1. **LIGO Data Analysis Systems**
   ***Data Management and Analysis***

2. **LIGO/LSC Analysis Software:**
   ***Case Study of Inspiral Analysis***

3. **Open Science Grid (OSG)**
   **Activities by LIGO and LSC**

**LIGO NSF Review**
**October 23-25, 2006**

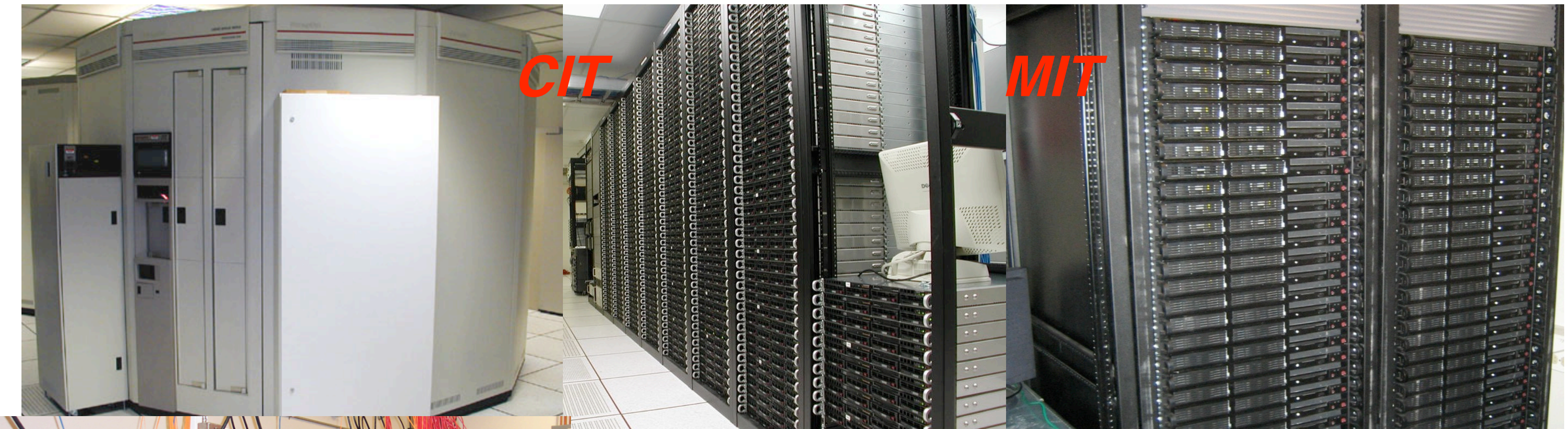
*Stuart Anderson, Kent Blackburn, Duncan Brown*

*LIGO Laboratory*
*California Institute of Technology*

# Outline

- *LIGO Data Management and Distribution (S. Anderson)*
  - Cluster access to data
  - Storage technology and dataflow
  - Size of problem (rates and volume)
  - Data and metadata replication
  - Hardware and Software updates during S5
- *LIGO Data Analysis Software (D. Brown)*
  - Software working groups
  - Example analysis pipeline
- *LIGO and the OSG (K.Blackburn)*
  - Overview
  - Participation
  - Applications
  - Challenges

# Data Management and Distribution





- ***Stuart Anderson (Group Leader)***
  - Greg Mendell (LHO Scientist)
  - Ben Johnson (LHO Systems Admin.)

  - Igor Yakushin (LLO Scientist)
  - Dwayne Giardina (LLO Systems Admin.)

  - Dan Kozak (CIT Systems Admin.)
  - Erik Espinoza (CIT Systems Admin.)
  - Phil Ehrens (CIT Systems Admin.)

  - Fred Donovan (MIT Systems Admin.)

# LIGO Data Analysis Systems

LIGO Laboratory operated systems:

- 4 Production systems
  - » 2 on-site: observatory archive and real-time processing (LHO/LLO)
    - – 1100 GHz P4-Xeon/100TB disk/6 STK9940B tape drives/700 slot robot each
  - » 2 off-site: central archive and multi-IFO processing (CIT/MIT)
    - – 2900 GHz Opteron-core/230TB disk/10 STK9940B tape drives/6000 slot robot (CIT)
    - – 450 GHz P4-Xeon/40TB disk (MIT)
- 1 Development system
  - » Run daily builds of LDAS from CVS
- 1 Test system
  - » Pre-release testing
  - » Integration testing of LSC/LIGO software

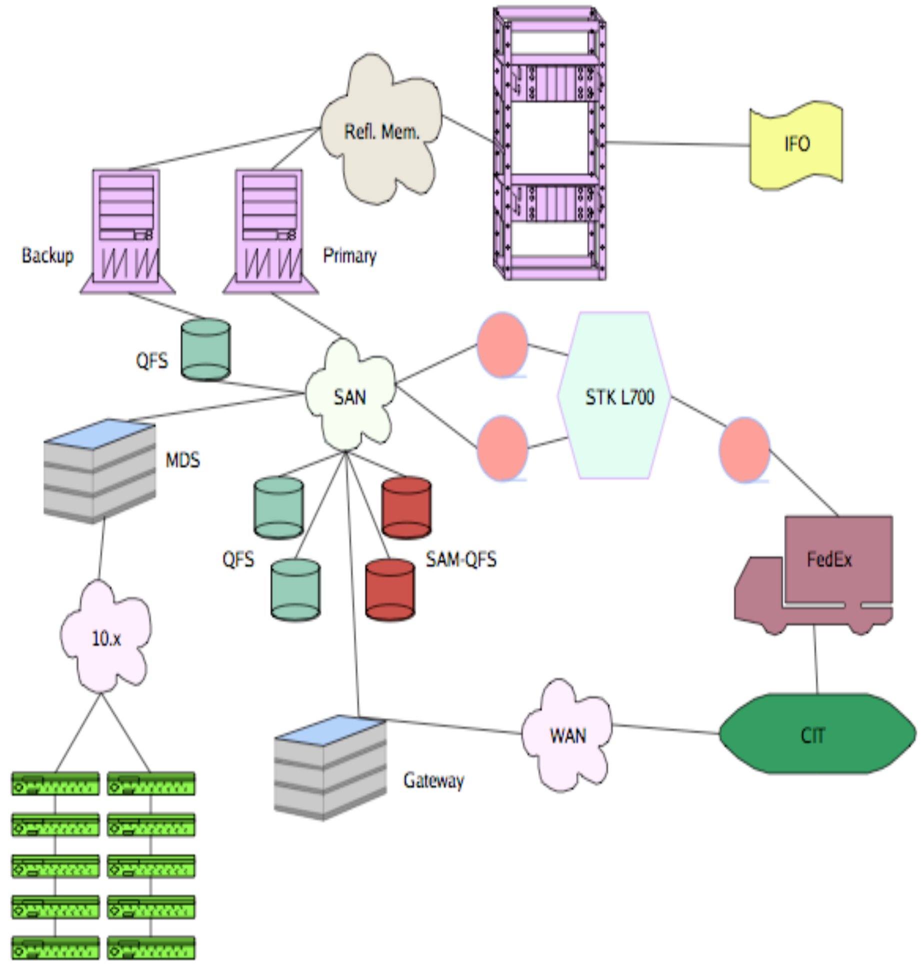
# Mass Storage Software

SAM-QFS is used to manage the IFO data and intermediate data products from cluster analysis

