LDAS From The Outside In: Database Access Tools

Peter Shawhan

LIGO/Caltech

NSF Review

May 9-11, 2000

LDAS From a User's Perspective

LDAS provides an analysis framework with certain capabilities:

- Archive of raw data in frame format
- Data conditioning (e.g. down-sampling, line removal, regression)
- Batch system for parallel processing
- Metadata database

LDAS as a self-contained system is geared to the demands of "production" analysis tasks

The components of LDAS also should provide support for "external" analyses and the interpretation of analysis output

LDAS specifies only the basic elements of user interfaces: communication protocols, file formats

Additional "external" tools are needed to facilitate fast-turnaround exploration, visualization, and statistical analysis

LIGO Datasets and Access Modes

Dataset	Data Volume	Typical Access Modes
Frame data	~100 TB/year (GW strain data alone is 3 TB/year)	• Program retrieves a few channels for a long time interval • Program retrieves many channels for a list of many short time intervals • User retrieves a few channels for a short time interval
Trend data	~1 TB/year	• User or program retrieves a few channels for a long time interval
Metadata	~50 GB/year	• User or program submits a query which returns a modest amount of information

Planned Usage of the LIGO “Metadata” Database

Database table definitions have been established to store various types of information:

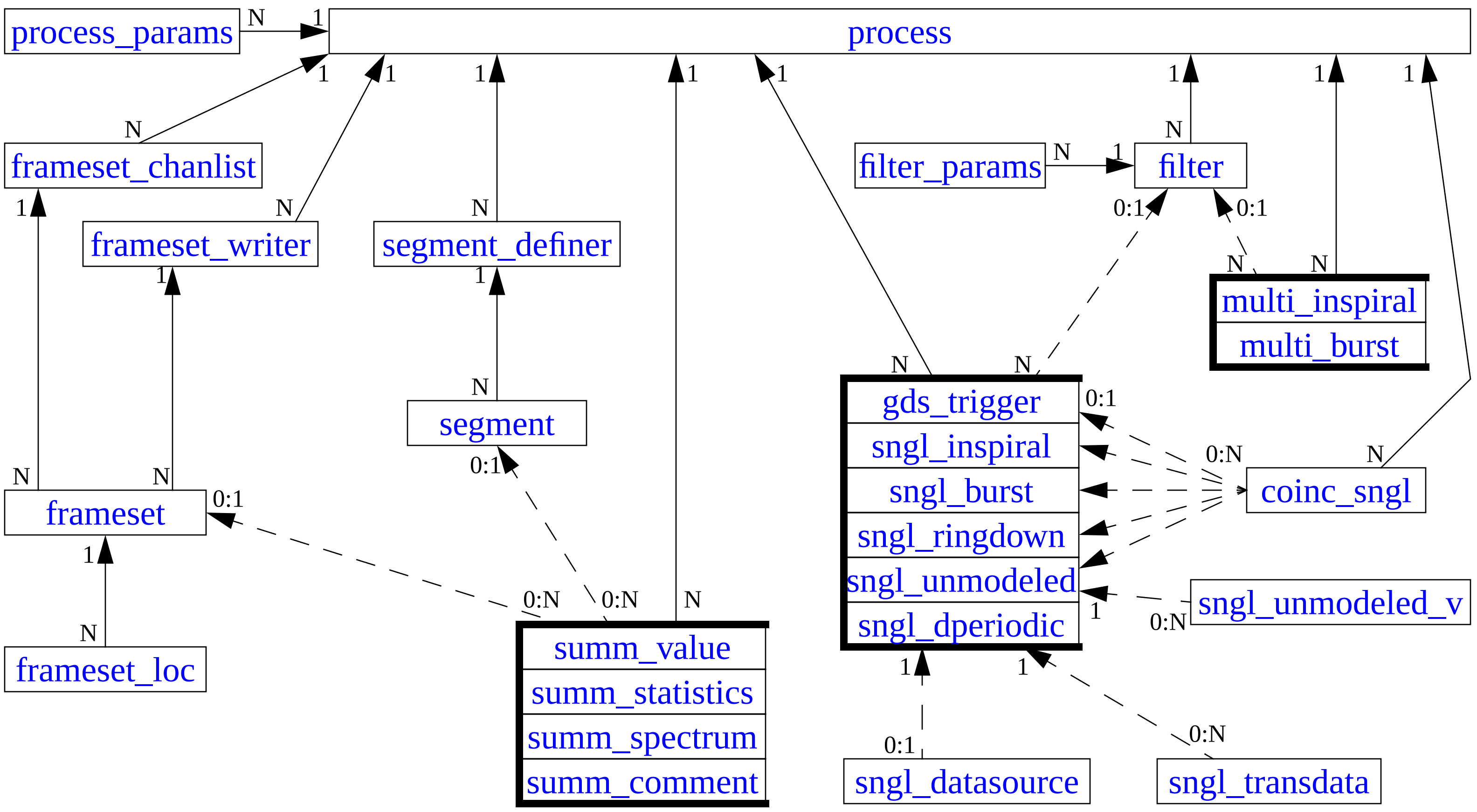
- Metadata about raw data (index of data files, detector state)
- Summary information for appropriate time intervals (named scalar values, statistical measures, spectra, comments)
- Diagnostic “triggers” (e.g. transients in environmental channels)
- Astrophysical event candidates of various types (inspiral, burst, ringdown, unmodeled)

