PPA Scientific Computing Applications – “Data Handling” group overview

Tony Johnson
tonyj@slac.stanford.edu

Contents

- Who are we?
- Fermi Pipeline and Web Tools
  - Web Tools
  - Fermi Processing Pipeline Overview
  - Data Catalog, Download Manager and Skimmers
  - Database tuning and Astro Server
  - WIRED event display
  - Generalizing Pipeline and Web Tools
    - Designing for portability
    - GWT experience
- Other Web Tools
  - Confluence, JIRA, Forums
  - Subversion, Sventon, Hudson and FreeHEP
  - Group manager and collaboration databases
- Java visualization, reconstruction and analysis
  - AIDA, JAS and WIRED
  - SLIC, org.lcsim and LCIO
- Future Desires and Plans
  - Interactions with CD

Who are we?

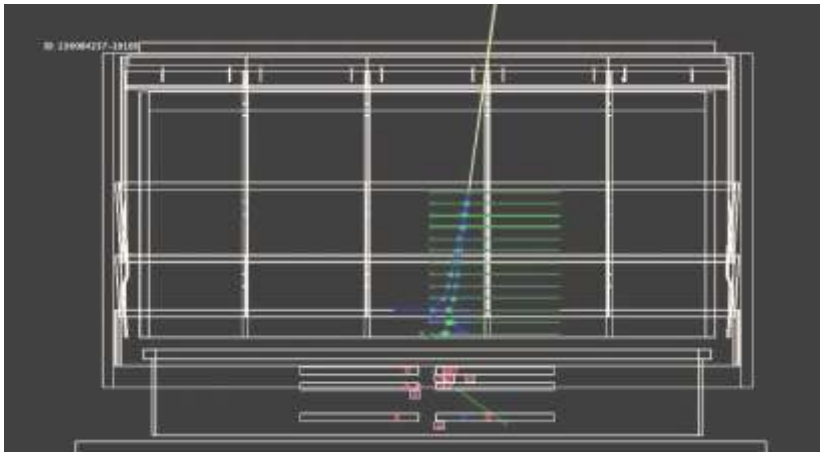
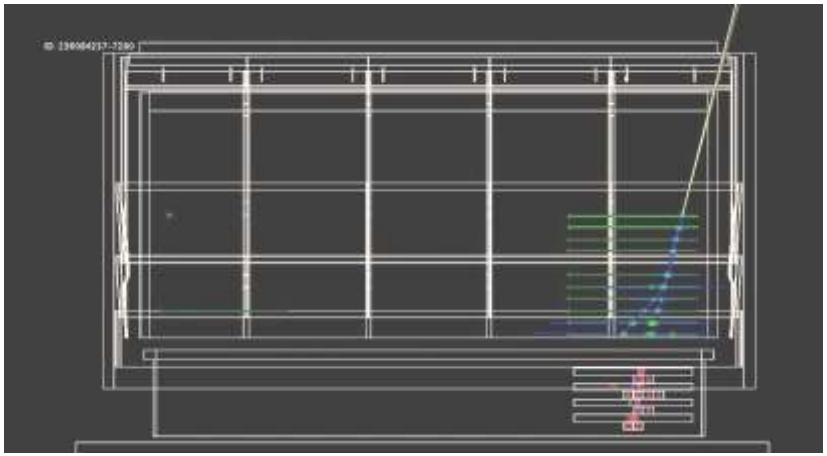
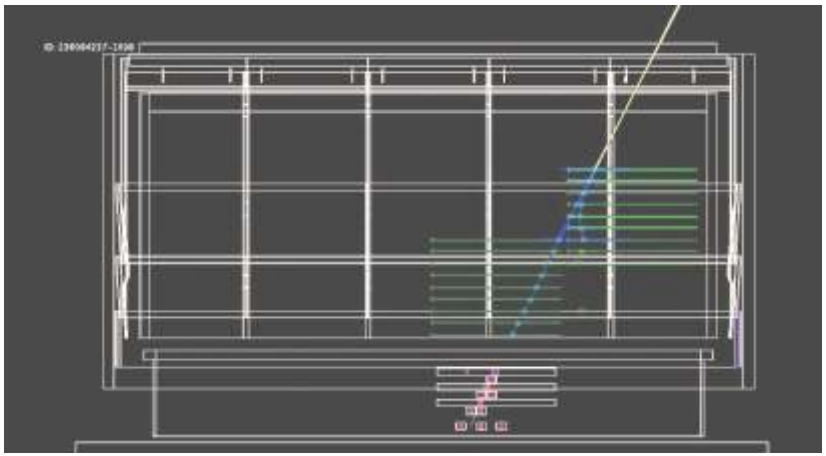
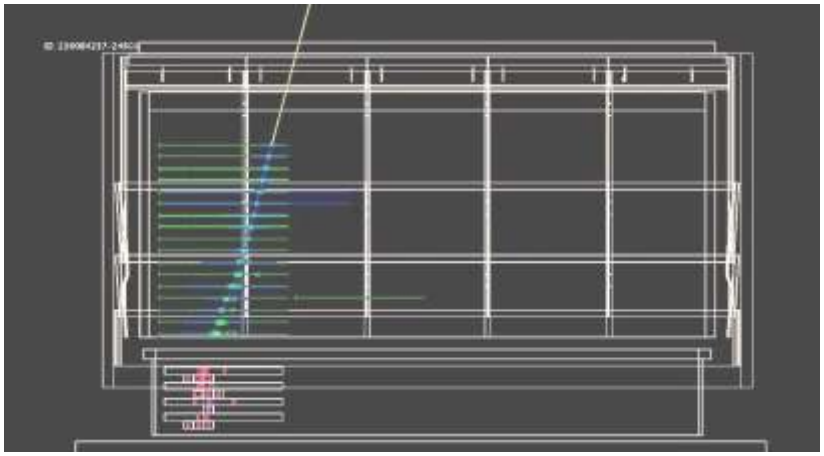
- Dan Flath
- Karen Heidenreich
- Charlotte Hee
- Tony Johnson
- Max Turri

- Much of what I describe here has been done in collaboration with many others including:
  - Tom Glanzman, Warren Focke, Richard Dubois, Norman Graf, Jeremy McCormick, …

Fermi Pipeline and Web Tools

Launched 11 June 2008 – LAT activated 25 June

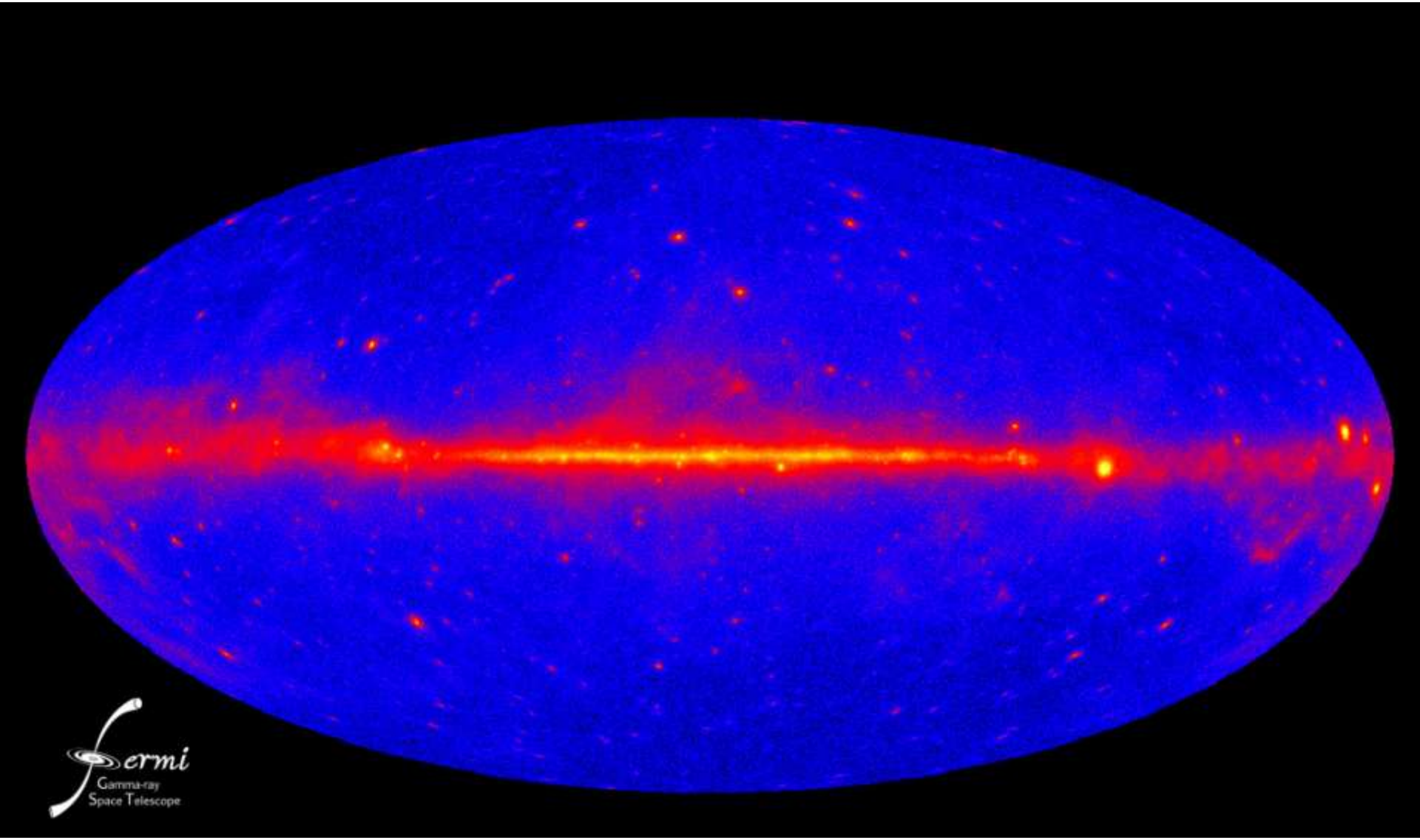
In Orbit: Single Events in the LAT

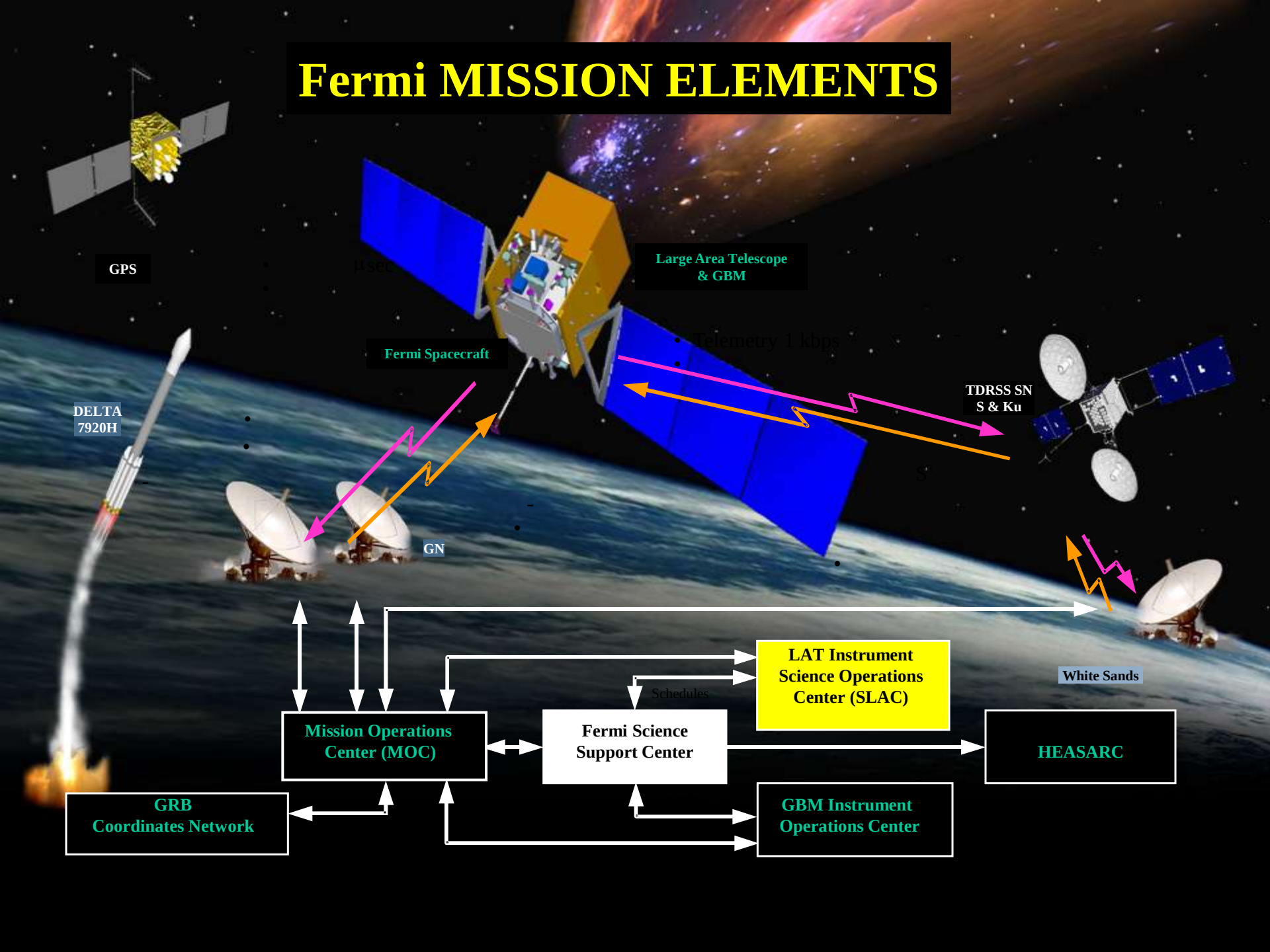
The green crosses show the detected positions of the charged particles, the blue lines show the reconstructed track trajectories, and the yellow line shows the candidate gamma-ray estimated direction. The red crosses show the detected energy depositions in the calorimeter.

