Group Communications and Database Replication: techniques, issues and performance

Matthias Wiesmann
PhD Thesis Exam
May 2002
Group Communications + Databases = Replicated Database
Outline

- Introduction
- Classification
- Failure Semantics
- Performance Simulation
- Conclusion
Outline – Introduction

- Introduction
  - Database Replication
  - Database & Group Communications
  - Problems
  - Solutions – Three axis approach
- Classification
- Failure Semantics
- Performance Simulation
- Conclusion
Database Replication

- One logical database
- N physical copies
- All copies are synchronised
  - Eager replication
- All servers enforce ACID properties
- Network links replicas
- Clients connect to the system
Database Replication
Eager vs Lazy

- Two ways to replicate databases:
  - Eager ⇨ all replicas are synchronised.
  - Lazy ⇨ replicas might diverge (violates ACID).
- Eager replication is considered expensive.
  - Gray et al, 1996
- New approach needed.
Idea: use group communication infrastructure.
- Use broadcasting primitives.
- Old idea (Chang 1984).

Re-use of work already done:
- Strong guarantees.
- Simplified design \(\rightarrow\) components.
- Better performance – less deadlocks.

Recent area of research
- DRAGON Project (EPFL & ETHZ)
- Explorative Work
  - Many techniques.
  - Are all found?
- Two communities:
  - Different terminology
  - Mismatched model (failures)
- Performance?
  - Group communications are considered slow…
Everything is different

<table>
<thead>
<tr>
<th></th>
<th>Distributed Systems</th>
<th>Database Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motivation</strong></td>
<td>Fault-Tolerance</td>
<td>Performance</td>
</tr>
<tr>
<td><strong>What is Replicated</strong></td>
<td>Processes</td>
<td>Data</td>
</tr>
<tr>
<td><strong>Operation Type</strong></td>
<td>Single (Events)</td>
<td>Multiple (Transactions)</td>
</tr>
<tr>
<td><strong>System Model</strong></td>
<td>Synchronous, Asynchronous etc...</td>
<td>Synchronous</td>
</tr>
<tr>
<td><strong>Consistency</strong></td>
<td>Causal, FIFO, Linearisability</td>
<td>1-copy Serialisability</td>
</tr>
</tbody>
</table>
Solution – Three Axis Approach

- **Structural Understanding**
  - Classification

- **Qualitative Understanding**
  - Study of failure semantics

- **Quantitative Understanding**
  - Performance simulation

Classification

Structural

Qualitative

Quantitative

Simulation

Failure

Semantics
Outline – Classification

- Introduction
- Classification
  - Introduction
  - Criterion
  - Examples
- Failure Semantics
- Performance Simulation
- Conclusion
Structural Classification of Techniques

- Highlights similar techniques.
- Systematic exploration of solution space.
- Classify existing techniques.
- Shows technical requirements for each category.
Existing Classifications

- CHKS94, CP92, WPS99
- Cannot handle non-voting replication
- Concentrate on primary back-up
- Do not use orthogonal criterion
- Include lazy techniques
  - Difficult to compare (relax ACID).
Classification – 3 Criterion

- **System Architecture**
  - Primary-Copy or Update-Everywhere.
  - Follows Gray's classification.
- **Number of Network Interactions**
  - $O(1)$ or $O(n)$ network interactions.
- **Transaction Termination**
  - Voting or non-voting.
Criterion 1 – System Architecture

- Where can transactions be submitted
  - Update Everywhere ➡ any server
  - Primary Copy ➡ primary server

- Important for conflict handling.
Criterion 2 – Number of interactions

- How many communications rounds?
  - $O(1)$ One interaction per transaction
  - $O(n)$ One interaction per operation
- Gives idea of network usage
  - We abstract precise protocol.
  - We avoid implementation details.
Criterion 3 – Transaction termination.

- How is the transaction terminated?
  - Multilateral agreement ➭ Strong-Voting
  - Unilateral agreement ➭ Weak-Voting
  - No agreement ➭ Non-Voting

- Is there a synchronization round?

