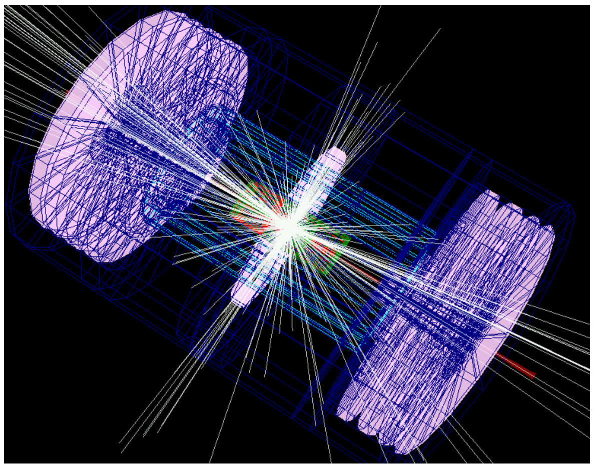
Grid Analysis Environment (GAE) Work at Caltech



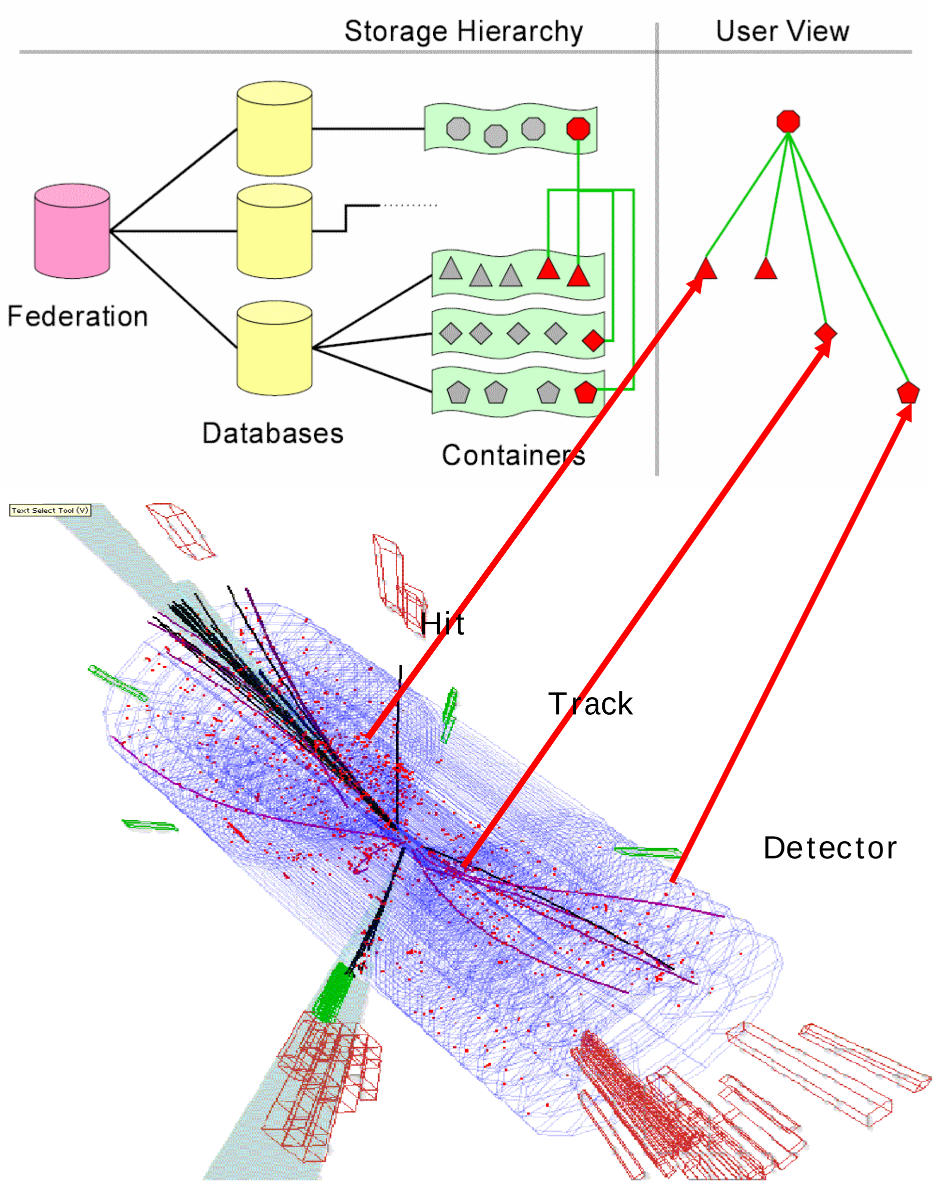
Mary Anne Scott's visit to Caltech
May 2002

E.Aslakson, J.Bunn, K.Holtman, S.Iqbal, I.Legrand, V.Litvin, H.Newman, E.Soedarmadji, S.Singh, C.Steenberg, R.Wilkinson

