LCG Database Deployment and Persistency Workshop
CERN 17-19 October 2005

Heterogeneous Database Replication

Gianni Pucciani

A.Domenici andrea.domenici@iet.unipi.it
F.Donno flavia.donno@cern.ch
L.Iannone lauraiannone@infinito.it
G.Pucciani gianni.pucciani@pi.infn.it
H.Stockinger Heinz.Stockinger@univie.ac.at

Introduction

- Oracle Heterogeneous Connectivity.
  - Streams based heterogeneous replication.
- The Constanza project.
  - RCS architecture.
  - RCS for Oracle to MySQL replication.
- Conclusions.

Tests on Oracle Heterogeneous Connectivity and Streams based replication

Laura Iannone's thesis available at:
http://etd.adm.unipi.it/theses/available/etd-0621005-153510/

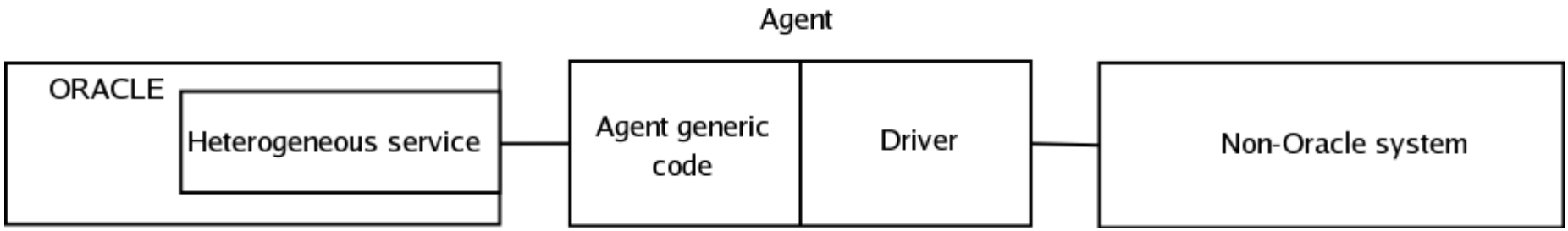
Oracle solutions to the Heterogeneous Connectivity problem:

- **Transparent Gateways**: only provides gateways for specific non-Oracle platforms like Sybase, MS SQL, Informix etc. MySQL is NOT supported.
- **Generic Connectivity**: more limited functionality than Transparent Gateways, connection only to local Oracle Database server, no distributed transactions.

Both provide the ability to transparently access data in non-Oracle databases from an Oracle environment.

Oracle Heterogeneous Connectivity

The Heterogeneous Connectivity process:



- Heterogeneous service: integrated in the Oracle server.
- Agent: provides connectivity to non-Oracle systems.

Transparent Gateways and Generic Connectivity are two types of agents.

Oracle Generic Connectivity test

Oracle machine located at CNAF (SL, Oracle 10.1.0.2.0)

MySQL machine at INFN Pisa (RH9, MySQL 4.1.9).

- Setting up Oracle Heterogeneous Services.
- Setting up ODBC Driver.
- Setting up Generic Connectivity Agent.

*Eventually, we were able to successfully update the remote MySQL database from the Oracle SQL*Plus console .*

Oracle to MySQL data sharing with Streams 1/2

In some Oracle documents we found that Streams can apply changes to a non-Oracle system via Transparent Gateways or Generic Connectivity.

- We set up an Oracle Streams environment with the *apply* process linked to the remote MySQL database.

- We created a simple table on both the Oracle and MySQL database.

- With the Streams processes activated we tried to make some DML changes to the Oracle table.

Oracle to MySQL data sharing with Streams 2/2

The *capture* process correctly captured changes but the *apply* process aborted. Oracle support has been contacted and they told us that Streams needs to have an Oracle Transparent Gateway for the Oracle to non-Oracle data sharing.

Results: altough we can access a remote MySQL database from an Oracle server, the Streams tool does not work with MySQL.

So, we need to look for different solutions to implement Oracle to MySQL synchronisation.

