Serializability with Snapshot Isolation under the Hood

Mihaela Bornea ¹, S. Elnikety ², O. Hodson ², A Fekete ³

¹IBM Research ²Microsoft Research ³University of Sydney
Motivation

Concurrency Control

Replication Model

Readset Certification

Evaluation

Conclusions
Transaction Processing in Replicated Databases

- **Database Replication:**
  - Higher availability & better performance
  - Maintaining consistency is challenging

- **State of the Art:**
  - GSI Replicated Databases.
  - Each replica uses Snapshot Isolation (SI).

- **Goal:**
  - Global One Copy Serializability.
  - Overall Isolation level stronger than the one of individual components.
  - The replicated system keeps its performance.
Transaction Isolation

- Isolation is a correctness criterion.
- Concurrency in the system.
- Multiple levels of isolation:
  - Snapshot Isolation.
  - Serializability.
Snapshot Isolation

- Multi-version concurrency control technique.
- Important
  - Used by Oracle, SQL Server, Postgres.
  - Sometimes the strongest isolation level available.
- Attractive performance
  - Read-only transactions never block or abort.
  - Read-only transactions do not block update transactions.
  - Updates might abort. Certification needed.
    - checks for ww conflicts.
## Anomaly under SI

<table>
<thead>
<tr>
<th>Time</th>
<th>R(X,Y) X=50 Y=50</th>
<th>W(X) X=−40</th>
<th>R(X,Y) X=50 Y=50</th>
<th>W(Y) Y=−40</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **X, Y balance of two bank accounts.**
- **$T_1$ and $T_2$ withdraw 90E from X and Y**
- **Logic:** $X + Y > 0$
Serializability

- The strongest DB isolation level.
- Illusion that transactions execute serially.
- Programmers want it:
  - As if there is no concurrency.
- Commonly implemented with 2PL.
  - expensive to achieve.
Serializability under SI

- **Centralized Database**
  - Modify database engine, SSI.
  - Use Fekete’s work [SIGMOD 2008, best paper]

- **Replicated Databases**
  - Open question.
  - No modification of the database engine.
GSI Replicated Database
SQL Transaction Model

A. SELECT $\textit{expr\_list}$ FROM $R_i$ WHERE $\textit{pred}(R_i)$
B. INSERT INTO $R_i$ VALUES ($\textit{values}$)
C. UPDATE $R_i$ SET $\textit{attr\_values}$ WHERE $\textit{pred}(R_i)$
D. DELETE FROM $R_i$ WHERE $\textit{pred}(R_i)$
E. SELECT $\textit{agg}(\textit{attr})$ FROM $R_i$ WHERE $\textit{pred}(R_i)$
   GROUP BY $\textit{group\_attr}$
   HAVING $\textit{pred}(\textit{agg}(\textit{attr}))$
F. SELECT $\textit{attr\_list}$ FROM $R_1\ldots R_i\ldots R_n$
   WHERE $\textit{pred}(R_1)\ LOP \ldots LOP\ \textit{pred}(R_i)\ LOP\ \ldots LOP\ \textit{pred}(R_n)$
   $\ LOP\ \textit{pred}(\textit{attr}_{i,j},\ \textit{attr}_{i,j})$
G. SELECT $\textit{attr\_list}$ FROM $R_1\ldots R_i\ldots R_n, \textit{SQ}$
   WHERE $\textit{pred}(R_1)\ LOP \ldots LOP\ \textit{pred}(R_i)\ LOP\ \ldots LOP\ \textit{pred}(R_n)$
   $\ LOP\ \textit{pred}(\textit{SQ})$
H. SELECT $\textit{attr\_list}$ FROM $R_1\ldots R_i\ldots R_n$
   WHERE $\textit{pred}(R_1)\ LOP \ldots LOP\ \textit{pred}(R_i)\ LOP\ \ldots LOP\ \textit{pred}(R_n)$
   $\ LOP\ \textit{pred}(\textit{attr}_{i, SQ})$
1SR Needs Readsets

- Snapshot Isolation (SI) → Generalized Snapshot Isolation (GSI)
  - Certify Writeset
- Serializability → One Copy Serializability (1SR)
  - Certify Writeset
  - Certify Readset
- Yes, we have a proof :)!
Writesets

- The Writeset contains modified tuples
- Introduced by UPDATE, INSERT and DELETE
- Includes both new and old tuple values
- All Writesets are managed at the Certifier.
- Writeset certification is required by both GSI and 1SR
  - checks if concurrent transactions modify the same item.
- It is well knows how to manage the Writesets
Readsets

- The Readset contains read tuples.
- Introduced by SELECT, UPDATE, INSERT and DELETE.
- Readsets certification is required by 1SR.
  - checks if a transaction reads data modified by concurrent transactions.
- Readset identification is challenging:
  - never done in replicated setting.
So far ...

- We introduced SI.
- Sometimes SI is not enough!
- Serializability needed:
  - Keep the nice properties of SI.
  - Open Problem for replicated databases:
    - Readset management is difficult!
Main Contribution - Readset Management

- Framework to manage the Readsets
- Observation: each SQL statement has a predicate.
  - The Readset is a list of predicates.
  - Readset certification requires predicate evaluation.
Certifier Design

- The Certifier manages:
  - persistent log.
  - main memory database, CertDB.
- The log is used for durability.
- CertDB is used to certify update transactions.
- CertDB maintains the Writeset of recently committed transactions.
- CertDB schema:
  - the replicated schema.
  - commit version attribute.
Readset Certification

- **Intuition:**
  - Ensures that if the transaction executes on the latest version it would read the same values.

