Replication Tools in the MADIS Middleware

L. Irún-Briz¹, J.E. Armendáriz², H. Decker¹, J.R. González de Mendívil², F.D. Muñoz-Escoí¹

¹ Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera s/n
46022 Valencia (Spain)
{lirun, hendrik, fmunyoz}@iti.upv.es

² Dpto. Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona (Spain)
{enrique.armendariz, mendivil}@unavarra.es
Replication Tools in the MADIS Middleware

Index:

- Introduction.
- MADIS Architecture.
- MADIS DBlayer.
- Consistency Manager.
- Performance Results.
- Conclusions.
Introduction

- Two approaches for building a replication support for databases:
  - Adding or modifying some components of the DBMS core at each replica (e.g., Dragon).
    - Advantages: Good performance and the addition of a minimal amount of code.
    - Drawbacks: DBMS-dependant.
  - Via middleware (e.g., C-J DBC or GlobData).
    - Advantages: Portability.
    - Drawbacks: Additional overhead.
Replication Tools in the MADIS Middleware

- Index:
  - Introduction.
  - MADIS Architecture.
  - MADIS DBlayer.
  - Consistency Manager.
  - Performance Results.
  - Conclusions.
MADIS Architecture

User Application

Consistency Manager (JDBC Driver)

Conflict Detector

Replication and Recovery Protocol

DBMS

JDBC Interface

Communication Service
Membership Service
Group Comm. System

Original Schema

Extended Schema
Replication Tools in the MADIS Middleware

Index:
- Introduction.
- MADIS Architecture.
- MADIS DBlayer.
- Consistency Manager.
- Performance Results.
- Conclusions.
MADIS DBLayer

- It is an extension of the original schema of a given database. It provides several tables, triggers and stored procedures.

- Each table \( T_j \) has an additional metadata table called \( \text{MADIS\_META\_} T_j \). It contains the following fields:
  - \text{Local\_ID}: local identifier; primary key.
  - \text{GLOBAL\_ID}: unique global identifier.
  - \text{VERSION}: the row’s version number.
  - \text{TRANSACTION\_ID}: ID of the last transaction that updated the row.
  - \text{TIMESTAMP}: most recent time where the row was updated.
A global TRREPORT table (or, per transaction, depending on the expected load).

- It is automatically filled each time an update, delete or insert operation is executed by the target transaction.

- It contains the following fields:
  - TR_ID: transaction identifier.
  - GLOBAL_ID: global row identifier.
  - FIELD_ID: Optional accessed field identifier.
  - MODE: access mode (read|insert|update|delete).
MADIS DBlayer. \textit{Writeset collection.}

- Triggers are introduced so as to provide the following:
  - Automation of the metadata generation: Version numbers, setting timestamps, writing the transaction ID of the last updating transaction.
  - \textbf{Writeset collection:} Several triggers are implemented to perform this task (WSC): \texttt{WSC\_I\_T}_j, \texttt{WSC\_D\_T}_j \textbf{and} \texttt{WSC\_U\_T}_j.
MADIS DBLayer. *Writeset collection*

**Trigger example:**

```
CREATE TRIGGER WSC_I_mytble
BEFORE INSERT ON mytable
FOR EACH ROW EXECUTE PROCEDURE tr_insert(mytble, getTrid(), NEW.l_mytble_oid);
```

- A table of the original schema
- Current Transaction ID
- Inserting Row ID
- Insert operation
Conflict Detection

Conflict detection:
- Needed for detecting conflicts between local active transactions and the updates received from remotely initiated ones.

Main goals:
- Report conflicting transactions to the protocol.
- Abort a conflicting transaction according to the protocol decision.
MADIS DBLayer. Conflict Detection

Multiple priority levels may be assigned to transactions:

- In case of conflict...
  - If the conflicting transactions have different priority levels, that with the lowest priority is automatically aborted.
  - If both transactions have the same priority level, none of them gets aborted, but the protocol is notified about this.
- In the BULLY protocol...
  - Priority 0 for local transactions that have not requested their commit.
  - Priority 1 for transactions that have sent their WS.
  - Priority 2 for transactions whose WS has been delivered.