0.25 CPU sec/event to reconstruct: downlink 500 Hz
Each photon event independent of others

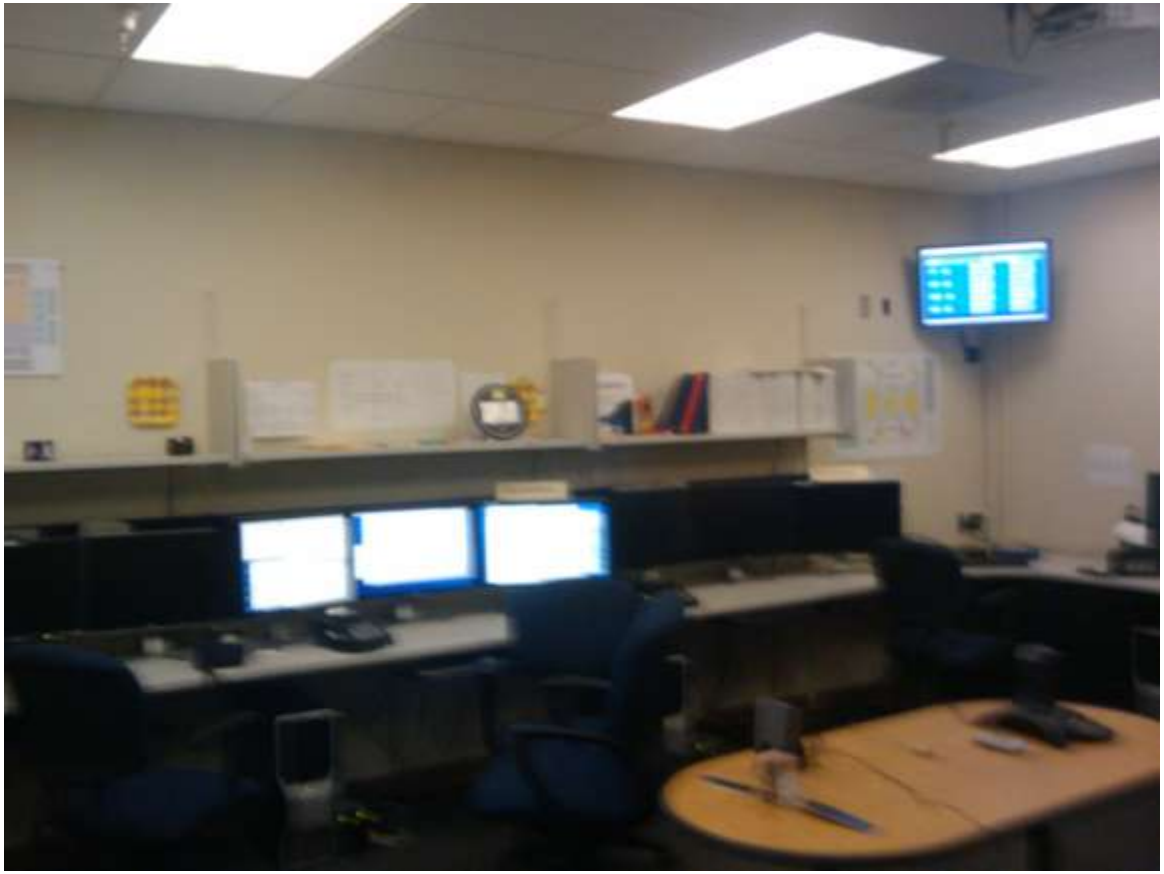
Fermi One Year All Sky Map



Fermi MISSION ELEMENTS

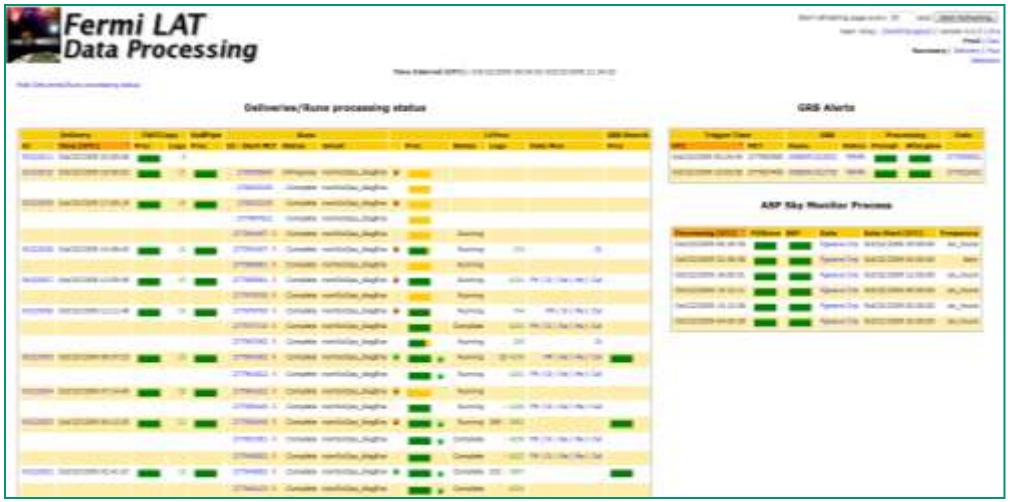


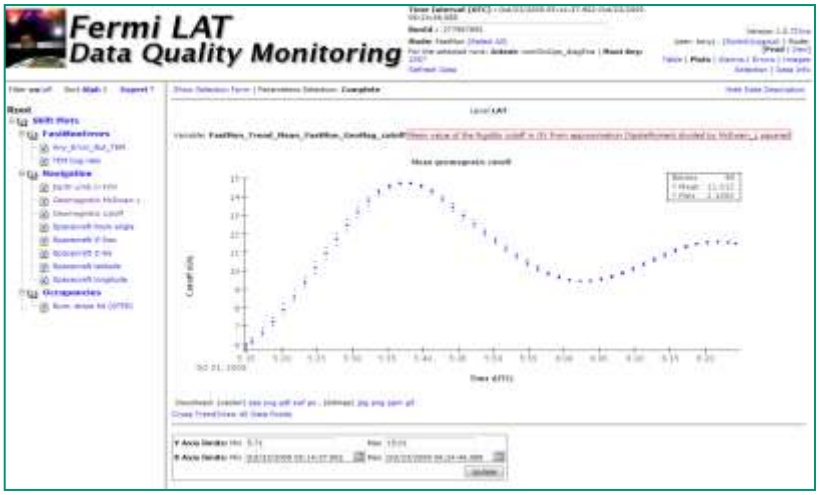
ISOC Control Room

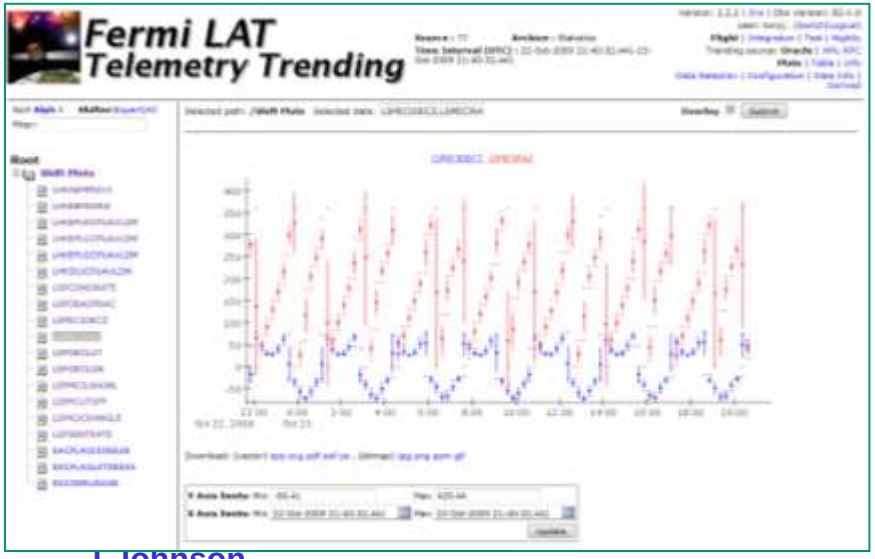


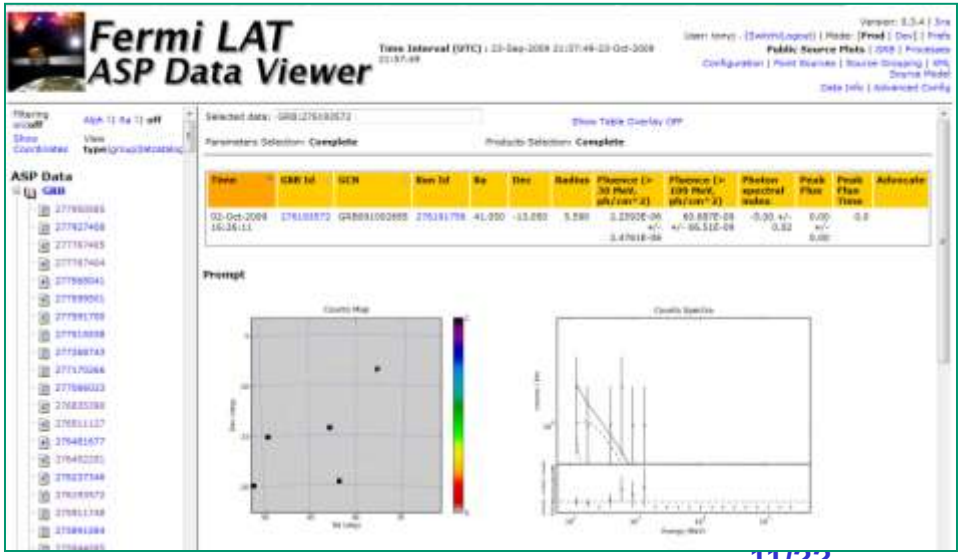
- All of the data processing and data quality monitoring can be done from the web
  - No need for anyone in the control room, monitoring load shared globally

Monitoring Pipeline + Data Quality



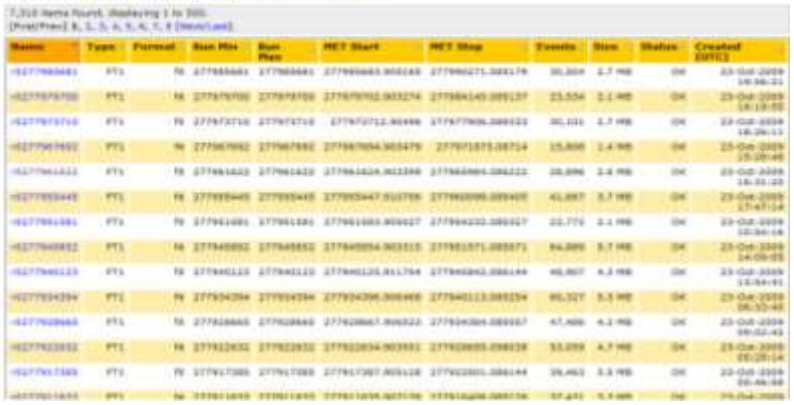


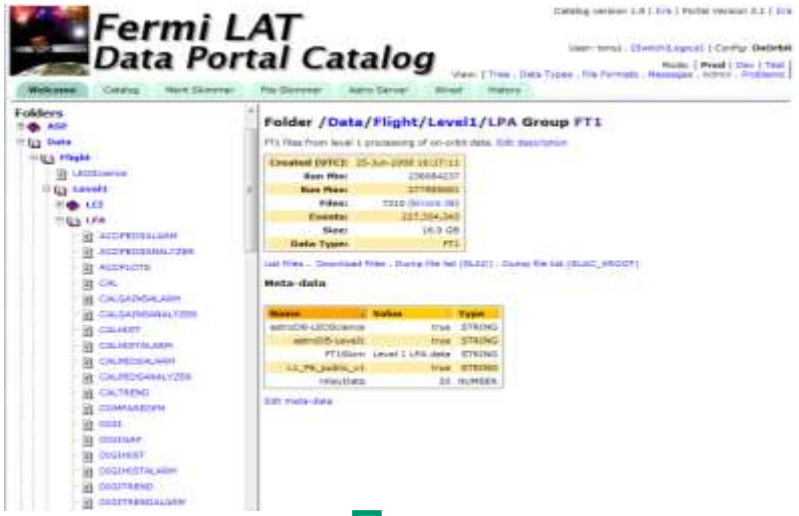




Data Access

Folder /Data/Flight/Level1/LPA Group FT1

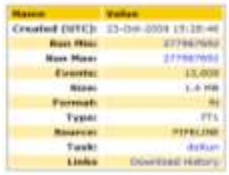




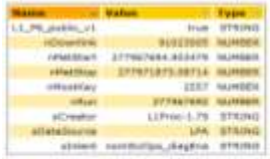
Folder /Data/Flight/Level1/LPA Group FT1

Dataset r0277967692 version 0

Standard Data



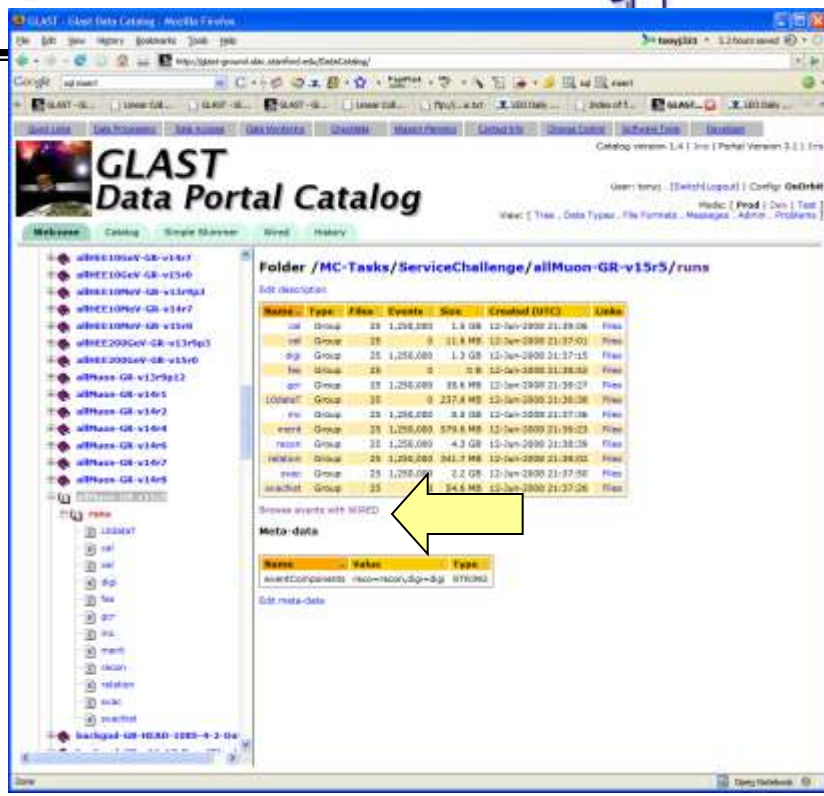
Meta-data

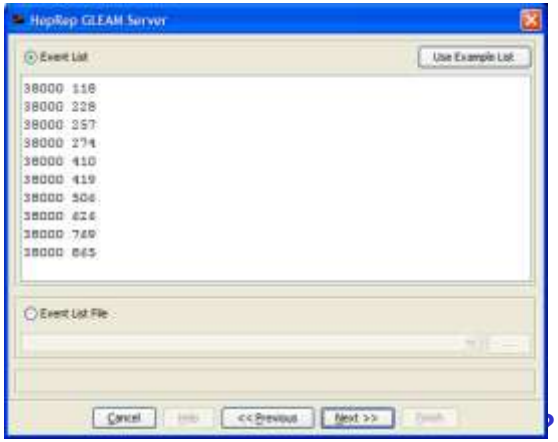


Location

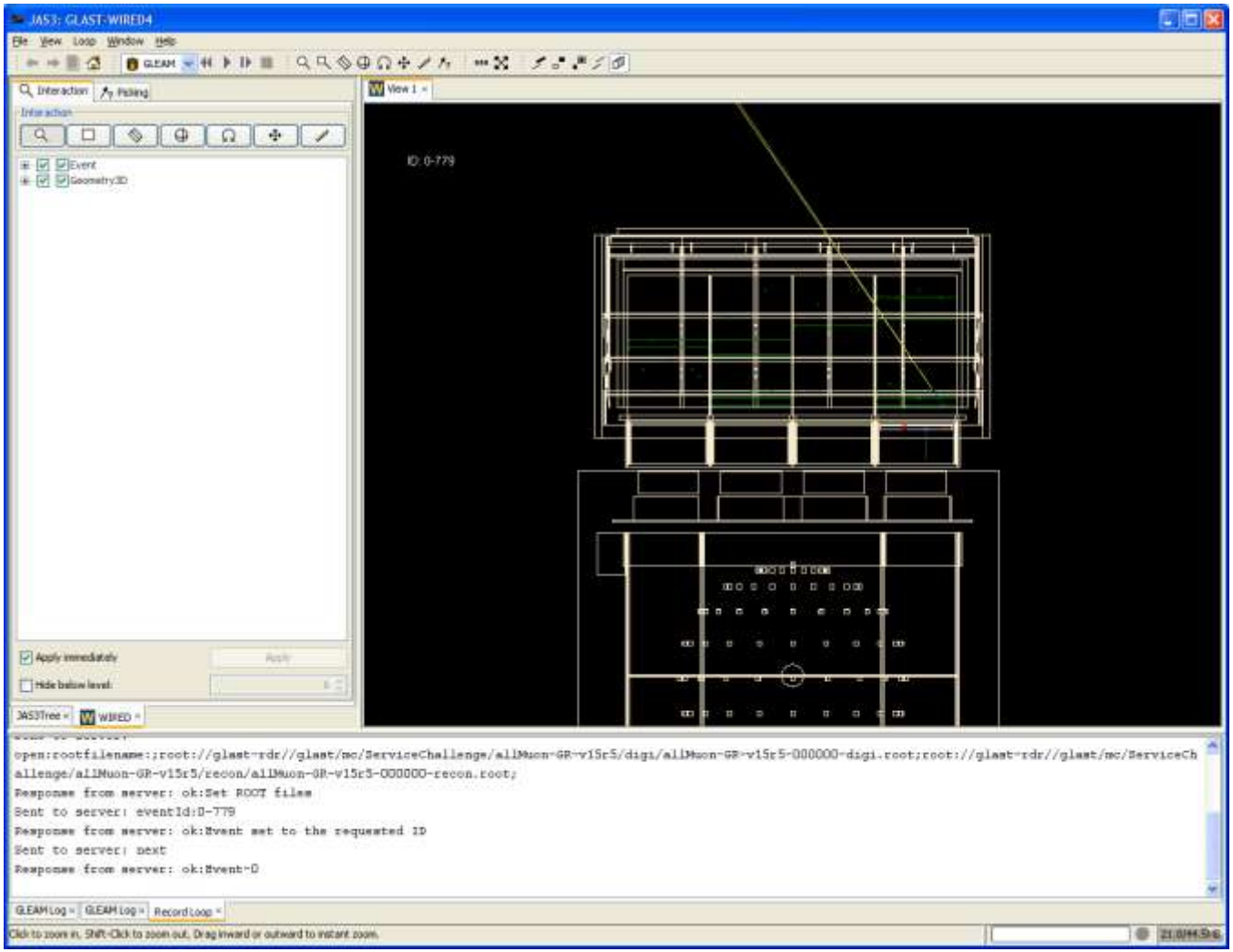
Wired4 Event Display for GLAST

- **WIRED4**
  - Java based experiment independent 3D event display
  - Many features including
    - Custom views, Dynamic Cuts
- **For GLAST set up as web start application integrated into data catalog**
  - Started by single-click on data catalog web interface. No prior installation required
    - Except Java itself – but always there thanks to EVO
  - Dialog allows user to select events of interest
  - Web service used to find location of files containing requested events from database
  - CORBA server used to fetch "HepRep" description of event
  - Event display appears…





