Evaluating Database Replication Protocols in the MADIS Middleware Architecture

J.E. Armendáriz¹, J.R. Juárez¹, J.R. Garitagoitia¹, J.R. González de Mendívil¹, F.D. Muñoz-Escoí²

Presents: José Enrique Armendáriz Iñigo
enrique.armendariz@unavarra.es

¹ Grupo de Sistemas Distribuidos
Dpto. Matemática e Informática
Universidad Pública de Navarra

² Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Outline

- Introduction
- BULLY Replication Protocol
- TORPE Replication Protocol
- MADIS Architecture
- Experimental Results
Introduction

- A middleware architecture providing DB replication
  - No lock management needed at the middleware layer
  - We rely for concurrency control on the DBMS
  - DBMS internals remain unaccessible
  - The consistency is maintained at the middleware layer
  - Client applications “see” a standard interface
- We propose two protocols
  - BULLY
    - Adaptation of the O2PL proposed by M.J. Carey and M. Livny
      - Distributed deadlock prevention schema based on:
        - Transaction’s priority
        - Transaction’s state in the system
  - TORPE
    - Adaptation of the 1-SER proposed by B. Kemme and G. Alonso
      - Transaction’s write operations are performed in the local phase
    - Based on total order delivery guarantees of GCSs
BULLY Replication Protocol

Remote Txn

create(t)
begin_operation(t,op)
end_operation(t,op)
{r(x),w(y),r(z),w(x)}

begin_operation(t,op)
WS = {x,y}
Send Updates

begin_commit(t)
WS = {x,y}
Send Updates

wait response
{j,k}

end_commit(t)
Commit

receive_remote(t,m)
Apply Updates
{x,y}

receive_commit(t,m)
Commit

receive_remote(t,m)
Apply Updates
{x,y}

receive_commit(t,m)
Commit

Remote Txn

pre_commit
committed
pre_commit
committed
pre_commit
committed
TORPE Replication Protocol

Remote Txn

create(t)
begin_operation(t,op)
end_operation(t,op)
\{r(x),w(y),r(z),w(x)}

begin_commit(t)
WS = \{x,y\}
TO-Mcast Upd.

end_commit(t)

receive_commit(t,m)
Commit

receive_remote(t,m)
Apply Updates
\{x,y\}

receive_commit(t,m)
Commit

receive_remote(t,m)
Apply Updates
\{x,y\}

receive_commit(t,m)
Commit

Remote Txn

\begin{align*}
\text{remote, } t, \text{ WS} \\
\text{remote, } t, \text{ WS} \\
\text{commit, } t \\
\text{remote, } t, \text{ WS} \\
\text{commit, } t \\
\text{remote, } t, \text{ WS} \\
\text{remote, } t, \text{ WS} \\
\text{commit, } t \\
\end{align*}
MADIS Architecture

- Java Client Program
- Servlet Container
- EJB Container
- Consistency Manager (JDBC Driver)
- Conflict Detector
- Consistency and Recovery Protocol
- Group Comm. System
  - Communication Service
  - Membership Service
- JDBC Interface
- DBMS
  - Extended Schema
  - Original Schema
Performance Analysis
Performance Analysis

![TORPE Protocol - Response Time Graph](image)
Performance Analysis

BULLY PROTOCOL - RESPONSE TIME

- Response time (ms)
- Number of clients
- 1 client, 2 clients, 4 clients, 8 clients, 12 clients, 16 clients
- 2 nodes, 4 nodes, 8 nodes
Performance Analysis

TORPE PROTOCOL - RESPONSE TIME

Response time (ms)

1 client, 2 clients, 4 clients, 8 clients, 12 clients, 16 clients

Number of clients

2 nodes, 4 nodes, 8 nodes
Evaluating Database Replication Protocols in the MADIS Middleware Architecture

J.E. Armendáriz¹, J.R. Juárez¹, J.R. Garitagoitia¹, J.R. González de Mendívil¹, F.D. Muñoz-Escoí²

Presents: José Enrique Armendáriz Iñigo
enrique.armendariz@unavarra.es

¹ Grupo de Sistemas Distribuidos
Dpto. Matemática e Informática
Universidad Pública de Navarra

² Instituto Tecnológico de Informática
Universidad Politécnica de Valencia