collaboration</td>
<td>In use</td>
</tr>
<tr>
<td>xml</td>
<td>Parameters</td>
<td>World standard</td>
<td>In use</td>
</tr>
<tr>
<td>Root 4.02.00</td>
<td>C++ object I/O</td>
<td>HEP standard</td>
<td>In use</td>
</tr>
<tr>
<td>Gaudi</td>
<td>Code skeleton</td>
<td>CERN standard</td>
<td>In use</td>
</tr>
<tr>
<td>doxygen</td>
<td>Code doc tool</td>
<td>World standard</td>
<td>In use</td>
</tr>
<tr>
<td>Visual C++/gnu</td>
<td>Development envs</td>
<td>World standards</td>
<td>In use</td>
</tr>
<tr>
<td>CMT</td>
<td>Code mgmt tool</td>
<td>HEP standard</td>
<td>In use</td>
</tr>
<tr>
<td>ViewCvs</td>
<td>cvs web viewer</td>
<td>World standard</td>
<td>In use</td>
</tr>
<tr>
<td>cvs</td>
<td>File version mgmt</td>
<td>World standard</td>
<td>In use</td>
</tr>
</tbody>
</table>
Instrument Simulation and Reconstruction

Source Fluxes

Particle Transport

"Raw" Data

Instrument data

Recon

3 GeV gamma recon

Background Rejection

Particle ID

Full geometry in xml with C++ interface
G4 discovers instrument from the xml

CAL Detail
Data flow in the Gaudi framework

- **Raw data**
  - Level 0
  - Level 1
  - Root

- **Transient Data Store**
  - Source Generators
  - Ntuple Service
  - Ntuple

- **Digitization Algorithms**
  - ACD, TKR, CAL, Trigger

- **Reconstruction Algorithms**
  - ACD, TKR, CAL

- **Simulation Algorithms**
  - G4

- **MC**
  - Real Data

- **Ready for astronomy**

R.Dubois
Example of Using Gaudi Tools

IEnergyCorr* m_lastLayerTool;
sc = toolSvc()->retrieveTool(m_lastLayerToolName, m_lastLayerTool);
m_lastLayerTool->setTrackSlope(slope);
m_lastLayerTool->doEnergyCorr((*it)->getEnergySum(),(*it));

Retrieve tool by name via base class
Refer to base class functions.
Does not know which concrete tool it is.

Concrete classes that customize behaviour

Tools id’ed by name in ascii config file (“jobOptions”)
Gaudi Interface to Geant4

http://www-glast.slac.stanford.edu/software/core/documentation/reviews/G4Generator/g4greview.pdf

R.Dubois
Data Challenges

• Ground software is amalgam of HEP instrument software and Astro FTOOLS

• Adopt HEP’s “Data Challenges” to create a series of end-to-end studies: create a progression of ever more demanding studies

• DC1. Modest goals. Contains most essential features of a data challenge.
  • 1 simulated day all-sky survey simulation
  • find GRBs
  • recognize simple hardware problem(s)
  • a few physics surprises
  • Exercise all the components

• DC2, start beginning of CY06. More ambitious goals. Encourage further development, based on lessons from DC1. One simulated month.

• DC3, in CY07. Support for flight science production.
DC Components

- Focal point for many threads
  - Orbit, rocking, celestial coordinates, pointing history
  - Plausible model of the sky
  - Background rejection and event selection
  - Instrument Response Functions
  - Data formats for input to high level tools
  - First look at major science tools – Likelihood, Observation Simulator
  - Generation of datasets
  - Populate and exercise data servers at SSC & LAT
  - Code distribution on windows and linux

- Involve new users from across the collaboration

- Teamwork!