Synchronization:
- Primary-copy: Update-everywhere
- Constant interaction
- Linear interaction
For each replication class

- Abstract Overview
  - Presents general structure
- Many replication techniques
  - List of relevant techniques
- Requirements:
  - On the communication system (order, uniformity)
  - On the database system (determinism)
Point of Determinism

- Determinism important issue
  - How do you quantify determinism?
- Point of Determinism (*PoD*)
  - Marks beginning of deterministic processing.
  - Related to the notion of serialization point.
  - Different databases have different *PoDs.*
Non-Voting Constant Interactions
Primary Copy

- Cold Standby Primary
  - Typical Commercial Configuration.
  - Needs Uniform FIFO Broadcast.
  - Cold Standby (no flow control).
  - Usually 1-Safe.

Primary Server

Other Server
Classical form of replication

- Read One Write All technique (ROWA).
- Each operation is sent to all replicas.
- The transaction is terminated by 2PC protocol.
Typical Group Communication Replication

- Needs total order broadcast.
- Needs a known point of determinism (PoD)
  - If the PoD at the start
    - Active Replication
  - If the PoD after start
    - Certification based replication
  - If the PoD in the middle
    - Possible – never proposed.
Classification – Results

- Classification helps:
  - Explore solution space.
  - Understand the relation between existing techniques.
  - Understand the requirements for:
    - communication system
    - database system.
  - Give Basis for comparing the techniques
    - Used as basis for simulation.
  - Earlier version quoted in Books (Coulouris, Tanenbaum)
Outline – Failure Semantics

- Introduction
- Classification
- Failure Semantics
  - Introduction
  - Roll-Forward Recovery
  - Roll-back recovery
  - Group Safety
- Performance Simulation
- Conclusion
Analysis of Fault Tolerance Semantics

- Group Communications vs Database
  - Different failure models.
  - What are the properties of the combined system?
- Database safety criterion: 1-safe & 2-safe
  - What kind of safety for group communication based database replication?
  - Better suited safety criterion?
- When is a client is notified of a commit?
  - When the transaction committed on one site.
    - 1-Safe
  - When the transaction will commit on all sites
    - 2-Safe
Group Communications based Database Replication

- Group communication model
  - Usually considered: dynamic crash no recovery (views).
  - Existing toolkits are in this model.

- Not adapted for 2-safety
  - Cannot tolerate total crash (majority needs to be up).
  - Recovery based on roll-forward recovery.
  - Even if the first issue could be addressed…
  - …the second issue remains…
- Basis of view based system.
- State if transferred from a good replica.
- Does not work if there is no good replica

Roll-Forward Recovery
To build 2-safe replication, we need:

- To tolerate a full crash
  - crash-recovery model with stable storage

- Roll-back based recovery
  - Messages need to be **successfully** delivered
  - Message are delivered, **and** processed by the application
  - If delivery is not successful ➫ deliver again
    - Message replay
Inter-Layer Messages

- Synchronisation needed between application and communication system
- We need to know when a message is successfully delivered.
A total crash can be recovered.
Classical group communications based replication

- Not 2-safe.
- Is it only 1-safe?

Classical 1-safe replication

- One crash ⇒ lost transaction.
- Group communications, this cannot happen.

We need another safety criterion.
Quantify the number of sites were a transactions is delivered.

<table>
<thead>
<tr>
<th>Transaction Delivered</th>
<th>No Replica</th>
<th>1 Replica</th>
<th>All Replica</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Processing</td>
<td>No Replica</td>
<td>1-Safe</td>
<td>2-Safe</td>
</tr>
<tr>
<td>No Safety (zero-Safe)</td>
<td>1 Replica</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group-Safe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Classification

- Structural
- Qualitative
- Quantitative

Qualitative: Simulation
Quantitative: Failure Semantics
Group Safety – Philosophy

- **2-Safe:**
  - Transaction is safe when *committed* on all sites

- **Group-Safe**
  - Transaction is safe when *delivered* on all sites

- **Durability**
  - Assumes one component never fails
  - Classical safety $\Rightarrow$ stable storage (disk)
  - Group-Safety $\Rightarrow$ group of servers
A technique can be:

- 1-safe and group-safe

What does 1-safety bring?

