

Status of the D1 (Event) and D2 (Spacecraft Data) Database Prototypes for DC1

Robert Schaefer – Software Lead, GSSC

- **Database Definitions review**
- **EGRET Databases**
- **DC1 plans**
- **D1/D2 Design Decisions**
- **D1 and access Prototype Architecture**
- **D2**
- **D1E & D2E**
- **Future**

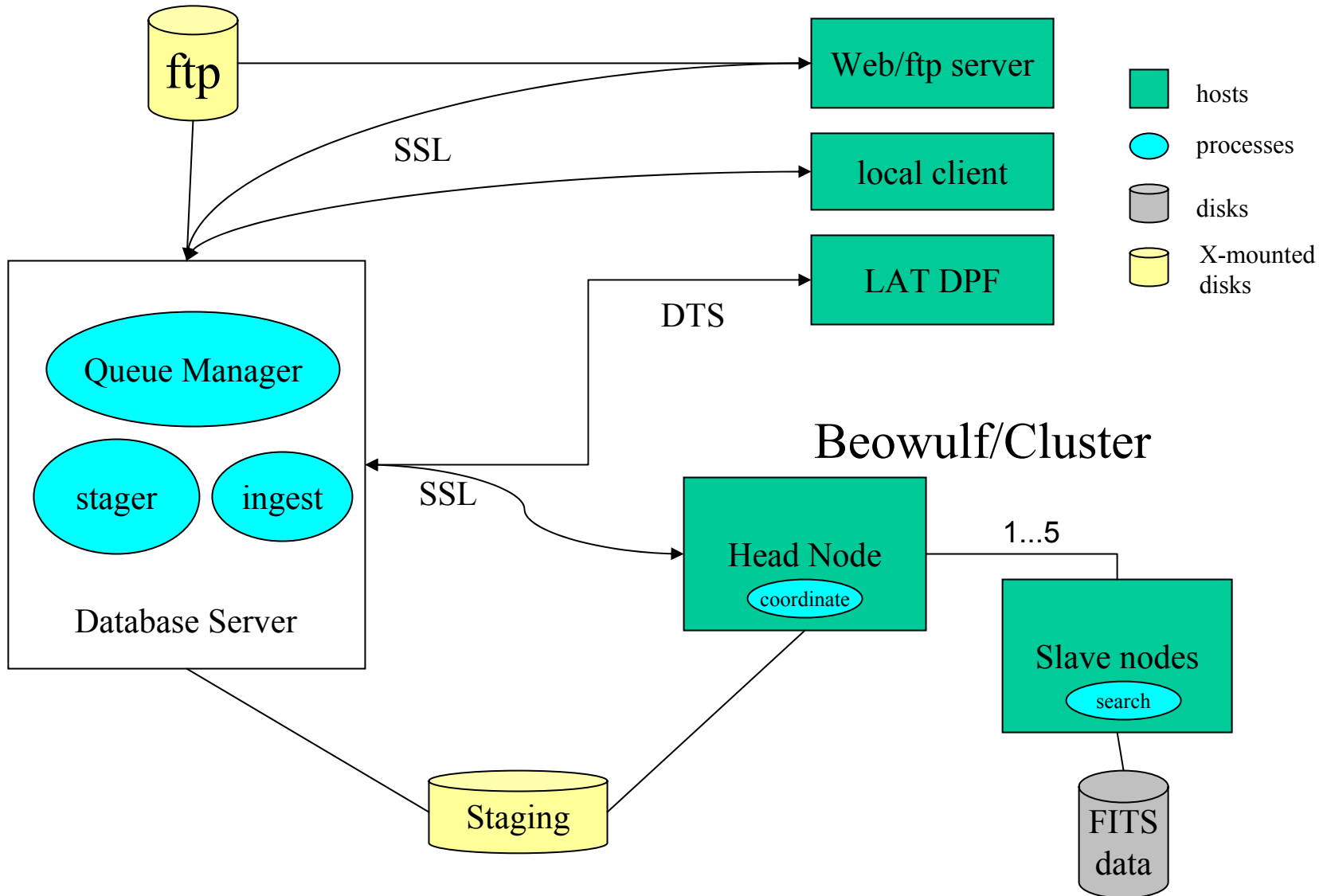
- **D1 - Event Database:** LAT Event Data queryable by region of the sky, energy, time, Earth azimuth, etc. A query results in an FT1 http://glast.gsfc.nasa.gov/ssc/dev/fits_def/definitionFT1.html) file which contains all data matching the selection.
 - **Actually two databases - a separate photon database (called photon database) and one containing all events (called event database).**
- **D2 - Spacecraft Database:** LAT pointing, livetime, and mode history data queryable by (most commonly) time range, spacecraft mode, and other spacecraft pointing parameters and returns the data in an FT2 file (http://glast.gsfc.nasa.gov/ssc/dev/fits_def/definitionFT2.html)

