DIRAC User Group Meeting
Oct. 29th – 31st 2012
Marseille, France

Interfacing the *Fermi*-LAT Dataprocessing Pipeline with Grid Services using DIRAC

Stephan Zimmer (Stockholm)

On behalf the *Fermi*-LAT Collaboration

with

A. Tsaregorodtsev, L. Arrabito and C. Lavalley

To appear in J.Phys.Conf.Proc (2012)
(CHEP 2012, in press)

Oskar Klein centre

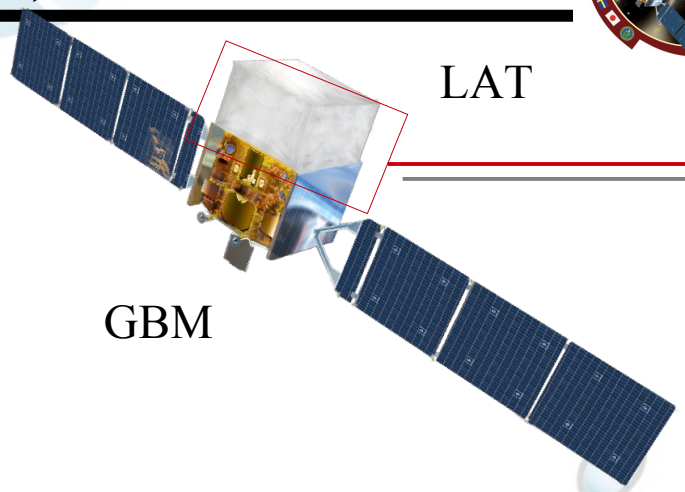
Stockholms Universitet
Stockholm University

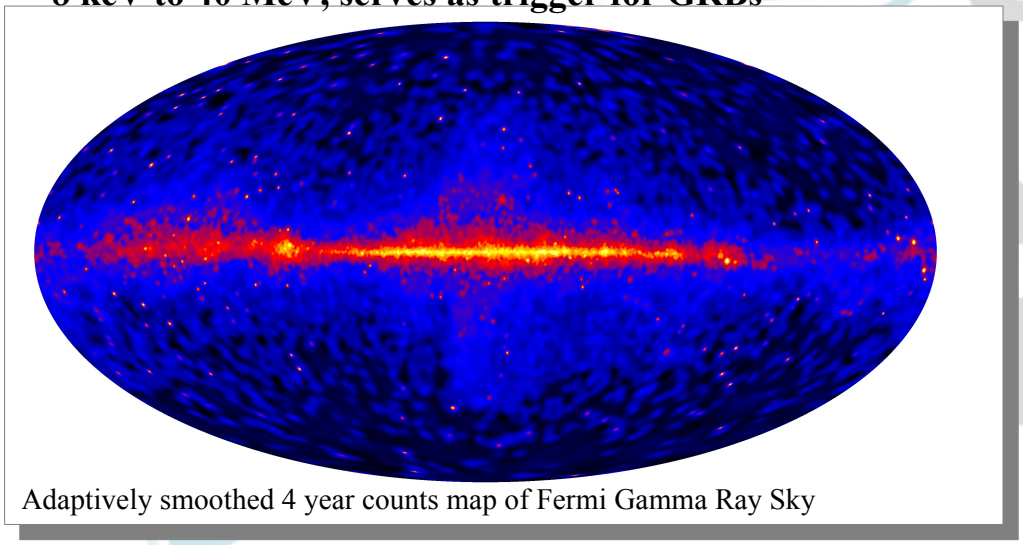
Outline

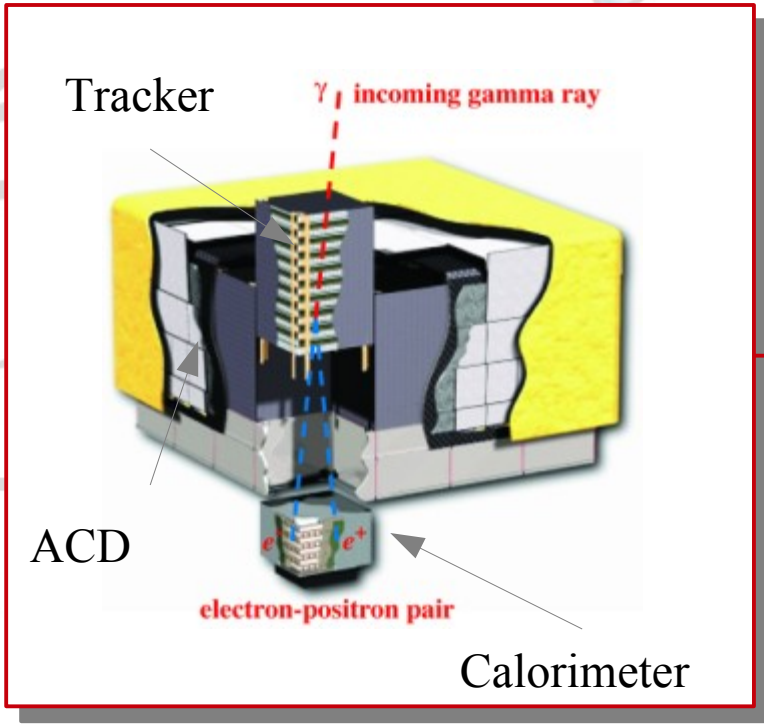
- The *Fermi*-LAT
- LAT Computing Requirements
  - Level 1 Data processing
  - Massive Monte Carlo Production
  - Data Reprocessing
- The *Fermi*-LAT Dataprocessing Pipeline
- Interfacing the Pipeline with the Grid
  - Design considerations
  - services and solutions provided by DIRAC
- Status

The Fermi Large Area Telescope (short *Fermi-LAT*)

- **Fermi Gamma-ray Space Telescope launched on June 11th, 2008 at Cape Canaveral, FL**
- **16 identical modules in a 4x4 array, consists of tracker (direction) & calorimeter (energy) → pair-conversion telescope**
- **18 tracker planes (TKR) 8.6 radiation lengths CsI(TS) (CAL)**
- **Energy Range: 20 MeV - 300 GeV**
- **Large effective area ~1m²**
- **All-Sky monitor ~3h for 2 orbits, FoV ~2.4 sr (@ 1 GeV)**
- **Good energy resolution (<15 % @ E>100 MeV)**
- **Gamma Ray Burst Monitor energy coverage 8 keV to 40 MeV, serves as trigger for GRBs**