The CONStanza project

- Developed at INFN, funded by the Italian FIRB project
- Site: http://www.pi.infn.it/~pucciani/constanza
- Main goals: developing a Replica Consistency Service (RCS) that maintains consistency of writable replicated datasets (files and databases) in a Grid environment.
- Current focus on:
  - heterogeneous database replication of VOMS DB from Oracle to MySQL
  - simple but reliable protocol for update propagation (single master).

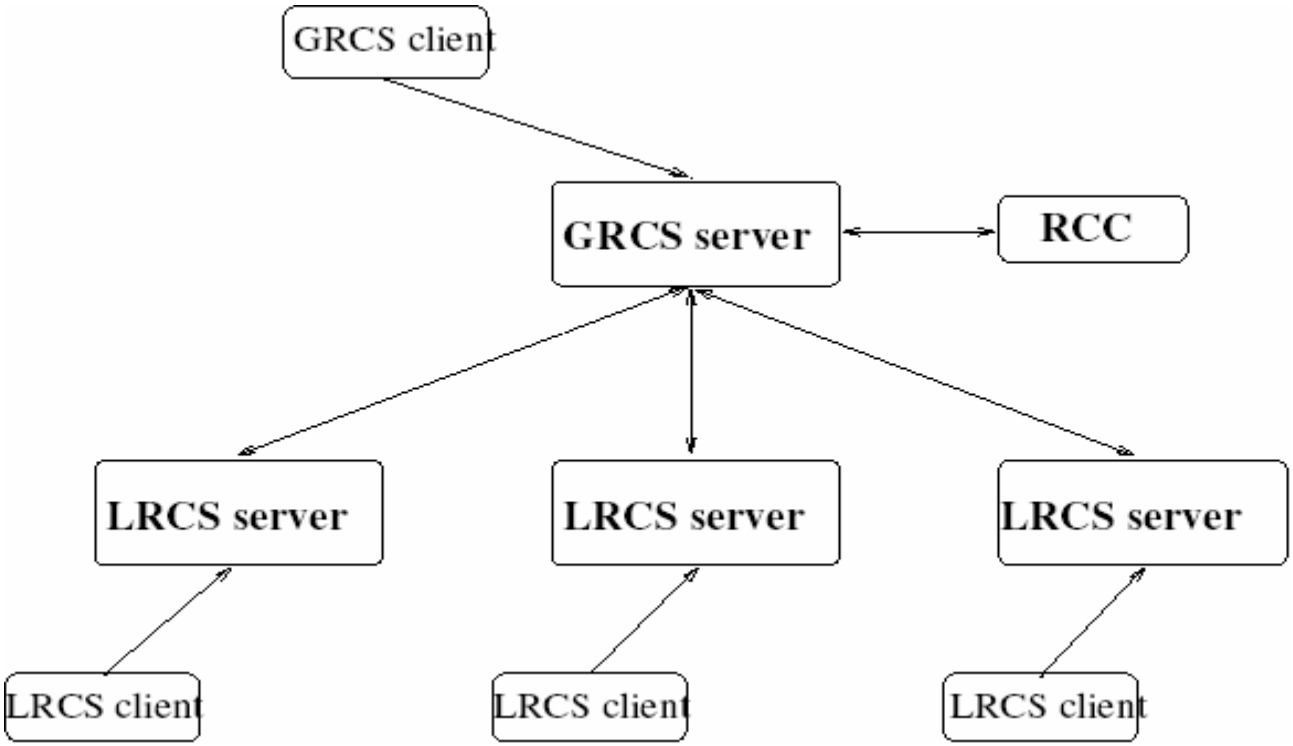
RCS: general features

- Designed as flexible as possible to support multiple update propagation protocols and update mechanisms.
- Prototype implemented in C++ with gSOAP as communication framework.
- Other used/to be used tools are: Autotools, doxygen, CppUnit with mockpp, CGSI.
- Using a configuration file we can set most RCS properties.