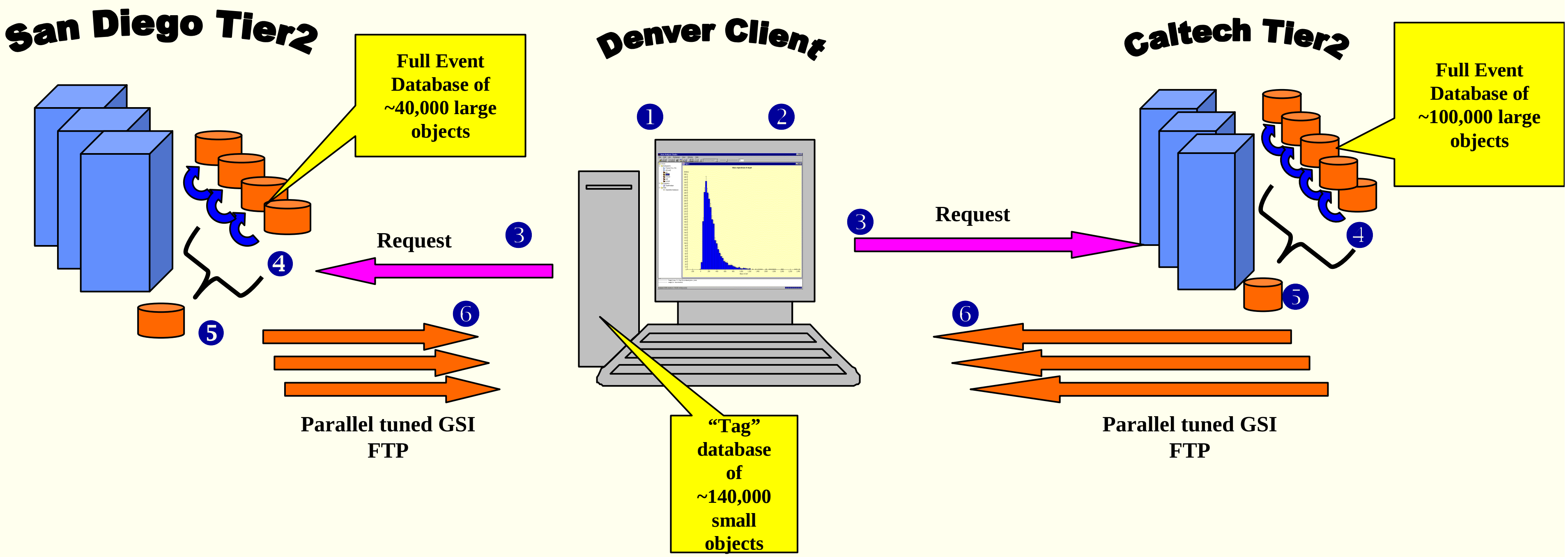
Overview

- A Historical Perspective (GIOD,Objects,Tags)
- Web Services: Why we think they are important
- Relational Databases: For AOD and Tags (at least)
- Grid Analysis: How does the Grid help Analysis?
- JETMET analysis ntuples: why did we pick them?
- CLARENS: Why Plug-in analysis components?
- ROOT: How does it fit?
- What is the Analysis Architecture?
- Who is doing what?

GIOD: Globally Interconnected Object Databases



- ➔ The GIOD Project broke new ground investigating distributed ODBMS for Physics
- ➔ We made a MultiTB OO Database Federation; used across LANs and WANs
- ➔ We exceeded a 100 MB/sec CMS Milestone (170MB/sec)
- ➔ We developed C++ & Java 3D OO Reconstruction, Analysis and Visualization Prototypes that Worked Seamlessly Over Worldwide Networks
- ➔ We deployed facilities and database federations as testbeds for Computing Model studies



Bandwidth Greedy Grid-enabled Object Collection Analysis for Particle Physics (SC2001 Demo)

Julian Bunn, Ian Fisk, Koen Holtman, Harvey Newman, James Patton

The object of this demo is to show grid-supported interactive physics analysis on a set of 144,000 physics events.

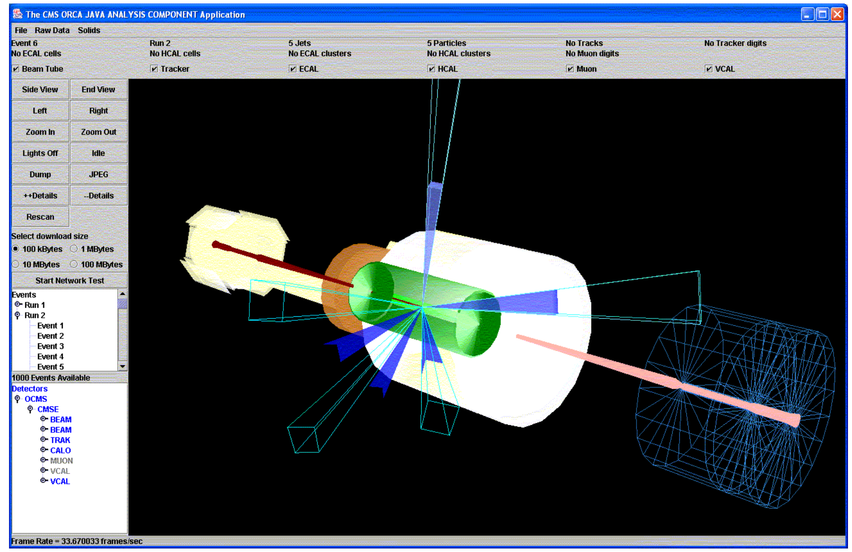
Initially we start out with 144,000 small Tag objects, one for each event, on the Denver client machine. We also have 144,000 LARGE objects, containing full event data, divided over the two tier2 servers.