Adaptively smoothed 4 year counts map of Fermi Gamma Ray Sky



LAT Computing Demands

- **Time-critical setup (data is publicly released as soon as it is processed)**
- **Event reconstruction rate 4 Hz, downlink rate 500 Hz need 125 computing cpus, peak 800 cores per downlink**
- **complex generic graphs of processes to be processed**
- **Raw data (15 GB/ day) reconstructed equals some 750 GB (processing, database storage, ~200 MB delivered to public)**
- **Peak usage 45.000 jobs per day (job = stream in the Pipeline language, complete procedure of batch jobs and scriptlets)**
- **In addition use Pipeline for Monte Carlo and scientific analysis jobs (e.g. GRB blind search)**
- **Small in comparison to LHC, but big for space mission**

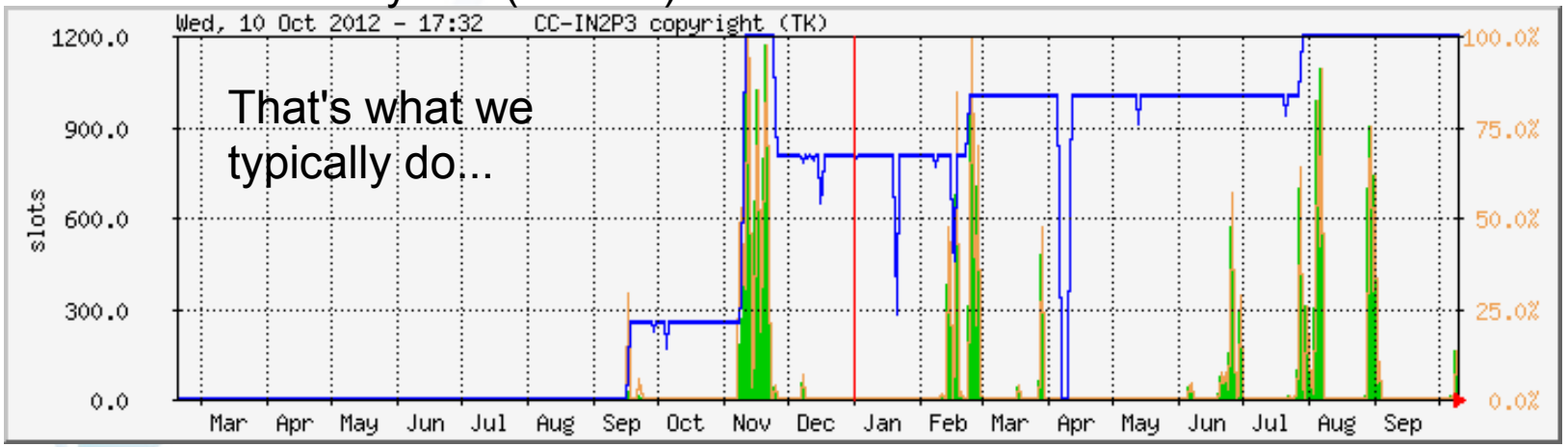
The Big Guy: L1 Processing

Very complex task that runs processing and monitoring without killing the resources: average throughput ~100 MB/s

The little Brother: Monte Carlo Simulations

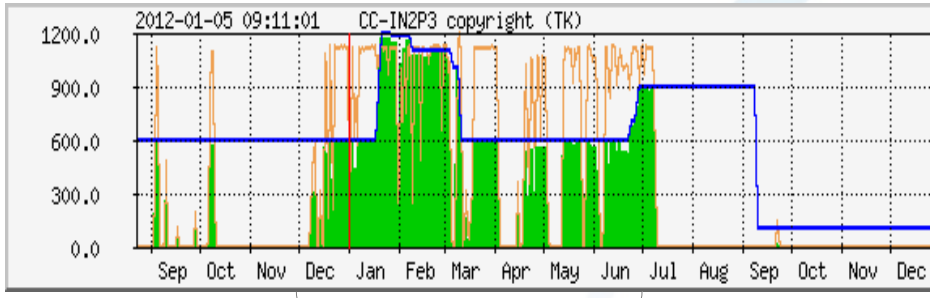
- **Use Geant4 simulation for Gammas & Backgrounds with varying complexity**
- **Simple task: runMonteCarlo → register data in Data Catalog**
- **Much less I/O critical but CPU intense, run most of MC at IN2P3 @ Lyon (Thanks again for many years of great collaboration!)**
  - **Usually a few peaks of increased cycle usage (new simulation releases)**
  - **IF running at SLAC compete with LAT team for resources**

SGE Production Cycles (current)



Previously at Lyon

BQS Production Cycles (old)



Massive All Proton Background Monte Carlo Run

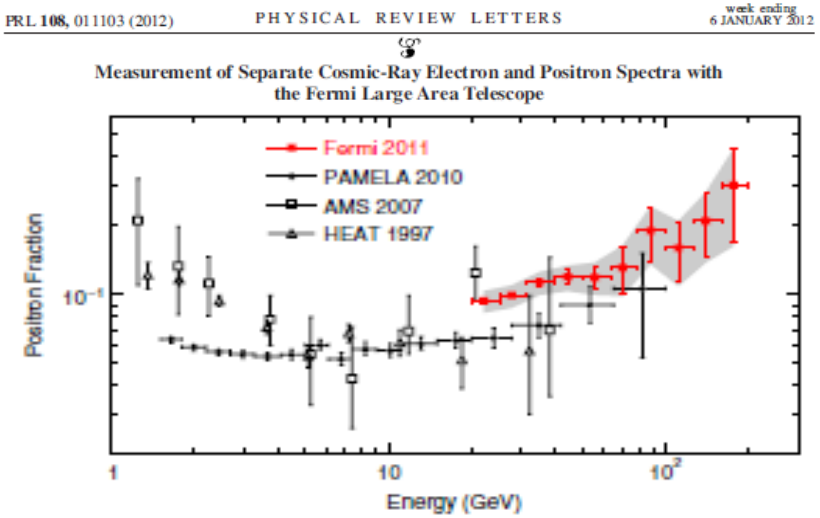


FIG. 5 (color online). Positron fraction measured by the Fermi LAT and by other experiments [7,14,16]. The Fermi statistical uncertainty is shown with error bars and the total (statistical plus systematic uncertainty) is shown as a shaded band.

