Database Replication with Oracle 11g and MS SQL Server 2008

Overview

- Introduction
- Theory
  - Conflict of goals
  - Consistency
- Replication strategies
  - Propagation strategies
  - Ownership strategies
  - Conflicts
- Mobile database replications
- Experiments
  - MS SQL 2008
  - Oracle 11g
- Questions / Discussion
Introduction – Seminar objectives

• Theme: Database replication with Oracle 11g and MS SQL Server 2008
• Author: F. Bolfing; Advisor: S. Keller
• Theory:
  – Architectures (incl. mobile DB’s)
  – Replication types
  – Algorithms and concepts
  – Consistency and conflict resolution
• Practical:
  – Implementation in Oracle 11g and MS SQL Server 2008
  – Evaluation with own implementation and test data
• Details:
  – Limitation on relational databases
  – Important paper: Jim Gray, “The Dangers of Replication and a Solution”, Weblink

Introduction – What is replication?

• Generally: Availability of data on different nodes
• Special case: Replication vs. ACID

• Why replicate?
  – Data availability
  – Fault tolerance
  – Performance by having better response times
  – Balance the load distribution
  – Use of offline applications

• Negative aspects:
  – Higher system complexity
  – Disk space
  – Update effort
Introduction – Today’s use

- Disaster recovery
- Data distribution to other locations (geographical distribution)
- Data consolidation from other locations (data warehouses)
- Bi-directional exchange of data with other locations
- High-availability (HA) clusters
- Load-balancing clusters
- Synchronisation of mobile clients

Theory – Conflict of goals

- Main requirements of replication
  - Availability
  - Performance
  - Consistency
- Requirements in conflict to each other (conflict of goals triangle)
Theory – Consistency (1)

- Main problem in replication: Ensure consistency in every copy of the database
- Consistency can be classified into strong and weak consistency

Theory – Consistency (2)

Strong Consistency:

All replicas have to have exactly the same content before any transaction can be performed

- Not easy in an unreliable network like the internet
- High latency if you have high amount of replicas
- Only suitable for systems with few replicas
Theory – Consistency (3)

Weak Consistency:

With weak consistency it is not necessary that all of the replicas have to have the same content to ensure transactions

- You need to ensure that the replicas sometime converge to a consistent state
  - Mostly bounded to a specific period of time
  - Use synchronization variables
  - Inside of a synchronization variable you have a critical section and there you allow inconsistent data
  - Inside of the critical section it is prohibited to do concurrent read/write which is guaranteed with locks

Replication strategies – Overview (1)

Classification of replication strategies (1):
- Classification based on differentiation of the ownership

<table>
<thead>
<tr>
<th>Ownership</th>
<th>Lazy</th>
<th>Eager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>N transactions</td>
<td>One transaction</td>
</tr>
<tr>
<td></td>
<td>N object owners</td>
<td>N object owners</td>
</tr>
<tr>
<td>Master</td>
<td>N transactions</td>
<td>One transaction</td>
</tr>
<tr>
<td></td>
<td>one object owner</td>
<td>one object owner</td>
</tr>
</tbody>
</table>
Replication strategies – Overview (2)

Classification of replication strategies (2):
- Classification based on the definition of their consistency

- Propagation strategies (1)
  - Synchronous replication (eager)
    - Update of all replicas before committing the transaction (two phase commit)
    - There exists no inconsistencies
    - Conflicts doesn’t exist (results in waiting loops or deadlocks)
    - Updates may fail due to the fact of existing deadlocks to prevent serialization errors
Replication strategies – Propagation strategies (2)

Asynchronous replication (lazy)
- No update of all replicas at the same time
- Propagation of the updates to the other nodes takes place after commit (by synchronizing)
- Synchronization happens either after a certain interval or at a certain point of time
- Increasing of performance
- Conflicts are inevitable

Replication strategies – Ownership strategies (1)

Most common ownership strategies:

1. Merge replication (update anywhere / multi-master):
   - Updates can be performed anywhere in the system
   - Weak consistency and asynchronous propagation
   - Timestamps / synchronization variable to prevent conflicts
Replication strategies – Ownership strategies (2)

2. **Primary copy (master-slave):**
   - Every update has to be sent to the primary copy node (publisher)
   - The updates can be propagated to the subscribers either by eager or lazy replication
   - Synchronization with version number check

Replication strategies – Ownership strategies (3)

3. **ROWA (Read One Write-All):**
   - Most used strategy in synchronous replication
   - Reading actions are preferred and allowed at any node
   - A replica can only be updated if all the others were locked
   - If the lock is accepted by any node the transactions happens on all nodes simultaneously
Database Replication with Oracle 11g and MS SQL Server 2008

