LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
–LIGO–
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note LIGO-T020169-00-R Nov. 5, 2002

Global use of LDAS metadatabase across LIGO and LSC sites

Igor Yakushin

Distribution of this draft:
LDAS

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project - MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW:http//www.ligo.caltech.edu/

Contents

1 Problem definition

1.1 How metadatabase is used for data analysis

We are currently using DB2 7.1 Universal Database [1] to store the following metadata [2]:

DMT vetoes Veto events are generated by various DMT [3] monitors in real time or offline.

LDAS Triggers Candidate gravitational wave events are generated by various DSOs running under LDAS [4] either in real time or offline.

Interferometer state There is also a DMT monitor that records into the database information about the interferometer state: was it locked or not at the moment.

Data analysis pipeline, for example, for burst [5] and inspiral [6] upper limits groups, currently looks like this:

- Clean[1] time segments are chosen from the metadatabase for analysis;
- Most of the triggers are vetoed by performing concidence analysis with local glitches at nongravitational channels generated by DMT monitors (the analysis can be done by either running an SQL query in the database or using something like EventTool [7]);
- Finally, coincidence analysis between the remaining unvetoed triggers produced by different interferometers is performed (currently it cannot be done with SQL since local databases from different sites don't communicate with each other, it is done with EventTool [7] or something else).

1.2 Problems with the current use of metadatabase

- At present time each LIGO/LSC site running LDAS[2] writes triggers/vetoes produced by DSO/DMT to its own database. There is no any communication between databases at different sites or even the same site:

[1] The interferometer is locked and its state has not been changed by operators during this time segment.

[2] LLO, LHO, CIT, UWM, MIT, PSU

  - One cannot run an SQL query that would involve metadata from several sites (which might be very desirable for coincidence analysis).
  - One cannot run an SQL query that would involve tables from different databases of the same site (if for some reason one would like to run a query involving E7 and S1 tables produced at LLO, one cannot do it at the moment).

- Some sites regularly backup metadata (LLO, LHO) and some sites do not do it. Therefore metadata might be lost. On the other hand, it might be a burden for each site to take care of backing up its metadatabase.

1.3 Proposed change

We propose to setup an automatic real time replication of metadata from all the sites to one central metaserver in such a way that would allow running SQL queries that can reference metadata from different sites on the central metaserver.

The central metaserver would also take care of backing up all the metadata.

If any LIGO/LSC site would want to have metadata from some other site in real or near real time, they should be able to subscribe for metadata replication from the central metaserver.

For most of the users it should be sufficient to be able to run their jobs on the central metaserver.

The local use of the metadatabases remains unchanged.

2 Relevant DB2 facilities

2.1 Replication

DB2 provides replication facilities [8] to maintain the same set of data in more than one locations. It works both between DB2 and some non-DB2 databases such as Oracle, Sybase, MS SQL Server.

The configuration consists of

- **Control tables** that can be stored anywhere: some database on a target or source servers, or even on some other computer, control server, whose database is cataloged[3] on a target and source servers;
- **Change-capture mechanism**: changes to the source database are captured from log files to temporary tables;
- **Apply program**: reads data from these temporary tables on the source server and copies data to the target database.

2.1.1 Configuration steps

To configure replication, one first must set *logretain* parameter of the source database to *capture* and, perhaps, increase the number of primary logs.

Second, one must declare which tables on the source server are available for replication (define replication sources).

Third, one creates a replication subscription which

- Maps a set of replication sources (tables, views) to targets;
- Defines for each source which columns and rows to replicate;
- Defines a type of a target table;
- Specifies when a replication occurs (either all the members of the subscription set are replicated or none at each replication cycle):
  - Interval timing,
  - Continuous timing,
  - Event timing,
  - On demand timing (supported only to Windows targets);
- Specifies which SQL statements to execute before or after each replication cycle (for example, one might want to clean various auxiliary tables).

[3] One must catalog a remote database to be able to access it locally. It is DB2 analog of NFS mount

Forth, one starts caption program on the source server: *asnccp sample cold noprune*. The capture program examines the log files and captures changes made to the source since the last replication to temporary tables.

Fifth, one starts apply program: *asnapply deptqual copydb*. It usually runs on a target server but can run on any computer that can connect to source, target and control servers.

During the first replication cycle the sources are fully copied to targets. During the consequent cycles only changes made to sources after the last replication cycle are copied to targets.

Most replication configurations support LOB data type[4].

One can configure target tables that are joins or unions of existing source tables.