R.Dubois
The Simulated DC1 Sky

Extragalactic diffuse

Galactic diffuse

Our Sky

EGRET 3EG

Fiddling 3C273/279

R.Dubois
GLAST Large Area Telescope:
Reconstruction

Tracy Usher
Stanford Linear Accelerator Center
usher@slac.stanford.edu

http://www-glast.slac.stanford.edu/software
GLAST Reconstruction
Anatomy of a “Typical” Event

Pair production is the dominant photon interaction in our energy range

- **Reconstruction Goals:**
  - Incident Gamma Direction and Energy
  - Reject Backgrounds
- Incident Gamma converts in the tracker
  - In particular, conversion occurs in one of the converter foils – ie at a well defined location
- Resulting electron-positron pair range out of tracker (TKR)...
  - No magnetic field, tracks are “straight lines”
  - Resulting two tracks “point” back to incident Gamma
- And into the CsI Calorimeter (CAL)
  - Measures total energy of electron-positron pair
    - = Gamma energy
- Surrounding Anti-Coincidence Detector (ACD) vetoes any wayward charged particles
GLAST Reconstruction
What makes it challenging...

• Track Opening Angle ~0
  – Resolve
    ~ 2 * 228 um / 30 mm = ~15 mr

< ~50 MeV photons to resolve tracks without “help”

• Looking for “v”s may not be the correct strategy for gamma direction reconstruction
  – Well... see next slides...
GLAST Reconstruction
What makes it challenging...

• Tracker has a lot of material
  – Actual tracker is ~ .3 rl
    • Could live with this...
  – Converter foils are ~ 1.1 rl
    • Love them: convert gamma
    • Hate them: tracking electrons
  – Total ~ 1.4 rl
    • For particles traversing active area of tracker
    • Does not include walls between towers, etc.

• Issues to deal with
  – Gammas can (and do) convert outside the foils
  – $e^+e^-$ pair interact with tracker
    • Multiple scatter
    • Primary $e^+$ or $e^-$ can stop in the tracker
    • $e^+$ and $e^-$ radiate energy
    • etc.
GLAST Reconstruction
What makes it challenging…

- Calorimeter Issues
  - Measure Event Energy – Not Track Energy(ies)
    - Don’t have resolution to separate
    - Large fraction of measured energy from Brems
    - Implications for determining gamma direction when you do have two track events…
  - Measure Fraction of Event Energy
    - Energy “loss”
      - in tracker
      - Leaking out of Calorimeter
    - Significant contribution at
      - lower energies (e.g. < 1 GeV)
      - for conversions starting higher in the tracker
    - Must augment total energy determination with contribution from tracker

Note energy flow in direction of incident Gamma
GLAST Reconstruction

- Summary: Slightly more complicated than first thought
- But still follow the “Standard” HEP Approach
  - Tracking
    - Change Goal slightly
      - Still look for two tracks
        » Multiple Scattering separates them
      - But emphasize the “longest, straightest” (highest energy) track
    - Algorithms to assign energy to tracks in final fits
    - Provide enough information to reject “bad” events
  - Calorimetry
    - Look for total event energy
    - Algorithms to correct for
      - Losses in the tracker
      - Leakage
      - Etc.
  - Both: Algorithms to help reject background
Tracking Overview
“Standard” HEP Approach

### Raw Data
1) Hit Strips
2) “ToT”

### 1) Clustering:
Associate adjacent hit strips to form clusters

- **Cluster Centroid** (Center of hit strips)
- **Hit strip** (strip pitch: 228 μm)
- **Cluster Width** (Number of hit strips)

### 2) Track Finding:
Assume start point, direction and energy,
Follow track and attach clusters allowing for deviations due to multiple scattering, etc.

**T. Usher**
Tracking Overview
“Standard” HEP Approach

3) Track Fit:
Kalman Filter Fit to get final Track Parameters

4) “Vertexing”:
Combine Tracks to determine conversion point and direction of the incident Gamma
Note: a “vertex” can consist of only one track

“Vertex” vector resulting from combining the two fit tracks in this event
Vertex Position = gamma conversion point
Vertex Vector = gamma direction

Clusters
Tracks
Errors at Clusters from Kalman Fit

T. Usher
Calorimetry Overview

"Raw" Data
Corrected Energy, "Position" per Crystal (Xtal)

Corrections/Calibrations applied before this step!