- Shared filesystem with metadata on shared disk for initial frame filesystem to minimize subsystem interface issues (security and reliability).
- Scalable to size of LIGO with multiple read/write servers
  - However, single tape drive server per filesystem
- Free for research use
- Presents fully POSIX compliant filesystem for ease and portability of application development
- Portable data format (GNU Tar)
  - Disaster recovery
  - Future migration of data if software product is terminated
- Seamless migration of physical tapes between Observatories and central archive at Caltech

# LIGO Data Management

- IFO data is stored in an International GW self-describing binary format (Frame).
- Duplicate servers write independent copies to 2 different QFS filesystems in different buildings for redundancy and quality check
- These filesystems are the interface betwee the data acquisition and archiving/analysis systems (no TCP/IP).
- Reduced data sets are created in near real-time and distributed over the WAN using gridftp.
- Two copies of the full IFO data are written to SAM-QFS tape and one read-only copy is sent to the central archive at Caltech.
- Local cluster analysis jobs have access to IFO data with a latency of ~1 minute, reduced data is available off-site with a latency of ~1 hour, and off-site raw data in ~1 week.



# LIGO Data Products

## Dual-copy tape utilization (STK9940B)

| LHO | Data Rate (MByte/s) | 1 Yr (TByte) | 1 Yr (tapes) | Comp. Ratio | Look-back |
|---|---|---|---|---|---|
| Level 0 raw | 9.063 | 272 | 1800 | 1.7 on tape | ~1 Month* |
| Level 1 RDS | 1.391 | 42 | 460 | 1.2 in files | ~most S5* |
| Level 3 RDS | 0.117 | 3.5 | 39 | 1.2 in files | all S5 |
| Level 4 RDS | 0.029 | 0.87 | 10 | 1.2 in files | all S5 |
| h(t) | 0.266 | 8.0 | 88 | -- | all S5 |
| SFTs | 0.032 | 0.96 | 11 | -- | all S5 |
| **LLO** | | | | | |
| Level 0 raw | 4.406 | 133 | 970 | 1.5 on tape | ~2 Month* |
| Level 1 RDS | 0.750 | 23 | 249 | 1.2 in files | ~most S5* |
| Level 3 RDS | 0.059 | 1.8 | 20 | 1.2 in files | all S5 |
| Level 4 RDS | 0.015 | 0.45 | 5 | 1.2 in files | all S5 |
| h(t) | 0.133 | 4.0 | 44 | -- | all S5 |
| SFTs | 0.016 | 0.48 | 5 | -- | all S5 |
| Totals: | 16.3 | 490 | 3701 | as above | All Science Runs at CIT |

** The remainder of these data are off-line in tape cabinets on-site and available on demand as well as being on-line at CIT.*

# LIGO Data Archive

| | CIT | LHO | LLO |
|---|---|---|---|
| Frame files | 29 M | 14 M | 25 M |
| Frame size | 950 TB | 610 TB | 270 TB |
| Cluster home files | 39 M | 25 M | 11 M |
| Cluster home size | 9.4 TB | 6.2 TB | 2.8 TB |
| Tape volumes (online/offline) | 4990/370 | 610/1700 | 670/550 |

# LIGO Data Distribution

- Reduced data products are distributed over the WAN using the LIGO Data Replicator (LDR), a grid tool built on top of the Globus Toolkit to manage both data and metadata transfers:
  - » Parallel data transfers are accomplished using GridFTP.
  - » Metadata uses the Replication Location Server to hold mappings of filenames to paths/URL's at a given cluster location.
- GEO calibrated strain data are replicated via LDR from Hannover, Germany to Caltech.
- Data are re-distributed from CIT to Tier-2 and Tier-3 centers via LDR.
- Discussions have started with VIRGO regarding joint data exchange and possible interoperability problems between US and EU grid middleware.

LIGO

# Data Transfer Performance

- Observatories to Caltech
  - LHO upgraded Esnet connection to GigE
    - Esnet upgraded backbone (SEA/SNV) to OC-192
    - Transferring frame data at 20-30 MByte/s
  - LLO upgraded to GigE connection to LSU
    - LSU connection to I2 is OC-3
    - 5-10 MByte/s
- CIT to Tier-2 centers
  - MIT 7 MByte/s
  - PSU 8 MByte/s
  - UWM 20 MByte/s

# Hardware Updates During S5

- Approximately doubled power and HVAC available for clusters.
- Upgraded LLO/LHO clusters from 70/140 to 210/210 dual-Xeon nodes by relocating CIT cluster.
- Installed a new CIT cluster (329 dual-socket dual-core Opteron).
- Adding additional GRID related servers (Linux) as grid model develops, e.g., OSG.
- Increasing SAM-QFS cluster home directory space as needed.
- Replacing T3 RAID system (5.5 yr old).
- Using shelf storage for tapes at Observatories and Caltech.
- Planned upgrade of SPARC/Solaris servers to Opteron/Solaris.

# Software Updates During S5

- Continuing with Solaris 10/Fedora Core 4 for S5.
  - » Still struggling to find the right compromise between the latest support and stability in Linux for post-S5.
- New CIT and UWM clusters are Opteron based running 64-bit FC x86_64:
  - » VDT 64-bit release on 2 Aug 2006 (1.3.11)
  - » Condor 64-bit was released on 16 Oct 2006 (6.8.2)
  - » Current LDAS development branch is running on 64-bit
- Porting LDAS software to Solaris 10 x86 to allow upgrading of floating point intensive servers from SPARC to Opteron.

# LIGO/LSC Analysis Software: Case Study of Inspiral Analysis

# Software Working Groups

The LIGO software is managed by the following teams:

- Control and Data Systems (CDS)
  - Real-time computer systems (VxWorks) and data distribution (Reflective Memory)
  - Writing IFO data to Frames (QFS)
- LIGO Data Analysis Software (LDAS)
  - Frame I/O library (FrameCPP)
  - Generating and distributing reduced data sets (RDS)
  - Managing cluster jobs (Condor)
  - Managing Metadata catalogues (Segment database and Q-replication)
- Data Analysis Software Working Group (DASWG)
  - LSC wide algorithm libraries (LAL, LALapps, Matapps, Root)
  - LIGO Data Replicator (LDR)
  - Online Analysis System (Onasys)
  - Detector Monitoring Tool (DMT)
  - LIGOtools/LDG software bundles
- OSG migration team
  - Working to port example LIGO pipelines to OSG using VDS
- Einstein@home
  - Use BOINC to harness home computers (currently ~50k machines)

# The Goal

There are 4 main search groups analyzing LIGO data using these tools for astrophysical publications:

- Inspiral (coalescing stellar binary systems, e.g., binary neutron stars)
- Stochastic (GW equivalent of microwave background)
- Periodic (Pulsars and other nearly periodic oscillators, e.g. LMXB's)
- Burst (unmodeled events, e.g., Supernovae explosion)

To highlight the various software components necessary to extract Astrophysical statements from LIGO IFO data consider the example of an inspiral analysis.