- **Implementation:**
  - Replica identifies the Readset:
    - Extracts the predicate of each SQL statement.
  - Replica expresses the readset as certification queries.
  - The certification queries are evaluated on CertDB.
  - Empty conflict set indicates serializable execution.
Concurrent Transactions

- Snapshot versions at originating replicas.
- Commit version of a transaction.
- CertDB contains the writesets and committed version.
- Consider a transaction T:
  - \( \text{version} > \text{snapshot}(T) \)
Readset for SELECT Statements

Transaction Queries

A. SELECT expr_list FROM Ri WHERE pred(Ri)

Certification Queries

A. SELECT * FROM Ri WHERE pred(Ri) AND
   version > snapshot(T)
Readset for UPDATE Statements

Transaction Queries

B. INSERT INTO $R_i$ VALUES ($values$)

C. UPDATE $R_i$ SET $attr_values$ WHERE $pred(R_i)$

D. DELETE FROM $R_i$ WHERE $pred(R_i)$

Certification Queries

B. SELECT * FROM $R_i$ WHERE $pk = @pk$ AND $version > snapshot(T)$

C. SELECT * FROM $R_i$ WHERE $pred(R_i)$ AND $version > snapshot(T)$

D. SELECT * FROM $R_i$ WHERE $pred(R_i)$ AND $version > snapshot(T)$

Certifying the Readset also detects ww conflicts.
Experimental Study

- Impact of providing 1SR vs. GSI:
  - Lower throughput and higher response time
  - Higher abort rate
- Replicated system with 8 replicas
- TPC-W
Workload

- TPC-W benchmark:
  - Web application (online book store).
  - Database schema consists of 10 tables.
  - Database size: 800 MB.
  - 13 transaction templates.
  - Ordering Mix (50% updates).
  - Browsing Mix (5% updates).

- Metrics:
  - Transactions per minute (TPM).
  - Response time.
  - Abort rate.
Scaling of SGSI with Replication Degree

Throughput of TPC-W Shopping Mix (20% updates)

Resp. Time of TPC-W Shopping Mix (20% updates)
Comparing SGSI to GSI

Scalability of TPC-W Shopping Mix (20% updates)

Certifier CPU Utilization TPC-W Shopping Mix (20% updates)

M. Bornea, S. Elnikety, O. Hodson, A. Fekete
IBM Research, Microsoft Research, Microsoft Research, University of Sydney

Serializability with Snapshot Isolation under the Hood
Sensitivity to Update Transaction Ratio

SGSI Throughput of TPC-W Mixes.

SGSI Response Time of TPC-W Mixes.

M. Bornea, S. Elnikety, O. Hodson, A. Fekete
IBM Research, Microsoft Research, Microsoft Research, University of Sydney

Serializability with Snapshot Isolation under the Hood
Abort Analysis via SmallBank

Conflicts, %

ww conflicts
rw conflicts
pivots

M. Bornea, S. Elnikety, O. Hodson, A. Fekete
IBM Research, Microsoft Research, Microsoft Research, University of Sydney

Serializability with Snapshot Isolation under the Hood
Conclusions

We introduced SGSI:
- 1SR in replicated databases.

Built a replicated system prototype.

Evaluated SGSI performance:
- SGSI is practical.
- Moderated cost for small degree of replication.
- Performance and scaling is comparable with GSI.
Readset for Joins

Transaction Queries

F. \text{SELECT} \ attr\_list \\
\text{FROM} \ R_1...R_i...R_n \\
\text{WHERE} \ pred(R_1) \ LOP ...LOP \ pred(R_i) \ LOP ...LOP \ pred(R_n) \\
LOP \ pred(attr_{i,j}, attr_{i,j})

Certification Queries

for each relation \( R_i \)

F. \text{SELECT} \ * \ \text{FROM} \ R_i \ \text{WHERE} \ version > snapshot(T)

- An upper-set of the Readset is certified.
- False aborts.
Data Managed at Certifier

- Accuracy depends the data maintained at the Certifier.
- False aborts:
  - not enough information to evaluate the Readset
- Solution:
  - manage a copy of relations at the Certifier.
  - physical design tuning problem.
Extended CertDB

- Each data item has several instances.
- New instance: UPDATE, INSERT.
- Expired: UPDATE, DELETE.
- Each copy relation is augmented with $V_{Start}$ and $V_{End}$.
- $V_{Start}$ and $V_{End}$ determine:
  - update predicate: $upd(R_i)$.
  - visibility predicate: $vis(R_i)$. 
Extended Certification

Transaction Queries

F. SELECT attr_list
   FROM R₁...Rᵢ...Rₙ
   WHERE pred(R₁) LOP ... LOP pred(Rᵢ) LOP ... LOP pred(Rₙ)
   LOP pred(attrᵢ,j, attrᵢ,j)

Certification Queries

SELECT * FROM R₁C...RᵢC...RₙC
WHERE (query_pred)
AND (upd(R₁C) ... OR upd(RᵢC) ... OR upd(RₙC))
AND (vis(R₁C) ... AND vis(RᵢC) ... AND vis(RₙC))