WIRED4 Event Display for GLAST



"Astro" Server Web Interface

The P6_public_v1 event sample currently contains 190,185,596 events covering the time period 2008-08-04 15:43:36 UTC (239,557,418 MET) to 2009-10-22 11:23:53 UTC (277,903,436 MET) .

Parameter	Value
Job Name	%u-%t-%n *Arbitrary name: %u=user name, %t=job type, %n=unique id*
Event Sample	P6_public_v1 *Event selection help*
Energy Range	Min: Max: *MeV (Leave blank for no limit)*
Time Range	Min: Max: *Mission elapsed time (MET) (Leave blank for no limit)*
Position	RA: 40.1 DEC: 61.225 *degrees (Leave blank for full sky)* or astronomical object: using NED *overrides ra, dec above help*
Radius	10.0 *degrees*
Event Class	Diffuse
Output (FT2 Files)	30 second (fits) 1 second (fits)
Output (Event Data)	FT1 (fits) LS1 (fits) Merit (root) Event-List (text)
Debug Mode	False
User Comment	LS I 61+303
Expert Options	*help*
	Proceed

Number of events selected: 369311

Parameter	Value
Job Name	%u-%t-%n
Event Source	P6_public_v1
Minimum energy	
Maximum energy	
Minimum MET	
Maximum MET	
RA	40.1
DEC	61.225
Galactic Object	
Radius	10.0
Event Class	Diffuse
Output (FT2 Files)	30-second
Output (Event Data)	FT1
Debug	false
User Comment	LS I 61+303
Expert Options	
Back	Submit

Astro job submitted

Your job tonyj-AstroServer-00040 has been submitted.

Your data will be available for download from ftp://ftp-glast.slac.stanford.edu/glast.u27/DataServer/1256243366055

You will be sent an e-mail at tonyj@slac.stanford.edu when your job has completed.

You can monitor your job's progress using the Pipeline

Note: Clicking on the Status column will take you to the pipeline task that ran the job. Clicking on the Job column will allow you to rerun this task, or a similar one. Clicking on the Output Directory column will take you to the output.

Up to higher level directory

Name	Size	Last Modified
tonyj-AstroServer-00040-README.html	4 KB	10/22/2009 8:31:00 PM
tonyj-AstroServer-00040-ft1.fits	33351 KB	10/22/2009 8:39:00 PM
tonyj-AstroServer-00040-ft2-30s.fits	144206 KB	10/22/2009 8:46:00 PM

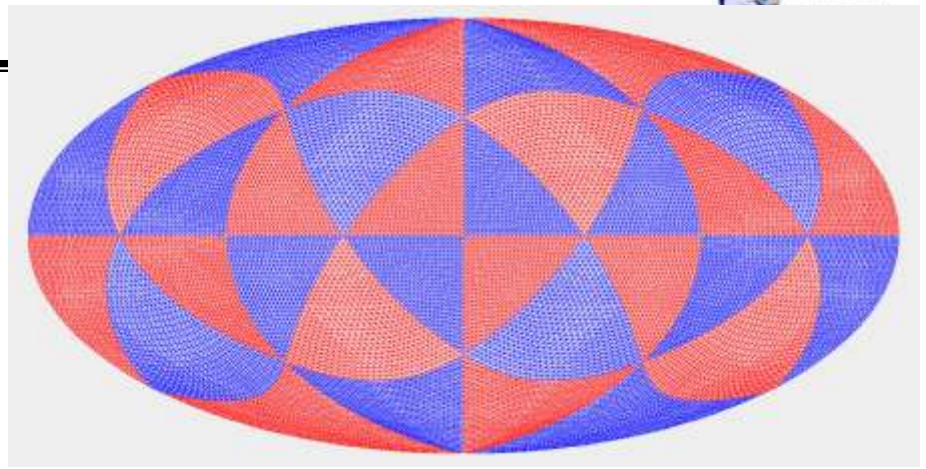
"Astro" Server Implementation

User Request
Time/Energy Range
Position in Sky
Quality

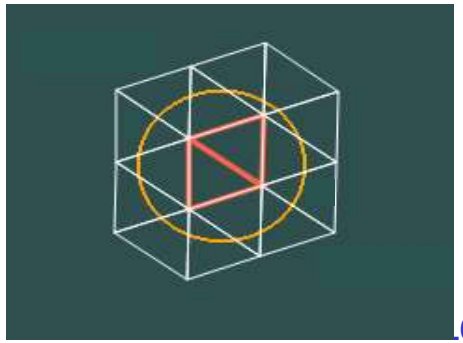
Dedicated
Oracle
Database
~200 Million
Photons

Event List

Xroot File Server
Root Tuples
Fits Files
~12 TB



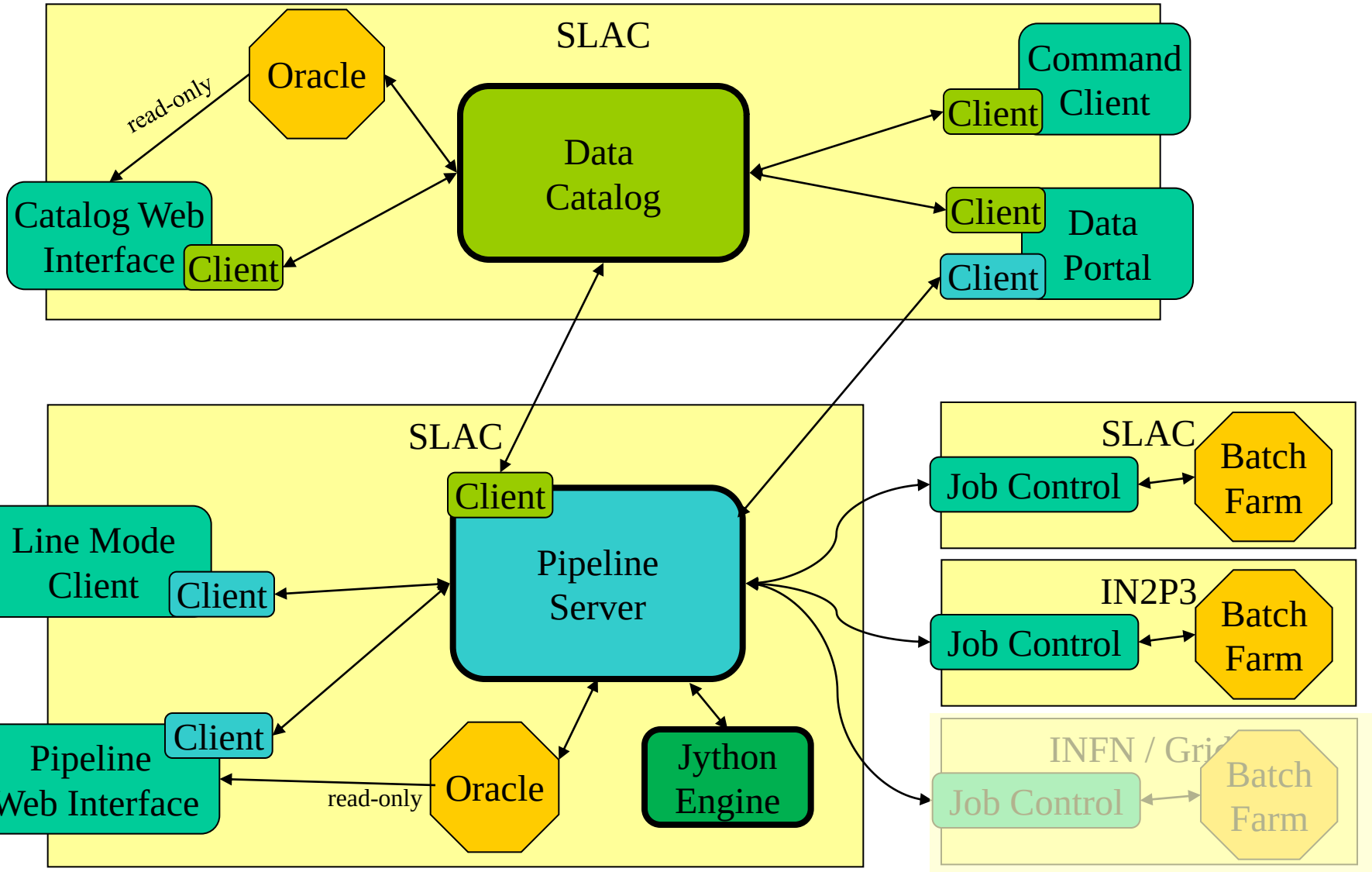
Within the database events are indexed by time, energy and position using a hierarchical triangular mesh (HTM). Database partitions are used to split the data into 1 week time bins and 32 position bins within each time bin, each containing 1024 HTM regions (shown above) . The use of HTM triangles makes it easy to identify which regions are entirely contained in the user request, and which are partially contained and require finer selection (below).



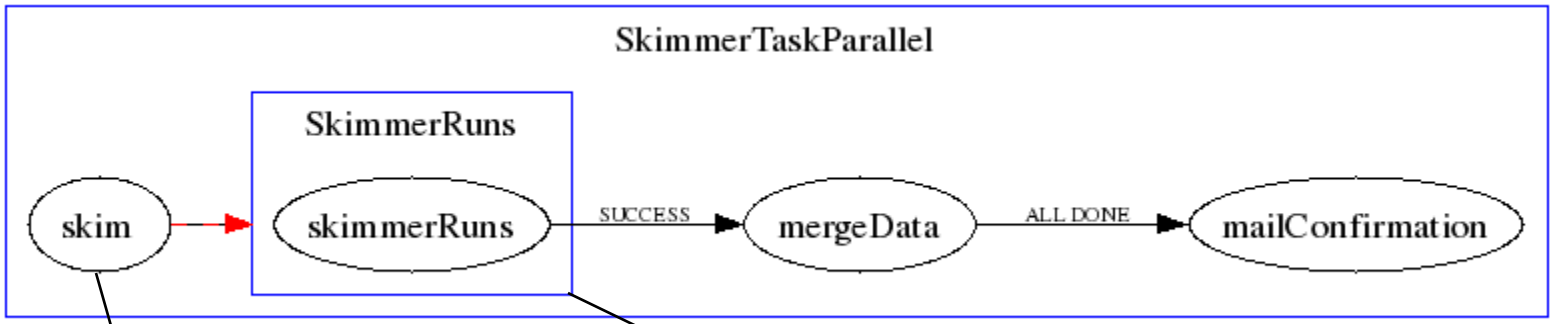
Pipeline Introduction

- Pipeline design goals
  - Automated submission and monitoring of batch jobs
    - Very high reliability
  - Ability to define graph of jobs to be run
    - Ability to parallelize processing tasks
    - Ability to perform simple computations as part of job graph
      - E.g. Compute how many parallel streams to create as a function of the number of events to be processed
  - Ability to “Roll Back” jobs (whether successful or not)
    - Capability to automatically compute sub-graph of jobs to rerun
  - Maintain full history of all data processing
  - Data catalog to keep track of all data products
  - Web interface for monitoring jobs and submitting new tasks
    - Plus command line client, and programmatic API

Pipeline and Data Catalog Components



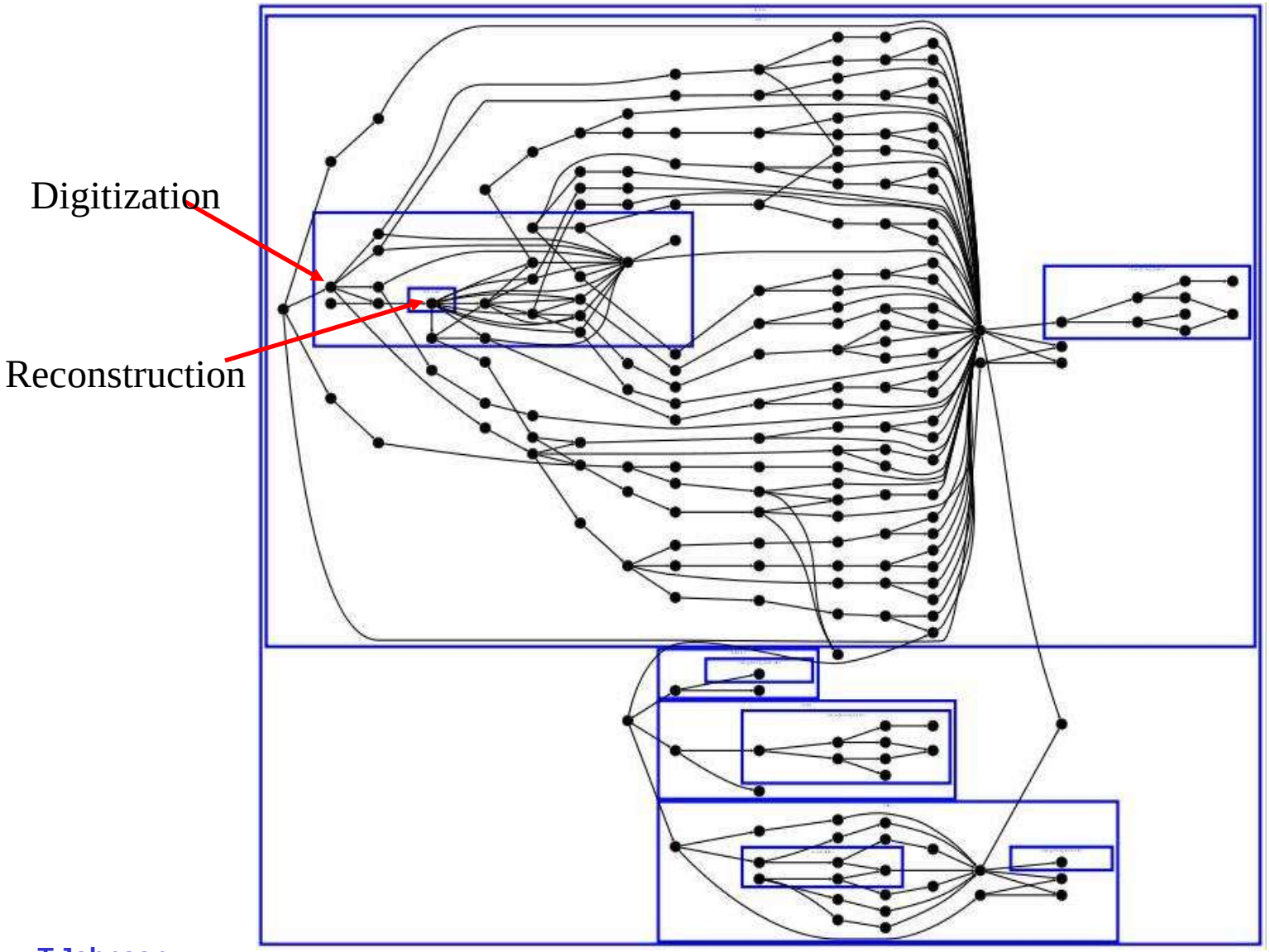
Pipeline Task specification (XML)