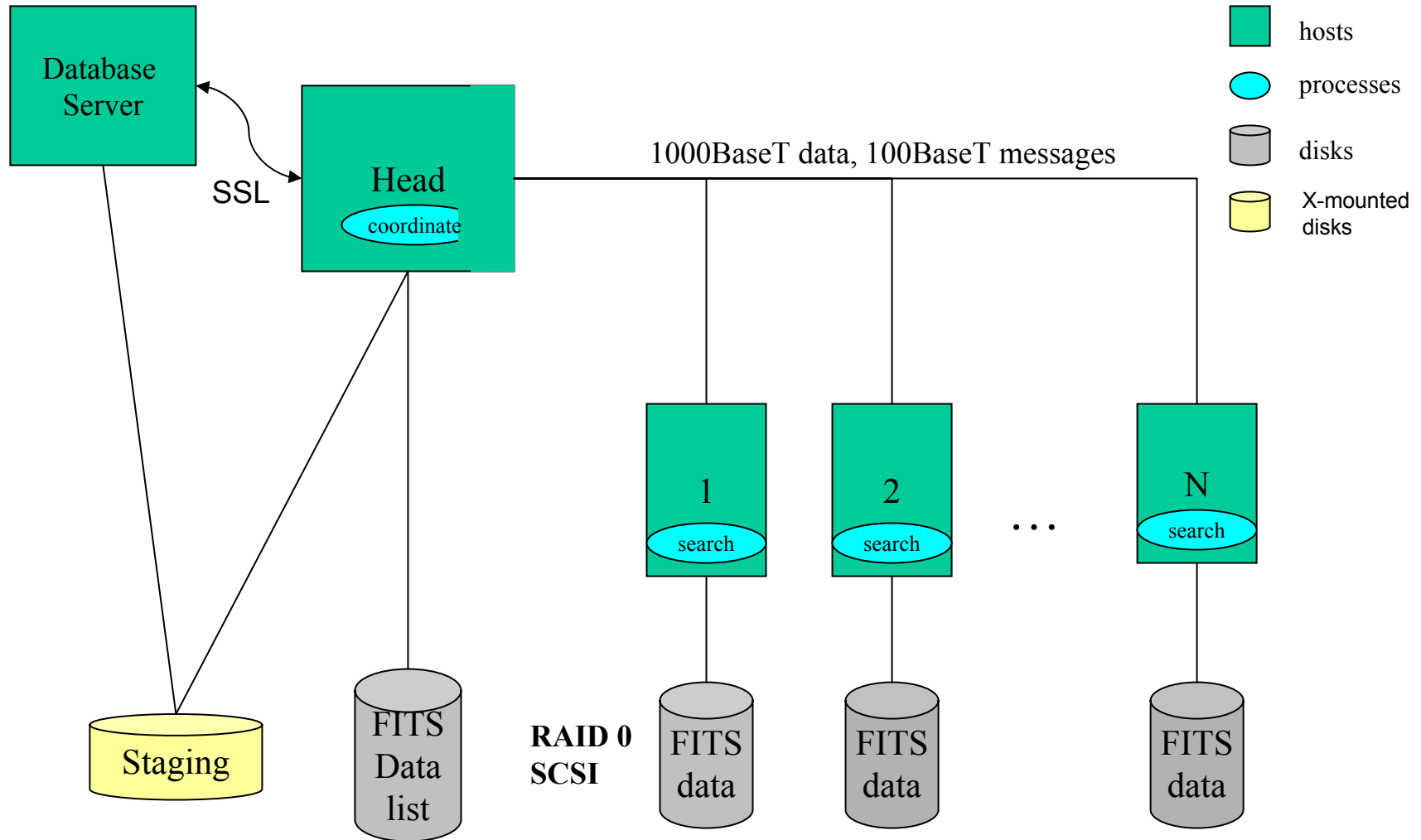
- Need to create “extractor” utilities, i.e., tools to send the query to the database and retrieve the data from it.
- There is also a “User level data extraction utility”. This is for refining data queries locally. This tool also has added benefits:
 - load is moved off of the database server.
 - Home version of D1/U1!
- Nomenclature:
 - U1 Event Data extractor
 - U2 User level data extraction utility
 - U3 Pointing/livetime history extractor

- **EGRET databases for analysis tools:**
 - **D1E EGRET photon database**
 - **D2E EGRET pointing history database**
- **Not necessary to force EGRET data strictly into FT1 and FT2 formats to add the analysis capability - Easier for implementing EGRET analysis.**

- Working prototypes of Databases (D1, D1E, D2, and D2E) for DC1
- User friendly extraction capabilities (U1, U1E, and U3, U3E)
- Possibly a stripped down U2 sub-selector tool which can also handle EGRET data.
- For DC1, a user will be able to log into the database web page, make a query, and ftp the data as FT1 and FT2 files to a local disk for analysis. All of these tasks will be done with the prototype architecture that is being discussed here.

