Continuous Consistency Management in Distributed Real-Time Databases with Multiple Writers of Replicated Data

Sanny Gustafsson, Sten F. Andler
Distributed Real-Time Systems Group
DeeDS Project
Work overview

• Replication protocol for DRTDB
  – Prototype: DeeDS
  – Distributed whiteboard architecture
    • Virtual full replication
  – Keep up with real-time progress of environment
    • Local timeliness, eventual global consistency
  – Support application tolerance of inconsistencies
• Presentation focus: conflict management
  – Continuous convergence
Example: introduction

Collaboration data:
- Table of survivors
  - ID
  - Position
- Route table
  - Waypoints
PRiDe
Protocol for Replication in DeeDS

- Simplifying assumptions:
  - Predictable message delivery time
  - No local overloads
  - Single-update transactions
  - Static set of nodes
Example: concurrent updates

<table>
<thead>
<tr>
<th>ID</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>(32,15)</td>
</tr>
</tbody>
</table>

Helicopter A (camera)

Helicopter B (camera)

Helicopter C (heat sensor)

survivor

rescue team

HQ

Propagation and integration
Conflict management

- Conflict detection
  - Version vectors
- Update qualification
  - Conflict sets, generations
- Conflict resolution
  - Forwards resolution only
  - Application-specific policies

Version vectors, ex.

A

\[
\begin{align*}
S & \quad <1,1,3> \\
S' & \quad <2,1,3>
\end{align*}
\]

B

\[
\begin{align*}
S & \quad <1,1,3> \\
S'' & \quad <1,1,4>
\end{align*}
\]

C

\[
\begin{align*}
S & \quad <1,1,3> \\
\end{align*}
\]

conflict
# Conflict sets & generations

- Conflict set: all *non-stable* updates to an object
- Generation: unit of conflict resolution
- Ensures deterministic resolution/update pruning
  - Assumes ordered messages between node pairs

<table>
<thead>
<tr>
<th>Gen. 1</th>
<th>Updates from A</th>
<th>Updates from B</th>
<th>Updates from C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add (S1,(32,15)) &lt;1,0,0&gt;</td>
<td>Add (S1,(30,16)) &lt;0,1,0&gt;</td>
<td></td>
</tr>
<tr>
<td>Gen. 2</td>
<td>Set (S1,(32,16)) &lt;2,1,0&gt;</td>
<td>Set (S1,(34,16)) &lt;1,2,0&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set (S1,(34,16)) &lt;3,2,0&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. 3</td>
<td>Set (S1,(37,16)) &lt;4,2,0&gt;</td>
<td>Set (S1,(36,12)) &lt;3,3,1&gt;</td>
<td></td>
</tr>
<tr>
<td>Gen. 4</td>
<td></td>
<td>Set (S1,(35,12)) &lt;1,2,1&gt;</td>
<td></td>
</tr>
</tbody>
</table>

- Add (S1,(31,15.5))
- Set (S1,(33,16))
- Set (S1,(34,16))
- Set (S1,(35,12))
- Set (S1,(36,12))
Example: conflict management

• Conflict/resolution examples:
  – Conflicting add operations
    • Resolution: merge or allow
  – Conflicting position updates
    • Resolution: merge, possibly weighted by confidence
  – Conflicting routing orders (planner/HQ)
    • Resolution: use confidence; prioritize HQ orders

• Application tolerance
  – Can exploit maximum-information position/routing data
  – Stable values can be used to, e.g., log movement
Protocol properties

• Local predictability
  – No global locks or commit protocols
  – No transaction rollback or update undo/redo

• Eventual global consistency
  – Deterministic update ordering & conflict resolution
    • (Real-time network for bounded-time stabilization)

• Support for application tolerance
  – Maximum-information and stable values
  – Future work: bound on deviation, confidence metric
Extensions

- Unbounded replication time/partitions
  - Stabilization messages
  - Reconciliation protocol
- Multi-update transactions
  - Transaction-level conflict sets
- Overload management
  - Lower priority of integration/propagation transactions
  - Reconciliate as necessary
Conclusions

- **Continuous convergence protocol**
  - Local predictability, eventual global consistency
  - Forward conflict resolution using conflict sets, generations

- **Application tolerance**
  - Maximum-information values
  - Stable values
  - Support for application-specific conflict resolution

- **Suitable for applications that can trade off strict consistency for predictability and progress**