# Overview of Inspiral Analysis

- LSC search for compact binary coalescence
  - Binary Neutron Stars
  - Binary Black Holes
  - Spinning Binary Black Holes
  - Primordial Black Holes

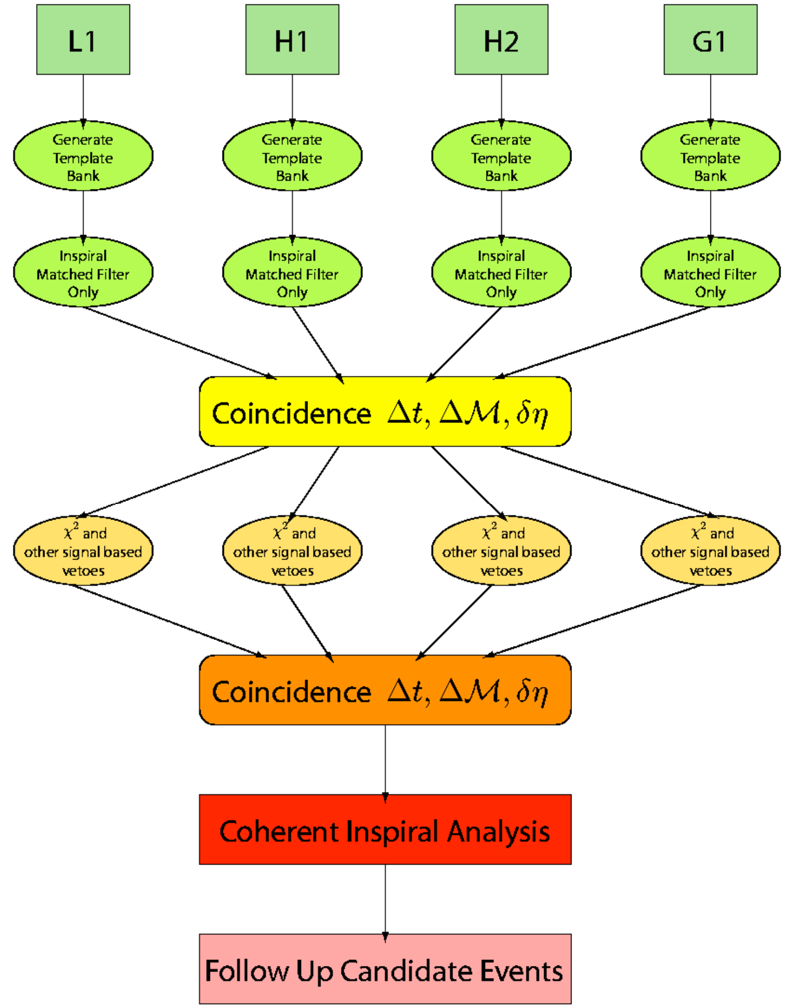
- Two components: online and offline analysis
  - Online single detector analysis for detector characterization
  - Offline analysis for multidetector analysis, pipeline tuning, Monte Carlo simulations, etc.
  - More of pipeline is migrating to online running as we gain a better understanding of analysis tuning

# Elements of Inspiral Analysis

- Programs which perform the individual aspects of the analysis
  - e.g. lalapps_tmpltbank to generate match filter template bank,
  - lalapps_inspiral to filter detector data through bank, find triggers, etc.
  - Stand alone C code written by scientists which can be run either on desktops or on clusters as part of large analysis pipeline
- Infrastructure which allows the construction of pipelines
  - Grid/LSC User Environment (GLUE): describes elements of workflows and allows construction of complex analysis pipelines
  - Can generate pipelines for LSC Data Grid or Open Science Grid
- Pipeline construction scripts
  - Written by scientists to implement desired analysis using various components
  - Use GLUE to create files needed to execute analysis

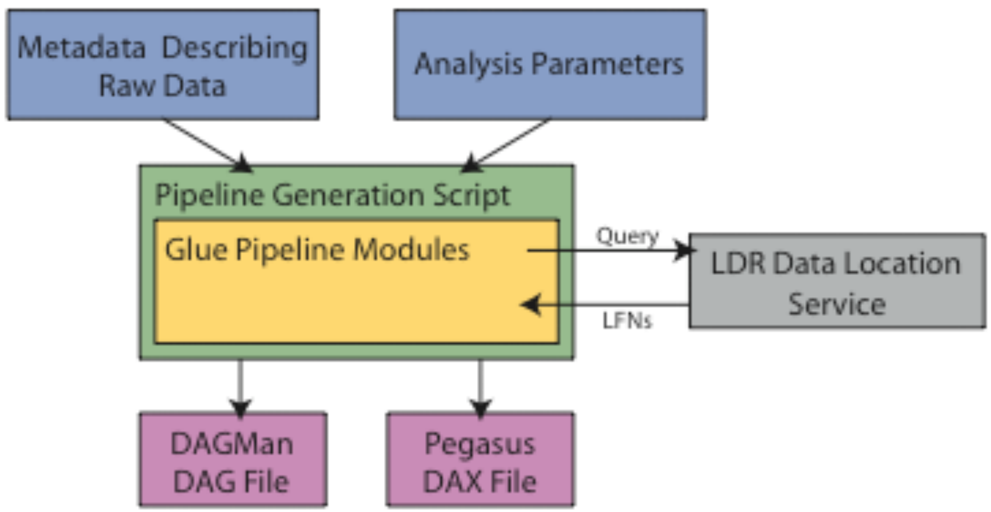
# Inspiral Analysis Pipeline

- Multidetector pipeline
  - Pipeline topology is same for all inspiral searches
  - BNS, PBH, BBH, spinning BBH
  - Different template/filtering code used for different searches
  - Can be used for LIGO-GEO and LIGO-VIRGO analysis
- Pipeline Description
  - Inspiral search run on each IFO
  - Look for triggers coincident in time and mass between detetors
  - Follow up with signal-based vetoes
  - Perform coherent analysis of surviving triggers
  - Follow up candidate events