- **D1: “Database and Related Utilities working group” settled on using a beowulf cluster to do parallel searches of time-ordered data in FITS files (using the CFITSIO library).**
- **D2: Since we are already writing all the software to search FITS files for D1, it was sensible to make D2 do the same. However, since we expect the vast majority of queries to be by time range only, we do not need a beowulf for this database.**





Note: only single disks (no RAID) for DC1 prototype

- **Development beowulf for prototype purchased (now in boxes at SSC)**
 - **5 slave + 1 head nodes (dual processor AMD, 2 ethernet cards, 73 Gb 15k rpm SCSI disk)**
 - **All nodes rack mounted, with kvm switch.**
- **Beowulf perl-based search code prototype presented at June 2002 ST workshop converted to C, and improved to use better communication and process handling. Tested on beowulf at UNM with PIII / 1 GHz processors.**
- **Next, get that code running on GSFC beowulf and then on SLAC beowulf.**

- In a RDBMS system, QM would be part of the DBMS server.
- Queue Manager is a large event loop which:
 - Tracks state of all submitted queries
 - Validates queries
 - Assigns priority to query (based on size and submission type a batch queue will be enabled)
 - Sends queries to database by priority and time in.
 - Checks for timeouts
 - Handles communication with query clients, database, and stager.
 - Logs all requests
- Beowulf + Queue Manager = “D1”

- **Reformats output of Database into proper FITS files.**
- **Moves data to ftp accessible area.**
- **Works under direction of queue manager.**
- **We would need a stager even if we were using an RDBMS**

- **Design is just like D1, only there is no beowulf.**
- **Communication and searching is done on a single node which mounts the data disk.**

- D1E can run within D1 on beowulf. Choosing EGRET data was already available with a command line option on the proto-prototype beowulf presented at last June's ST workshop.
- D2E can also easily run within framework of D2.
- Note: parameters for EGRET are not all the same as for GLAST, so the stagers must be different.
- **Fun with MySQL.** Since the amount of data in EGRET's pointing data is so small (40 Mb - see <http://glast.gsfc.nasa.gov/ssc/dev/egret2glast/d2e.html>) we loaded it into a MySQL database. D2E has been implemented this way (web page front end, D2E stager, MySQL table), but there are still some bugs in it.

- **For DC1,**
 - **U1 and U3 are the web pages that allow database querying.**
 - **Also will have the local client to send queries directly to the Queue Manager. (This tool could be an exclusive LAT team tool for use with the SLAC Bewoulf.)**
- **Later, U1 and U3 will expand to include local tools that will**
 - **allow a user to construct the query fill in the Web page form**
 - **read the resulting web pages and**
 - **ftp the resulting data back to the person's local host.**

- **Beowulf prototype search code written - needs testing**
- **Queue Manager nearly out of design phase - coding about to begin**
- **Stager, Ingest programs not yet written but should be easy to do.**
- **D2 search engine (without beowulf) needs to be done - but should also be straightforward.**
- **Test D1 with 389 days of photon event data expanded to FT1 size.**
- **Implementation of “E” versions easy**
- **D2E as MySQL almost ready for querying**
- **Use Cases being updated.**

- Queue Manager is complicated will take a couple of months to have it working.
- Beowulf needs to be setup and have software installed and debugged.
- Stager, Ingest programs need to be written.
- Issues for future
 - Should single QM handle D1 and D2?
 - Just what parameters should be query-able via web page?
 - Get performance benchmarks with newer hardware
 - How big is D2 (how often do we need position information updates?)
 - Use of pixelization of data for D1 (to improve search speed).

- **The main science product database will be a Beowulf cluster of machines which will do parallel spatial searching of data stored in FITS files. Why not use a standard DBMS?**
 - **Data access will be simple read-only queries; we do not need many of the features of a DBMS (database management system).**
 - **DBMS power comes with an overhead. We benchmarked search speeds of FITS files using CFITSIO and three DBMS systems - searching FITS files was fastest. (same results found by ASTROGRID project in the UK - <http://wiki.astrogrid.org/bin/view/Astrogrid/DbmsEvaluations>)**
 - **Database will be stored in FITS format - desired by HEASARC, and flexible - can easily accommodate modifications of data content.**
 - **Reprocessing will replacing FITS files rather than finding and deleting old photons and inserting new photons.**
 - **Downside is that we have to write search and interface software, but we are well on the way.**