Database Components

Tom Stephens
GSFC

November 2003
Data Flow
Server Design

- Operates as TCP/IP server for connections from Queue Manager and Ingest program.
- Parallel program using MPICH as message passing interface.
- Scalable to any number of processes, but can run on a single computer.
- Written in C
  - ease of implementation and interface with cfitsio and MPICH libraries
  - Maximum performance
Program Flow I

Start

System wide initializations

Control node code

Search node code

Message traffic

End
D1/D2 Internal Process Structure

Control Process

Query Request
Search Results

Selected Data

Search Process
Photon Data

Query Parameters
Files to Search

Search Process
Photon Data

Search Process
Photon Data

Search Process
Photon Data

Search Process
Photon Data

Search Process
Photon Data

Search Process
Photon Data
Program Flow II – Search Node

Start

Wait for message from control node

Send list of output files to control node

Apply search criteria to data in each file and write results to new output file

Get list of files to search or ingest

Ingest message?

yes

Shutdown message?

yes

no

no

End

Copy new files from ingest area into data area
Program Flow III – Control Node

Start

Read in metadata, open the log file and initialize listen port for communications

Wait for message to appear on an open TCP/IP socket.

Process all the received messages

Shutdown message? yes

End

no
Program Flow IV – Conducting a Search

Start

Receive NEW_QUERY message from QM

Parse query parameters and check validity

Is this a good query?

no

Send BAD_QUERY message to QM

yes

Send QUERY_STARTED message to QM

Send QUERY_RECEIVED message to QM

Send QUERY_FINISHED message to QM with list of output files

Were there any problems?

no

yes

Send QUERY_FAILED message to QM

End

Build FITS row filtering query string

Build list of data files to search for each search node

Send query string and file list to each search node

Receive list of output files from each search node

Receive NEWQUERY message from QM

Parse query parameters and check validity

Is this a good query?

no

yes

Send QUERY_STARTED message to QM

Send QUERY_RECEIVED message to QM

Send QUERY_FAILED message to QM

Were there any problems?

no

yes

Send QUERY_FAILED message to QM
Stager Design

- Operates as TCP/IP server for connections from Queue Manager and Ingest program.
- Must merge and sort data from search nodes and return an FT1/FT2 file to the user.
- Overall structure similar to control node of server but without parallel processing.
Program Flow – Merging Query Data

Start

Receive NEW_MERGE message from QM

Read list of files to merge into final data file

Data read successfully?

Send MERGE_NOT_RECEIVED message to QM

Merge and sort data from files with same times

Sort input file list by data start times

Send MERGE_RECEIVED message to QM

Merge and sort all data into single file

Remove duplicate entries in GTI extension.

Update the appropriate header keywords

Were there any problems?

Send MERGE_FAILED message to QM

Send MERGE_FINISHED message to QM with FTP location of output

Remove all intermediate files from staging disk

End
D1 Ingest Design

- Operates as TCP/IP client to Server and Stager
- Prepares data output from LAT Level 1 processing pipeline for ingest into databases.
- Breaks sky into regions using Hierarchical Triangular Mesh (HTM) indexing.
- Will probably be rewritten in C++ using Goodi once final decision on HTM is made.
Program Flow – Ingest New Data

Start

1. Initalize system and read in Metadata
2. Wait for NEW_DATA_ADDED message
3. Open input file and read header information
4. Is the data new?
   - yes: Send message to D1 Server with list of new files
   - no: Send error message
5. Split data in subfiles by HTM pixelization
6. Move files from temporary ingest area to archive
7. Does data need to be merged with previous data?
   - yes: Merge the appropriate files into single file
   - no: Update Metadata
8. Send message to Server with new and old files
9. Update Metadata

End
Sample Configuration File

# Sample configuration file
# These are the settings used on the hygd cluster.
# Any line that begins with a '#' character is a comment and is ignored.
# Edit the second column to reflect your specific site configuration.
# The order of the keywords does not matter but spelling does.

HOME_DIR /data/nodes/GLAST/D1
FINAL_DIR /data/head/GLAST/repository/D1
FTP_HOST heasarcdev.gsfc.nasa.gov
URL_PREFIX ftp://legacy.gsfc.nasa.gov/glast
FTP_DIR /FTP/glast #ftp directory on heasarcdev
INGEST_DIR /data/head/GLAST/ingest/D1
NEW_DATA_DIR /data/head/GLAST/new/D1
LOG_DIR /data/head/GLAST/logs
ARCHIVE_DIR /data/head/GLAST/archive/D1
HEAD_DATA_DIR /data/head/GLAST/D1
SERVER_PORT 45278
STAGER_PORT 45280
Code Layout

Databases

D1

v1r2

cmt
config
doc
include
src

D1Server.c
D1ServerFunctions.c
D1Stager.c
D1StagerFunctions.c
D1Ingest.c
D1IngestFunctions.c
generalFunctions.c

D2

v1

cmt
doc
include
src
To Do

- Write ingest for D2
- Add code to ingest reprocessed data
- Refine ingest process for D1
- Benchmark D1 both with and without HTM pixelization