MC Display of hit Crystals:
Divides Xtal longitudinally into eight sections for display
Shows Hit Xtal, does not show energy deposited - can't really see shower development

Clustering:
Associate hit Xtlas together to form clusters

Gamma Energy: All hit Xtlas are from incident Gamma - Energy is sum over all hit Xtlas

Backgrounds: Isolated sets of hit Xtlas from own "cluster"

T. Usher
Energy Correction Algorithms:
1) Shower Profile
2) Inefficiencies due to Geometry
3) Leakage
4) etc.

(see next slides)
Energy Correction Algorithms:
Inefficiencies due to Geometry

Given the direction of energy flow (from the reconstructed Gamma direction), can apply geometric corrections to account for energy “lost”, e.g. between towers
Putting It All Together

• Chicken…
  – Track Finding/following needs starting values:
    • Initial Position
    • Initial Direction
    • Initial Energy
  – This from the Calorimeter…

• Or Egg?
  – Energy Correction algorithms need gamma direction
  – This from the tracking…

• Solution: Iterative Reconstruction
  – First Pass
    • Calorimeter Reconstruction
      – Through Clustering
        » Total Energy
        » Cluster Centroid
        » Cluster Axis
    • Tracker Reconstruction
      – Track Finding/Following
      – Track Fit and Vertexing
        » Good enough for 2nd pass
  – Second Pass
    • Calorimeter Reconstruction
      – Energy Correction Algorithms
    • Tracker Reconstruction
      – Track Fit and Vertexing
        » Use “improved” energy
Background Rejection
Example: Charged Particles in Tracker

- Project Track to plane of struck tile
- Calculate distance to nearest edge
- Sign
  - Positive if track projection inside the tile
  - Negative if track projection outside the tile
- Reject if inside the tile

Extra: Min I signature in Calorimeter

"Active Distance"
inside tile boundary
outside tile boundary
no tile hit

T. Usher
End of Reconstruction

Two Levels of Output at end of Reconstruction:

- **Root Trees**
  - Basically, all the output of all steps of reconstruction
  - Enough information to read back in and continue reconstruction from that point
  - Detailed offline analysis for reconstruction algorithm improvements
  - Main component of System Tests

- **Output Ntuple with two branches**
  - A detailed branch which contains enough information for checking of reconstruction performance
  - The analysis branch which is passed on to the next stage…

T. Usher
LAT Science Tools for Gamma-Ray Astronomy

James Chiang
GLAST Science Support Center
jchiang@slac.stanford.edu
The Gamma-Ray Sky

- EGRET All-Sky Map and 3rd EGRET Catalog:

<table>
<thead>
<tr>
<th>Source</th>
<th>3EG</th>
<th>GLAST</th>
<th>3EG</th>
<th>GLAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGNs</td>
<td>94</td>
<td>67</td>
<td>∼3000</td>
<td>Unids</td>
</tr>
<tr>
<td>Pulsars</td>
<td>5</td>
<td>∼10s</td>
<td>Sol. Flare</td>
<td>1</td>
</tr>
<tr>
<td>galaxies</td>
<td>1(LMC)</td>
<td>&gt;1?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

J. Chiang: Dark Matter, SNRs, etc.
Example Source Class: Blazars

- Spectral Energy Distribution (SED)

Blazars (cont.)

- Radio morphology and its evolution implies a relativistic outflow (jet):

  ![Blazar Morphology](http://chandra.harvard.edu/photo/2002/0157/more.html)

Multi-wavelength Observations

- Light curves and rapid variability across wavebands, e.g., 3C279 in 1996 (Wehrle et al 1998) require coordinated monitoring efforts with other missions and ground-based teams
- After 1st year, all data becomes public immediately

∴ It must be straight-forward to analyze LAT data by investigators outside of the collaboration.
Data Analysis for High Energy Astronomy

- Framework driven by
  - Desire for uniformity between missions
  - Guest observer support (HEASARC)
  - Aggregate nature of the data: events are (almost) never analyzed individually

- Standardized software and data formats
  - FITS files for images and tabular data
  - FTOOLS for examining and manipulating contents
    - Can be mission-specific
    - User interfaces – “parameter interface layer”, ballistic operation
    - Often scripted (Tcl, Perl), some GUI use
  - High level analysis applications: Xspec (from Xanadu suite), Sherpa, ISIS (from CIAO), etc.
  - Unix-based tradition; GLAST pushing for Windows support
Instrument Response Functions (IRFs)