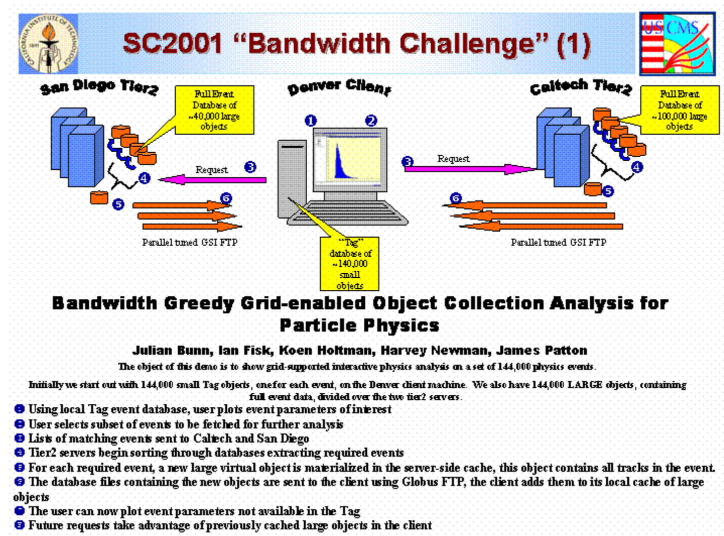
❶ Using local Tag event database, user plots event parameters of interest
❷ User selects subset of events to be fetched for further analysis
❸ Lists of matching events sent to Caltech and San Diego
❹ Tier2 servers begin sorting through databases extracting required events
❺ For each required event, a new large virtual object is materialized in the server-side cache, this object contains all tracks in the event.
❻ The database files containing the new objects are sent to the client using Globus FTP, the client adds them to its local cache of large objects
❼ The user can now plot event parameters not available in the Tag
❽ Future requests take advantage of previously cached large objects in the client

http://pcbunn.cacr.caltech.edu/Tier2/Tier2_Overall_JJB.htm

Simple Tags

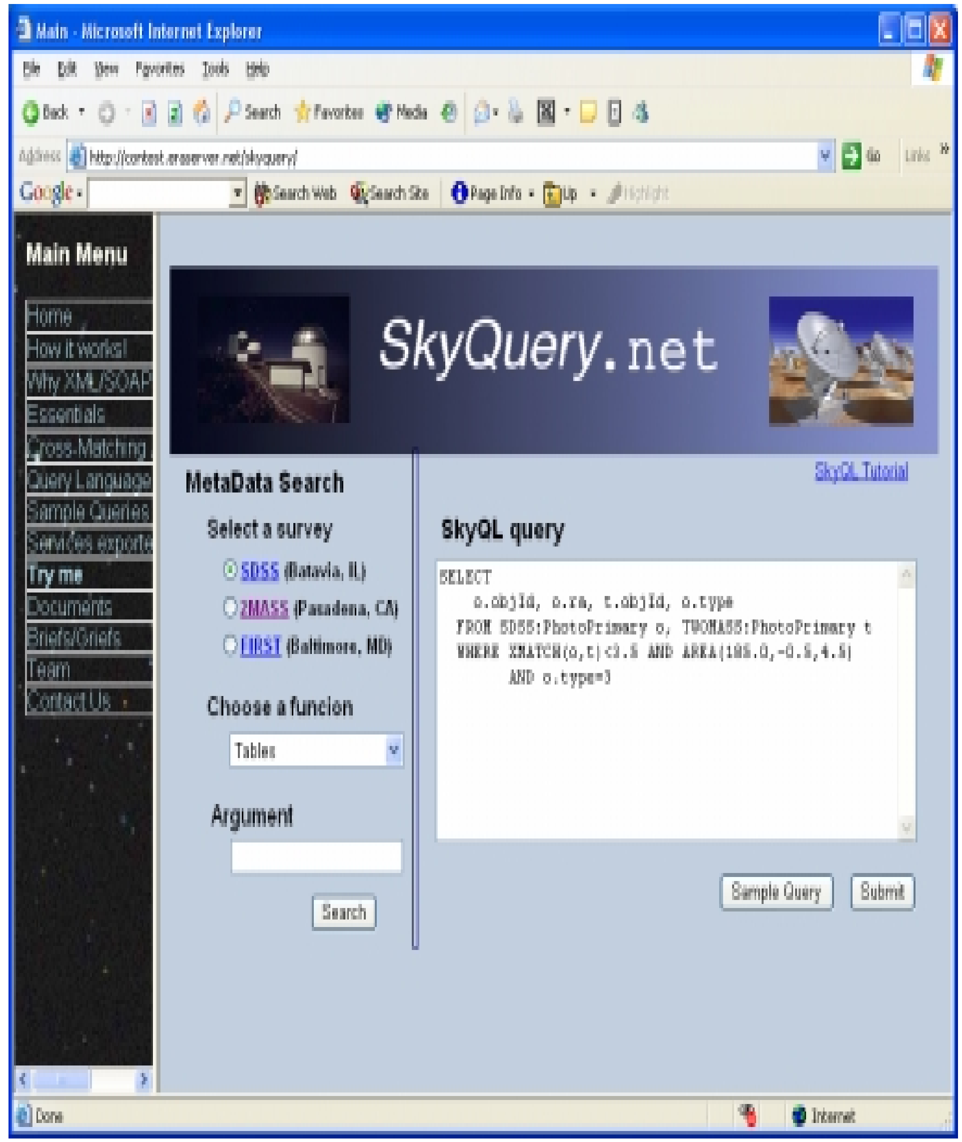
- We have ~180,000 Tag objects derived from di-jet ORCA events
- Each Tag: run & event number, OID of ORCA event object, then E,Phi,Theta,ID for 5 most energetic particles, and E,Phi,Theta for 5 most energetic jets
- These Tag events have been used in various performance tests and demonstrations, e.g. SC2000, SC2001, comparison of Objy vs RDBMS query speeds (GIOD), as source data for COJAC, etc.
- The COJAC Java3D client opens a remote Objy database containing the Tags, and allows the user to show the individual particles and jets, overlaid on the most recent CMS geometry (directly read from CMSIM rz file)