Draft document with detailed definitions (LIGO-T990101-02) has been circulated within LIGO/LSC analysis groups

The existing table definitions are thought to fulfill most data storage needs, but more tables can be added as necessary

The LSC has the authority to determine the scope of the database

Database Tables and Relationships



process_params

process

frameset_chanlist

frameset_writer

segment_definer

filter_params

filter

multi_inspiral

multi_burst

segment

gds_trigger

sngl_inspiral

sngl_burst

sngl_ringdown

sngl_unmodeled

sngl_dperiodic

coinc_sngl

sngl_unmodeled_v

frameset

frameset_loc

summ_value

summ_statistics

summ_spectrum

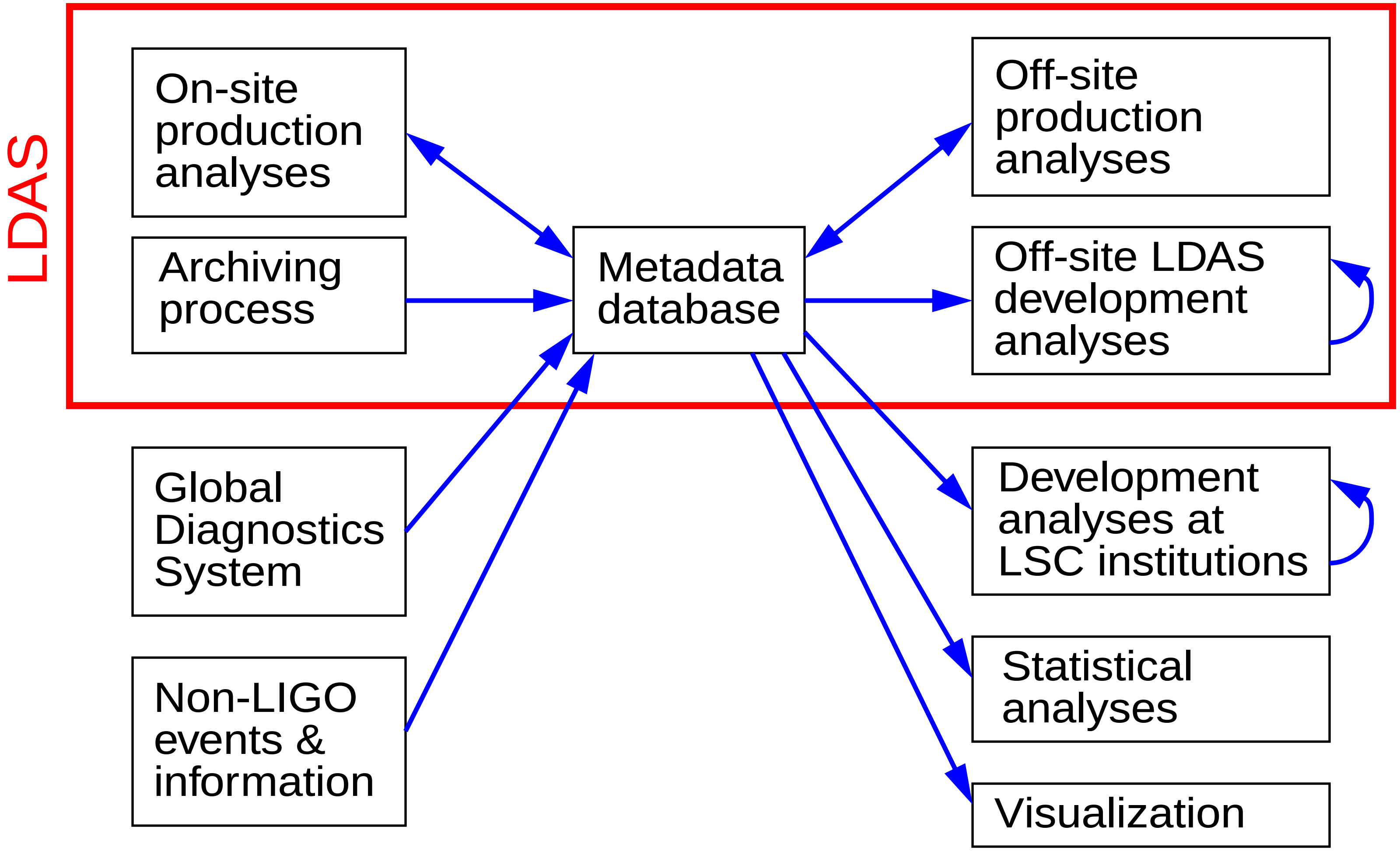
summ_comment

sngl_datasource

sngl_transdata

LIGO

Metadata Flow Diagram

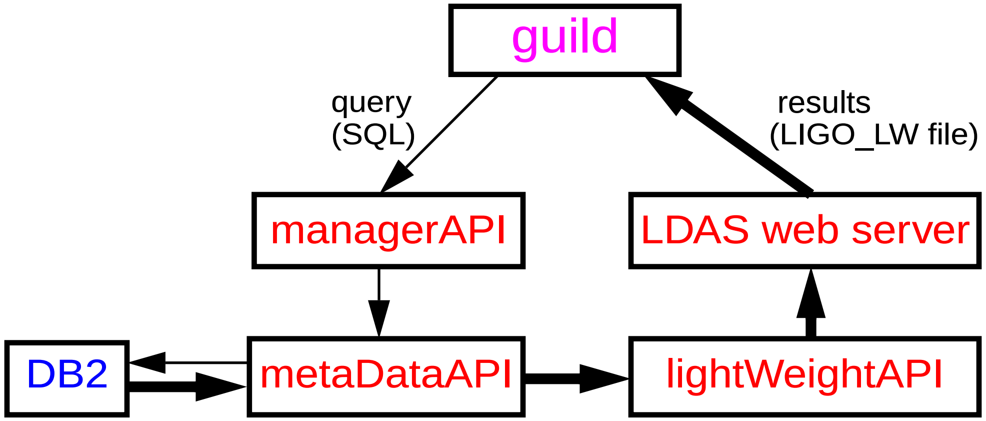

LDAS
On-site production analyses
Archiving process
Metadata database
Off-site production analyses
Off-site LDAS development analyses
Global Diagnostics System
Non-LIGO events & information
Development analyses at LSC institutions
Statistical analyses
Visualization

Graphical User Interface to LIGO Database — “guild”