# Building Pipelines with GLUE

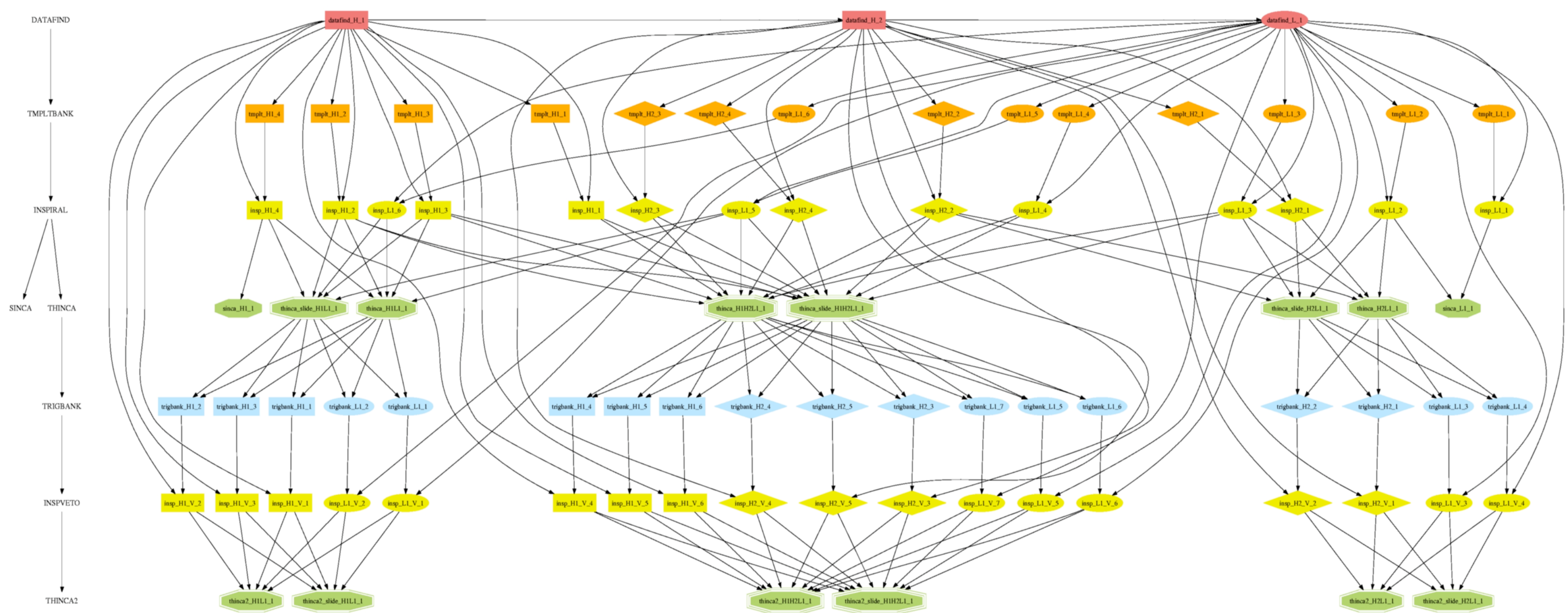
- Scientist writes pipeline generation script to implement desired topology
- Input to pipeline script:
  - Metadata describing when detectors were in science mode
  - Parameters of analysis (e.g. mass range, thresholds, coincidence windows etc.)




- Output is workflow that can be run on LSC clusters or OSG

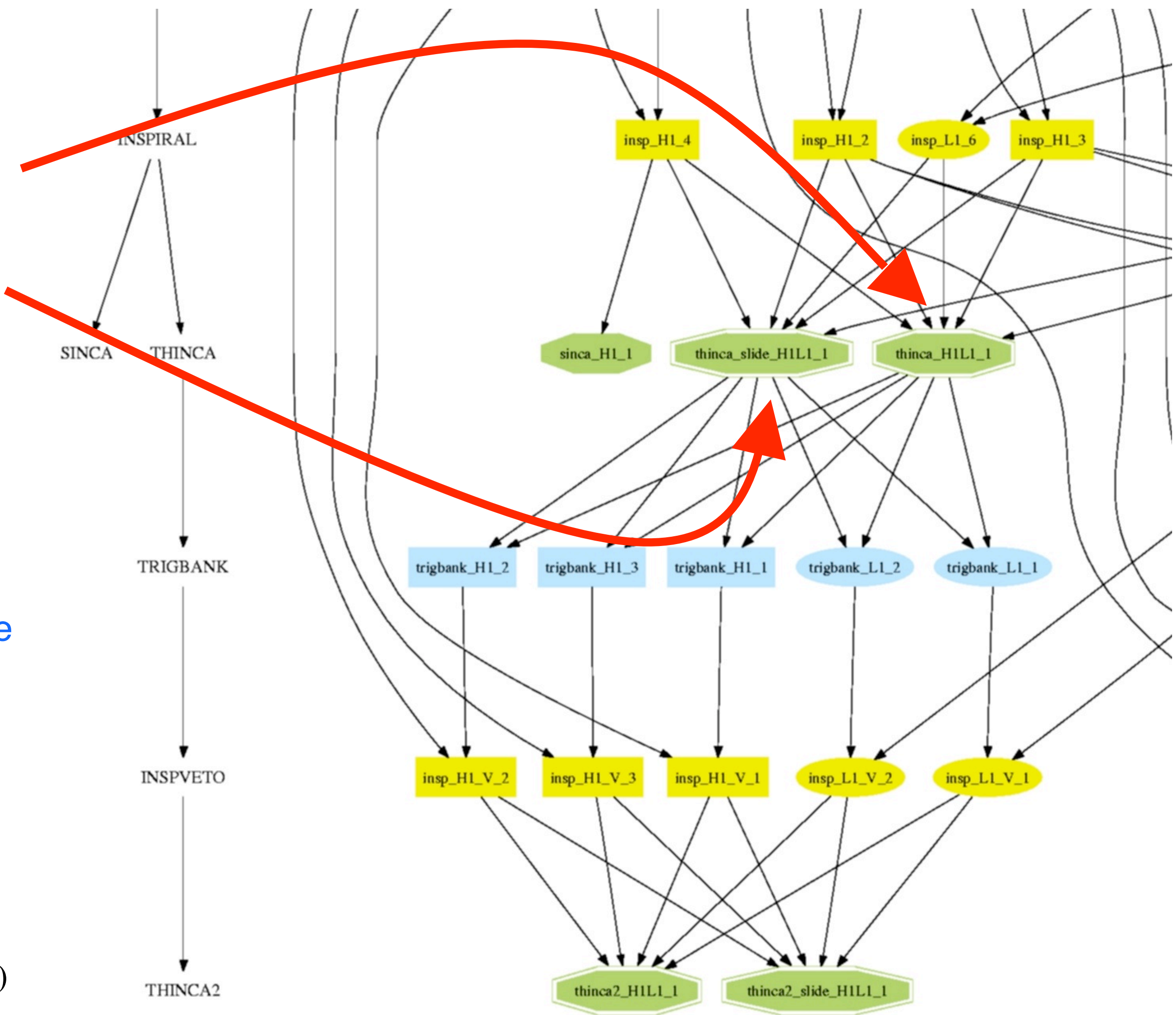
# Executing Analysis Pipeline

- Workflow written contains instructions for Condor to execute analysis on cluster
- Figure shows small subset of inspiral Directed Acyclic Graph (DAG)