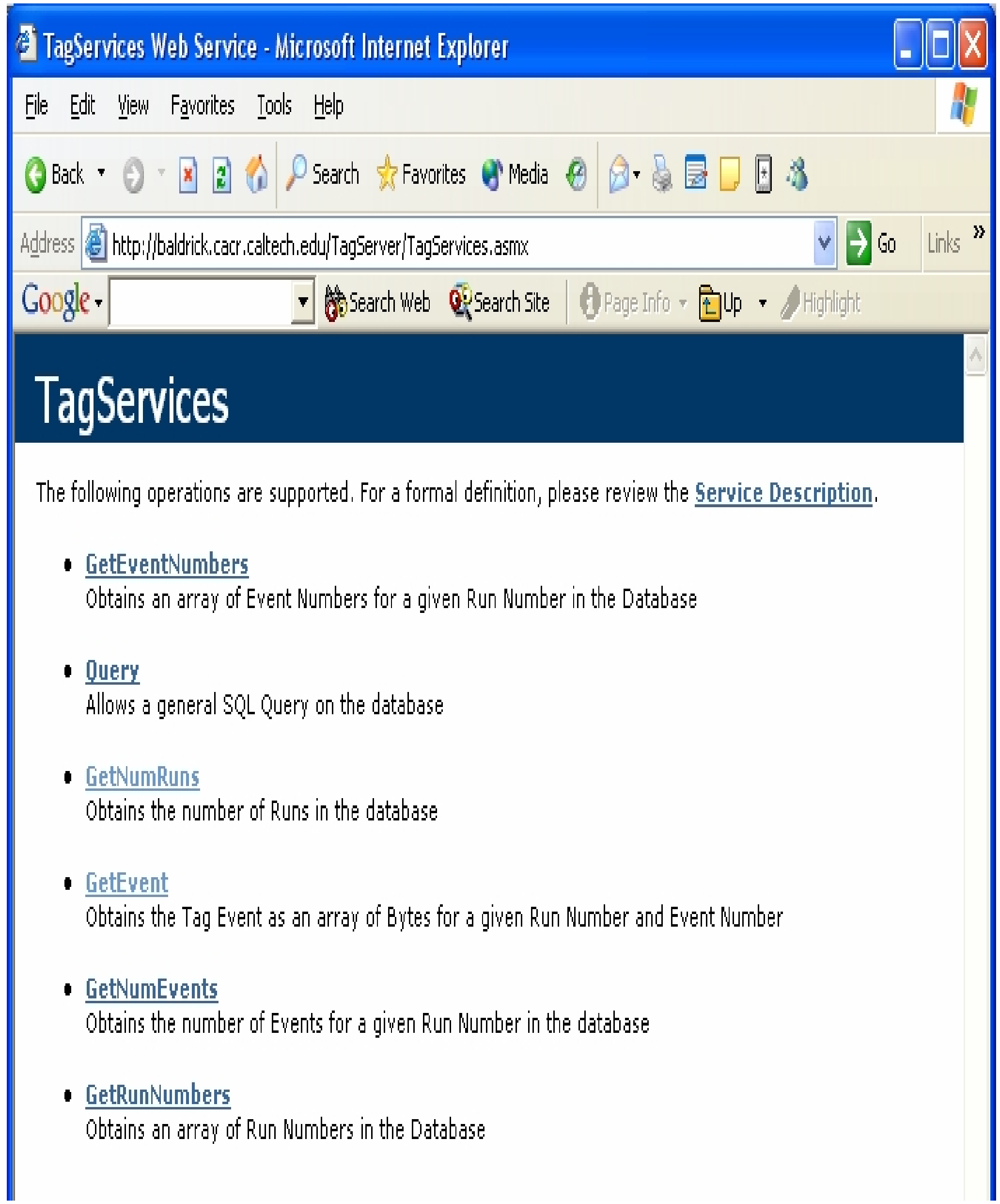
Web Services

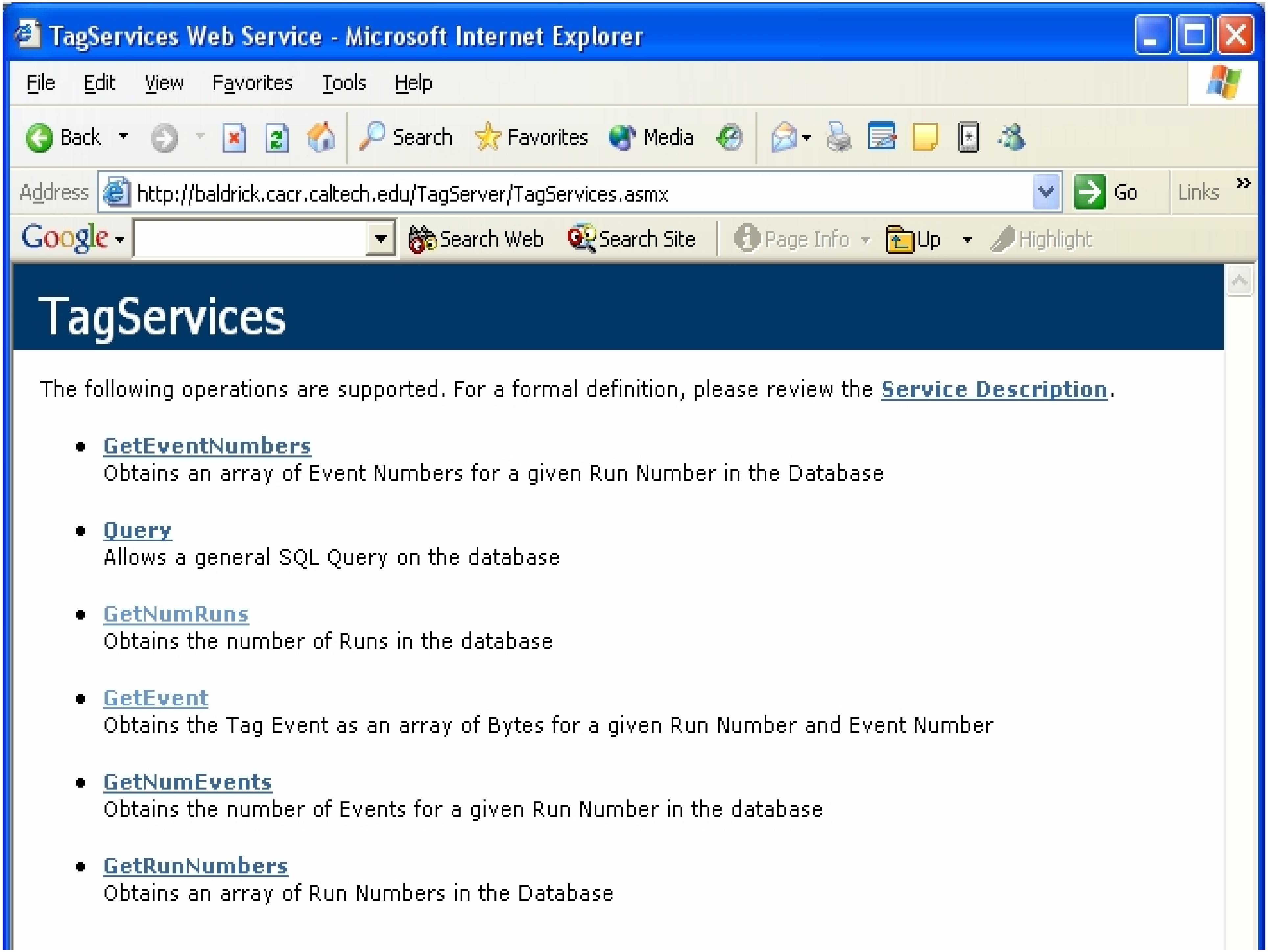
- Are Web Services just hype, or are they really useful to us?
- Interesting work at JHU/FNAL/Caltech with Web Service based access to astronomy surveys
- ~150MB datasets at FNAL, JHU and Caltech
- Can be individually or simultaneously queried via Web interface
- Simplicity of interface hides considerable server power (from stored procedures etc.)
- We investigated whether we could build Web Services to access our Tag data