- The **linchpin** between the event reconstruction and Science Tools
- The IRFs are a statistical description of the performance of the LAT for measuring photon properties, e.g., a transition matrix.
- They are derived from real calibration runs using a photon source (e.g., real data + recon) and/or from Monte Carlo simulations using GlastRelease (GEANT 4 + recon).
- The total response, $R$, is usually factored into three components:

$$ R(E', p'; E, p) = A(E, p) \, D(E'; E, p) \, P(p'; E, p) $$

<table>
<thead>
<tr>
<th>Measured 4-momentum</th>
<th>Effective Area (cm²)</th>
<th>Point Spread Function (sr⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True photon 4-momentum</td>
<td>Energy Disp. (MeV⁻¹)</td>
<td></td>
</tr>
</tbody>
</table>

- $R(E', p'; E, p)$
- $A(E, p)$: Effective Area
- $D(E'; E, p)$: Energy Dispersion
- $P(p'; E, p)$: Point Spread Function

J. Chiang
Example IRF Generation: DC1

- ~5 M “AllGamma” events are generated covering $2\pi$ sr and spanning energies 20 MeV to 200 GeV.
- Effective Area -- detector “cross-section” as a function of energy:

LAT performance is strongly impacted by cuts on particle background
Point Spread Functions (PSFs)

- Data are partitioned into broad energy and inclination bins
- Angular deviations are scaled by $\sim E^{-1}$ to account for multiple scattering
Energy Dispersion

Plots from /home/chiang/StarAnalysis/Data/energy_fit_thin.root Mon Jul 5 19:02:52 2004
Using the IRFs for Simulation

• Assuming perfect knowledge of incident photons, the distributions of measured quantities should (ideally) be identical for all three ways of obtaining them:
  – Real observations + Gleam reconstruction
  – Gleam simulation + recon
  – IRFs

• For Science Tools development and testing, we have developed a high level observation simulator that reads in the same sky model as Gleam, but uses the IRFs to produce simulated events:
  – Source flux (photons cm$^{-2}$s$^{-1}$) \times A (cm$^2$) = rate of detected events
  – True photon 4-momentum & $P$ & $D \Rightarrow$ Apparent photon 4-momentum, i.e., smeared by instrument response
IRF Simulation vs Gleam
Typical Steps in a LAT Analysis

- Acquire data
  - download from GSSC server
- Preliminary visualization
  - counts and exposure maps
- Analysis-specific data selections
  - GTIs, ROI, event type
- Source identification
  - Source detection and identification: image processing techniques, wavelet analyses, etc. (should be fast).
- Source characterization
  - Maximum Likelihood estimate (MLE) of source properties—flux, spectrum, position (computationally expensive).
  - Multi-wavelength spectral fitting (using Xspec).
South Atlantic Anomaly (SAA) passages handled by “good time intervals” (GTIs)...

event data are partitioned into “regions-of-interest” (ROIs)...

and by event type, eg. “front” vs “back” (depends on IRF granularity)
Source Identification

- De-noising and deconvolution (wavelets, etc.)
- Source finder (preferably automated)

Input counts map
1 week simulation time

Deconvolved map using EM algorithm ⇒ MLE
Galactic Diffuse and Source Confusion

Emission results from cosmic ray interactions with interstellar gas.

Models rely on HI & CO observations for the gas distribution.

These observations reveal structures on angular scales similar to the PSF:
\[ \sim 3.5^\circ \text{ @ } 100 \text{ MeV} \]
\[ \sim 0.1^\circ \text{ @ } 10 \text{ GeV} \]
Source Characterization

- Maximum likelihood for ascertaining source parameters
  - flux, spectral index, source position
  - > 50 parameter fits for a single ROI are common
Types of Gamma-Ray Sources

• Pulsars
  – Rapidly rotating neutron stars \((P \sim 10^{-3}-10\text{s}, (dP/dt)_{\text{Crab}} \sim 10^{-13}\text{ s/s})\) with \(B \sim 10^9-10^{12}\text{ G}\)
  – Ephemerides from radio observations