RCS: particular features

- Dynamic replica subscription.
- Update requests can be both blocking and non-blocking.
- Quorum constraint on update requests.
- Fault tolerant communication.
- For DB replicas: *limited SQL dialect/data types translation.*

RCS Architecture

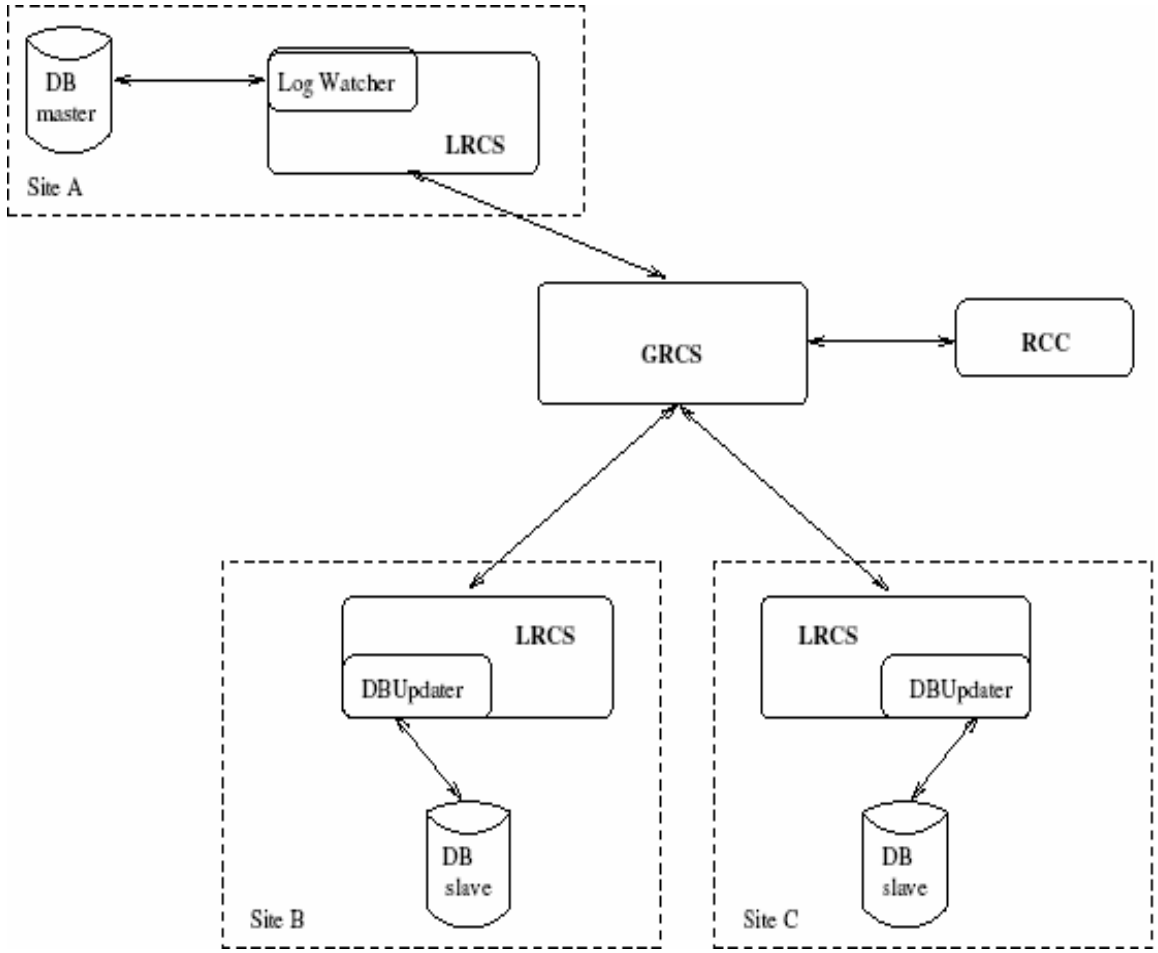


- GRCS, entry point for user requests (bottleneck and single point of failure issues will be addressed if needed).
- LRCSs, distributed servers “close” to replicas.