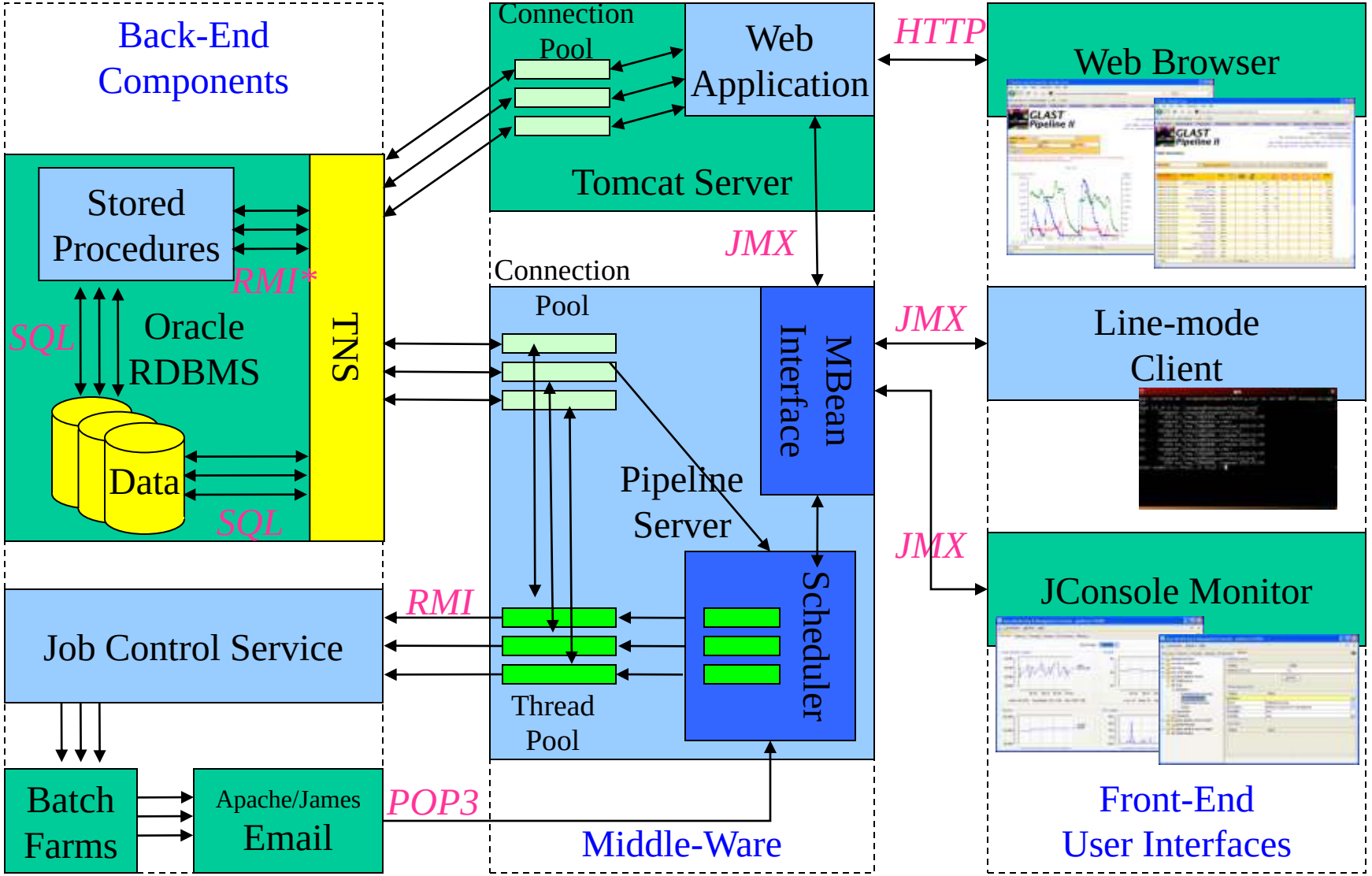

```
<process name="skim">
    <script>
        <![CDATA[
 from java.util import HashMap
 start = 0
 chunk_size = DP_FILE_LINES/DP_SUBTASKS + 1
 for i in range(DP_SUBTASKS):
    vars = HashMap()
    vars.put("DP_START", start)
    end = min( start + chunk_size, DP_FILE_LINES )
    vars.put("DP_END", end)
    if end>start:
       pipeline.createSubstream("SkimmerRuns",i,vars)
    start = end
 ]]>
    </script>
    <createsSubtasks>
        <subtask>SkimmerRuns</subtask>
    </createsSubtasks>
</process>
```

```
<task name="SkimmerRuns" type="Data" version="1.0">
    <process name="skimmerRuns">
        <variables>
            <var name="streamID">${format(pipeline.stream,"%04d")}</var>
        </variables>
        <job maxCPU="${DP_SKIMMER_MAXCPU}">
            cat ${DP_FILE_LIST} | head -${DP_END} | tail -$((DP_END-DP_START)) > partlist
            export SK_FILE_LIST_FILE=partlist
            export SK_OUT_DIR=${DP_OUT_DIR}/${streamID}
            export SK_ENFORCE_OUTPUT_FILES=false
            export SK_MAX_FILE_SIZE=0
            export SK_OUT_FILE_BODY=${DP_JOBNAME}-${streamID}
            mkdir -p ${SK_OUT_DIR}
            $SK_DIR/bin/skimmer
        </job>
    </process>
</task>
```

Level 1 Task Specification



Pipeline Implementation



Pipeline Web Interface

Version 2.9 | Jira (Front-End) (Server) | Help

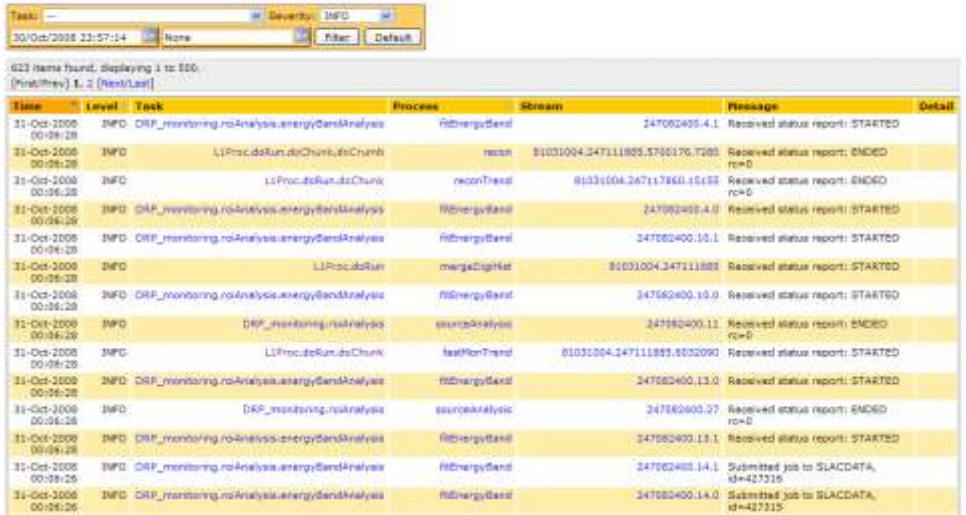
Page updated: 10/31/2008 00:07:15

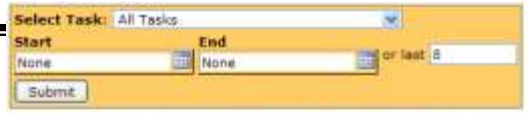
Start refreshing page every 60 secs Start Refreshing

User: dflath . [Switch|Logout] Mode: [Prod | Dev | Test] Preferences

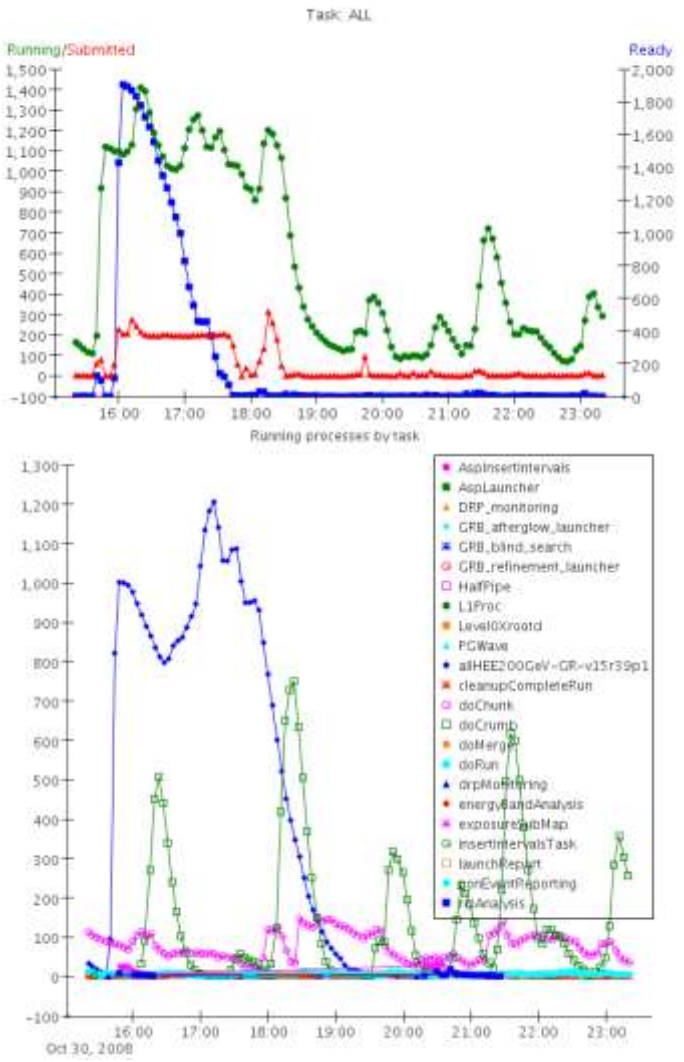
Task List . Message Viewer . Usage Plots . Fair Share Plots . Admin . JMX

Message Viewer



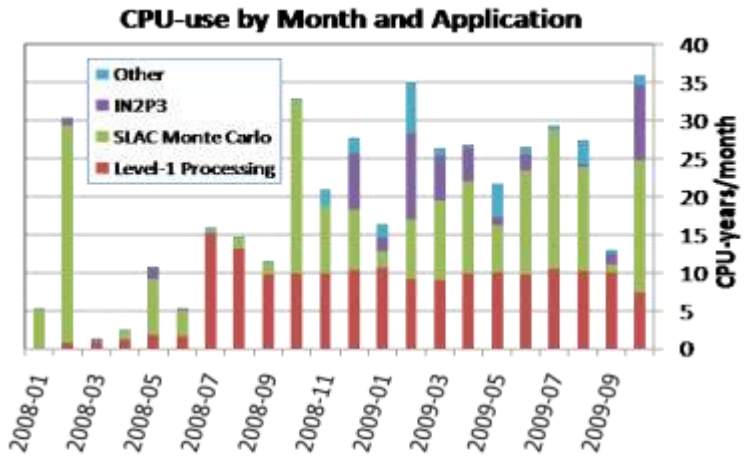


Starting Date: Thu Oct 30 15:20:00 PDT 2008 - Ending Date: Thu Oct 30 23:20:00 PDT 2008
121 records found from table Minutes with group by 4

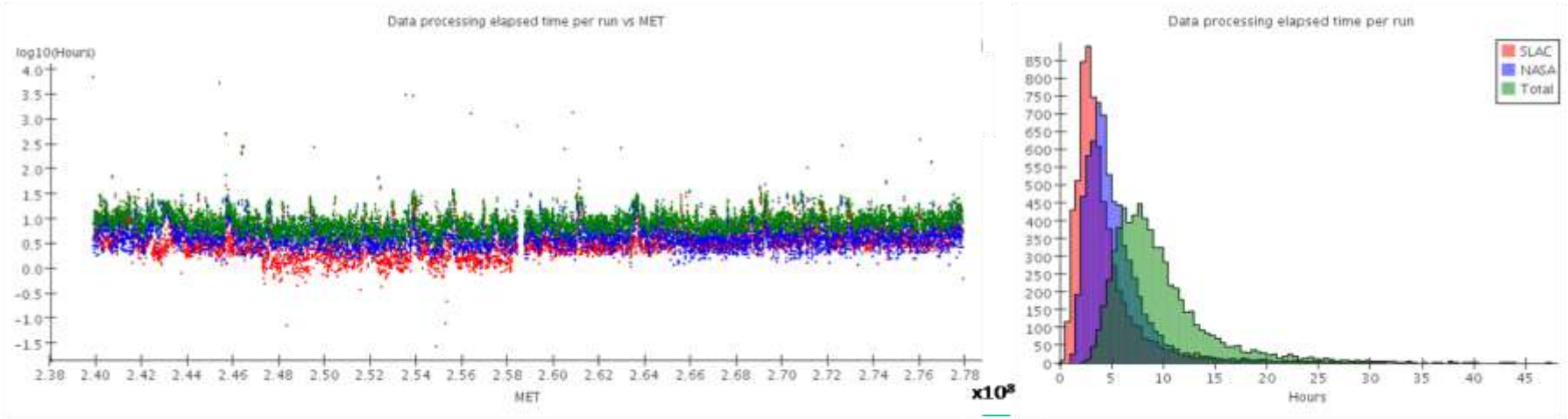


Pipeline Performance and Reliability

Pipeline reliability. AutoRetry allows failed jobs to be rerun without manual intervention.

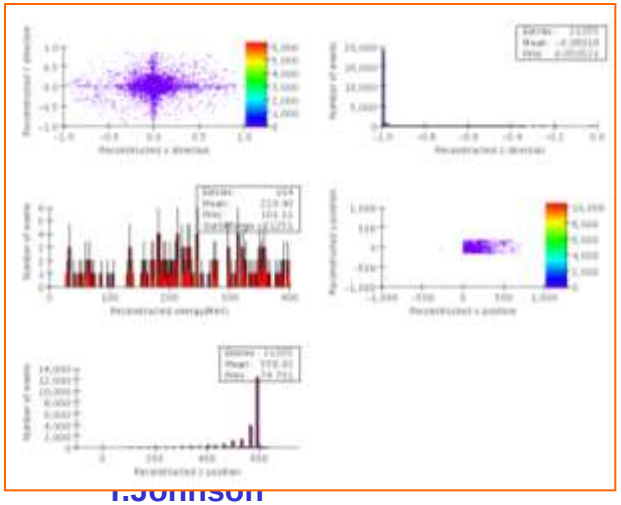


CPU-years delivered by the pipeline per month.



Elapsed time between data being recorded on satellite and arriving at SLAC (red), and between arriving at SLAC and being totally processed (blue), and total elapsed time (green). Most data is fully processed <24 hours after being taken.