• Blazars
  – Variability over a wide range of time scales (hours to months)
  – Multi-wavelength monitoring is crucial

• Gamma-ray Bursts
  – Very short time scales, < 10s of seconds

• Diffuse/extended emission
  – Milkyway galaxy, LMC, supernova remnants
  – Extragalactic diffuse may comprise unresolved discrete sources such as blazars

• New physics:
  – Dark Matter sources

J.Chiang
Pulsars

Gemini Pulsar

Crab Pulsar

J. Chiang
Gamma-Ray Bursts
For more on ScienceTools...


http://glast.gsfc.nasa.gov/cgi-bin/ssc/LAT/STCDataQuery.cgi
GLAST Large Area Telescope:
Exploring the $\gamma$-ray Sky

Daniel Flath
Stanford Linear Accelerator Center
dflath@slac.stanford.edu

http://www-glast.slac.stanford.edu/software
Data Handling: Outline

- **Automation: Pipeline**
  - Guaranteed 1 hour turnaround
  - 24x7x365x10
  - Parallel processing of data – Workflow
  - Web based monitoring
- **System Tests**
  - Monitoring data and software
  - Web based
- **Data Server**
  - Public data server, for public
  - Glast data server, for collaboration
    - Why? (pull region of sky from many orbits)
    - Users don’t (want to know) much about recon/simulation
- **Technologies (to be) used**
  - 3rd Generation Web Application Containers
    - Web applications that work like desktop applications
  - Mix of commercial products and Open Source projects
    - JIRA
      - Bug tracker
      - Project management tool
    - Confluence
      - Documentation repository
      - “Super WIKI”
  - Commercial tools
    - Themselves based on Open Source projects
    - Examples of 3rd Generation web applications
L0 Data Arrives At Data Processing Facility (Pipeline)

Input Dataset(s) logged to Processing Database (PDB)

Processing orders and details retrieved from PDB Processing Orders and details forwarded to Batch Submitter

Batch Submit Server Notifies DPF of Processing Progress

Process scheduling info logged to Batch Submit Database

Process submitted to LSF

Process exit status and statistics logged to PDB

Datasets made available to Data Servers

---

DS: Data Server

BS: Batch Server

LSF: Load Sharing Facility

SP/F: Storage Pool/Files

Oracle: Database System

MySQL: Database System

GSFC: Goddard Space Flight Center

SLAC: Stanford Linear Accelerator Center

NASA: National Aeronautics and Space Administration

GLO: Global Orbit

66/85: Page Number
Pipeline Intro

• What is the pipeline?
  – Envisaged as tool to provide a tree of processing on a given input dataset
  – Full bookkeeping to track what happened
  – Archive all files touched

• Used by whom?
  – Online
    • for sweeping integration data out of the clean room and to tape
    • populate eLogbook
  – SVAC (Science Verification and Calibrations)
    • for doing digi, recon
    • creating reports
    • Preparing for calibrations
  – Generic MC
    • DC2, background runs etc etc
  – ISOC (Instrument Science Operations Center)
    • Flight operations
    • What about environmental testing, at Spectrum Astro, KSC?
Sample Processing Chain

- JobOpt.txt → MC → MC.Root → MCRreport → MCRreport.out
- DigiReport → DigiReport.out
- Recon1 → Recon1.root
- Recon2 → Recon2.root
- ReconN → ReconN.root
- Recon.root → ReconReport → ReconReport.out
Current Pipeline: Major Components & Tech Used

- **RDBMS** (relational database management system)
  - Oracle
  - Contains all processing and data product history and relationships
- **Data Exchange Layer**
  - Oracle PL/SQL
  - Compiled SQL queries provide read/write access to tables
- **DB Access Layer**
  - Perl::DBI
  - Auto-Generated subroutines wrapping every public stored function and procedure
  - Provides simple, seamless DB interface to Perl Utilities
  - Also Perl classes representing each record type
- **Scheduler, utilities**
  - Perl
  - Higher level code to manage data and processing
  - Little dependency on actual table structure gives developer freedom to write maintainable, extensible code
Pipeline Component Dependencies