- An execution thread per transaction.
Replication Tools in the MADIS Middleware

Index:
- Introduction.
- MADIS Architecture.
- MADIS DBlayer.
- Consistency Manager.
- Performance Results.
- Conclusions.
Consistency Manager

- It is the actual middleware layer, providing a JDBC interface.
- It provides support for the replication protocol.
  - We have multiple protocols that can be plugged into this consistency manager.
- It uses the JDBC interface provided by the underlying DBMS. This DBMS holds the support described as DBlayer.
Consistency Manager

Internal architecture:

- Several JDBC classes have been extended:

- Some of these classes are able to provide a parsing of the SQL statements.

- The replication protocol may use this parsing for retrieving some metadata in read or write accesses:
  - The SQL statement has to be extended before it is transferred to the underlying DBMS.
  - The ResultSet must be read and modified once the statement has been executed by the DBMS.
Consistency Manager

Protocol interface:

- The consistency manager provides an interface for the plugged replication protocol.
- It will be able to notify certain events to this replication protocol. Such protocol has to implement a “Listener” interface.
- There are several ways of requesting an event:
  - In the registration of the replication protocol.
  - With specific operations, at any time.
- Events:
  - RECOVERED, UPDATED, UPDATE_PRE, UPDATE_POST, QUERY_PRE, QUERY_POST, ACCESSED, TREE.
- Besides the events, the WS and metadata may be explicitly got using some retrieving operations.
- WS and metadata for each transaction are deleted once the transaction has locally terminated, either with commit or abort.
Replication Tools in the MADIS Architecture

Index:
- Introduction.
- MADIS Architecture.
- MADIS DBlayer.
- Consistency Manager.
- Performance Results.
- Conclusions.
Performance Results

- Overhead Description. It can be classified into four main categories:
  - Insertion.
    - Insertion of a row into TRREPORT and the respective MADIS_META_Tj tables.
  - Update.
    - Insertion of a row into TRREPORT and updating the MADIS_META_Tj table.
  - Deletion.
    - Insertion of a row into TRREPORT and deleting the row from the MADIS_META_Tj table (deferred mode).
  - Selection.
    - There is no reason to alter the MADIS_META_Tj table. Although it may be possible to insert a row into the TRREPORT table.
Performance Results. Mean Overhead
Performance Results. Mean Overhead

overhead (in %)

% 700 600 500 400 300 200 100 0

numrows

numtr 90 80 70 60 50 40 30 20 10 0 1.5 2 2.5 3 3.5 4 4.5

MADIS I
MADIS U
MADIS D
Replication Tools in the MADIS Architecture

Index:
- Introduction.
- MADIS Architecture.
- MADIS DBlayer.
- Consistency Manager.
- Performance Results.
- Conclusions.
Conclusions

- MADIS is a middleware designed to support a wide range of replication protocols.
  - Minimal database extensions.
  - Replication applications.
- The Consistency Manager uses the automatically collected information in the DBMS.
  - It notifies user applications accesses to the replication protocol.
  - Its implementation allows user applications to access in a JDBC fashion to the replicated database.
  - No need to change their codes.
- We plan to use MADIS for testing several replication protocols:
  - Object replication and reconciliation protocols designed in the DeDiSys project.
  - Mobile databases protocols.

The DeDiSys project (http://www.dedisys.org/) investigates the trade-off between availability and consistency in partitionable environments. It uses synchronous replication protocols in healthy systems and asynchronous replication in case of partitions. It is based on constraint consistency, and on support for nested transactions.
Replication Tools in the MADIS Middleware

L. Irún-Briz\textsuperscript{1}, J.E. Armendáriz\textsuperscript{2}, H. Decker\textsuperscript{1}, J.R. González de Mendívil\textsuperscript{2}, F.D. Muñoz-Escó\textsuperscript{1}

\textsuperscript{1}Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera s/n
46022 Valencia (Spain)
{lirun, hendrik, fmunyoz}@iti.upv.es

\textsuperscript{2}Dpto. Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona (Spain)
{enrique.armendariz, mendivil}@unavarra.es