Written as a standalone Tcl/Tk script, for use anywhere

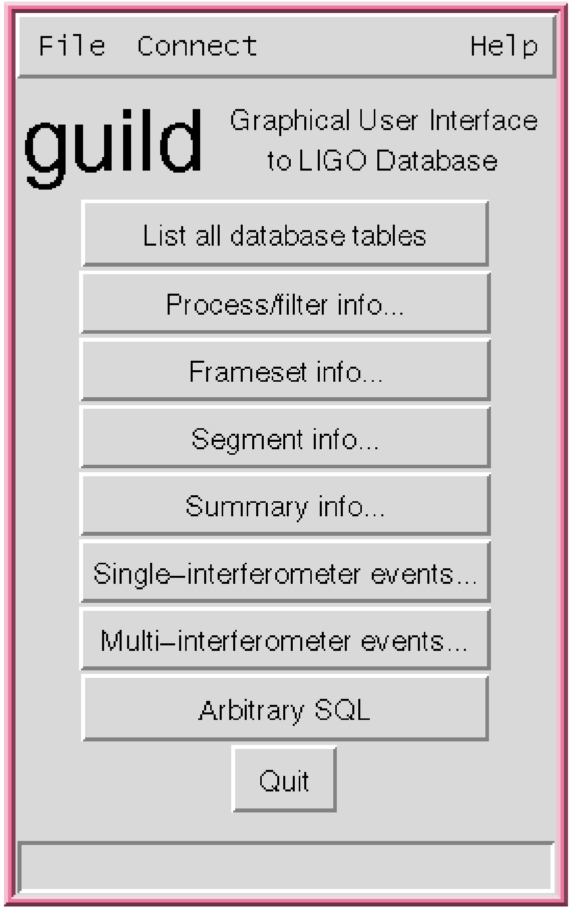
Provides a point-and-click method to build database queries (in DB2’s native SQL language), with various optional qualifiers

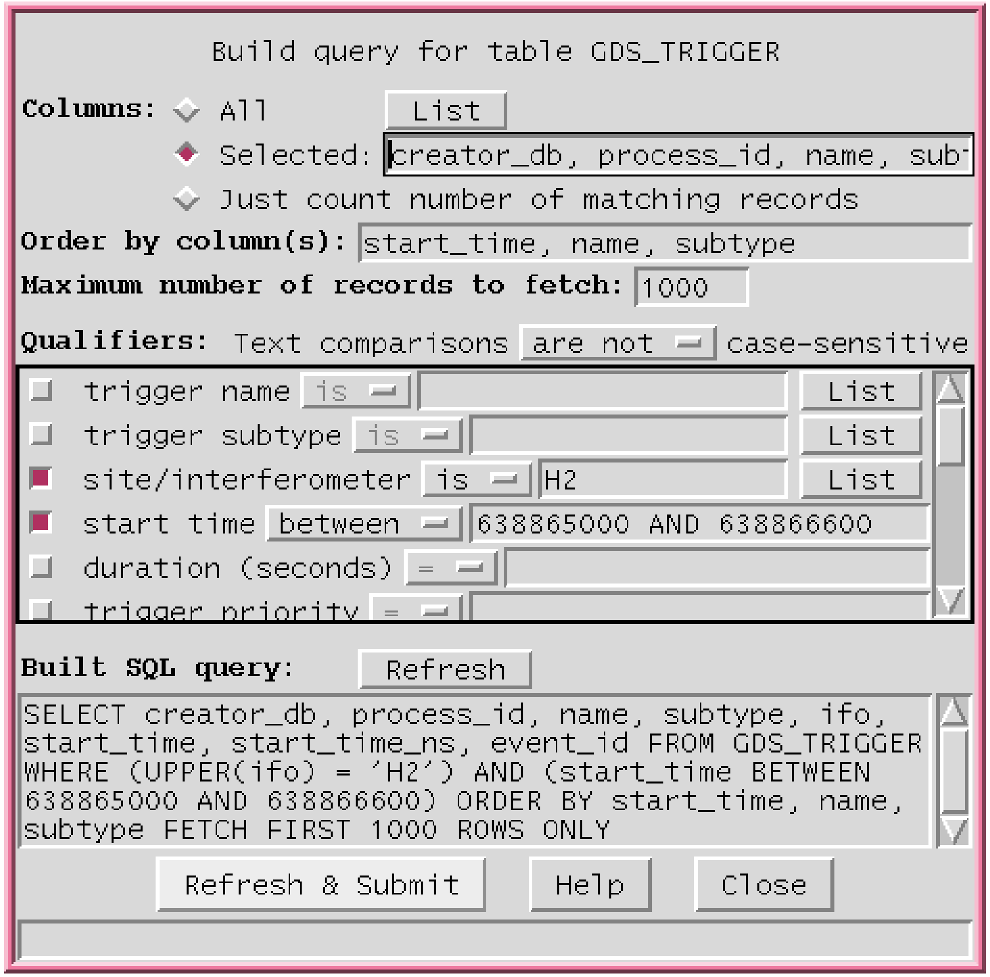
Sends query to the LDAS managerAPI, retrieves file of matching database records via http, and displays results as a scrollable table