- Oracle Data
  - Oracle Stored Procedure Code (PL/SQL)
- Perl Stored Procedure Wrapper Library (auto-generated)
  - Perl Data Storage Classes
  - Perl Data Management Code (logical operations)
- Pipeline Scheduler & Management Utilities (Perl)

D.Flath
System Tests

- **Goals**
  - Provides mechanism for validating:
    - Software releases (now)
    - Data quality (after launch)
  - Run (automatically) after each software release
    - Compares plots to references and flags problems
- **Web based access to system tests results from any platform**
  - No software install needed
  - Accesses data from combination of
    - Oracle database tables
    - Root files
  - Implemented using JAIDA, xrootd, JSP, Tomcat
System Tests

D. Flath
System Tests

D. Flath

IT&Tea May 12 2004
Data Server

- Glast will run two data servers
  - One for the public at Goddard Space Flight Center
  - One at SLAC for Glast collaborators
- Glast Physicists will access data via Data Server
  - Pulls events associated with
    - Particular region of the sky
      - Satellite doesn’t stay still so this is spread throughout data.
    - Energy range
    - Time Period
  - Removes need for users to know how/where data is stored
    - For most astrophysics measurements physicists only need to know about photon direction and efficiency, details of reconstruction/simulation are largely irrelevant
  - Should be able to download data in various formats
    - List of run/events
    - Tuples (FITS, root, possibly with choice of number of columns)
    - Full root trees
  - Should be able to browse events
    - with web based event display (WIRED)
  - Should be able to store personal favorite searches
    - Should be able to download incremental updates to data
- Expect to get 100M events/year for 10 years
  - Small compared to Babar, but we want fast turnaround
Data Server

You selected 39383844 events
Change Criteria
Add “TCut”
Browse Events
Download: Event Selection
Compressed Tuple
Full Merit Tuple Full Root Tree

Web Form
Region in Sky:
Time Range:
Energy Range:
Gammas/Events:

In memory meta-data
Binned by sky position, time, energy

Root Event Store

D.Flath
Data Server

Glast Data Server Home

Enter GDS.

This application allows you to select a subset of Glast events, to visualize their data and/or to download the associated datasets.

To access this web service you must be registered in the Glast authentication server.

The GLAST Ground Software portal presents the list of application currently available.

Selection by parameters

Parameters: text

Energy range: min 1.0 max 10000000 (12.0 - 642792.6)

and

Quality range: min max (0.0 - 10.0)

and

Location: ra 20.0000 decl 0.0000

and

Area to search: ra 15.0000 decl 15.0000

and

Observation

D.Flath
Data Server

Selection by TCut

TCut: [Input field]

Notify: [Input field]

apply

TCut command sent

TCut: [Input field]

Your Job Id is: 525144.
Results will be found in http://ftp-glast-slac.stanford.edu/glast-a13/DataServer/1115258555942.

Done
Future Data Server

- Future plans for Data Server:
  - Instead of delivering data via FTP, use real-time streaming
Back to the Future:
Adopt 3rd Generation Web Applications

• Program as little as possible, declare as much as possible
  – State your requirements and intentions in configuration files
  – Write code only for your specific problem domain, leave the rest to the container
Brief History of Web Application Platforms

First Generation 1994-1997
Dynamic Web Pages

- PHP
- Perl-CGI
- ASP

Second Generation 1998-2002
Application Servers

- J2EE
- .NET

Third Generation 2002-Present
Lightweight Containers

- Java
- Spring
- Hibernate

Approach Adopted by GLAST Offline

References
- http://www.springframework.org
- http://perl.apache.org/about/history.html
MVC type 2
Model View Controller

1. Request
Browser

2. Create
Controller: Java Class

3. Database access

WebApplication Server

4. SQL
Data

Service
(Data Access)

5. Forward To
Model: Java Class

6. Use-
View (JSP, etc)

Guaranteed workflow (page order)
Guaranteed security (force login, HTTPS, Role membership, etc.)
Coarse-grained security (directory, page, page fragment)
Fine-grained security (member functions, arguments, element in returned array)
Auto-populate web forms
Data validation (date ranges, run ranges, etc.)
JIRA Web Application