Replication strategies – Conflicts (1)

• Conflicts only exist in asynchronous replication
• Types of conflict:
  – Update conflicts
  – Uniqueness conflicts
  – Delete conflicts
  – Ordering conflicts
• Conflicts can be handled by automated conflict handler or by hand

Replication strategies – Conflicts (2)

Update conflict:
• Replication updates on a data object which is in conflict with another update at the same time
• Example: Two different updates want to update the same data record at different replicas at almost the same time
• Can be detected by comparing the two timestamps

Uniqueness conflict:
• Exists when the replication of an update violates entity integrity like PK or unique constraint
• Example: Two transactions which are originated from different nodes insert data which uses the same PK
**Replication strategies – Conflicts (3)**

**Delete conflict:**
- Occurs if two transactions from different nodes happen, with one transaction deleting a data record and the other wants to update or also delete it

**Ordering conflict:**
- Can occur in replication environments with more than 3 master sites
- Example: Propagation to master site x is blocked -> updates can continue on other master sites
- Resume of propagation to available master site x can happen in different order than to the other master sites

---

**Mobile database replications (1)**

- Nowadays a lot of applications demand for availability and also mobility
- Mobile databases are not attached to a specific location and therefore their position always changes
- Synchronous replication is not suitable where the nodes are often disconnected
- Mobile databases need asynchronous replication
- Best practice solution: Two-tier replication scheme
Mobile database replications (2)

Two-tier replication (1):
• Two different kind of nodes:
  – Mobile nodes:
    • Mostly disconnected
    • Store a replica of the database and may create tentative transactions
    • Every mobile node is mastered at some base node
  – Base nodes:
    • Store a replica of the database
    • Are usually the master of data items
• Two different kind of data items:
  – Master version:
    • True master at the object master
  – Tentative version:
    • Most recent value created by local updates

Mobile database replications (3)

Two-tier replication (2):
• Two different kind of transactions:
  – Base transactions:
    • Can only be performed at master data
    • Produce again master data
    • There can be at most 1 mobile node and several base nodes involved
  – Tentative transactions:
    • Use only local tentative data
    • Not processed as a base transaction until the mobile node is reconnected to the base node
Experiments - Scenario

Goals:
• Database replication with the specific product (MS SQL Server 2008 and Oracle 11g)
• MS SQL Server 2008: Merge replication implemented
• Oracle 11g: Multi-Master Replication with 1 Master
• Conflict behaviour of the implementation
• Mobile databases are not considered

Experiments - MS SQL Server 2008 (1)

Replication in MS SQL Server 2008:
• Snapshot publication:
  – A copy of the entire database is published out to the subscribers. Further changes to the snapshot are not tracked.
• Transactional publication:
  – Transaction log is monitored for changes and those (the transactions) are sent to the distributor. The subscribers either pull the changes or they are pushed to them.
• Merge publication:
  – Publisher and Subscribers can update the published data independently. Changes are merged periodically. Changes made at the publisher or the subscribers are tracked and periodically synchronized between them. If the same data has been modified differently, the synchronization will result in a conflict
Experiments - MS SQL Server 2008 (2)

Test setup in MS SQL Server 2008:
- Database tables: AdventureWorks
- Only table Person.Address from database AdventureWorks is replicated
- Implementation: Merge replication

Experiments - MS SQL Server 2008 (3)

3 different conflict resolutions in MS SQL Server 2008:
- The first method states that the publisher always wins
- The second states that the subscriber always wins
- In the third case, you are asked to resolve each conflict individually through a SQL Server 2008 interface.
Experiments - MS SQL Server 2008 demo

MS SQL 2008 demo

Experiments – Oracle 11g (1)

Types of replication in Oracle 11g (Advanced replication):
• 4 different types
  ▶ Mult-master Replication
  ▶ Updateable Materialized View Replication
  ▶ Read-only Materialized Views
  ▶ Create Template for mass deployment of Materialized View sites
Types of replication in Oracle 11g (Advanced replication):

- Multi-master Replication
  - Applications can update any replicated table at any site in a multi-master configuration
  - Asynchronous and synchronous propagation possible

Types of replication in Oracle 11g (Advanced replication):

- Materialized View Replication (Snapshot)
  - A materialized view are tables based on queries
  - Often read only
Experiment – Oracle 11g (4)

Types of replication in Oracle 11g (Advanced replication):
- Multi-master and Materialized View Hybrid Configurations
  - Hybrid configurations can have any number of master sites and multiple materialized view sites for each master.