Simple Tag Web Services

- Using .NET and SQLServer, we loaded all the Tags into a single DB table
- We have also loaded the Tags into Oracle @ CERN
- We then wrote some simple C# Web Service methods (could have used C++, but we already had Java code available which we could cut&paste!)
- Access to the resulting Web Services can be from a Browser or other program.
- We added the necessary code to COJAC so that it could fetch the Tag events via the Web Services, rather than via Objectivity





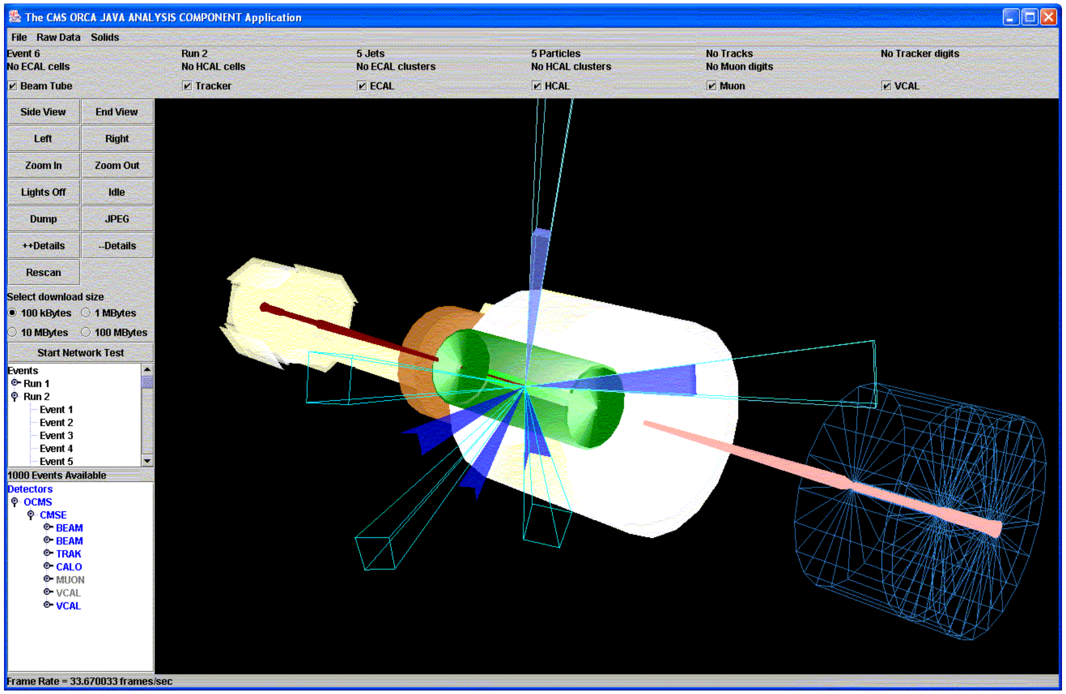
TagServices Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://baldrick.cacr.caltech.edu/TagServer/TagServices.asmx
Go Links
Google Search Web Search Site Page Info Up Highlight
TagServices
The following operations are supported. For a formal definition, please review the Service Description.
GetEventNumbers
Obtains an array of Event Numbers for a given Run Number in the Database
Query
Allows a general SQL Query on the database
GetNumRuns
Obtains the number of Runs in the database
GetEvent
Obtains the Tag Event as an array of Bytes for a given Run Number and Event Number
GetNumEvents
Obtains the number of Events for a given Run Number in the database
GetRunNumbers
Obtains an array of Run Numbers in the Database

COJAC – The CMS ORCA Java Analysis Component

Written in Java2, COJAC uses Java3D for high-speed rendering, rotating, zooming, panning and lighting of the CMS detector geometry.