RCS: architecture for DBs

Two specialised components have been added to extract/apply updates to databases:

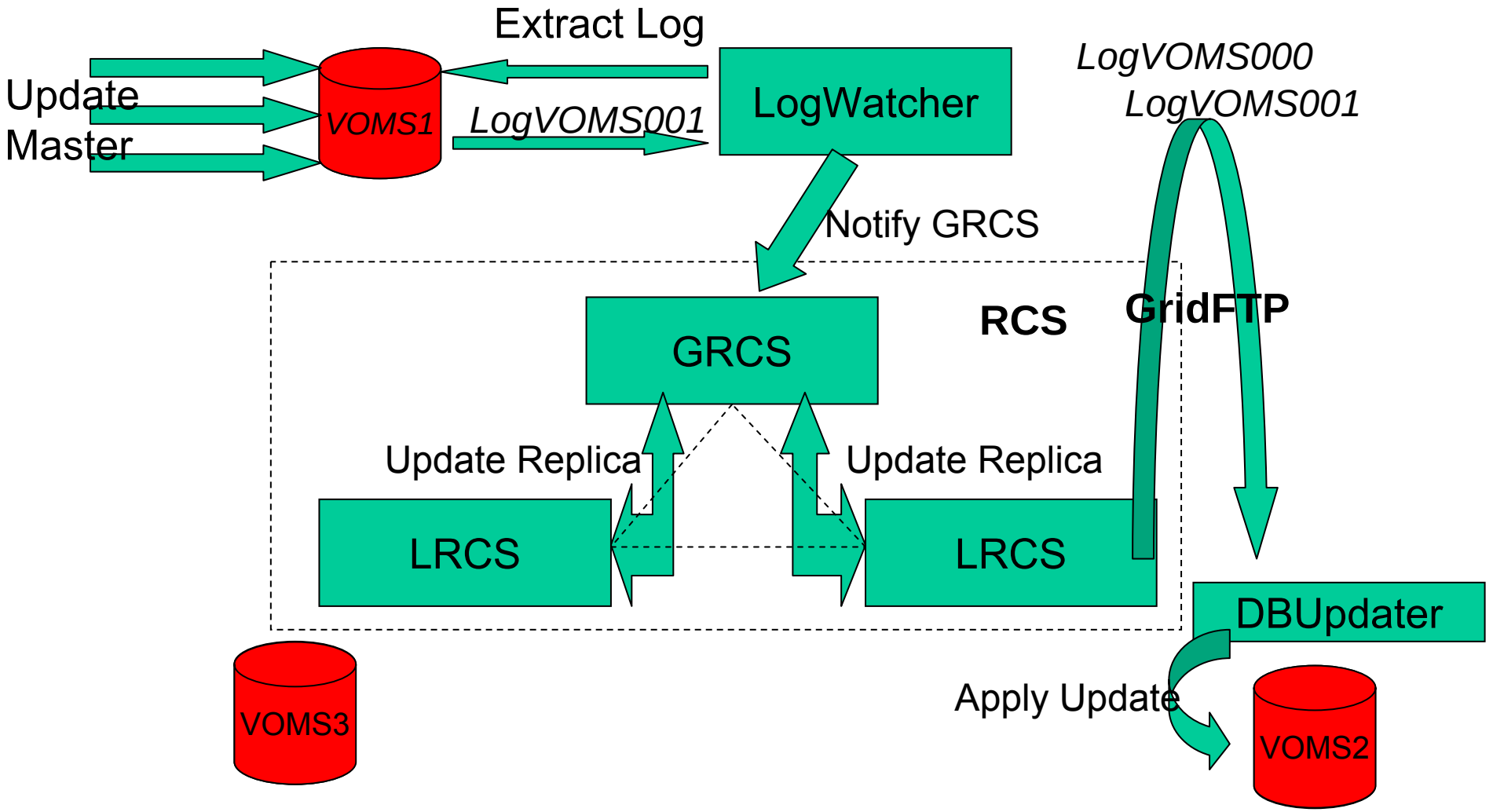
- Log Watcher
- DB Updater



RCS: current status

- The current architecture works with homogeneous MySQL databases.
- It monitors the master DB log file, extracts last updates and propagates them to the RCS registered secondary replicas. Updates are then applied to each secondary replica.
- If a replica is temporarily unreachable, a tentative update can be repeated later on.