# Detail of Analysis Pipeline

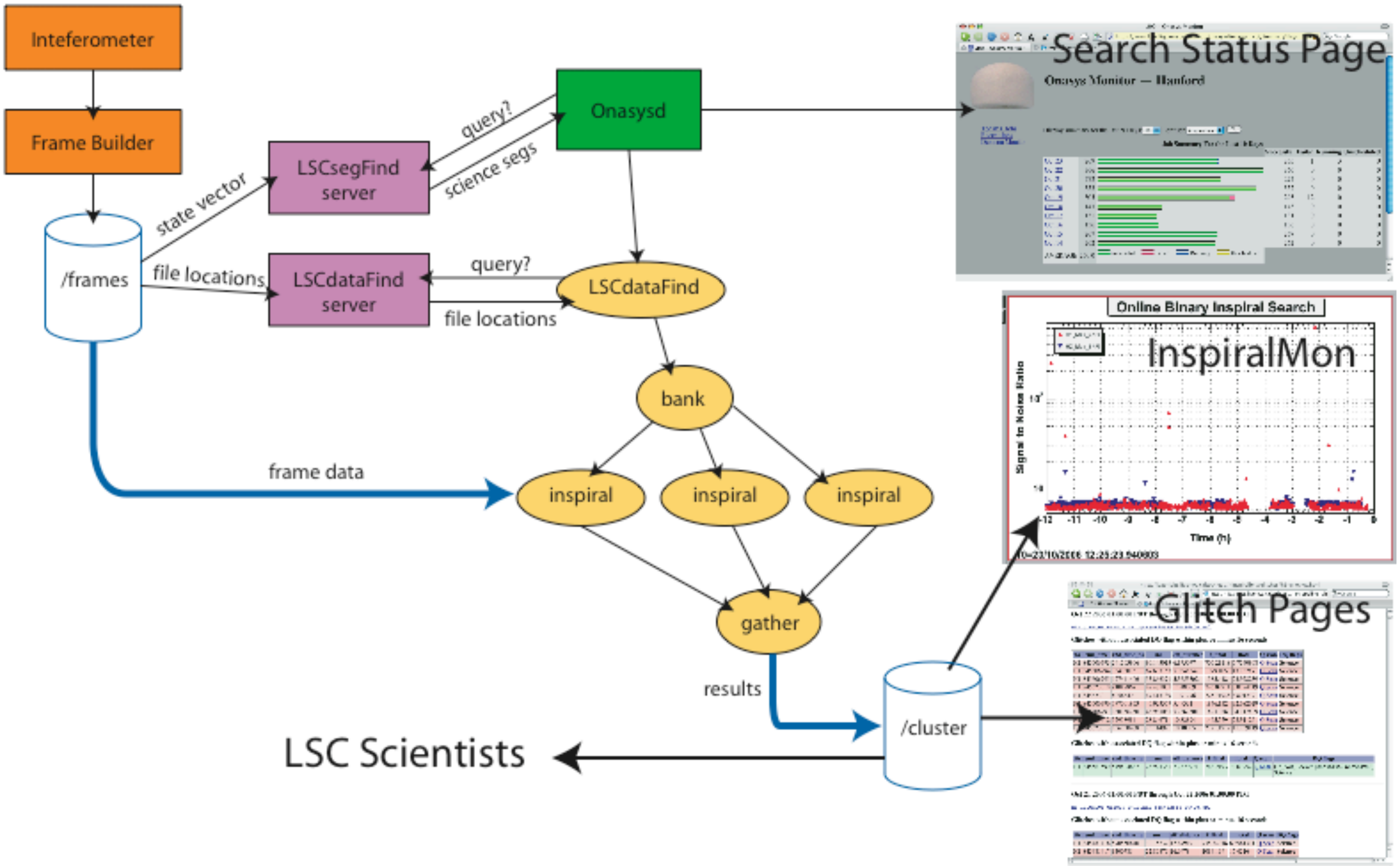
- Close up of section of pipeline
  - » "thinca" is coincidence code
  - » Timeslide analysis to extimate background is built in
- Condor handles all job scheduling, management, etc.
  - » Reports success of failure of DAG to user
  - » Creates "rescue DAG" if any jobs failed



# Online Inspiral Analysis

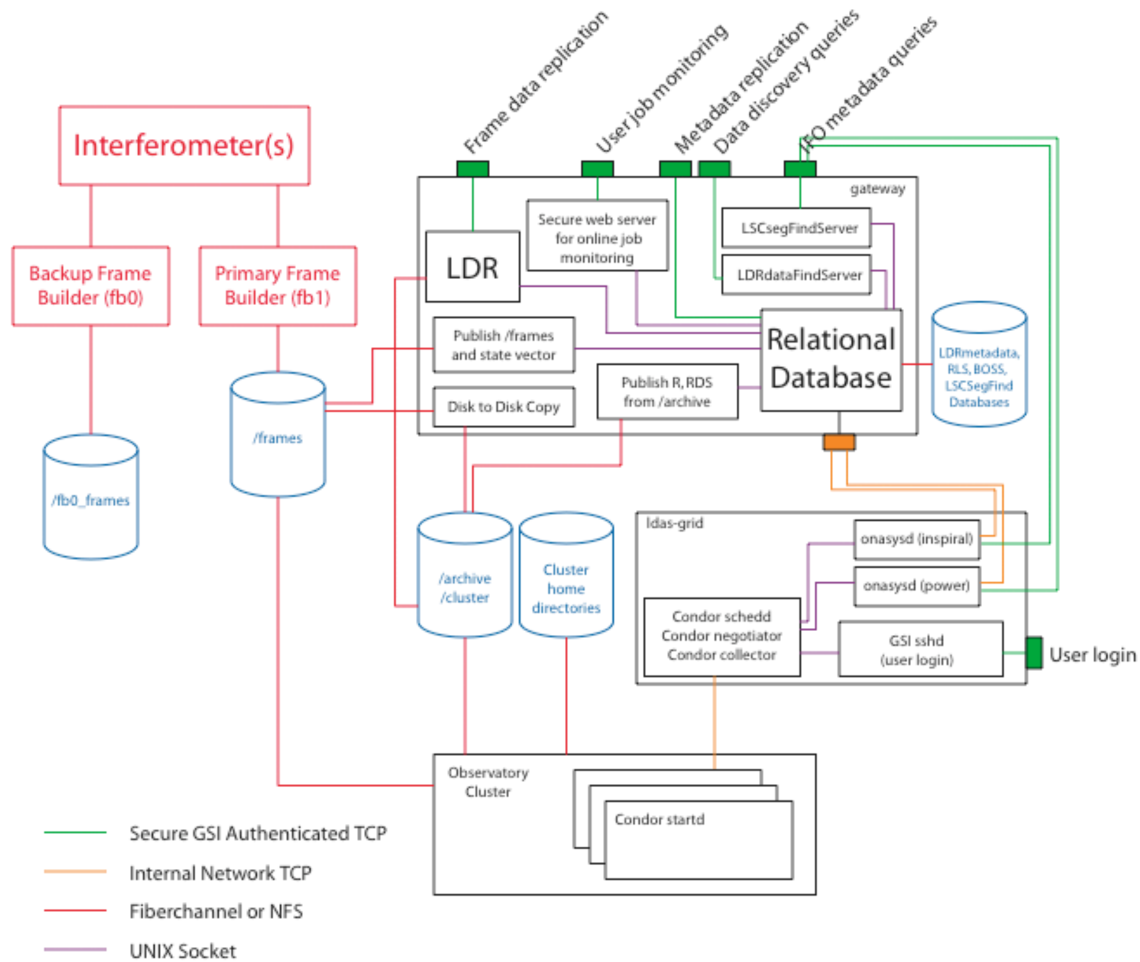
- Inspiral analysis runs online at observatories for detector characterization
  - » Provide rapid feedback to operators and scientists in control room
  - » Inspiral "figure of merit," analysis of hardware injections, "loudest glitch" pages
- Executes analysis in near-real time
  - » ONline Analysis SYStem (Onasys) launches inspiral DAGs when data is available
  - » Subset of inspiral analysis pipeline runs on observatory cluster
  - » Inspiral code analyzes data and produces triggers in XML format
  - » ~ 256 second latency for production of triggers
  - » Triggers summarized, displayed in control room and made available to LSC for additional analysis