Java Native Interface (JNI) methods are used to extract the very latest CMS geometry from any CMSIM rz file. The geometry can be displayed at various detail levels, and all individual detectors can be included or excluded from the display.

The Objectivity ODBMS Java binding is used to access a remote (or local) database of simulated particle collision events. Each event is rendered in Java3D, and can be overlayed on the CMS detector geometry.



The Jspheon-Jsoap SOAP library is used to fetch simulated events from a .NET Web Service, as an alternative to the ODBMS access.

A Network Test feature allows the user to continually fetch simulated event objects from the server until a preselected transfer size has been reached. This Test gives an idea of the speed at which Analysis tools such as COJAC can operate in a WAN environment.

Direct Access to ORCA Databases

- We would like to extract ORCA persistent objects from an ORCA database, without using ORCA itself
- Then we could do various interesting things:
  - ➔ Export a selection of events into an alternative persistency scheme
  - ➔ Extract individual reconstructed objects on an event by event basis (e.g. tracks and Tracker digis for a given event)
  - ➔ Provide some Web Services for the above
- We started by creating an Objy Active-schema based application that could pull out some selected objects associated with an event. This application used no ORCA code whatsoever.
- We then wrapped that application in a Web Service (we worked on both Apache and .NET)
- We plan to continue this work later, when it becomes more important for the JETMET work we are doing

(In that context we want a remote client to be able to re-run an ORCA reconstruction with e.g. a new algorithm, and then remotely access the newly created ORCA objects. Think example: apply new jetfinding algorithm on the fly)

JETMET Analysis

- We sought a group of end user CMS physicists who are working with analysis of large amounts of ORCA data.
- We want to see how existing analysis techniques (PAW,ROOT etc.) are going to evolve into Grid-enabled analysis, and what benefits are evident from this evolution
- Our discussions with the FNAL JETMET people (notably Pal Hidas and Shuichi Kunori) led to us selecting the JETMET hlt studies as an excellent test case.
- The JETMET analysis is interesting because:
  - ➔ It involves iterative access over a very large number of ntuple files and events (up to 500,000)
  - ➔ It is lengthy: Pal's analysis task takes several hours for example
    ↳ A significant analysis speed-up would thus be very beneficial
  - ➔ It uses PAW and KUMACs (although there are plans to move to ROOT-based analysis at the next production)
  - ➔ The existing body of ntuples that are used are generated by ORCA, and exist in large numbers both at FNAL and at CERN (and now at Caltech)

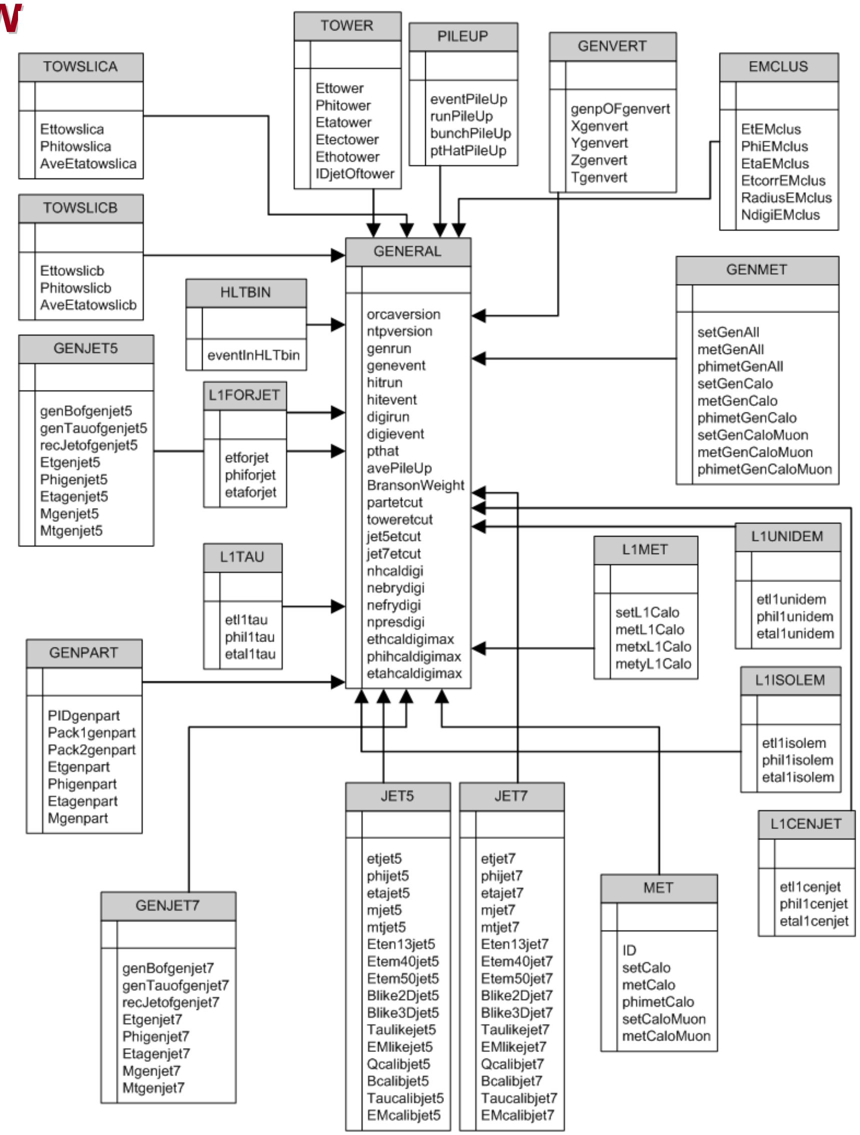
JETMET Analysis (II)