Test Name	Date	CPU (secs)	Memory (MB)	Plots (All/Fail)	Links
ACDDigi	Apr 10, 2006	0	NA	0 / 0	
ACDTop	Apr 10, 2006	0	NA	0 / 0	
AllGamma	Apr 11, 2006	8805	377	111 / 17	log meta-data
BackGndAvg	Apr 10, 2006	10	1	0 / 0	log meta-data
CALSingleCrystal	Apr 10, 2006	5	1	0 / 0	
VerticalGamma100GeV	Apr 11, 2006	30596	278	110 / 18	log meta-data
VerticalGamma100MeV	Apr 11, 2006	8138	416	110 / 23	log meta-data
VerticalGamma10GeV	Apr 11, 2006	8926	318	110 / 14	log meta-data
VerticalGamma1GeV	Apr 11, 2006	8657	391	110 / 25	log meta-data



DisplayTag

Root files

xrootd

freehep-rootio

User's request

http://glast-ground.slac.stanford.edu/

IIS

JSP in Tomcat Servers

jdbc

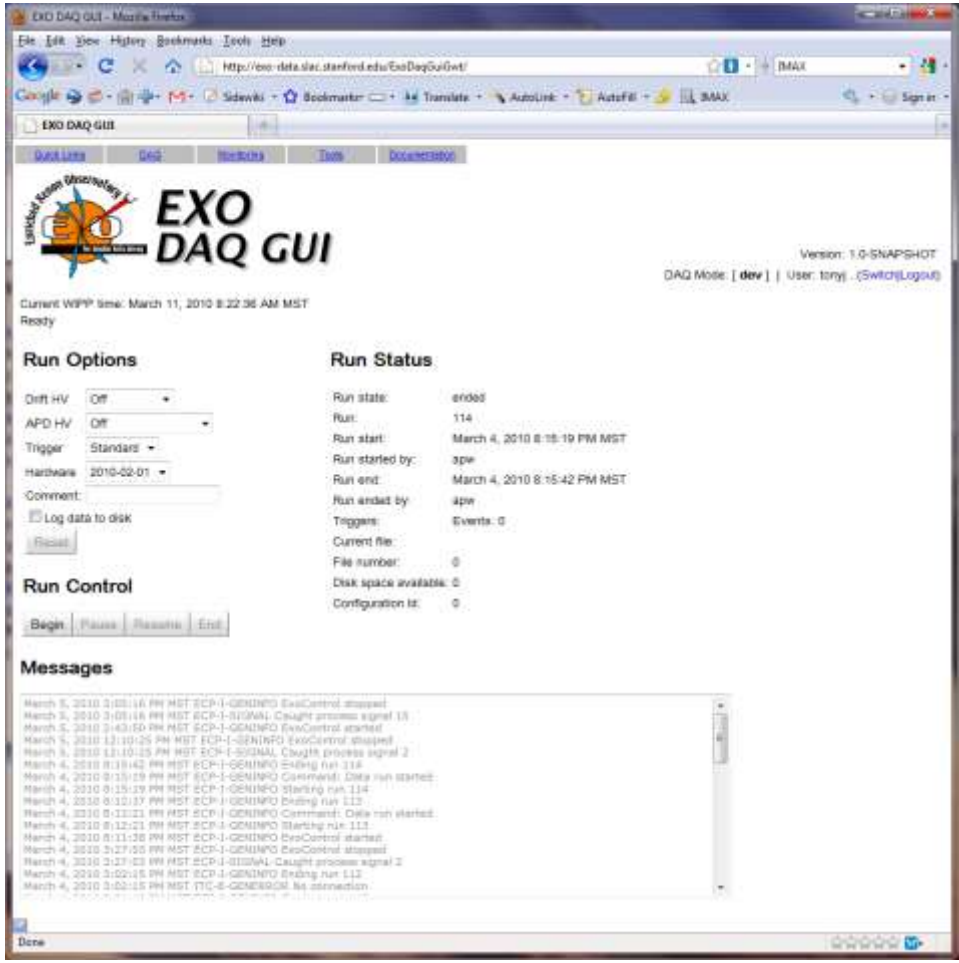
DB

AIDATLD

AIDA/JAIDA

Adding interactivity to web applications

- Most of our Fermi web applications deliberately built with simple technology
  - Plain JSP pages + tag libraries
  - SQL embedded in JSP
    - Available at the time we started Fermi
      - Uniform technology
    - Easily accessible to non experts
    - Easy to rapidly customize
- We have recently started looking at “Google Web Toolkit” GWT for adding interactivity (AJAX) to web applications
  - Looks promising



Web Tools/Pipeline for other Experiments

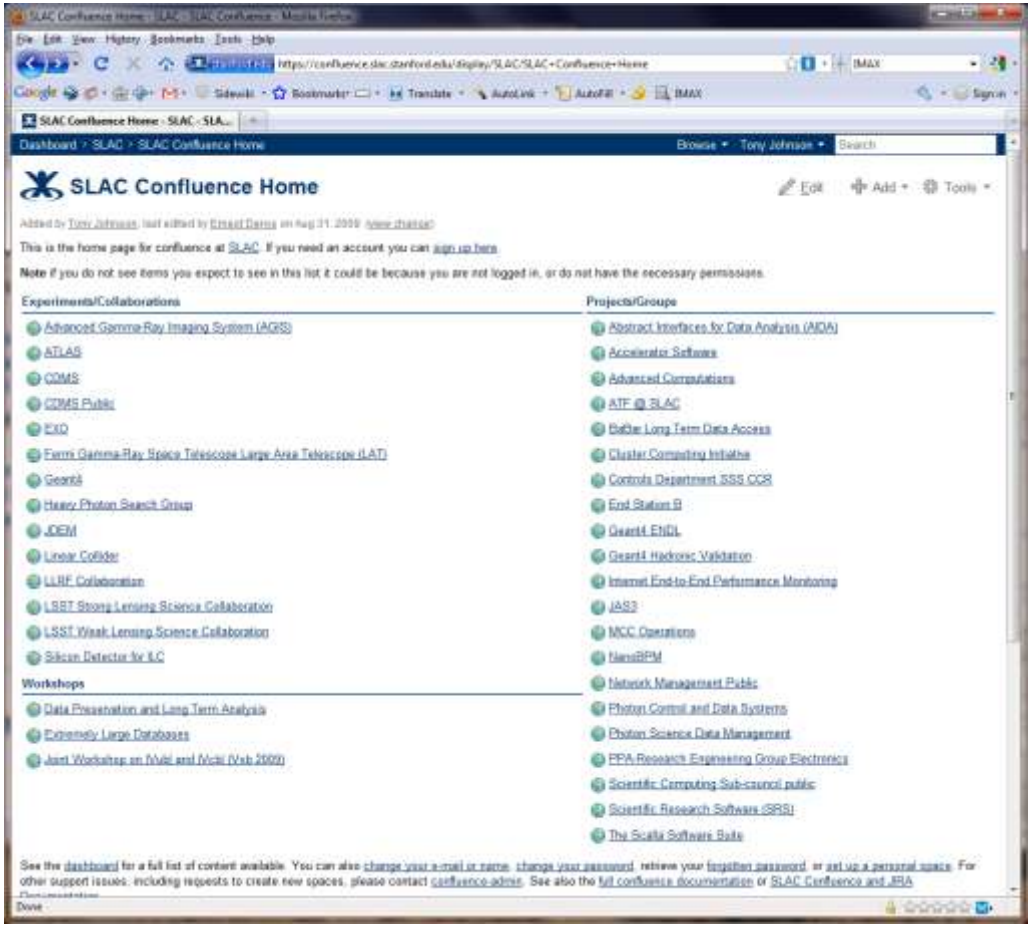
- We have cleaned up many of our web tools to remove any vestigial Fermi dependence, and set up a portal at:
  - http://srs.slac.stanford.edu/
  - Can easily be configured for different experiments, e.g.
    - http://exo-data.slac.stanford.edu/
- We have set up a second "non-Fermi" version of pipeline server and data catalog
  - Currently starting to be used for EXO, CDMS and AGIS
    - EXO will use pipeline for data processing starting next summer
    - CDMS are using the pipeline for MC data processing
- We are working with Fermi Italian and French collaborators to create a job submission daemon to work with the Grid
  - We would like to extend pipeline to be able to submit jobs to the "Cloud"

Other Web Tools

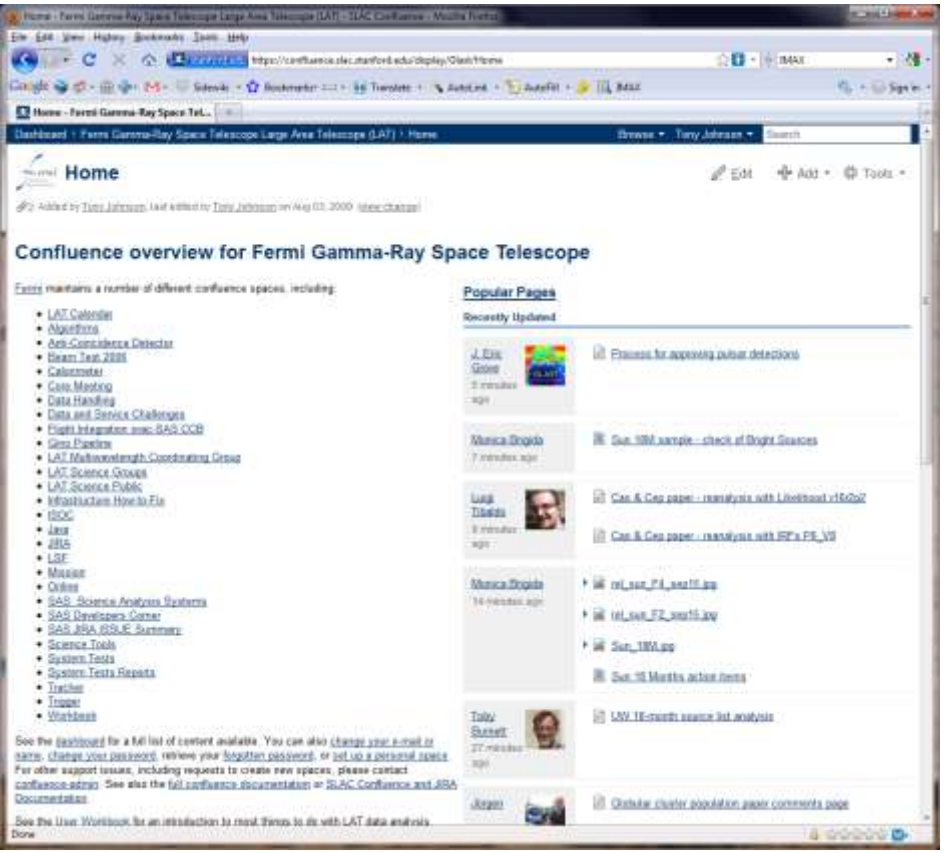
In addition to building our own web tools we try to introduce existing commercial and open-source tools especially those which enhance distributed collaboration

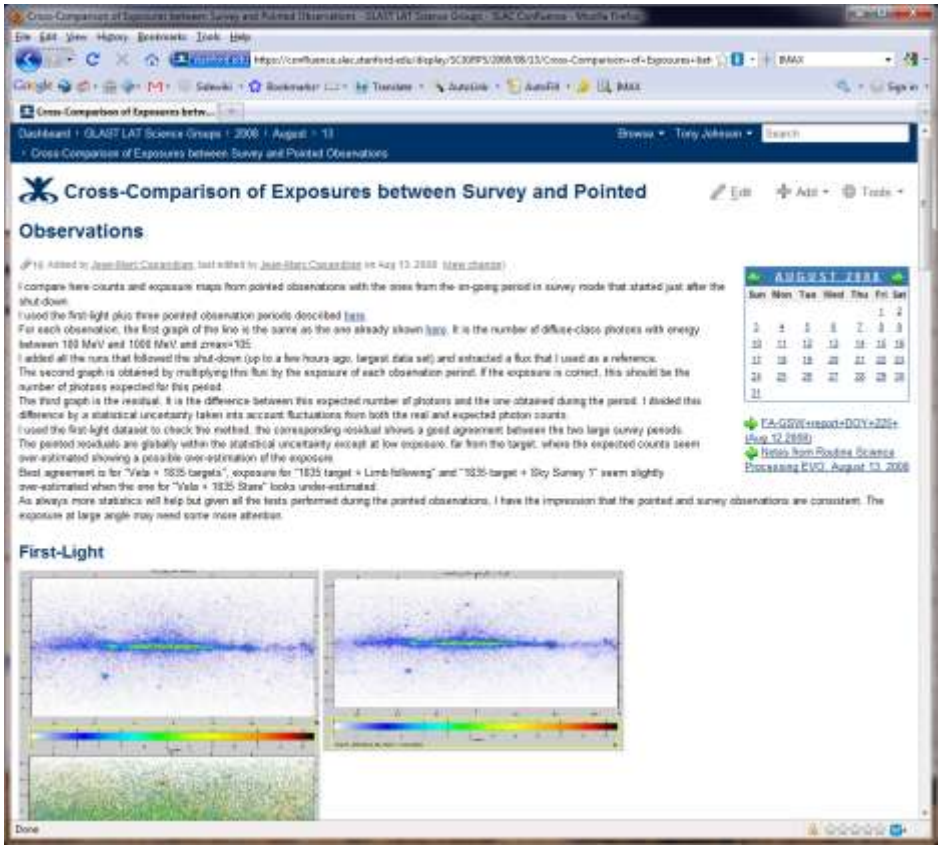
Confluence

- **Confluence first used at SLAC by Fermi.**
  - **Used very extensively for scientific collaboration, discussion of results.**
- **Adopted by many other groups at SLAC**
  - **Confluence and JIRA now maintained at SLAC by CD (Ernest Denys)**



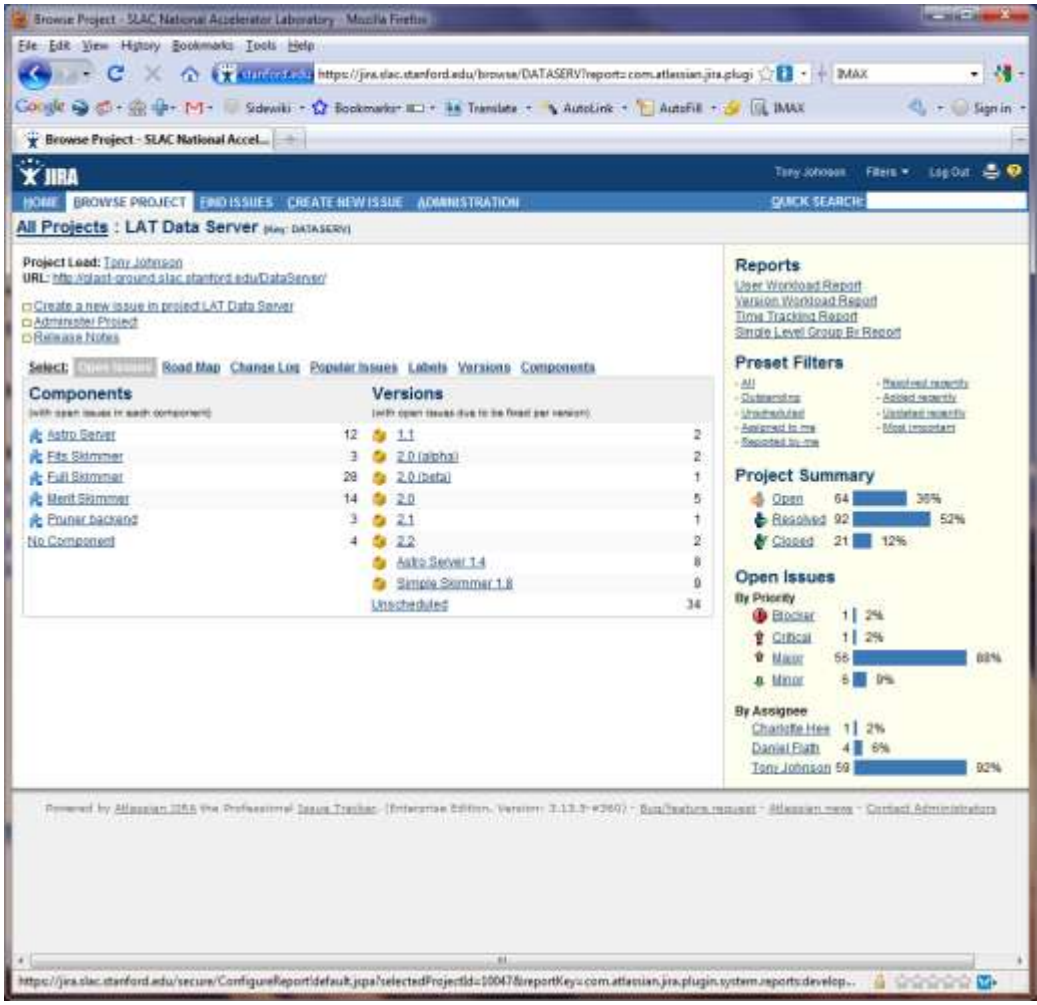
Confluence


Home
Edit
Add
Tools
Confluence overview for Fermi Gamma-Ray Space Telescope
Popular Pages
Recently Updated