# Analysis View of Online Inspiral Search



# Infrastructure View of Online Inspiral Search

- Framebuilder (Solaris) running CDS realtime inferface to FrameCPP I/O library writes detector data to disk
- Frame file locations discovered using LDAS diskCache API
- Detector state metadata extracted from frames, stored in relational database and replicated to other observatory and CIT
- Data locations published into LIGO Data Replicator (LDR)
- Onasys periodically launches DAGs to analyze data
- Condor DAGman manages execution of workflows on cluster
- Results made available to users



# Offline Inspiral Analysis

- Scientists use Glue and pipeline scripts described previously to create Condor DAGs
- Glue writes necessary DAGman files to launch jobs (language of Condor)
- Scientist runs DAG under Condor on LSC Data Grid cluster
  - » comes back a few (sometimes many) days later to look at progress and/or results
- This is how our current scientific results for publications are obtained

- Glue can also write “Abstract DAG” (DAX) files which describe workflow (language of OSG Virtual Data System)
- Use Grid middleware “Pegasus” (Planning for Execution in Grids) developed by USC ISI to turn DAG into Condor-G DAGs for OSG
- Pegasus writes Condor-G DAG with necessary data staging and retrieval steps for running on non-LSC computing clusters

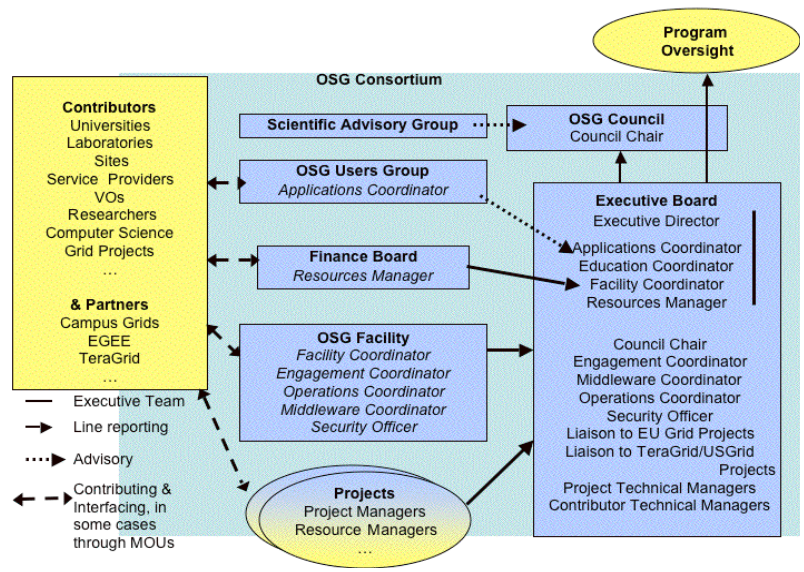
# Running Inspiral Analysis on the OSG

- Using Pegasus has some problems for LSC scientists
  - Knowledge of VDS and/or OSG issues is required
- Solution: LIGO has developed an interface around Pegasus that will take the users pipeline DAX, discover the necessary OSG information, use Pegasus to plan the DAG and execute it using Condor-G
- LIGO/OSG team currently working with Pegasus team and OSG sites to run inspiral analysis on OSG
  - Currently dealing with issues such as available space for data, etc.
- Current application used for OSG development is inspiral software injection Monte Carlo
  - Same analysis used by LSC scientists on LIGO data grid
  - Has been run successfully on several OSG sites

# Open Science Grid (OSG) Activities by LIGO and LSC

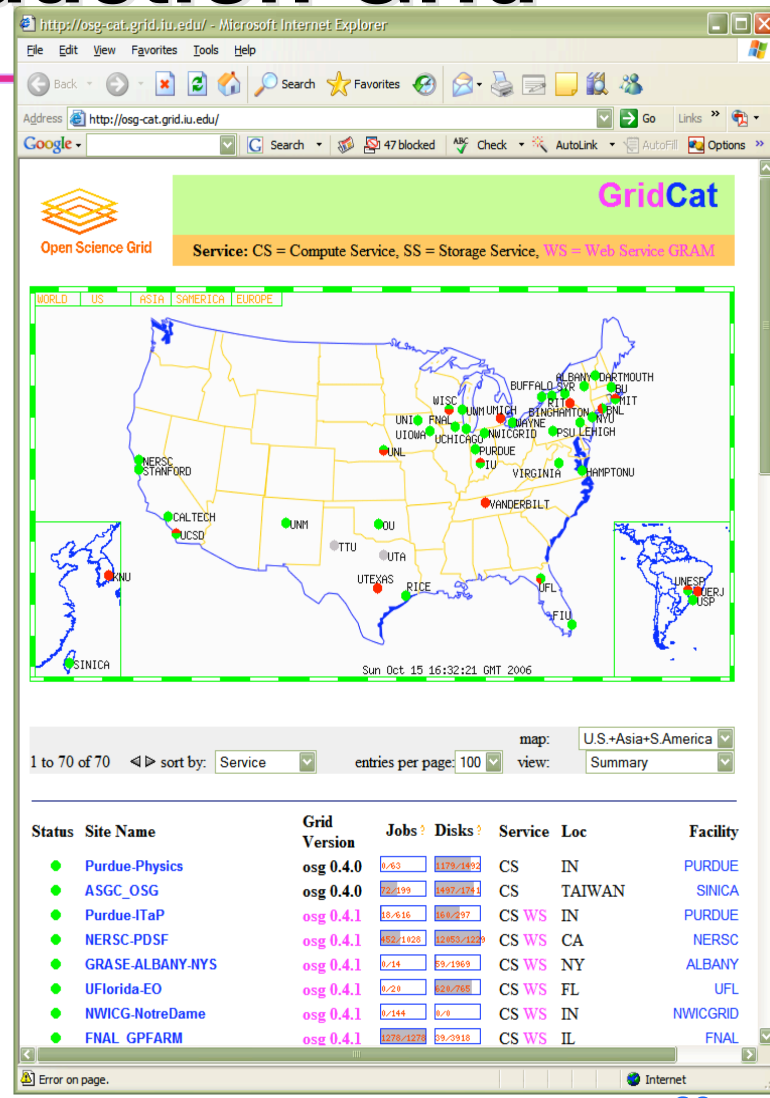
# OSG Organization

- Kent Blackburn (Caltech) recently appointed to OSG Executive Board as the co-Resources Manager w/ Paul Avery
- Kent Blackburn is LIGO liaison to OSG and participates in many OSG activity groups
- Warren Anderson (UWM) recently appointed LIGO Virtual Organization (VO) Council Member to the OSG
- Murali Ramsunder (PSU) has provided the LIGO VO Support Center Role
- David Meyers recently asked to take a lead role in the Integration Testbed Activities and to provide leadership to the new Validation Testbed Activity