- **Above shows most extensive LAT Monte Carlo Simulation to date**
  - **Simulations running simultaneously at Lyon AND SLAC since December 25th 2010 (until Mid June 2011)**
  - **Close to a billion triggered events, ~2 TB of disk (simulations are small)**
  - **Full use of our resources at Lyon & SLAC**
- **Can't we delegate this to the GRID?**
  - **Simpler than L1 and perhaps easier to implement, frees resources at Lyon/SLAC**

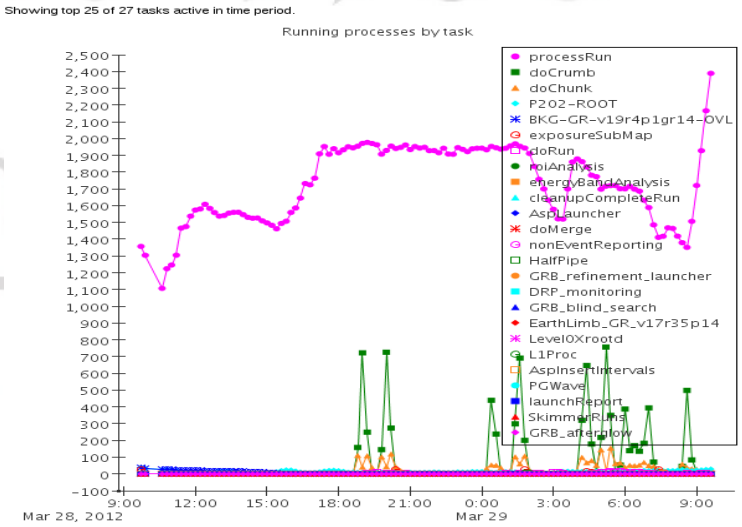
The new kid in town: Data Reprocessing

- Data released as soon as it reaches the ground, available to the whole world
- New improvements in detector understanding, calibrations need to be distributed as well → need for reprocessing of data
- First reprocessing done at SLAC to include on-orbit experience of LAT
  - I/O intense operation that requires various inputs at SLAC, porting to other site complicated
  - ~1.5 million total batch jobs
  - Elapsed time: 112 days of active running (on SLAC farm)
  - CPU time (dole-class machine): 175 CPU years
- Would be good to port this to Lyon (work in progress)
  - Maybe on Grid... but lots of question marks

Folder /Data/Flight/Reprocess/P202

Edit description

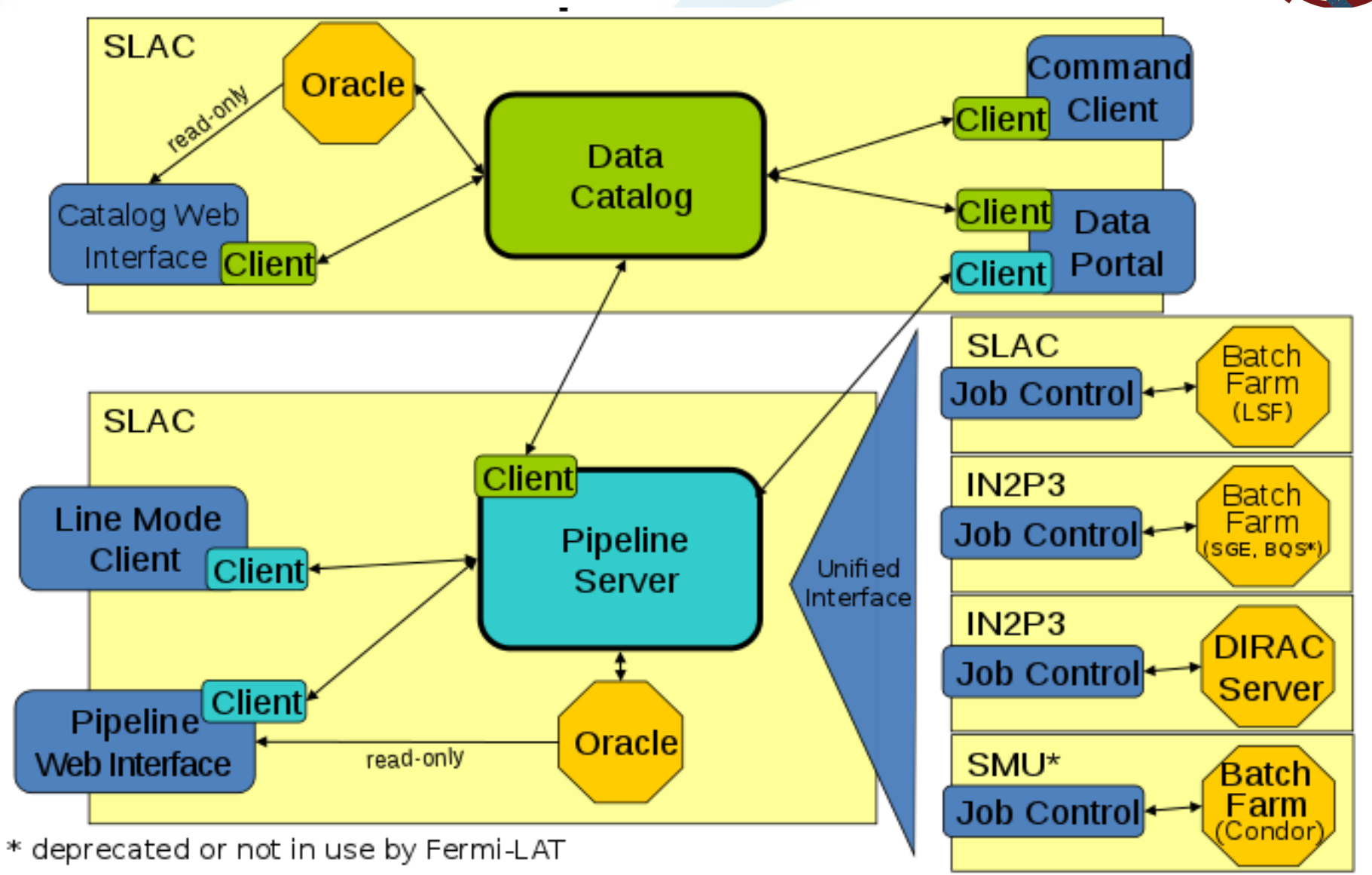
Name	Type	Files	Events	Size	Created (UTC)	Links
ELECTRONFT1	Group	20229	0	8.5 GB	02-Mar-2012 00:06:07	Files
FT1	Group	20229	189,323,074	17.8 GB	02-Mar-2012 00:06:06	Files
LS1	Group	20229	1,325,204,821	215.3 GB	02-Mar-2012 00:06:08	Files
ELECTRONMERIT	Group	22065	98,783,773	223.2 GB	25-Jan-2012 00:53:32	Files
EXTENDEDFT1	Group	20229	6,291,424,926	574.7 GB	02-Mar-2012 00:06:09	Files
EXTENDEDLS1	Group	20229	6,291,424,926	1,020.1 GB	02-Mar-2012 00:06:09	Files
GCR	Group	22065	48,094,684,158	1.0 TB	25-Jan-2012 00:53:31	Files
FILTEREDMERIT	Group	22065	6,882,654,108	5.8 TB	25-Jan-2012 00:53:29	Files
MERIT	Group	22065	48,094,684,158	38.5 TB	25-Jan-2012 00:53:30	Files
CAL	Group	22065	48,094,684,158	140.1 TB	25-Jan-2012 00:53:31	Files
RECON	Group	22065	48,094,684,158	642.5 TB	25-Jan-2012 00:53:33	Files



