Database Replication Techniques: A Three Parameter Classification

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)
  - Common Interest ➔ Replicated Databases

- Goal:
  - build synergy between both communities.

- Results:
  - Understanding the other community.
  - New Replication algorithms & techniques.
  - Exploring Replication Techniques ➔ this paper.
Replicated Database

- One Logical Database
- \(N\) logical replicas
- Network connects Replicas
- All replica are synchronous (eager replication)
- Clients connect to the logical Database.
- Replicas processes transactions (ACID properties)
Database Replication Techniques
Eager vs. Lazy

- Two ways to replicate a database: eager & lazy
  - Eager       Replica are kept coherent
  - Lazy        Replica might diverge (violates ACID)

- Eager is traditionally considered too costly [Gray 96]

- Search more efficient eager replication.

- Systematically explore solution space
  ➔ Classification
Classification

- Many Classifications tentatives
  - [Wiesmann & All ERSADS’99]

- Classification must:
  - Be applicable for all considered techniques
  - Group together techniques with same communication patterns
  - Group together techniques with similar network load.

- Permits to:
  - Systematically Explore Parameter Space
  - Find out requirements for each Element in Parameter Space

- 3 Orthogonal Criterion
  - Server Architecture, Server Interaction, Transaction Termination
Parameter ①
Server Architecture

- Reflects the way the replica are organized in the System.
- Where clients send their requests.
Parameter ②  
Server Interaction

- Represents the traffic generated for one transaction

<table>
<thead>
<tr>
<th>Constant number of interactions</th>
<th>$O(1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear number of interactions</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>$n = \text{number of operations}$</td>
<td></td>
</tr>
</tbody>
</table>
## Parameter ③

Transaction Termination

- Determines how the outcome of transaction is decided:

<table>
<thead>
<tr>
<th>Voting</th>
<th>There's No Atomic Commitment Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>NonrVoting</td>
<td>There's No Atomic Commitment Phase</td>
</tr>
</tbody>
</table>
The Classification
Requirements for Each Class

- Each class contains many replication techniques
- Each class implies some requirements:
  - On the communication primitives (order, reliability, uniformity)
  - On the database system (determinism)
- A few Examples:
  - Present the general protocol for three classes
Non Voting – Constant Interaction – Primary Copy

“Cold Standby” Primary Copy

- Typical Commercial configuration [Gray 93 Transaction Processing…]
- Needs FIFO Broadcast
- Cold Standby (back-up might not have applied changes)
- 1 Safe - 2 Safe with certain constraints on the communication primitives
Update Everywhere – Linear Interactions – Voting

- Classical form of replication
  - Read One Write All technique [Bernstein 87 Concurrency Control…]
  - Each Operation is Sent to All replicas
  - The Transaction is terminated by 2PC
Update Everywhere – Non Voting – Constant Interaction

- Needs a known determinism point.
- Needs total order broadcast.
- If the $dp$ at the start ➜ Active Replication [Schneider 90 ACM Survey]
- If the $dp$ after start ➜ Certification based replication [Kemme & all ICDCS’ 98 - Pedone & all EuroPar 98]
- If the $dp$ in the middle ➜ Possible - never proposed.
The Issue of Determinism

- How do we express the constraints on the server?
- One important issue *determinism*.
- We need something more than a boolean.
- Notion of Determinism Point (*dp*).
- Execution from this point is deterministic.
Conclusion

- Classification helps:
  - Explore Solution Space.
  - Understand the relation between existing techniques.
  - Understand the requirements for communication and database system.
  - Give Basis for comparing the techniques (for instance by simulation).

- A case for eager replication
  - Lazy replication usually compared to expensive eager replication.
  - Should be compared to comparable eager replications.
  - Non-voting - Update everywhere techniques are very promising.