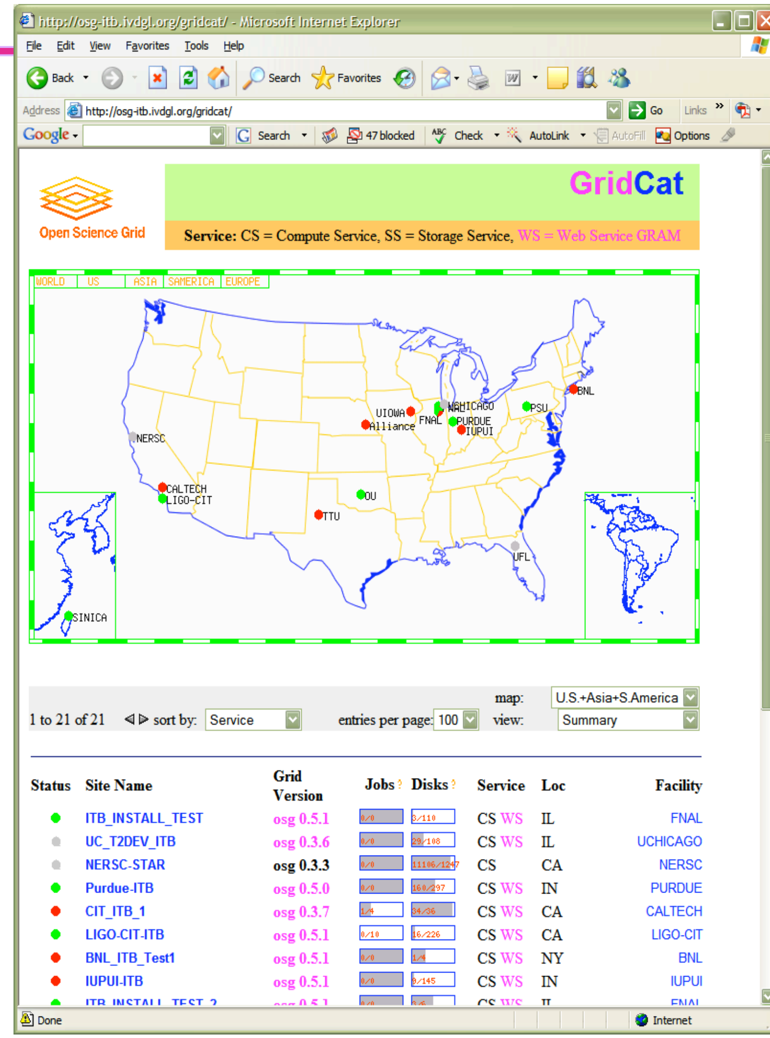
# OSG Production Grid

- Evolved out of GRID3
- Roughly 70 sites on OSG Production Grid
  - UWM & PSU from LIGO
- Roughly 25000 CPUs
  - UWM & PSU: 175+312 CPUs
- Used to support VO specific application runs for simulations and data analysis
- Current OSG release version 0.4.1.
  - Preceded by 0.2.0 and 0.4.0
  - Improving patch technology
  - 0.6.0 expected by March 2007



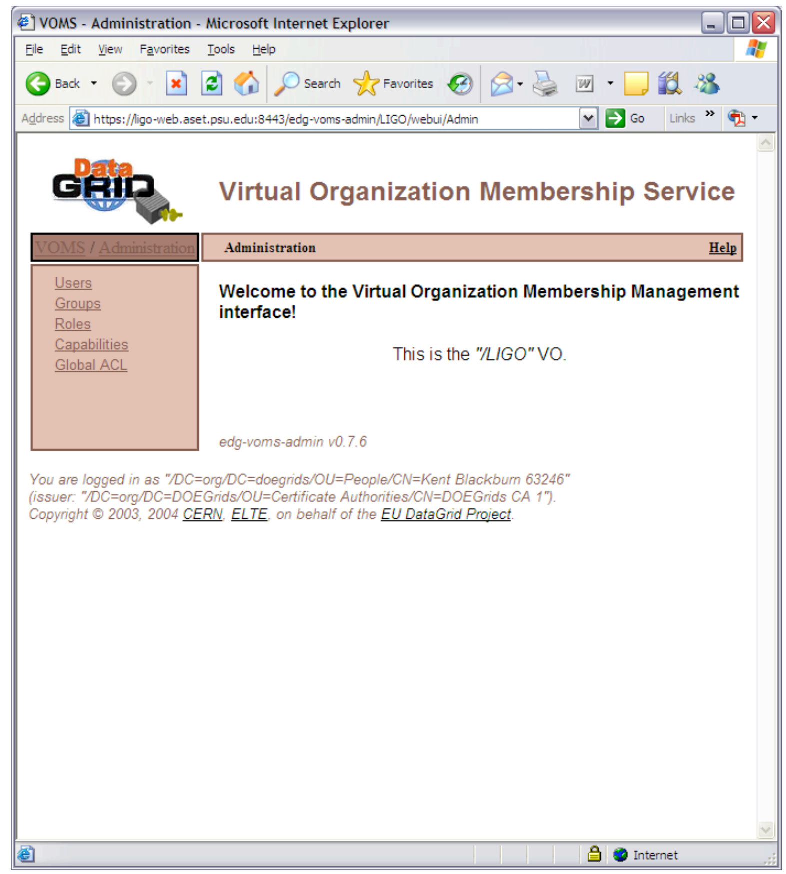
# OSG Integration Testbed

- Roughly 20 sites currently in the OSG
  - » LIGO-CIT & PSU from LIGO Scientific Collaboration
- Almost 5000 CPUs (some shared with production sites)
- Used to as testbed to validate new OSG software stacks and develop new VO applications.
- Adding a Validation Testbed to validate software stack at the level of the grid itself
  - » Initially will be composed of three sites
  - » David Meyers (LIGO/Caltech) asked to lead this effort



# Virtual Organization

- PSU hosts LIGO's VOMS server
- As of October 16th, 2006 there were 147 OSG:LIGO VO certificates registered
- Two productions sites supported:
  - » *UW Milwaukee*
  - » *PSU*
  - » *LIGO Lab considering setting up another site at MIT*
- Two LIGO integration testbed (ITB) sites:
  - » *Caltech*
  - » *PSU*