The Fermi-LAT Dataprocessing Pipeline

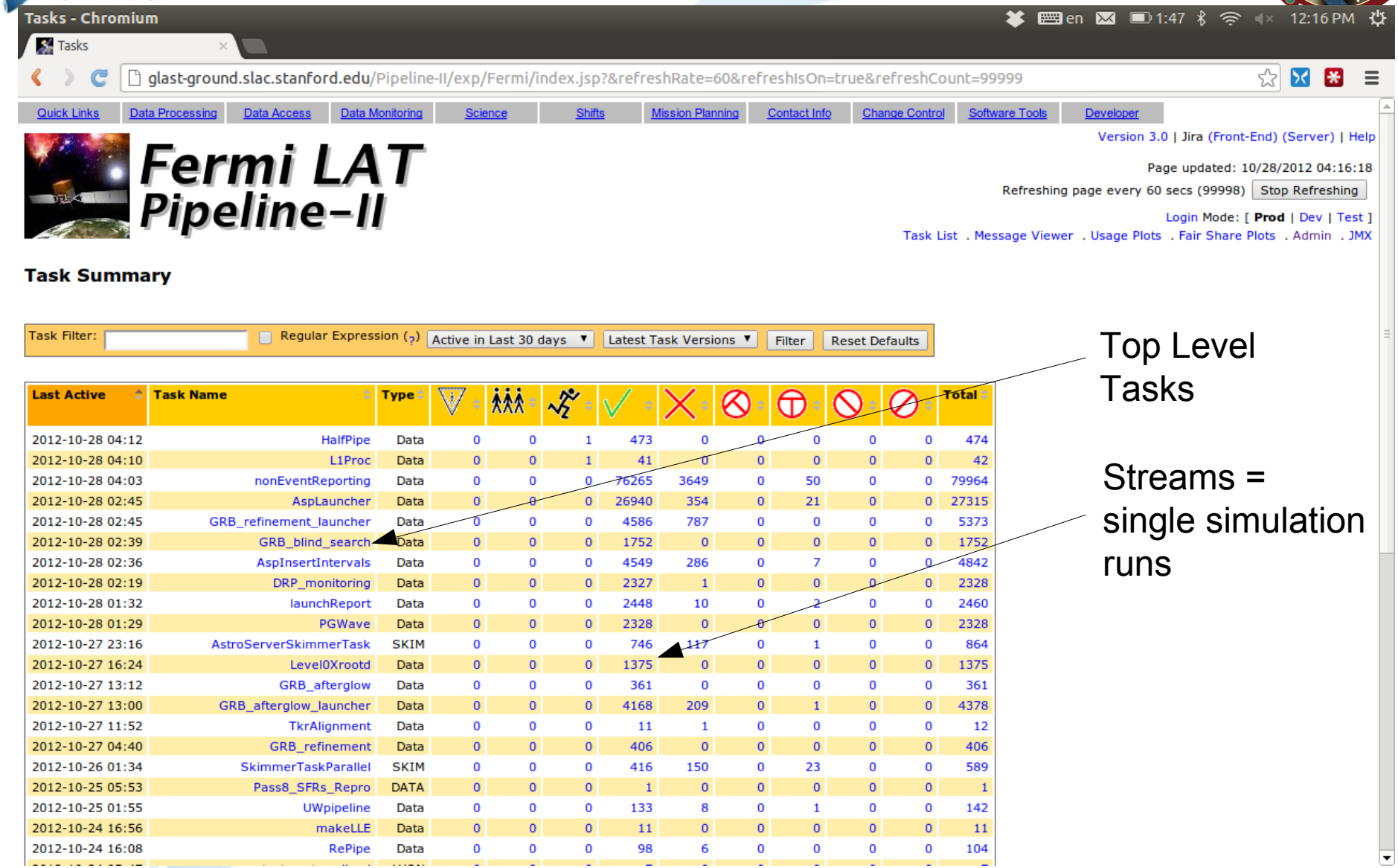
- Can manage arbitrarily complex tasks (type of data) and streams (actual computing instances, batch runs)
- Broker for all centralized data handling:
  - Manages requests and delegates to worker pool
  - Tightly coupled to LAT Data Catalog (plugin)
  - Interfaces with on-site batch solutions (and hopefully soon the Grid)
- Interface to batch farms done through JobControlService (daemon)
  - Supported batch types to date: BQS, LSF, SGE, Condor, PBS
  - Each daemon implements job submission, status tracking and deletion of jobs (streams)
  - Batch node workers send mails to central Pipeline Server to report their status
- Both Web & Command Line Interfaces to interact with Pipeline Server

Pipeline and Data Catalog Implementation



* deprecated or not in use by Fermi-LAT

Pipeline Web Interface – Task Summary



Last Active	Task Name	Type										Total
2012-10-28 04:12	HalfPipe	Data	0	0	1	473	0	0	0	0	0	474
2012-10-28 04:10	L1Proc	Data	0	0	1	41	0	0	0	0	0	42
2012-10-28 04:03	nonEventReporting	Data	0	0	0	76265	3649	0	50	0	0	79964
2012-10-28 02:45	AspLauncher	Data	0	0	0	26940	354	0	21	0	0	27315
2012-10-28 02:45	GRB_refinement_launcher	Data	0	0	0	4586	787	0	0	0	0	5373
2012-10-28 02:39	GRB_blind_search	Data	0	0	0	1752	0	0	0	0	0	1752
2012-10-28 02:36	AspInsertIntervals	Data	0	0	0	4549	286	0	7	0	0	4842
2012-10-28 02:19	DRP_monitoring	Data	0	0	0	2327	1	0	0	0	0	2328
2012-10-28 01:32	launchReport	Data	0	0	0	2448	10	0	2	0	0	2460
2012-10-28 01:29	PGWave	Data	0	0	0	2328	0	0	0	0	0	2328
2012-10-27 23:16	AstroServerSkimmerTask	SKIM	0	0	0	746	117	0	1	0	0	864
2012-10-27 16:24	Level0Xrootd	Data	0	0	0	1375	0	0	0	0	0	1375
2012-10-27 13:12	GRB_afterglow	Data	0	0	0	361	0	0	0	0	0	361
2012-10-27 13:00	GRB_afterglow_launcher	Data	0	0	0	4168	209	0	1	0	0	4378
2012-10-27 11:52	TkrAlignment	Data	0	0	0	11	1	0	0	0	0	12
2012-10-27 04:40	GRB_refinement	Data	0	0	0	406	0	0	0	0	0	406
2012-10-26 01:34	SkimmerTaskParallel	SKIM	0	0	0	416	150	0	23	0	0	589
2012-10-25 05:53	Pass8_SFRs_Repro	DATA	0	0	0	1	0	0	0	0	0	1
2012-10-25 01:55	UWpipeline	Data	0	0	0	133	8	0	1	0	0	142
2012-10-24 16:56	makeLLE	Data	0	0	0	11	0	0	0	0	0	11
2012-10-24 16:08	RePipe	Data	0	0	0	98	6	0	0	0	0	104

