Understanding Replication in Database and Distributed Systems

M. Wiesmann  F. Pedone  A. Schiper
Swiss Federal Institute of Technology – Lausanne

B. Kemme  G. Alonso
Swiss Federal Institute of Technology – Zürich

http://lsewww.epfl.ch/~dragon/
Background - Dragon Project

- Joint research project between Zürich and Lausanne
- Cooperation between Database people (Zürich) and Distributed System people (Lausanne)

Goal:
- Build synergy between both communities.

Results:
- Understanding the other side.
- New algorithms & techniques.
- Comparison Framework ➔ this paper.
Difference between Database and Distributed Systems

- Different Goals
- Different Models
- Different Settings
- Different Terminology
Difficulties between Database and Distributed Systems

- Difficult to transpose concepts.
- Difficult to compare techniques.
- Difficult to do synergy.
Goal of the Paper

- We want to compare replication techniques.
- Find what’s common between all techniques.
- Abstract framework.
### Model Mismatch

<table>
<thead>
<tr>
<th></th>
<th>Database Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motivation</strong></td>
<td>Performance</td>
<td>Fault-Tolerance</td>
</tr>
<tr>
<td><strong>What is Replicated</strong></td>
<td>Data</td>
<td>Processes</td>
</tr>
<tr>
<td><strong>Operation Type</strong></td>
<td>Multiple (Transactions)</td>
<td>Single (Remote Invocations)</td>
</tr>
<tr>
<td><strong>System Model</strong></td>
<td>Synchronous</td>
<td>Synchronous Asynchronous etc...</td>
</tr>
<tr>
<td><strong>Consistency</strong></td>
<td>1 copy serializability</td>
<td>Sequential Consistency Linearizability</td>
</tr>
</tbody>
</table>
## Database World Classification

<table>
<thead>
<tr>
<th>Update Location</th>
<th>Eager Primary Copy</th>
<th>Lazy Primary Copy</th>
<th>Eager Update Everywhere</th>
<th>Lazy Update Everywhere</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Propagation</td>
<td>Eager</td>
<td>Lazy</td>
<td>Eager</td>
<td>Lazy</td>
</tr>
</tbody>
</table>
## Distributed System Classification

<table>
<thead>
<tr>
<th>Server Determinism Needed</th>
<th>Server Determinism Not Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td><strong>Passive</strong></td>
</tr>
<tr>
<td>Server Failure <strong>Not</strong> Transparent for the Client</td>
<td></td>
</tr>
<tr>
<td>Server Failure Transparent for the Client</td>
<td></td>
</tr>
</tbody>
</table>

**Not** means it is not transparent for the client.
The Framework

- Key Idea: Find what is common in all techniques.

- Neutral Point of view.

- Abstract Framework.

- Decomposes techniques in logical steps.

- Emphasise Logical Structure.

- Hide Model differences.
Typical Execution

Replica 1  Replica 2  Replica 3

Client

Client

time
Execution in the Framework

Understanding Replication in Database & Distributed Systems
The Phases

- Each algorithm is decomposed in 5 phases.
- Steps are optional.
- Steps can be looped.
- Steps can be reordered.
Client Request Phase

- Client submits operation to the system
- Client either contacts
  - The whole group
  - One representative of the group
The Servers try to find an order to execute the operations.

This phase depends a lot of the model
- Dependencies and conflicts change with the model.
The actual execution of the operation

- Execution can happen on one or more replicas

- The applying of update is typically done during the agreement coordination phase.
Agreement Coordination Phase

- Replicas make sure that the outcome is correct
- Replicas make sure that they all can commit the changes

- Two “coordination” phases
  - one before ➔ server coordination
  - one after ➔ agreement coordination
Response to the Client

- The system sends the response to the client
- The response can come from any replica (or all)
- Gives to the client the outcome of the operation
Database example: Eager Update Everywhere Distributed Locking

Client contacts Server
RE

Servers Coordination SC

Transaction Executes EX

Agreement Coordination AC

Server answers Client END

Replica 1 Replica 2 Replica 3

Broadcast

Update

Update

Update

2 Phase Commit
Distributed System example: 
Active Replication
# Comparison

<table>
<thead>
<tr>
<th></th>
<th>Distributed Systems</th>
<th>Database Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Active</td>
<td>Passive</td>
</tr>
<tr>
<td><strong>RE</strong></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>SC</strong></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>EX</strong></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>AC</strong></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>END</strong></td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

One operation transactions - Eager Techniques
Conclusion

- Frameworks make comparison possible
- Puts into evidence model differences between worlds
- Lot of common algorithms
- Basis for quantitative comparison
- For more information:

  http://lsewww.epfl.ch/~dragon/