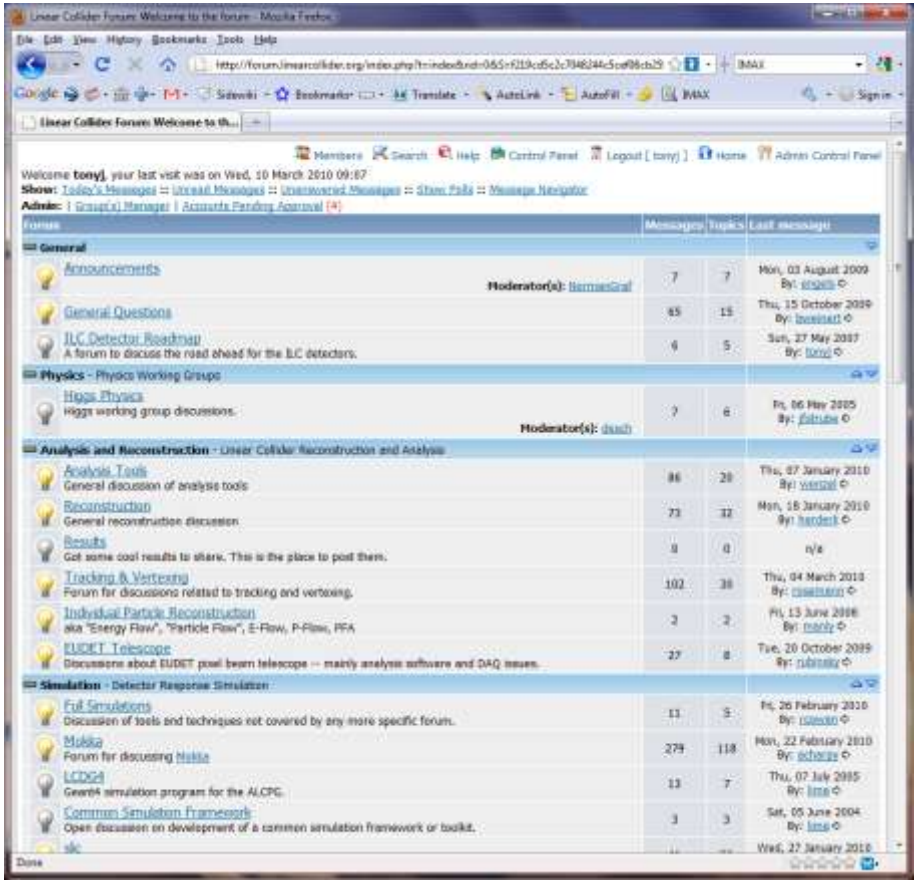

Cross-Comparison of Exposures between Survey and Pointed
Observations
First-Light

JIRA

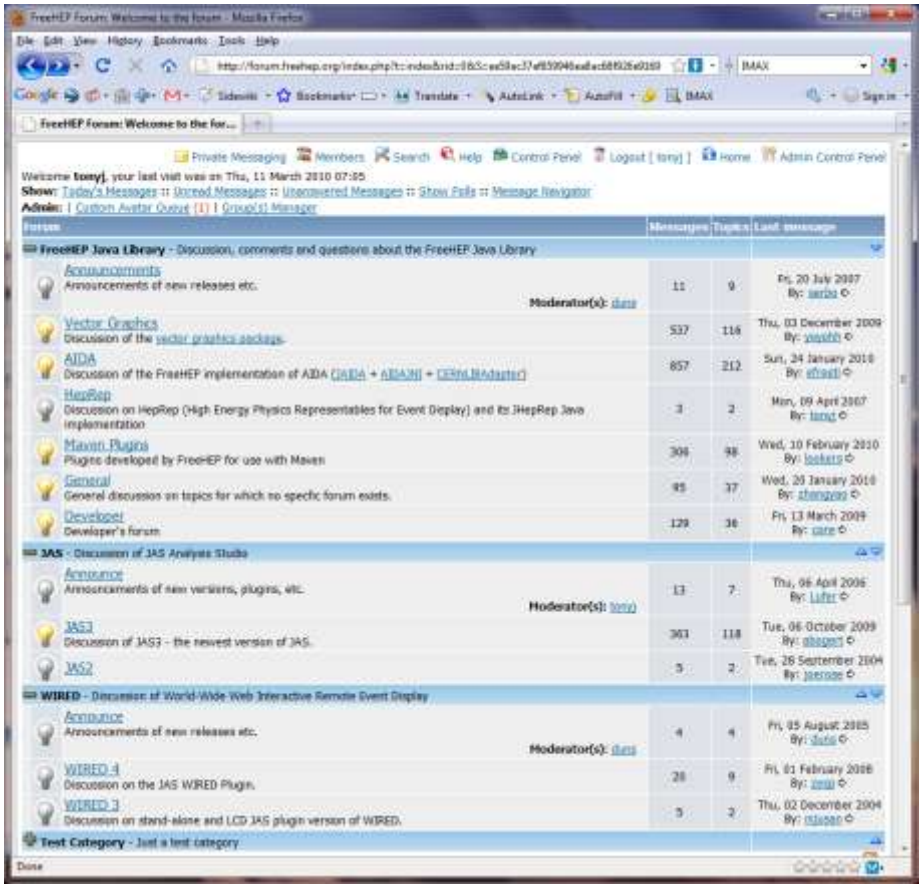
- JIRA also used by Fermi and many other groups
- Excellent web-based issue tracking system
- Can be extensively customized
  - For instance with custom workflows we use it to keep track of release approval process and simulation requests



Discussion forums



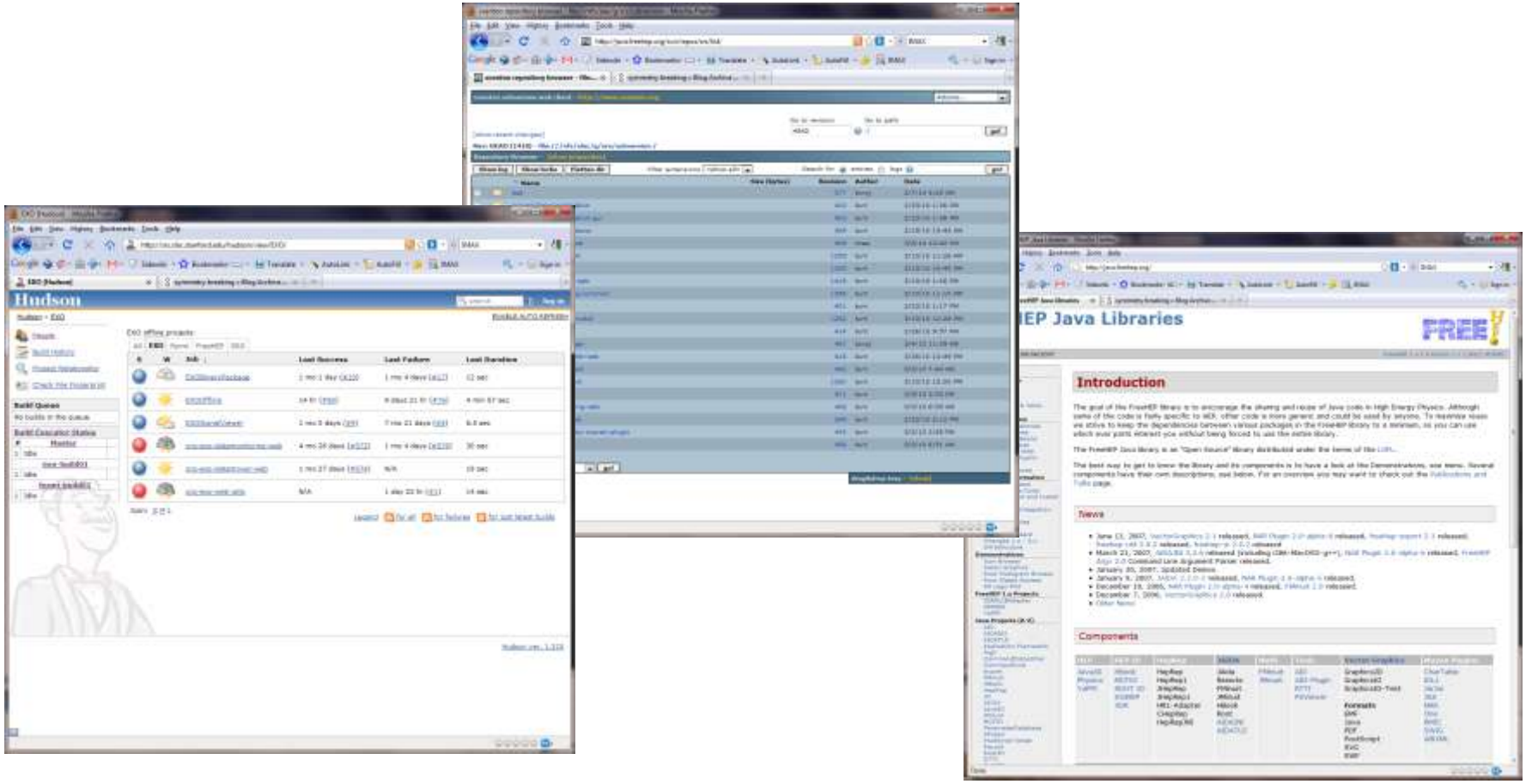
forum.linearcollider.org



forum.freehep.org

Other collaborative tools

- CVS, Subversion, Sventon, Nagios, Hudson and FreeHEP



Collaboration User Databases

Java for visualization, reconstruction and analysis

Why Java for HEAP?

- Java is a pure Object Oriented Language
- Simpler to learn and use than C++
  - Language design emphasizes ease-of-use, programmer productivity
    - Not hampered by historical baggage backwards compatibility with C
  - Lack of direct access to pointers eliminates large source of common errors in C++
    - Especially for less expert developers
  - Garbage collector takes care of freeing unused objects
    - Avoids many common programming errors
    - Avoids distorting OO design by removing need for “ownership” of Objects
  - Very powerful standard libraries build-in
    - Cross-platform GUI development
    - Huge number of open-source libraries
    - Libraries for scientific computing
      - Apache commons-math, JSci, FreeHEP, …
  - Physicist gets to concentrate on writing clean OO code to perform analysis tasks
    - Not understanding core dumps and learning difference between a pointer and a reference.
- Java is increasingly a (maybe the) mainstream OO language
  - Taught in many university courses
  - Overtaken C++ in popularity for “open-source” projects
  - Very widely used especially in the areas of
    - Web application development
    - Graphical enterprise applications
    - Other fields of science, especially astronomy, biology, …

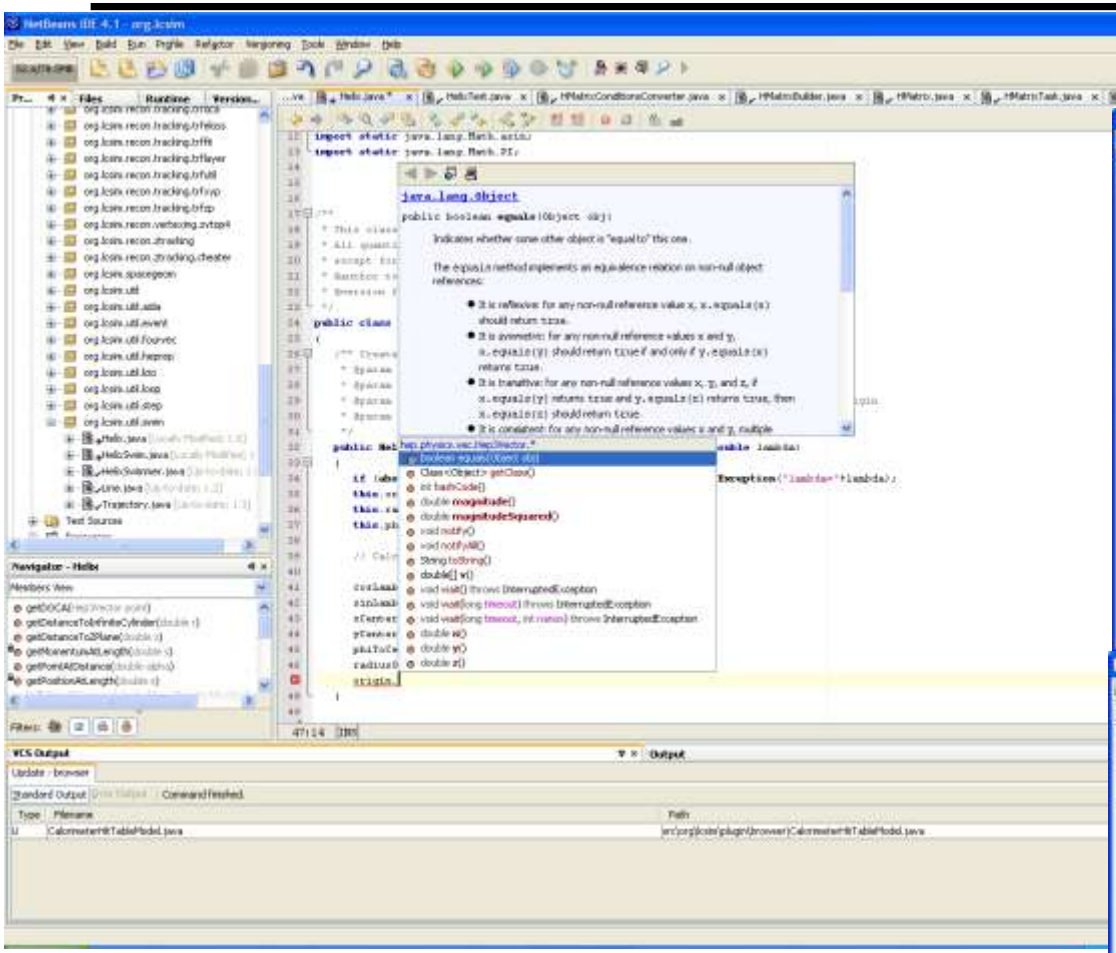
Why Java for HEAP?