- Transaction committed on one disc.
- In case of total crash: last chance.

Problem:

- We must block (wait) for this chance.
- Not very useful in practice.
Advantages of Group-Safety alone...

- Decreased latency
  - We do not wait for any stable storage.
  - Writes are executed outside transaction.
- **Group Safe & 1 - Safe**
- **Group Safe**
- **Lazy replication**
  - Considered optimum performance
Group Safety vs Lazy Replication (1)

- Group safety: good alternative to lazy replication
  - Good performance.
  - ACID not violated if less than $f$ crashes occur.

- Orthogonal Approaches
  - In each case, we relax a slow link.
  - Lazy replication $\Rightarrow$ link between servers.
  - Group Safe replication $\Rightarrow$ link with stable storage.
  - Network faster than Disk I/O (LAN).
Group Safe vs Lazy Replication (2)

Structural
Qualitative
Quantitative
Classification
Simulation
Failure Semantics
Group communication based database replication

- Usually not 2-safe (toolkit model) $\Rightarrow$ 1-safe.
- 2-safe is possible (but toolkit is not available).
- But more than 1-safe.
- Group safety is more adapted.
- Group-safe replication (without 1-safety)
  - Offers better performance
Performance – Outline

- Introduction
- Classification
- Failure Semantics
- Performance Simulation
  - Simulator
  - General
  - Scalability
  - Query Load
- Conclusion
- Understand performance of techniques
  - Behaviour with different loads
  - Scalability, load balancing etc...
  - Use of different resources (disk, cpu, network)
  - See practical issues (concurrency, garbage collection)
- Discrete event simulation
- Uses C-Sim (c++ version)
- Low-level resources simulated
  - Disks, CPU and network
- High-level operations executed in the simulator
  - Locking, transaction processing, communication protocols.
- ≈ 35'000 lines of code.
Simulated Techniques

- Follows classification
  - At least one technique per category (except one).
- Classical techniques:
  - Distributed locking, primary copy, lazy
- Group Communication techniques:
  - Active replication, certification, Ser-D.
- Optimisations
  - Group safe, optimistic…
- **Transactions:**
  - 5-15 operations 50% queries

- **Load:**
  - 10 -20 transaction / second

- **System:**
  - 9 Servers and 36 clients

- **Servers:**
  - 2 CPU, 2 Disks, fast ethernet interface (100 Mb/s)
General Performance – Results

- Distributed Locking
  - Network not issue.
  - Synchronisation is.
- Active Replication
  - Serialisation phase
- Primary backup
  - Primary is bottleneck
- G. Com. Based techniques
  - Close to lazy (optimum)
- Clients: 36
- Servers:
  - 2-36
- All technique scale
- Distributed locking
  - Performance degrades when to many servers.
- Low load (10 trx/second)
- Changing query proportion
  - (0% - 100%)
- Active replication
  - Better than primary copy
  - Response collection
- Group Communication
  - Close to lazy (optimum).
Simulation – Conclusion

- Simulation gives insight on behaviour
- Network is **not** bottleneck
  - But synchronisations has impact on performance.
- Group communication technique perform well
  - Practical issues: garbage collection, serialisation, lock contention etc…
Group Communication based database Replication:

- Good approach for database replication.
- Database specific techniques offer good performance.
- Can be made 2-safe (need more work).
- Group Safe replication offers increased performance.
- Many improvements & optimisations possible.
Future Works

- New replication techniques:
  - Shown possible by the classification.
  - Better integration with the communication system.

- Better group communication system
  - Clearer interface with the application.
  - More "hooks" for application optimisations.