Top Level Tasks

Streams = single simulation runs

Pipeline Web Interface Task details

Task Summary: BKG-GR-v20r4p1-OVL-NO-Prescale - Chromium

Tasks | Task Summary: BKG-GR-v2

glast-ground.slac.stanford.edu/Pipeline-II/exp/Fermi/task.jsp?task=91512755

Quick Links | Data Processing | Data Access | Data Monitoring | Science | Shifts | Mission Planning | Contact Info | Change Control | Software Tools | Developer

Fermi LAT Pipeline-II

Version 3.0 | Jira (Front-End) (Server) | Help

Page updated: 10/28/2012 04:19:29

Start refreshing page every 60 secs Start Refreshing

Login Mode: [**Prod** | Dev | Test]

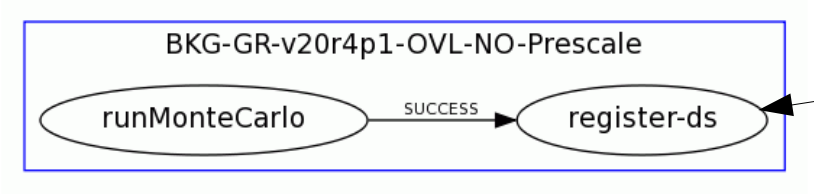
Task List . Message Viewer . Usage Plots . Fair Share Plots . Admin . JMX

summary / BKG-GR-v20r4p1-OVL-NO-Prescale

Task Summary: BKG-GR-v20r4p1-OVL-NO-Prescale 1.2 (XML) ← Task defined as XML

Created by zimmer at 2012-09-28 09:25:46.0 with comment: *clone of background P8 SIMREQ-97 task, no ODF or trigger prescales*

Versions: (1.1) **(1.2)**



Task Layout (c.f. L1)

Top Level Processes

Graph Orientation: Left/Right Top/Bottom . Full Diagram . Diagram source

Task Summary: Canceled: 0, Canceling: 0, Failed: 24, Queued: 0, Running: 0, Success: 2976, Terminated: 1, Terminating: 0, Waiting: 0, Total: 3001

To filter by status click on the count in the status column. To see all streams click on the name in the Name column.

Show running jobs . Show streams . Summary plots

Show processes by status: Waiting Ready Queued Submitted Running Success Failed Terminated Canceled Skipped [ALL] [All not SUCCESS]

Task	Version	Process	Type											Total	Links
BKG-GR-v20r4p1-OVL-NO-Prescale	1.2	runMonteCarlo	Batch	0	0	0	0	0	2976	24	1	0	0	3001	Plots
		register-ds	Script	0	0	0	0	0	2976	0	0	0	25	3001	Plots
			Totals	0	0	0	0	0	5,952	24	1	0	25	**6,002**	

Pipeline States

A word on Pipeline States

- **Pipeline does not resolve detailed batch states, only:**
  - **Waiting (not yet submitted to batch)**
  - **Ready (submitted to batch)**
  - **Queued (*that* is waiting on batch)**
  - **Submitted (just after being queued)**
  - **Running (on the batch)**
  - **Success (email sent successfully)**
  - **Failed (email sent, but contains failure)**
  - **Terminated (email delivery failed or Job Control Exception)**
  - **Canceled/ Skipped (Job Control Exception/ Warning)**

Match to batch-status

- **Any Interface implements "status" query, need to query Queued/Running**
- **No need for DIRACs extended Job Statuses, but could still be useful**

Pipeline Development

- **Core Development**
  - **Tony Johnson**
  - **Dan Flath (now LCLS)**
  - **Brian Van Klaveren**
- **Job Control Service(s)**
  - **Claudia Lavalley (now CTA)**
  - **Stephan Zimmer**

- **Not exclusively for *Fermi*-LAT anymore:**
  - **SRS pipeline used by EXO/LSST/CTA/CDMS**

The VO glast.org

VO manager: Michael Kuss, Francesco Longo
software manager: Johan Bregeon, Michael Kuss, Francesco Longo
VOMS server: voms2.cnaf.infn.it, voms-02.pd.infn.it (replica)
WMS: wms-multi.cnaf.infn.it (6 server load-shared at CNAF, Catania, and Ferrara), prod-wms-01.pd.infn.it (backup)

Site	Place	CPU	Reserved	disk / TB	Fermi SW installed
INFN-PISA	INFN Pisa	3238	87	2	y
TRIESTE	INFN Trieste	2243	150	1.5	y
INFN-T1	CNAF/Bologna	8482	175	60	y
GRIF	LAL/POL Paris	3204	(200)		y
MSFG-OPEN	Montpellier	200	(<=100)		y
PERUGIA	INFN Perugia	166	2 + 24(*2)		y
INFN-NAPOLI-PAMELA	INFN Napoli	184	?		y
SNS-PISA	SNS Pisa	632	(30)		n, down till July
OBSPM	Paris Meudon	112	?		n, configuration issue
INFN-BARI	INFN Bari	3617	75		n
ROMA2	INFN ROMA2	?	?		n, decommissioned?
INFN-CNAF	CNAF/Bologna	8	?		n
CNR-ILC-Pisa	CNR Pisa	4	?		n