RCS: current scenario

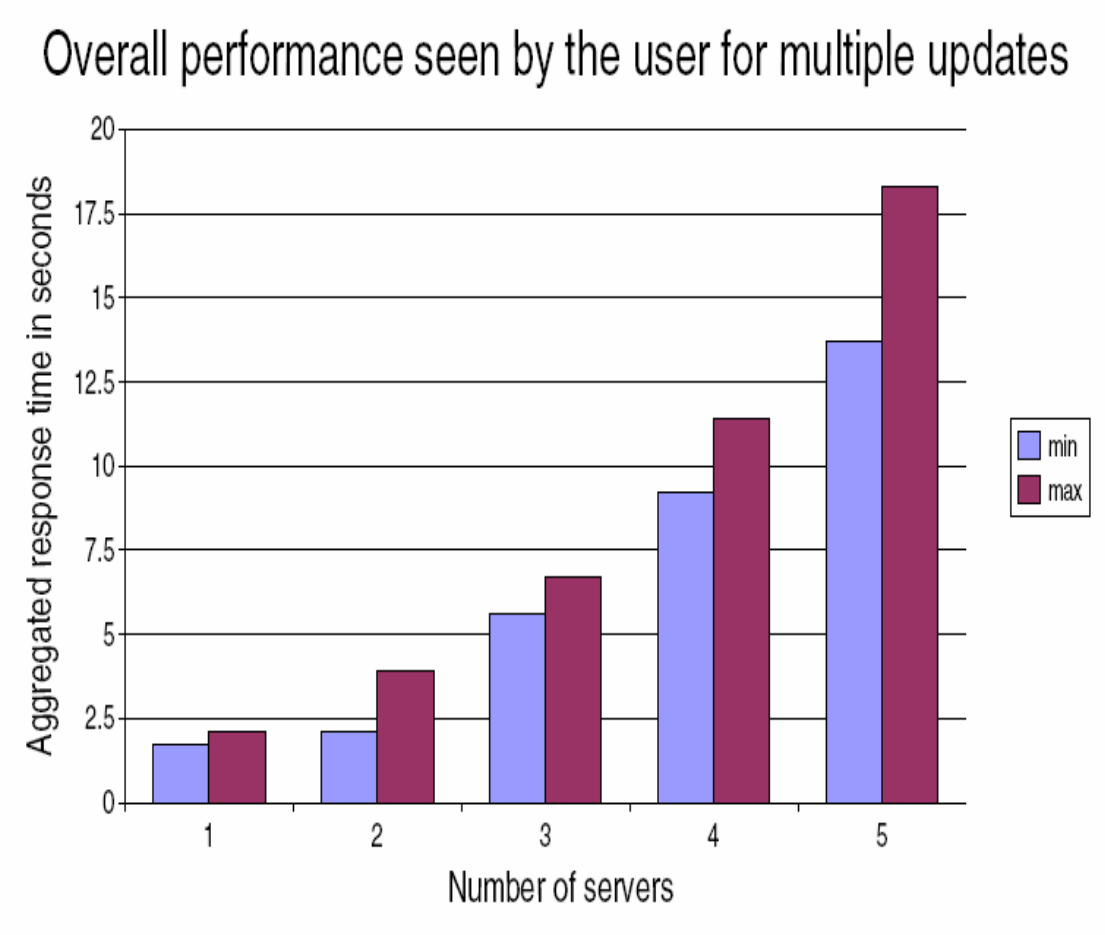


RCS: fault tolerant links

- GRCS→LRCSs: when an LRCS is not reachable, the update operation involved is stored in a queue to be executed later on, periodically or upon an LRCS request.
- Log Watcher → GRCS and DB Updater → DB links will use a similar system.

RCS: first results

Preliminary performance tests done using a master DB located in Pisa and secondary DBs at CERN (50ms RTT) and SLAC (165ms RTT).



RCS: Oracle to MySQL replication

- The LogWatcher component can be specialised to work with Oracle as it works with MySQL using the Oracle **LogMiner** utility and the OCCI interface (work in progress).
- The difficult problem of SQL dialect translation will be addressed for special cases (starting with VOMS use case). A general SQL statement translation module is almost impossible to build. Solutions can be:
  - Reducing the use of non-standard SQL statetements.
  - Agreeing on a common application-level interface for log storage/exctraction/application.

Conclusions

- Oracle Streams does not work with MySQL.
- Can a Transparent Gateway driver be done for MySQL as well?
- CONStanza RCS.
- Using Log Miner RCS should be able to synchronise an Oracle DB with MySQL DBs.
- The problem of non-standard SQL dialects must be addressed.