Project: Pipeline Front End

GLAST Pipeline Front End

Key: PFE
URL: http://glast-ground.slac.stanford.edu/
Lead: Matt Langston
Default Assignee: Project Lead
Notification Scheme: None
Permission Scheme: Glast
Issue Security Scheme: None
Field Layout Schemes: System Default Layout
Workflow Scheme: None
CVS Modules: None
Mail Configuration: Mail notifications from this project will come from the default address
Project Category: Glast Offline Infrastructure

Components
- Add new component, select assignees for components
- Run Summaries (Lead: Matt Langston)
- XML Configuration Files (Lead: Matt Langston)

Versions
- Manage versions:
  - v0r1p0
  - v0r2p0
  - v0r3p1
  - v0r3p4
  - v1r0p0
  - v1r1p0
  - v1r2p0

Project Configuration

Define Upcoming Versions and Assign Issues
### JIRA as a Project Management Tool

#### Manage Versions

On this page you can manage the versions for the Pipeline Front End project.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Release Date</th>
<th>Schedule</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0r1p0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v0r2p0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v0r3p1</td>
<td>Support XML upload to test database in addition to prod and dev.</td>
<td>11/Jan/05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v0r3p4</td>
<td>Bug fix release.</td>
<td>22/Mar/05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1r0p0</td>
<td>First version using JSP.</td>
<td>12/May/05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1r1p0</td>
<td>Integrate outstanding issues</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1r2p0</td>
<td>Integrate outstanding issues</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Add Version

Add a new version to the project Pipeline Front End.

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Author</th>
<th>Status</th>
<th>Date</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFE-42</td>
<td>Dataset Catalog doesn't like Run containing non-numeric character</td>
<td>Matt Langston</td>
<td>Open 05/May/05</td>
<td>UNRESOLVED</td>
<td>v1r3p0</td>
</tr>
<tr>
<td>PFE-44</td>
<td>Corrupted log file display</td>
<td>Matt Langston</td>
<td>Open 05/May/05</td>
<td>UNRESOLVED</td>
<td>v1r3p0</td>
</tr>
<tr>
<td>PFE-45</td>
<td>Add multiple tasks in one XML file</td>
<td>Matt Langston</td>
<td>Open 05/May/05</td>
<td>UNRESOLVED</td>
<td>v1r3p0</td>
</tr>
<tr>
<td>PFE-46</td>
<td>Add a &quot;include pipelines with no runs&quot; toggle to main stats page</td>
<td>Matt Langston</td>
<td>Open 11/May/05</td>
<td>UNRESOLVED</td>
<td>v1r3p0</td>
</tr>
<tr>
<td>PFE-47</td>
<td>Can't configure Tasks on dev server</td>
<td>Matt Langston</td>
<td>Reopened 22/Apr/06</td>
<td>UNRESOLVED</td>
<td>v1r0p0</td>
</tr>
<tr>
<td>PFE-51</td>
<td>Sorting only affects items in waved page</td>
<td>Matt Langston</td>
<td>Open 05/May/05</td>
<td>UNRESOLVED</td>
<td>v1r0p0</td>
</tr>
<tr>
<td>PFE-52</td>
<td>Can't view full view</td>
<td>Matt Langston</td>
<td>Open 05/May/05</td>
<td>UNRESOLVED</td>
<td>v1r0p0</td>
</tr>
<tr>
<td>PFE-53</td>
<td>Add run statistics summary by task</td>
<td>Matt Langston</td>
<td>Open 05/May/05</td>
<td>UNRESOLVED</td>
<td>v1r0p0</td>
</tr>
</tbody>
</table>
Overall Summary

- GLAST offline software represents a confluence of HEP and Astro communities
  - Looks like HEP for instrument simulation
    - C++; Gaudi; Geant4; Root; Kalman filter tracking etc
  - Looks like a telescope for analysis of the sky
    - FTOOLS, FITS etc
  - We have a small group
    - Trying to automate as much as we can
    - Trying for good gui tools; good user doc
    - Release Manager, System Tests
    - processing pipeline
      - Backbone of Science Ops Center
  - Modest data volumes
    - Keep it all on disk
    - Provide smart data servers for the collaboration