Knows about the relationships between tables, and provides cross-reference links

guild Query-Building Screens





Build query for table GDS_TRIGGER

Columns: ○ All
● Selected: creator_db, process_id, name, sub
○ Just count number of matching records

Order by column(s): start_time, name, subtype

Maximum number of records to fetch: 1000

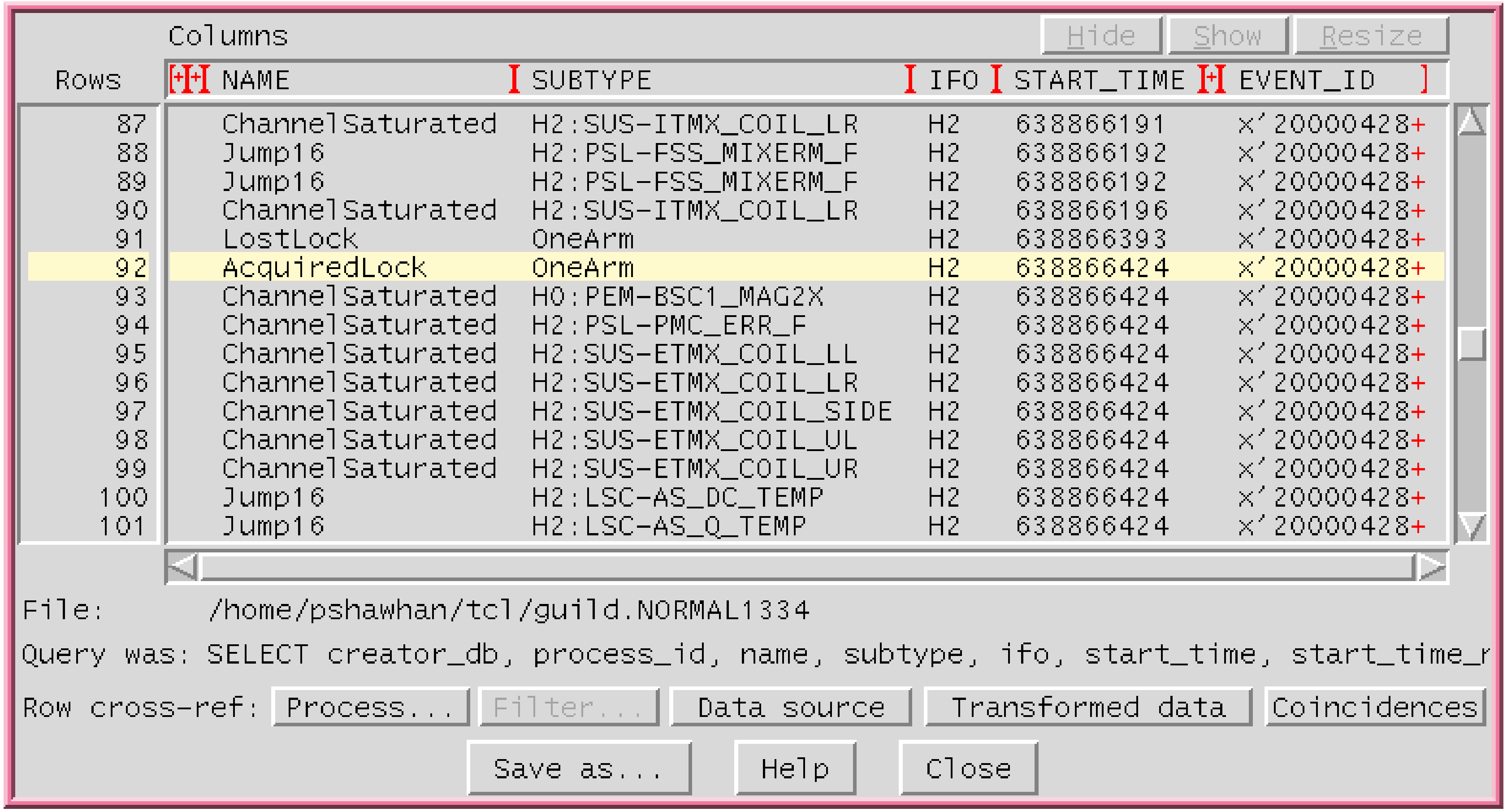
Qualifiers: Text comparisons are not case-sensitive