- **Platform independent, compile once just runs everywhere**
  - **Linux, Windows, Mac OSX**
    - **Saves a lot of time when supporting code on many platforms**
  - **Makes possible tools such as Java WebStart where user clicks a button on a web page to start an application with no need to have pre-installed any software.**
- **Full runtime access to information about classes (methods, member variables *etc*)**
  - **Directly usable by scripting languages, analysis toolkits, IO**
  - **Replaces need to invent these mechanisms ourselves (c.f. CINT dictionary)**
- **Performance of Java code is close to that of C++**
  - **Although Java is initially compiled to machine independent "byte-codes" these are converted to machine code at runtime**
  - **Dynamic (runtime) optimization can take into account actual usage patterns**
    - **Not available to static optimizers used by Fortran, C++**
  - **Garbage collection often more efficient than user written malloc/free (or new/delete)**
  - **Many benchmarks available on the web**
    - **Some show C++ faster than Java, others show Java faster than C++**
    - **Our experience is that overhead of garbage collection and factors like array bounds checking makes (well-written) Java slightly slower than (well-written) C++, but overhead is typically small**
    - **Often raw performance is irrelevant compared to savings in development time**
      - **Which in turn can lead to cleaner more optimized code to begin with**

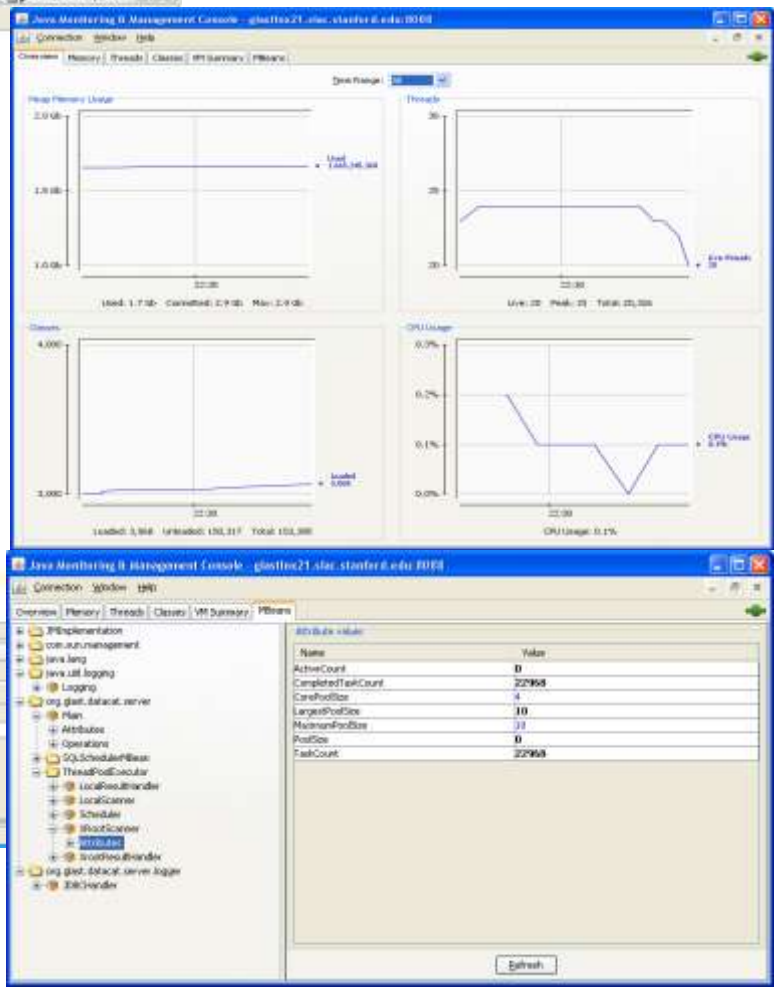
Why Java for HEAP

- **Excellent tools (in many cases free)**
  - **IDE's**
    - **Eclipse, Netbeans, IDEA, …**
    - **Typically integrate editing, code completion, documentation viewing, refactoring, debugging, WYSIWYG GUI development, performance analysis, …**
    - **These IDE's now support many languages including C++**
      - **Java has lead the way in excellent IDE support**
  - **JMX**
    - **Ability to connect to running program to view statistics, memory usage and control program execution,**
    - **Perform memory and thread dumps on running programs**
    - **Support for user defined "mbeans" for dynamic access to program functionality**
  - **Build tools, ant, maven**
    - **Maven allows a project to be build from source with a single command that:**
      - **Downloads correct versions of all dependencies**
      - **Compiles the code**
      - **Runs unit and integration tests**
      - **Deploys library to allow it do be used as dependency for other projects**
    - **Also allows web site, documentation and reports to be generated and deployed in a single command**
    - **Configured with a single declarative project description (in XML)**

Why Java for HEAP



Netbeans IDE

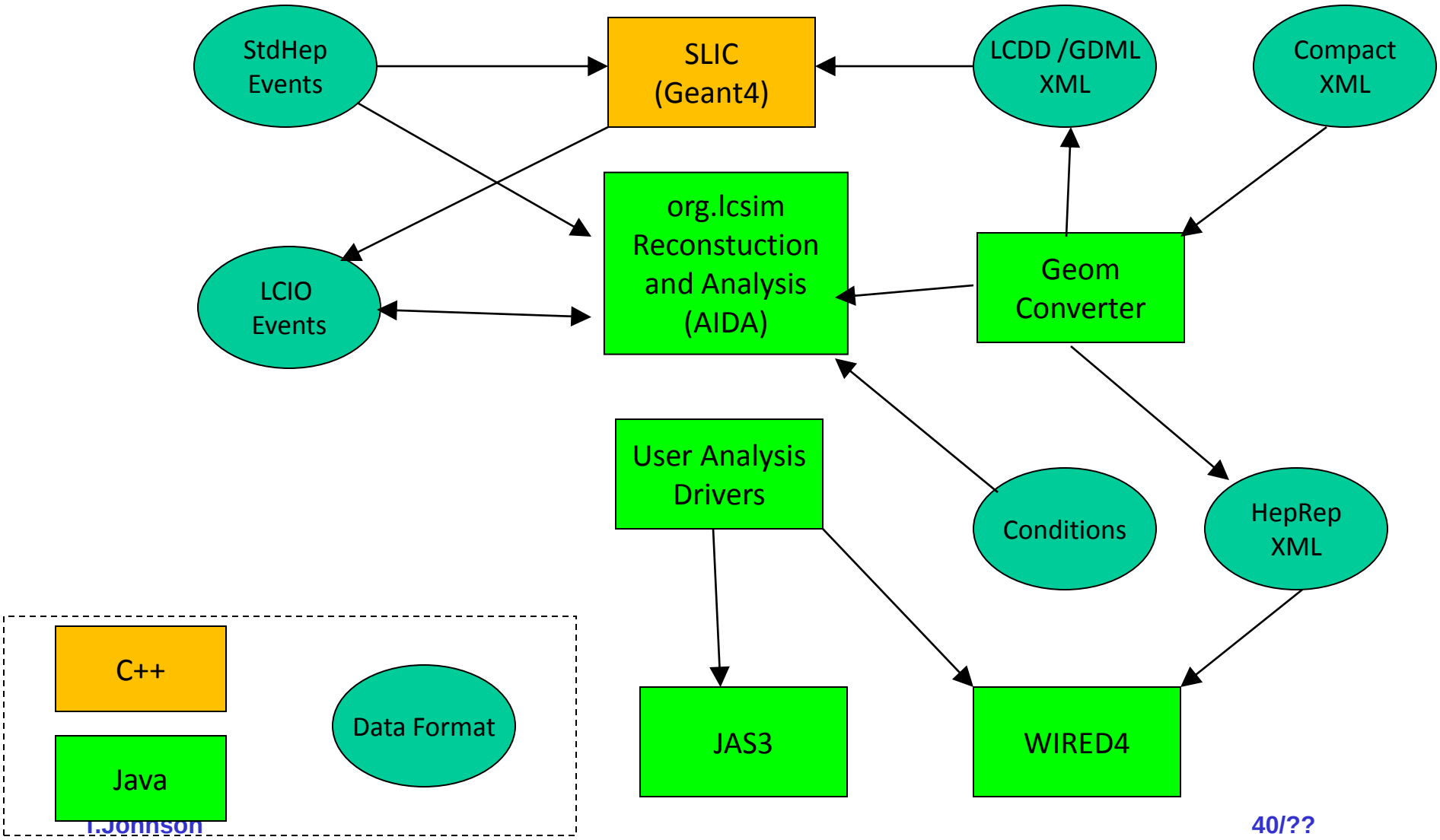


Jconsole

ILC Detector Development

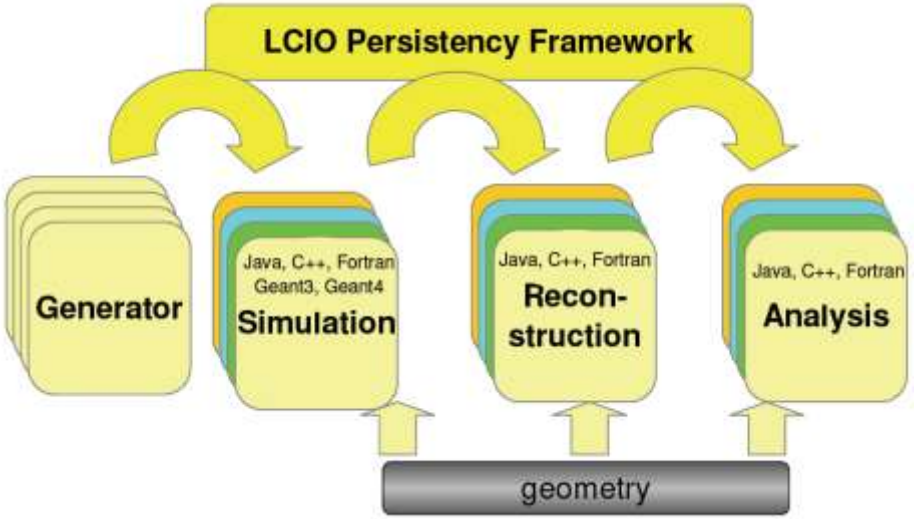
- Goal of ILC detector development is to study a variety of different possible detector configurations/technology for a future linear collider.
- Work has been ongoing for many years
  - Typically involves people who only work part time on this project
  - Students, post-docs typically work for 1 or 2 years then move on to something else
- Software needs to be flexible, very easy to learn and use
  - At past workshops we have distributed software suite on CD with goal “15 minutes from zero to physics”.
    - Windows, Mac, Linux

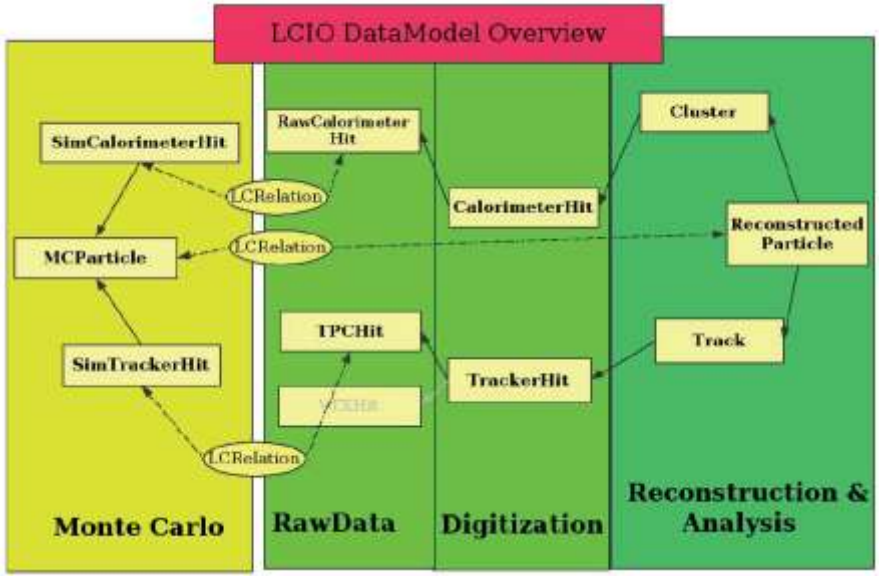
ILC Reconstruction/Simulation/Analysis Framework as used by SiD detector