400 reserved, 1000 possible

Status: 04/2012

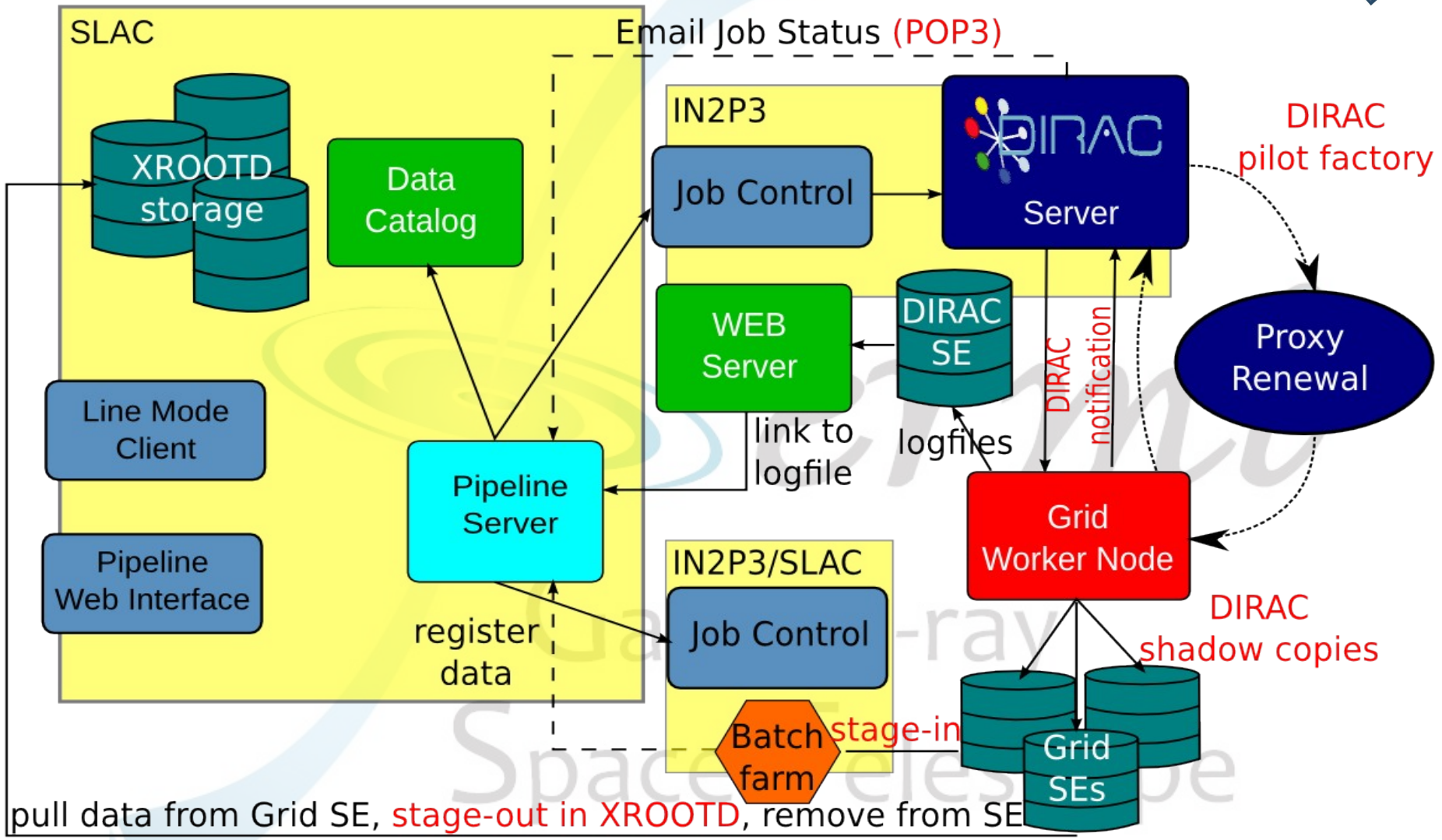
Bringing the Pipeline to the Grid … should be easy?

- **Grid Issues:**
  - **plethora of different Grid middleware (want to be as general as possible), existing glast.org VO uses gLite, could imagine use of OSG/SweGrid cycles**
  - **Inherent Grid deficiencies**
- **Pipeline Design Issues:**
  - **Pipeline was not developed with Grid in mind!**
  - **Workers use emails to tell server about status of job**
  - **Batch jobs usually submitted under one generic login (glastpro, glastmc, etc.)**
  - **A stream in the pipeline may contain several batch instances (different batch job IDs)**
  - **All Data stored at SLACs xrootd (need to get data from workers to SLAC)**
    - **Making xrootd available to outside is difficult (have custom set of bbcp to handle firewalls)**
    - **Instead, use "split-layout" to re-use setup at SLAC/Lyon to perform pull request**
- **LAT enters extended mission time:**
  - **Number of developers dwindle, manpower greatly reduced**

DIRAC as Grid Interface

- **Actually misnomer, should be interface to DIRAC server @ Lyon**
  - **Renew certificates through pilot factory mechanism**
  - **Re-direct emails from batch workers via DIRAC notification scheme to DIRAC server and broadcast to SLAC**
  - **Store data on GRID SE and pull from SLAC**
  - **Store logfiles on DIRAC SE at Lyon (dCache) and allow web access to those**
- **Goodies:**
  - **DIRAC extended status: can be used to more effectively monitor jobs (and debug them)**
  - **DIRAC shadow copies allow automatic data replication**
  - **Can extend VO resources to OSG/SweGrid sites (almost) for free**
- **Focus on bringing MC to the Grid for now, more complex tasks later... (if needed)**

DIRAC Pipeline Integration



Low Level Implementation I – and status

- Any Job Control Service implements interface: submit, status, delete
- dirac-submit (in progress):
  - Needs to report job-ID and registers job in DIRAC
  - Tunable parameters: cputime, environment
- dirac-status (done!):
  - 2 functions: tell reaper whether a job is alive and give (more detailed) information to user
  - Avoid excessive calls, so report all running jobs at once (and cache, unaware of all IDs)
    - Core information: submission details, resource usage
- dirac-delete (done!):
  - Enables deletion of a job, require input job-ID