□ trigger name is
□ trigger subtype is
☒ site/interferometer is H2
☒ start time between 638865000 AND 638866600
□ duration (seconds) =
□ trigger priority =

Built SQL query:

```
SELECT creator_db, process_id, name, subtype, ifo,
start_time, start_time_ns, event_id FROM GDS_TRIGGER
WHERE (UPPER(ifo) = 'H2') AND (start_time BETWEEN
638865000 AND 638866600) ORDER BY start_time, name,
subtype FETCH FIRST 1000 ROWS ONLY
```

guild Table Display



Rows	NAME	SUBTYPE	IFO	START_TIME	EVENT_ID
87	ChannelSaturated	H2:SUS-ITMX_COIL_LR	H2	638866191	x'20000428+
88	Jump16	H2:PSL-FSS_MIXERM_F	H2	638866192	x'20000428+
89	Jump16	H2:PSL-FSS_MIXERM_F	H2	638866192	x'20000428+
90	ChannelSaturated	H2:SUS-ITMX_COIL_LR	H2	638866196	x'20000428+
91	LostLock	OneArm	H2	638866393	x'20000428+
92	AcquiredLock	OneArm	H2	638866424	x'20000428+
93	ChannelSaturated	H0:PEM-BSC1_MAG2X	H2	638866424	x'20000428+
94	ChannelSaturated	H2:PSL-PMC_ERR_F	H2	638866424	x'20000428+
95	ChannelSaturated	H2:SUS-ETMX_COIL_LL	H2	638866424	x'20000428+
96	ChannelSaturated	H2:SUS-ETMX_COIL_LR	H2	638866424	x'20000428+
97	ChannelSaturated	H2:SUS-ETMX_COIL_SIDE	H2	638866424	x'20000428+
98	ChannelSaturated	H2:SUS-ETMX_COIL_UL	H2	638866424	x'20000428+
99	ChannelSaturated	H2:SUS-ETMX_COIL_UR	H2	638866424	x'20000428+
100	Jump16	H2:LSC-AS_DC_TEMP	H2	638866424	x'20000428+
101	Jump16	H2:LSC-AS_Q_TEMP	H2	638866424	x'20000428+

File: /home/pshawhan/tcl/guild.NORMAL1334

Query was: SELECT creator_db, process_id, name, subtype, ifo, start_time, start_time_r

Summary

Have an “existence proof” of metadata generation, storage and retrieval

Metadata handling within LDAS environment is in good shape

guild is very mature, and will be distributed soon

guild has already been useful in shaking down the LDAS system

Now starting to put database tables into active use;
will revise table definitions based on early experience

Still need a interface tool for non-LDAS analysis programs to read and write table data, e.g. for statistical analysis of event candidates; will be a part of the LIGO/LSC Algorithm Library

Also need a user interface tool for data in the frame archive (front end to frameAPI and dataConditionAPI (?))

Together, these tools will help define the “analysis environment” in the broad sense, and facilitate scientific inquiry