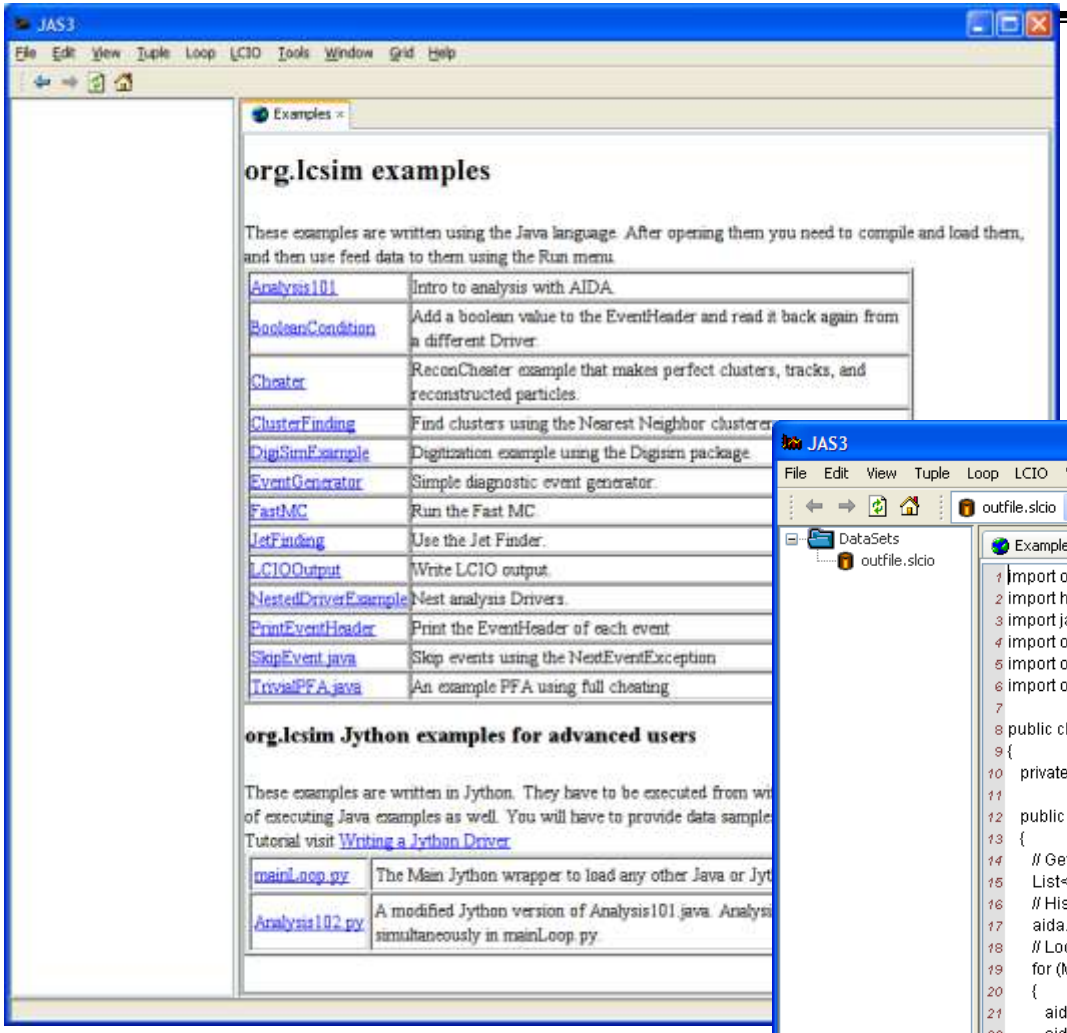
LCIO Persistency Framework

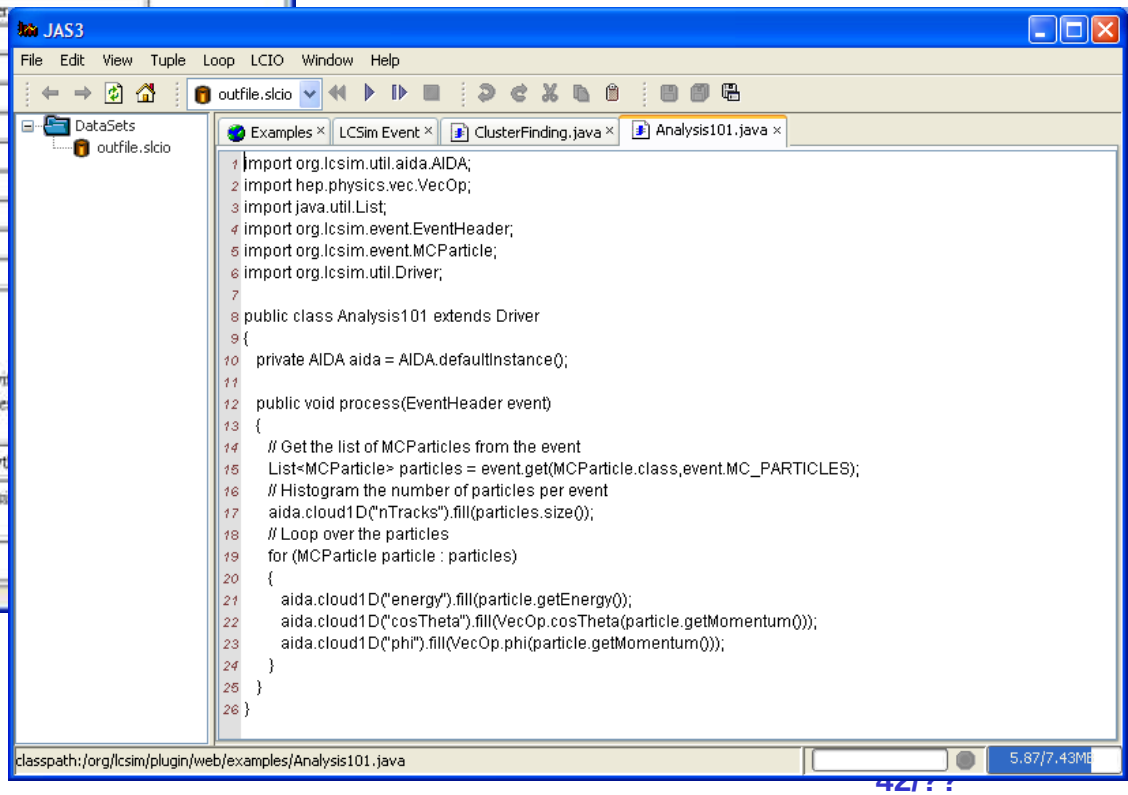
- Object model and persistency
  - Events
    - Monte Carlo
    - Raw
    - Event and run metadata
  - Reconstruction
  - Parameters, relations, attributes, arrays, generic objects, …
- All the ILC simulators write LCIO
  - Enables cross-checks between data from different simulators
  - Read/write LCIO from
    - Fast MC / Full Simulation
    - Different detectors
    - Different reconstruction tools



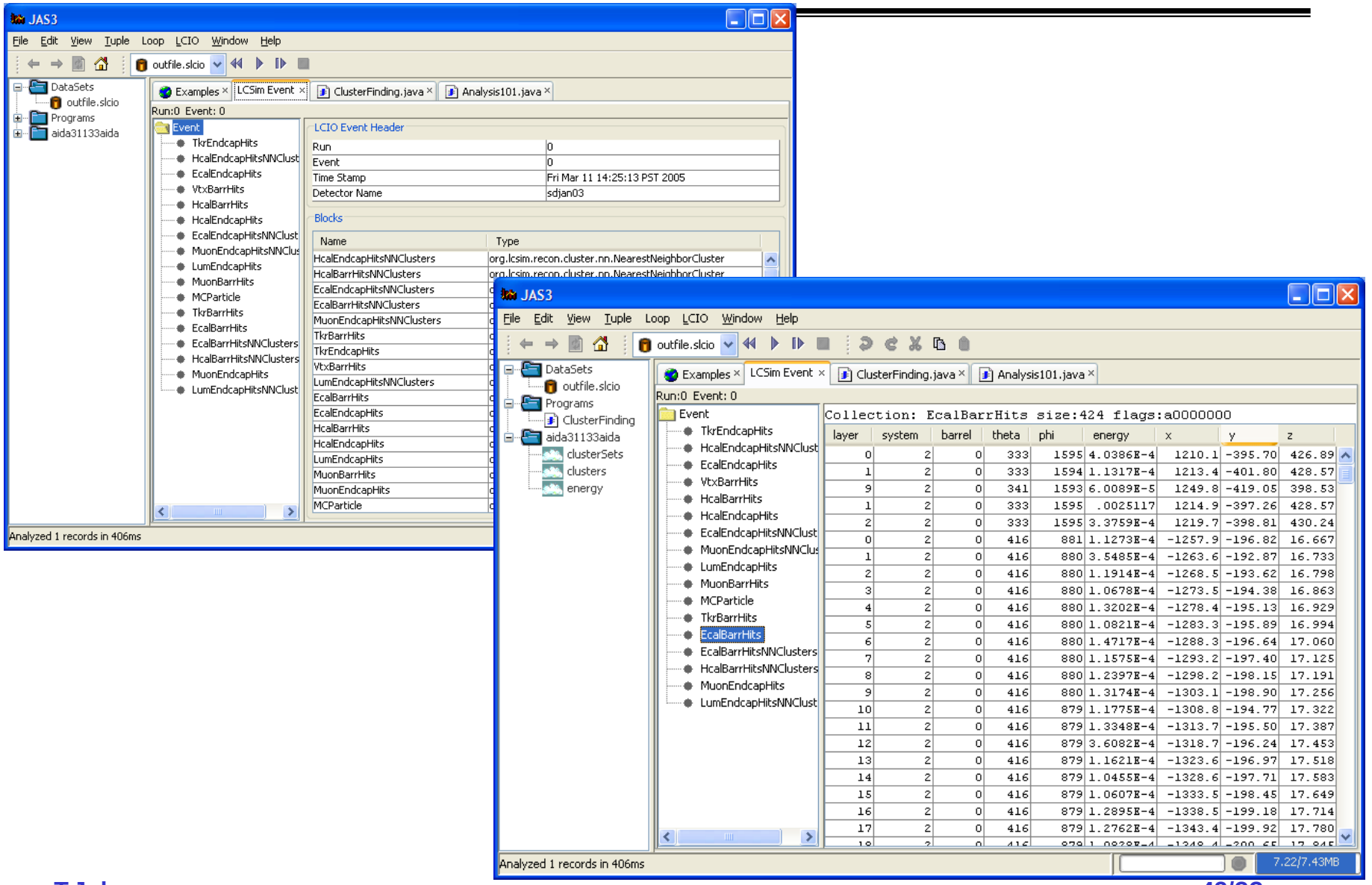


org.lcsim: Examples

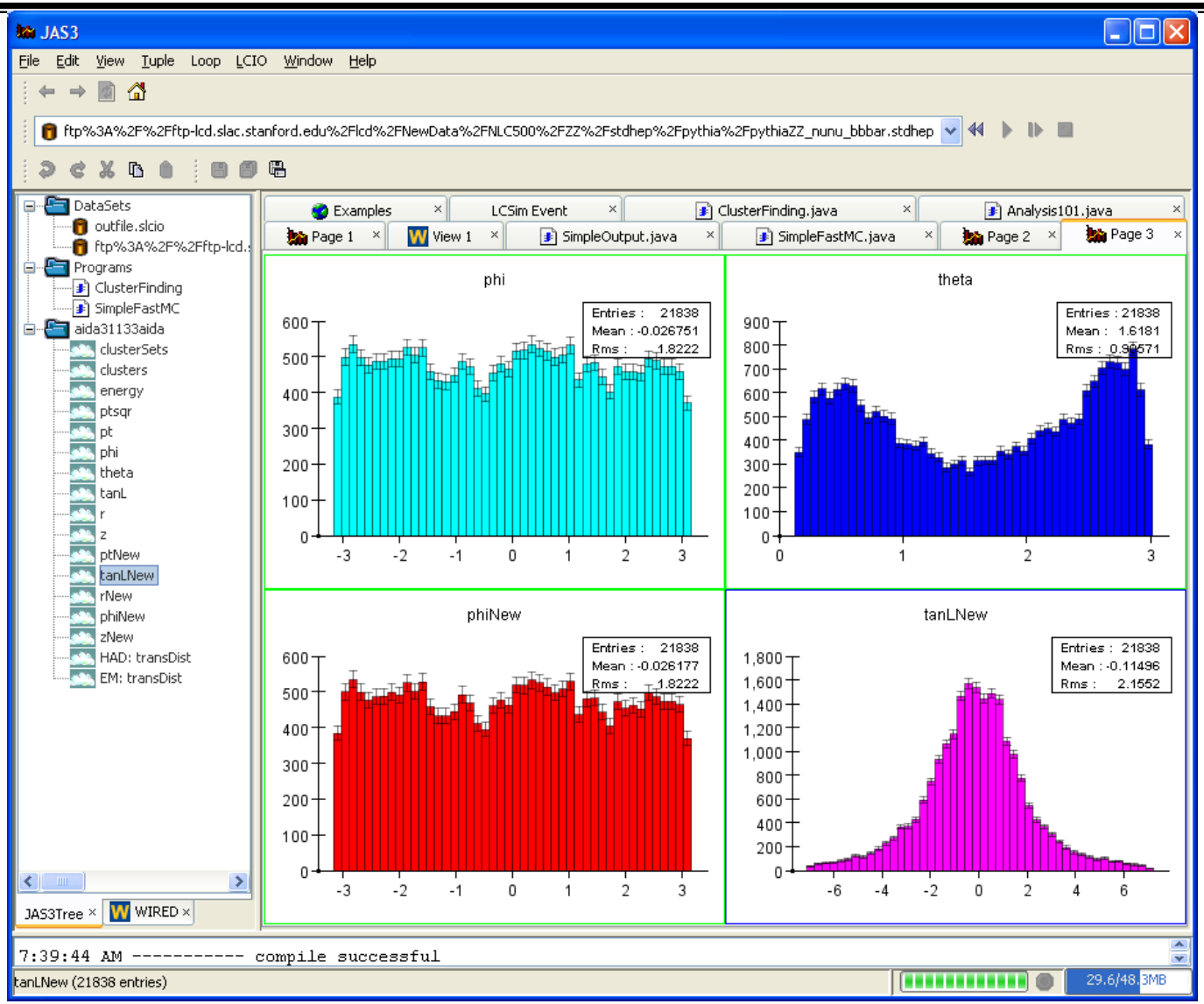




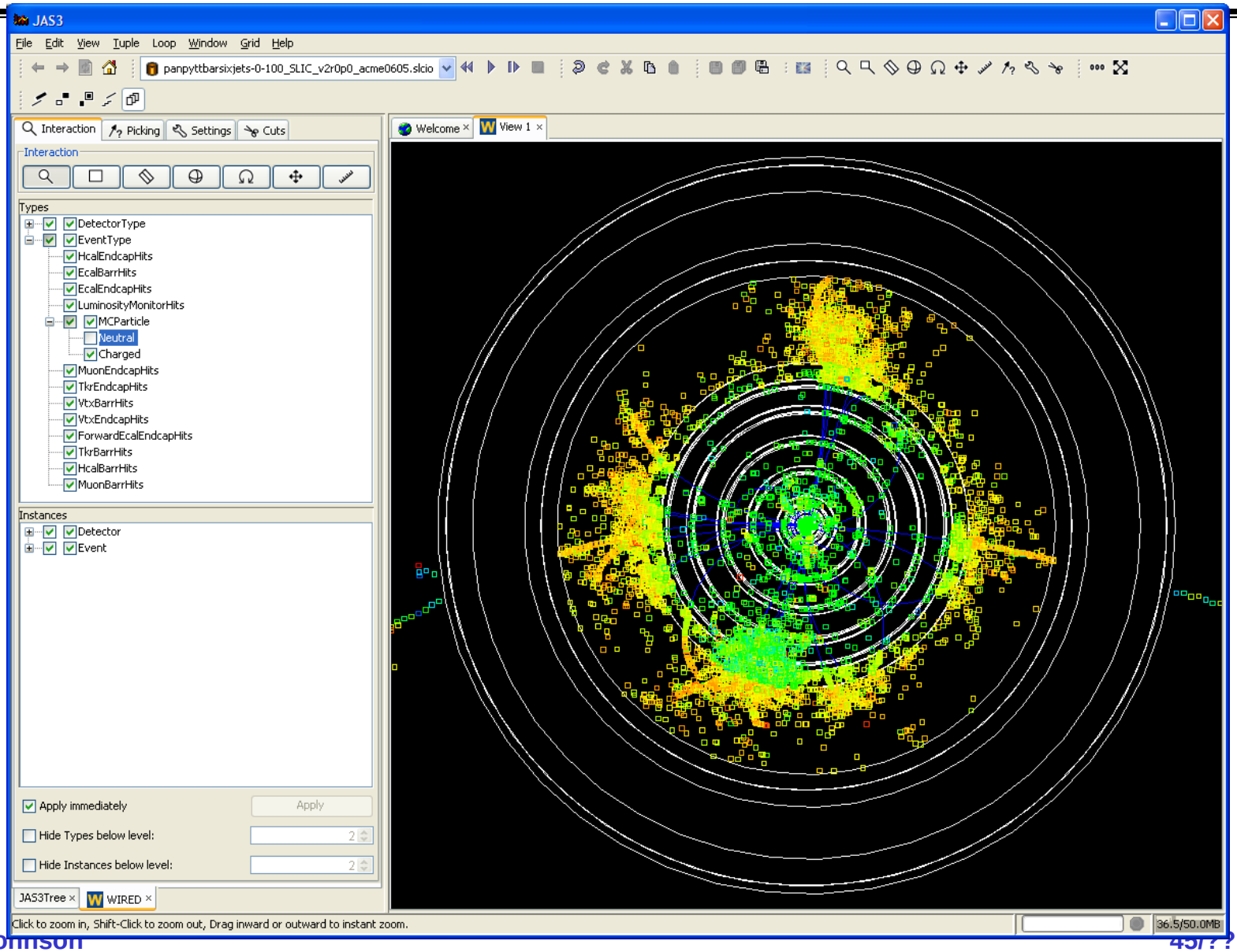
org.lcsim: Examples



org.lcsim: Plot Viewing



Using org.lcsim with WIRED4



Design Goals

- Keep technology simple for flexibility
  - Goals change frequently
  - Plain JSP pages + tag libraries
  - SQL embedded in JSP
- Design for portability and reuse
  - Always manpower limited
  - E.g. pipeline
- Modular systems with loosely coupled interfaces
  - AIDA, HepRep, LCIO
  - Make use of industry standard solutions wherever possible
  - Maintain ability to move forward with new technologies
  - Small independent applications which work seamlessly together
    - Developed in independent timescales

Future Plans

- We have clearly developed more stuff than we can support with the small group that we have
  - Many items are barely/not supported now
    - JAS, WIRED, AIDA, Plotter
- But still new items we would like to explore
  - More interactive web applications
    - GWT looks like a promising direction
  - Database optimization
    - More usage of Oracle 11g features
    - Optimization using Oracle Streams AQ
  - Ability to exploit many-processor architecture
    - Threading support and concurrency utilities in Java make this an obvious target
  - Cloud computing
    - Using cloud computing tools at SLAC
    - Use of remote cloud computing facilities
  - Collaborate with Root team, others….

Interactions with CD

- We have worked very closely with CD in many areas:
  - Unix support
  - Batch farm
  - Oracle/mysql support
  - Confluence/JIRA support
- There remain some areas where I think collaboration could be enhanced
  - General support of web tools and technology
    - E.g. Single sign-on
  - Collaborative tools (e.g. forum software)
  - Account management
  - Security issues

Conclusions

- We attempt to design software which can be reused by many experiments
  - Particularly appropriate as SLAC moves away from the ERA of one large experiment towards many smaller experiments
- Most of our interactions with CD are very positive
  - We seek to build strong collaborations with CD and other groups at the lab