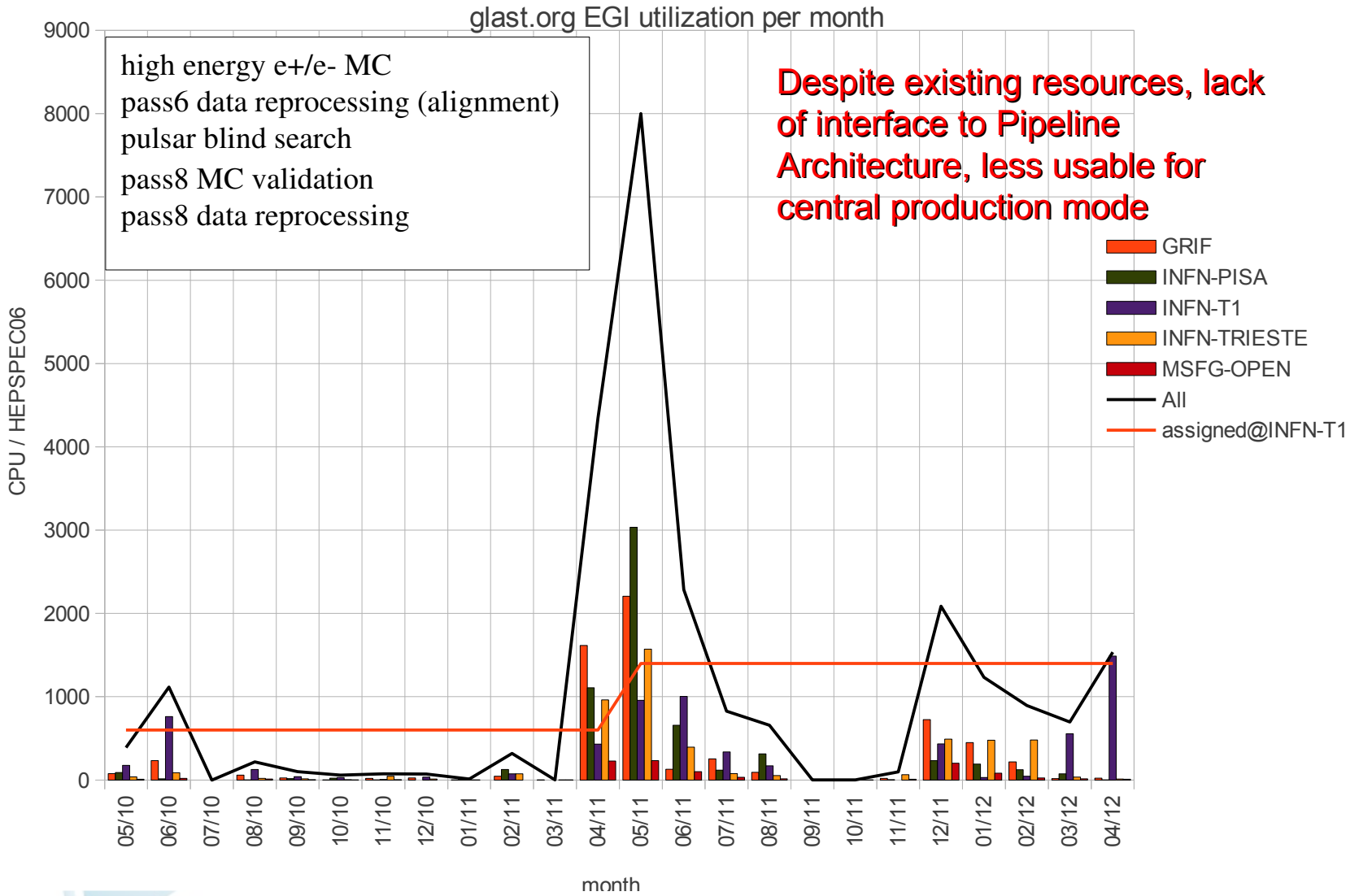
Summary & Conclusions

- *Fermi*-LAT uses processing pipeline to handle all collaboration wide efforts:
  - L1 processing, reprocessing & MC-tasks
  - Tight integration of the pipeline system with the Data Catalog
- Pipeline connects to batch systems through dedicated Job Control Services that implement interface to:
  - Submit, status, delete
- Data needs to be stored on central xrootd @ SLAC
- Submissions done through generic user id glast, glastmc or glastpro
- Use DIRAC to simplify Grid access to existing VO resources:
  - DIRAC notification layer to communicate between batch nodes & DIRAC server (relay to Pipeline server)
  - DIRAC pilot factory for proxy renewal
  - DIRAC SE @ Lyon to store (small) Log files (and access via web-server)
  - Shadow replications on Grid SEs
  - Split-layout to pull data from Grid SE and transfer to xrootd @ SLAC
- Once done, extend resources to OSG/SweGrid?

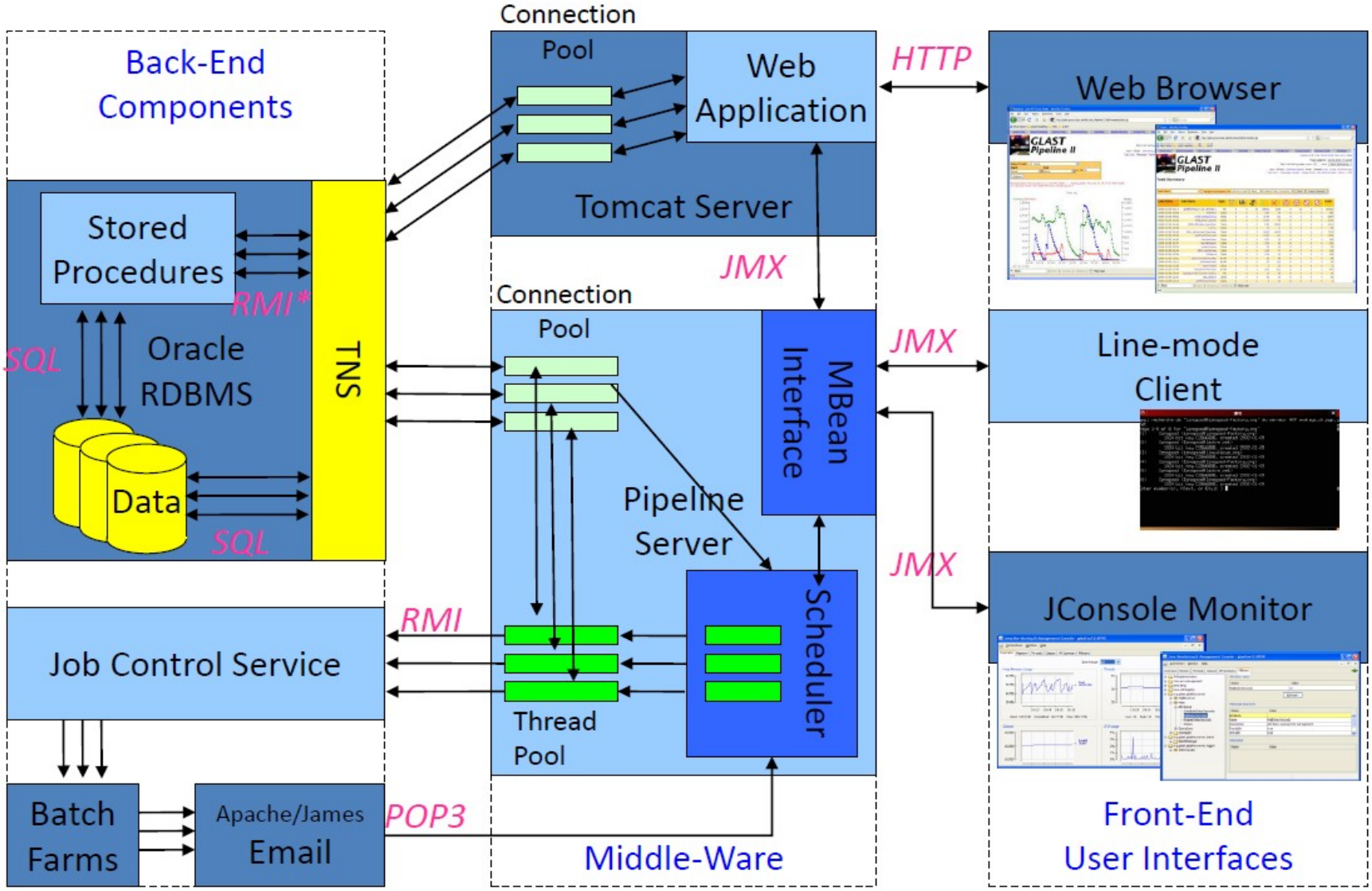
Thank you for organizing this Workshop!