- We started by taking a significant subset of the low luminosity ntuples that were generated in the 2001 Fall Production. These were copied to RAID arrays on the Caltech Tier2.
- With Pal's assistance, we took a complete analysis chain (basically KUMAC scripts and Fortran selection routines used with PAW) and installed it on the Caltech Tier2, modifying it where appropriate so that it worked correctly with the Caltech ntuple subset
- We then made detailed timing measurements for the analysis, and compared the analysis results with those obtained at FNAL by Pal (to ensure consistency).
- We looked then in detail at the ntuple format (v2.06):

http://computing.fnal.gov/cms/jpg/ntuple_maker/Ntuple_content_206.html

JETMET Analysis (III)

- From the ntuple description, we drew an ER diagram that represents the content. This can be thought of as an AOD description for the JETMET analysis
- It can readily be seen that this is a sophisticated schema with a lot of detail
- At this point we would like to
  - ➔ Automatically generate the ER diagram from the ntuple itself
  - ➔ Automatically generate a set of SQL CREATE TABLE commands to create suitable RDBMS tables
  - ➔ Generate SQL INSERT or bulk load scripts from the ntuple that will enable population of the tables created in 2.
  - ➔ Generate a “Tag” table of data that captures the most often used columns in the ntuple



JETMET Analysis (IV)

- The ntuple filenames themselves contain useful/important metadata, e.g.
  - ➔ jm_hlt230300_1.ntup
  - ➔ jm_bigmb123_run161000.ntup
- Which needs to be stored in a metadata catalogue table, with appropriate relationships to the event tables
- We investigated CERN IT Division's ROOT file to RDBMS converter application.
- We have imported a subset of the ntuples into SQLServer at Caltech
- We will shortly import to Oracle 9i at CERN.
- Eventually, we want to import ntuples into a database at FNAL, so that it can be used directly by the JETMET team.
- Then we will be in a position to offer a distributed JETMET ntuple analysis "Service".

JETMET Analysis (V)

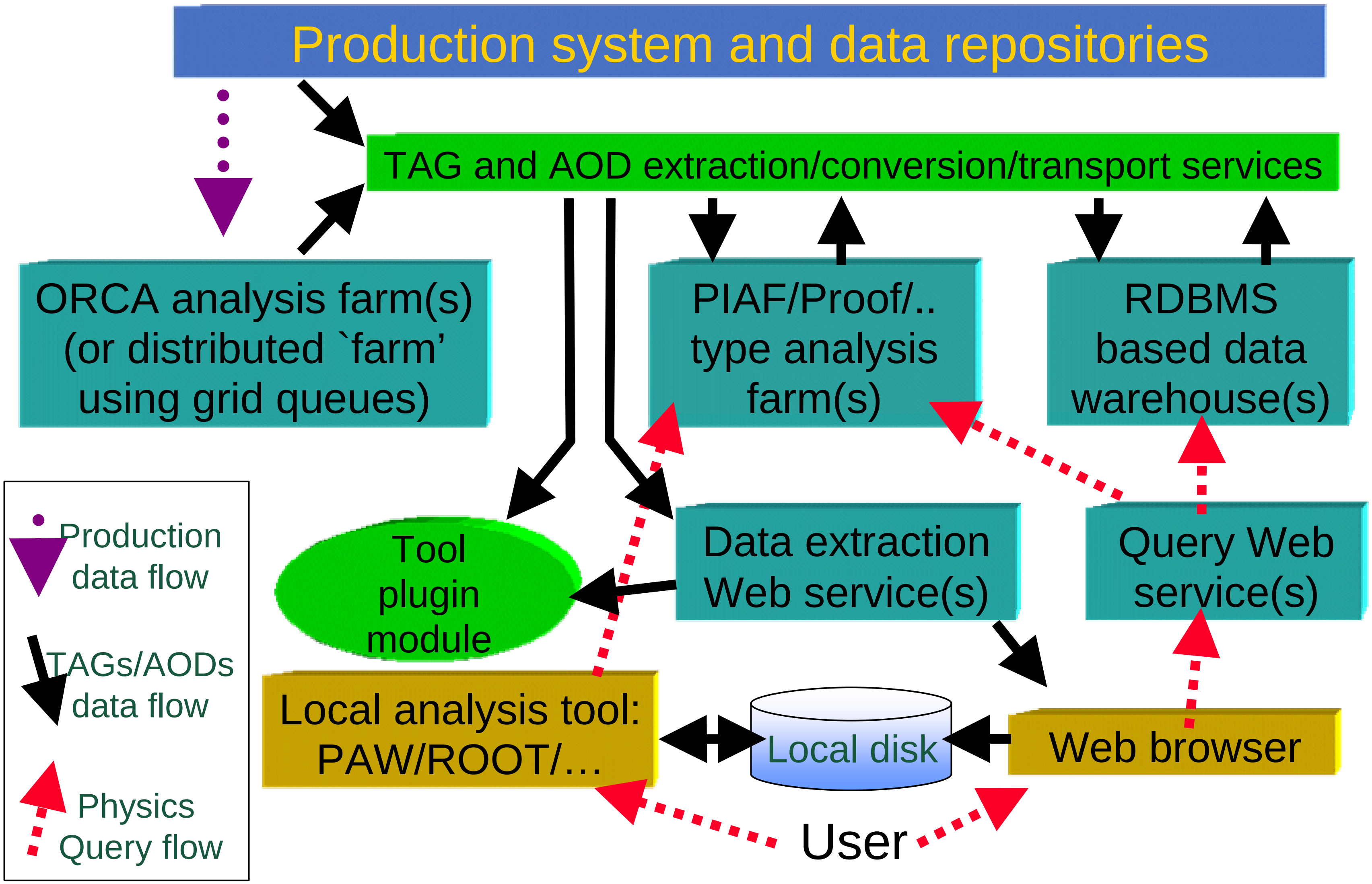
Using the RDBMS-based JETMET data, we expect to be able to re-work the PAW-based analysis:

- Selections of the ntuple files based on file name will be replaced by a query on the metadata table(s), providing direct access to the required AOD events in the RDBMS
- Cuts on the data will be replaced by queries and suitable SQL selects
- Interpreted Fortran COMIS functions will be replaced by stored procedures on the RDBMS server
- We expect a significant speed-up due primarily to
  - → Only having to access fractions of the data for each event
  - → Stored procedures
  - → Optimised query execution plans generated by the RDBMS

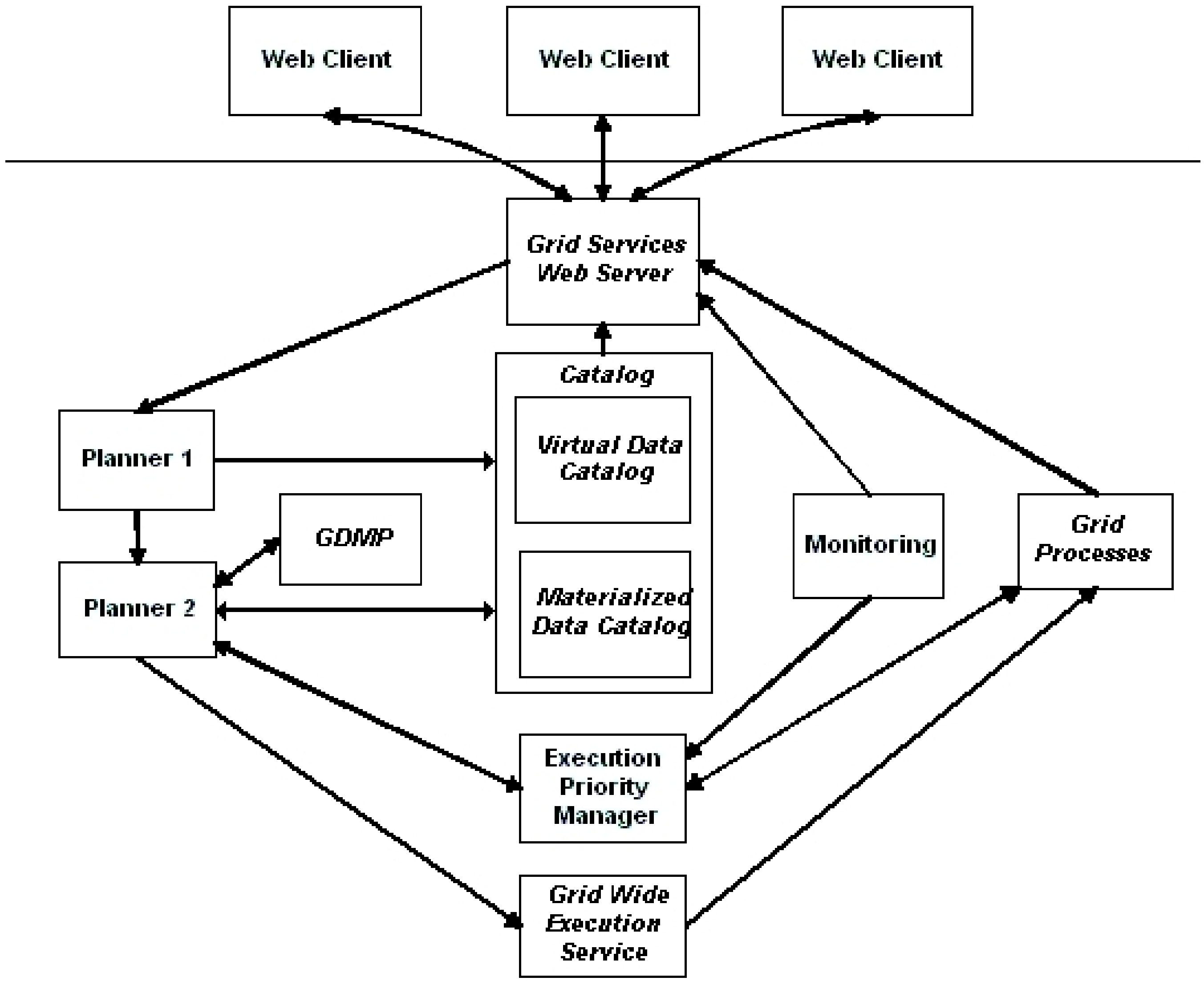
JETMET Analysis (VI)

- In a separate activity, we are investigating the replacement of PAW by ROOT as an analysis tool for the JETMET ntuples
- We have developed a ROOT analysis that mimics the existing PAW JETMET analysis.
- Instead of using ROOT to directly access the ntuples (which is possible and supported) we prefer a solution that involves accessing the JETMET RDBMS
- This means accessing the RDBMS from ROOT.
- To do this we are developing a CLARENS environment that incorporates Grid authentication and which allows a ROOT client to access the JETMET data.
- A target capability is a CLARENS module that can import remote ntuples, and this has been completed.

Analysis Architecture Candidate



Analysis Architecture Candidate (CAIGEE)



Who is Doing What?

- Strategic Oversight and Direction; Harvey Newman
- Analysis Architecture and Grid integration: Koen Holtman
- Ntuple → RDBMS Conversion (JETMET and General ntuple). Eric Aslakson
- Reproduction of PAW-based ntuple analysis on Tier2, timing measurements, identification of possible optimization. Edwin Soedarmadji
- ROOT version of PAW-based ntuple analysis, reproduction of results, timing measurements, data access via CLARENS server. Conrad Steenberg
- RDBMS population with ntuple data. Julian Bunn (SQLServer), Saima Iqbal (Oracle 9i), Eric Aslakson (Tools), Edwin Soedarmadji (Optimisations/stored procedures)