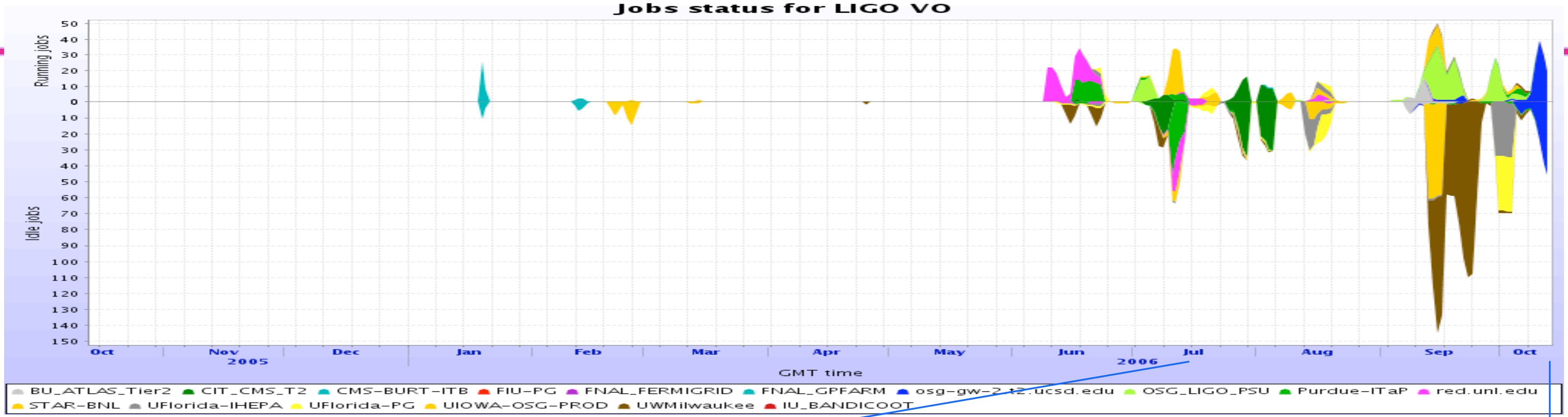
# LIGO's ITB Cluster

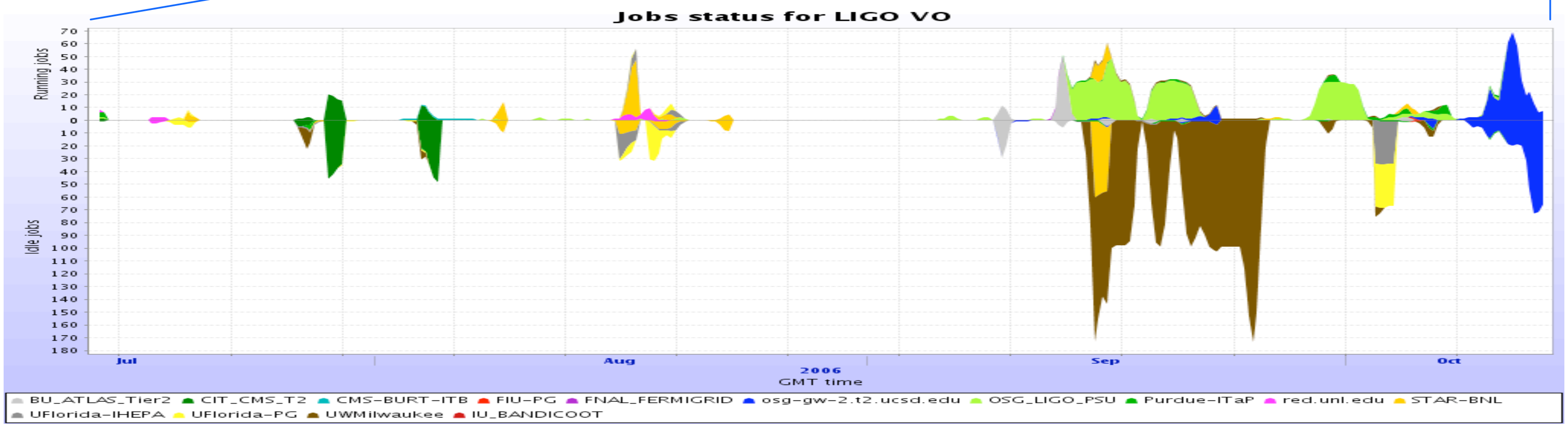
- Recently moved into main computer room at Caltech
- Eight dual CPU nodes
  - All are 32 bit hardware
  - Plan to add two more nodes
- 1.6 TB disk space
- Used to test new releases of the OSG software stack (& propose future changes supporting LIGO's OSG application development)
- Being used to develop LIGO applications before the OSG Production Grid
- Will also be part of the new validation testbed (VTB)

# LIGO Applications on OSG

- Focusing on binary inspiral searches
  - Some support from two caltech software developers, caltech post-doc, Pegasus developers
- First successful test run on October 1st, 2005 on LIGO's integration testbed cluster using PIPE, a test DAG of 940 dag-nodes using 5.1 GBs of data.
  - *Two weeks later, succeeded in running on LIGO's OSG production sites*
  - *Demonstrated able to run this workflow at 10 different OSG Production sites*
- This summer we began running a much larger binary inspiral workflow (HIPE)on a few larger OSG sites (PSU and UCSD)
  - *Close to 700 gigabytes of data involved*
  - *Over 79,000 DAG nodes (two orders of magnitude larger; 40m-lab -> LIGO)*
  - *Run time is typically a week on these sites*
  - *Improvements to Pegasus workflow planner needed*
    - *Implemented a scaled down test version of workflow called Nano-HIPE with 356 DAG nodes*
  - *Approaching scale necessary for conducting science*
    - *Need to expand template space by several to reach science*
- Began prototype activities to understand Burst analysis workflows on the OSG
  - *Primary difference is that this group is not working in the with abstract DAG framework*
  - *Based on compiled Matlab and need to have Matlab runtime libraries managed as part of the workflows.*
  - *Have only run this workflow on the Caltech Integration Testbed site*

# LIGO Jobs on the OSG



# OSG Challenges

- Ease of porting LIGO data analysis applications from LDG to OSG
- Efficiency of utilization of other OSG VO sites
- More frequent releases (or updates) of software stack to better support VO and application needs
- Down the road issues:
  - » *Now that it is funded …*
    - – *Organization challenges*
    - – *Staffing challenges (LIGO staff in place!)*
  - » *Storage Management*
  - » *Interoperability with TeraGrid and EGEE as well as LDG*

# OSG Summary

- LIGO and the LSC have contributed significantly to the OSG process, providing personnel at all levels from oversight to administrators, developers and users.
- OSG has only released two (plus patched) production software stack, 0.2.0, 0.4.0, 0.4.1
  - *LIGO has worked on individual sites (mostly within the CMS VO) to get the LIGO VO supported during the release interim*
- LIGO has now developed an application based on the binary inspiral search code for running on the OSG
  - *Requires patches beyond officials OSG 0.2.1 software stack*
  - Data transmission times significantly mask CPU benefits for LIGO application
  - Storage requirement greatly constrain sites which can support LIGO binary inspiral application until Storage Elements (SE) are integral part of OSG sites