Backup

Utilization

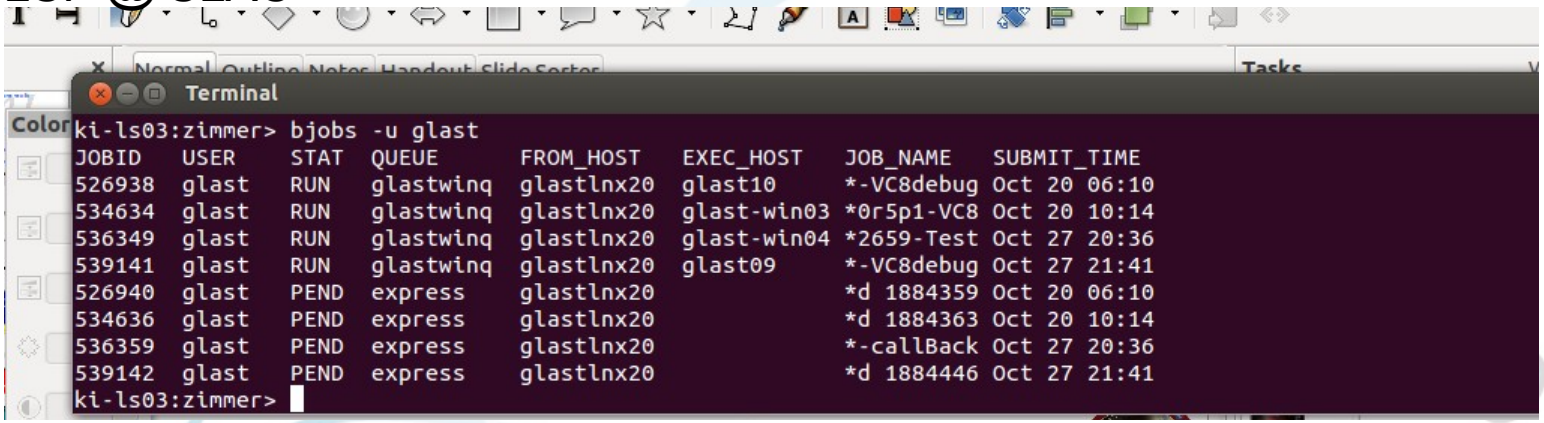


Pipeline Technologies

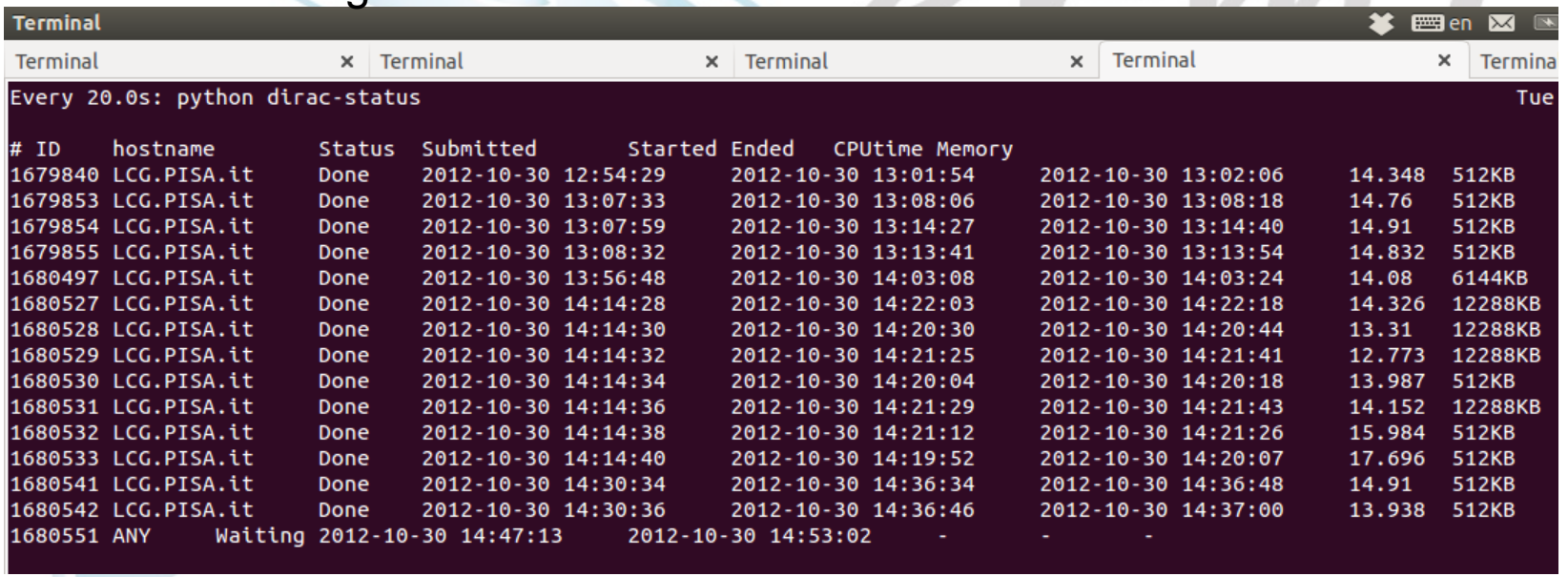


Thanks to DIRAC folks here: dirac-status

LSF @ SLAC



dirac-status using the DIRAC API



Low Level Implementation II – 3 possibilities

- **Java JobControlService calls python module that implements DIRAC API and handles submission**
  - **What are the inputs for this "pipeline-dirac-submit"? How are LFNs handled?**
- **JobControlService builds JDL (bindings for java?)**
  - **DIRAC.Interfaces.API.Job implements _toXML, can we also use xml import?**
  - **Is there an xml schema file (xsd) that we could use to build our xml outside of DIRAC?**
- **Use Jython to expose relevant parts of API to Job Control Service**
  - **Could be added to contributed code to the DIRAC community**
  - **Most difficult for me since I don't know Java well enough...**