Conflict resolution in Oracle 11g:
- Always the receiving site performs conflict detection and resolution:
  - Update conflicts: Detected by comparing the differences between old value of replicated row and current value at receiving site
  - Uniqueness conflicts: Detected if a uniqueness constraint violation occurs during INSERT or UPDATE
  - Delete conflict: Detected if the primary key for the updated row does not exist
- Inbuilt conflict resolutions:
  - Latest (earliest) time stamp: Most recent (oldest) update wins
  - Overwrite: Overwrites current value with new value
  - Discard: Ignores the new value
  - Additive: Difference of the two values is added to the current value (current value = current value + (new value - old value))
Experiments – Oracle 11g (6)

Test setup in Oracle 11g (Advanced replication):
• Advanced replication
• Multi-master replication with one master site
• User repadmin

```sql
-- Setting up a multi-master replication environment
-- Connecting to root user and granting access to user
CONNECT system with password identifying user sys

-- Creating replication administrator at site acc11_win-c9ri3psxxova
-- Creating user MNSUSER at site acc11_win-c9ri3psxxova
GRANT SYSDBA, CREATE SESSION TO "MNSUSER";

-- Setting up when-pushing to MNSUSER at site acc11_win-c9ri3psxxova
EXECUTE BULKLOAD_related_to "MNSUSER";

-- Setting current table to "EMAIL"
-- setting lock table to "EMAIL"
-- set current query to "EMAIL"
-- set enable map to "EMAIL"
-- setting push schedule of a database link
```

Experiments – Oracle 11g (7)

Test setup in Oracle 11g (Advanced replication):
• Push schedule of a database link

```sql
/* Scheduling link MNS12_WORLD at site acc11_win-c9ri3psxxova... */
BEGIN
  DBMS_REPCATv.SCHEDULE_PUSH(
    destination => 'MNS12_WORLD',
    interval => '1/5',
    next_date => TO_DATE('Dec 16, 2008 19:01:41 OK', 'DD-MON-YYYY HH24:MI:SS UT'),
    stop_on_error => false,
    delay_seconds => 0,
    parallelism => 1);
END;
/
Experiments – Oracle 11g (8)

Test setup in Oracle 11g (Advanced replication):
Configure master groups
• Replicated schema: SCOTT
  • Master object: orcl1

```sql
EXECUTE DBMS_REP.CHECK_SUBSYSTEM_RS
  WHERE (GROUP = 'SCOTT_RS', SUBSYSTEM = 'SCOTT_RS');
END;
EXECUTE DBMS_REP.CHECK_MASTER_RS
  WHERE (GROUP = 'SCOTT_RS', SUBSYSTEM = 'SCOTT_RS',
          MASTER_OBJECT = 'SCOTT',
          ORACLE_NAME = 'ORCL1',
          DATABASE_NAME = 'ORCL1',
          PRIMARY_RS = 'SCOTT_RS',
          SECONDARY_RS = 'SCOTT_RS',
          PROPAGATION_MODE = 'NORMAL');
END;
EXECUTE DBMS_REP.CHECK_MONITOR_RS
  WHERE (GROUP = 'SCOTT_RS', SUBSYSTEM = 'SCOTT_RS');
END;
EXECUTE DBMS_REP.REVERT_MASTER_RS
  WHERE (GROUP = 'SCOTT_RS', SUBSYSTEM = 'SCOTT_RS');
END;
EXECUTE DBMS_REP.REVERT_MONITOR_RS
  WHERE (GROUP = 'SCOTT_RS', SUBSYSTEM = 'SCOTT_RS');
END;
```

Experiments – Oracle 11g (9)

Test setup in Oracle 11g (Advanced replication):
• Table SALGRADE

```sql
CREATE TABLE SALGRADE
  ( GRADE NUMBER,
    LOCAL NUMBER,
    SISAL NUMBER );
INSERT INTO SALGRADE VALUES (1,200,1200);
INSERT INTO SALGRADE VALUES (2,120,1200);
INSERT INTO SALGRADE VALUES (3,120,1200);
INSERT INTO SALGRADE VALUES (4,120,1200);
INSERT INTO SALGRADE VALUES (5,120,1200);
COMMIT;
```
Experiments – Oracle 11g demo

Oracle 11g Demo

Conclusion

- Replication is a useful tool to increase the performance of a database
- Asynchronous replication can create conflicts
- Replication is used to keep mobile databases up to date
- Conflict resolution is not trivial
- To much replication increases the system complexity
Bibliography