2.1.2 Target table types

The following types of target tables exist in DB2:

User copy tables - read-only copies of sources;

Point-in-time tables - read-only copies of sources with an extra timestamp column that shows when the last update was made;

Aggregate tables - read-only tables that use SQL column functions (sum, avg, etc.) to compute various summary information:

Base aggregate tables - summary of source content

Change aggregate tables - summary of changes to source content;

Consistent-change-data (CCD) tables - contain data from committed transactions (many different types of CCD tables), can be used in between source and target to speed things up (e.g. source → CCD → multiple targets);

Replica tables - the only target tables that are not read-only: applications can write to them, changes are propagated back to the source and from there to all other replica tables, replica do not support LOBs;

[4] LOB stands for Large Object and allows, for example, to store raw binary data in a relational database table. This functionality is currently used in LDAS to store various time or frequency series

replica requires conflict handling if, for example, source and target or two targets update the same row;

User tables - a source table for replica tables is a target table itself.

2.2 Federation

Federated system [1] is a server that supports queries that reference data located in different local or remote databases as if the corresponding tables were located in one database: for example, one can do joins or unions between table A from DB2 database on the remote server, table B from DB2 database located on the same machine as the federated system and table C from the remote Oracle database. DB2 supports federated system configuration.

Federated system provides location transparency: users refer data objects by nicknames and do not have to know where they are physically located when submitting a query to a federation system. Data can be moved, nicknames updated without any changes to applications

Federated system is just another client for data sources, it does not monopolize or restrict an access to the sources. Data sources do not need to know that they have been used in a federation system.

Distributed queries are limited to read-only operations. Some DB2 specific utility operations are not supported: load, import, reorg, etc. LOBs are not supported in DB 7.1 but are supported starting from DB2 7.2. One can, however, use a pass-through facility to submit SQL statements directly to data sources in their native SQL dialect. One can also write to the source databases using pass-through access.

2.2.1 Configuration steps

To configure the federated system, one must install DB2 with 'Distributed Join' support enabled.

If non-DB2 databases will be used as sources, DB2 Connect (not free even under IBM Scholar Program license!) should be installed.

Set the database manager configuration parameter *federated* to *yes*.

Create wrappers. A wrapper loads a library used to access a particular class of data sources. For DB2 data source load choose DRDA wrapper.

Create servers. A server describes a data sources: its type, location, wrapper to use, authorization information and other server options.

Create nicknames for tables, views, aliases for the objects in the data sources.

You might also need to define user mapping, data type mapping, function mapping, index specifications (to improve performance of the distributed queries), etc.

For more details about relevant DB2 features see [9].

3 Implementation details

Central metaserver with fast network connectivity running LDAS and DB2 7.2 is installed, for example, at Caltech.

LDAS systems from different sites are connected through VPN and therefore metaservers can communicate with each other directly.

All important databases at sites are set to be continuously or periodically replicated to separate databases on the central metaserver using DB2 replication facilities.

User copy target tables are used.

On the central metaserver all the replicated and local databases can be accessed through federation database configuration so that one can run SQL queries that include tables from different databases.

Replicas on the central metaserver are accessed in read-only way through federated system or directly. The results of the queries are recorded into local database.

We can either continue allowing access to the databases only through LDAS using guild, or we might want to consider allowing users to use DB2 directly if it turns out that using guild is too restrictive.

The central metaserver can also serve as a replication source in case some other sites would want to have metadata from other sites. We anticipate though that at least in the nearest future anybody who needs to have metadata from different sites at one place would simply run their queries at the central metaserver.

Central metaserver takes care of backing up and archiving metadata regardless of whether it is done at sites or not.

Currently we have a *creator_db* entry in the process table that is supposed to indicate at which site the metadata was generated. It is not used at the moment but should be for the described configuration to work.

During the first stage of implementing this plan, let us just set up the central metaserver and replicate data from LLO and LHO the way it is described above.

Once we work out the first stage, we can proceed replicating data from more sites and allowing other sites to replicate data from the central metaserver.

References

[1] IBM.
DB2 Administration Guide.

[2] Peter Shawhan.
Table Definitions for LDAS Metadata/Event Database.
Technical Report T990101-02-E, LIGO, Caltech, Nov 1999.

[3] Data Monitor Tool Project web page.
http://www.ligo.caltech.edu/ jzweizig/DMT-Project.html.

[4] LDAS web page.
http://www.ldas-sw.ligo.caltech.edu/.

[5] Peter Saulson.
E7 Burst Search Status Report.
Technical Report G020404-00-Z, LSC, Jun 2002.

[6] Patrick Brady and Gabriella Gonzalez.
E7 Inspiral Search Status Report.
Technical Report G020424-00-Z, LSC, Sep 2002.

[7] Event Tool web page.
http://emvogil-3.mit.edu/ cadonati/EventTool/EventTool.html.

[8] IBM.
DB2 Replication guide and reference.

[9] Igor Yakushin.
Selected Features of DB2 and Metadata Replication.
Technical Report G020413-00-L, LIGO, Caltech, Apr 2002.