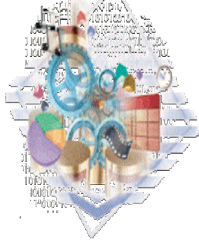




DB2 UDB



# Discussion of *Middle-Tier Database Caching for e-Business*

Joon Wong  
DB2 UDB System Testing  
IBM Toronto Lab

October 2, 2002



DB2 UDB



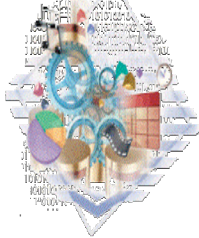
## Outline

- ↗ Limitations and Restrictions
- ↗ Counter Examples
- ↗ Performance Issues
- ↗ Experimental Results
- ↗ Misc. Comments
- ↗ Tips and Tricks
- ↗ Bibliography

October 2, 2002



DB2 UDB



## Limitations and Restrictions

### ↪ Restriction for Replication on EEE

- ◆ "You can capture changes on DB2 Enterprise - Extended Edition *only* if the source table is nonpartitioned and it resides on the catalog node. Any replication control tables must also be nonpartitioned and reside on the catalog node." - Replication Guide and Reference.
- ◆ If DBServer is a MPP system, Replication may not work well.

October 2, 2002



DB2 UDB



## Limitations and Restrictions - Cont.

### ↪ Replication : Source table must have a Primary Key(PK) or Unique Index

- ◆ Source table refers to the table in DBServer.
- ◆ This limitation implies that all of the cached table must have PK or Unique Index

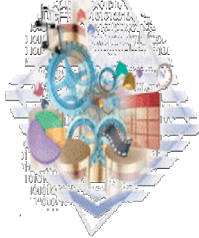
### ↪ Federated : Nickname cannot be locked

- ◆ <DB Cache>
  - ✦ db2 create nickname tab1 for remServ.jcwong.tab1
  - ✦ DB20000I The SQL command completed successfully.
  - ✦ db2 lock table tab1 in share mode
  - ✦ SQL0156N The name used for this operation is not a table. SQLSTATE=42809

October 2, 2002



## DB2 UDB



## Limitations and Restrictions - Cont.

- ↪ **Federated : Cannot create a nickname on a table with long varchar column.**
  - ◆ <DB Server>
    - ✦ db2 create table tab2 (i long varchar)
    - ✦ DB20000I The SQL command completed successfully.
  - ◆ <DB Cache>
    - ✦ db2 create nickname tab2 for remServ.jcwong.tab2
    - ✦ SQL3324N Column "I" has a type of "LONGVAR" which is not recognized.
  - ◆ No longer a restriction in DB2 UDB V8.1

October 2, 2002



## DB2 UDB



## Limitations and Restrictions - Cont.

- ↪ **Federated : A problem in lob for federated.**
  - ◆ <DB Server>
    - ✦ db2 create table tab3 (i blob(10))
    - ✦ DB20000I The SQL command completed successfully.
  - ◆ <DB Cache>
    - ✦ db2 create nickname tab3 for remServ.jcwong.tab3
    - ✦ DB20000I The SQL command completed successfully.
    - ✦ db2 select \* from tab3
    - ✦ SQL1822N Unexpected error code "-351" received from data source "REMSERV".
- ↪ **Originally, I want to show that we cannot create a nickname on table with 10 lob columns.**

October 2, 2002



DB2 UDB



## Counter Examples

### ↪ Update Contention (maybe handled by auto-passthru but doubtful)

- ◆ Supposed that two users update a value simultaneously, one can override the others, which lead to data inconsistency.
- ◆ Proper method : lock table in exclusive mode, then update the values accordingly.
- ◆ Since a nickname cannot be locked, one must count on auto-passthru. How is this implemented?

October 2, 2002



DB2 UDB



## Counter Examples - Cont.

### ↪ Lob Table Problem

- ◆ Supposed that this is a table with ten blob columns.
- ◆ We cannot create nickname on this table.
- ◆ We cannot cache (replicate) this table.
  - ✦ Cannot create a PK on a lob column.
  - ✦ Cannot create a Unique Index on a lob column.

### ↪ Rename Table Problem

- ◆ Supposed that a user program tries to rename a table.
- ◆ Nickname fails to work.
- ◆ Cached table fails to work.
- ◆ **Adaptivity Problem.**

October 2, 2002



## DB2 UDB



## Counter Examples - Cont.

### ↳ User Specific Passthru Problem

- ◆ <DB Server>
  - + db2 set passthru farRemSv
  - + DB20000I The SQL command completed successfully.
  - + db2 create table far\_table1 (col1 integer)
  - + DB20000I The SQL command completed successfully.
  - + db2 set passthru reset
  - + DB20000I The SQL command completed successfully.
  
- ◆ <DB Cache>
  - + db2 set passthru remServ
  - + DB20000I The SQL command completed successfully.
  - + db2 set passthru farRemSv
  - + SQL0204N "FARREMSV" is an undefined name. SQLSTATE=42704

### ↳ I am doubtful on how the auto-passthru can solve this User Specific Passthru Problem.

October 2, 2002



## DB2 UDB



## Performance Issues

### ↳ No index is physically created for nickname

- ◆ with "specification only" clause.

### ↳ db2 and websphere running together

- ◆ websphere consumes a lot of memory.
- ◆ db2 will need to compete with websphere for memory.

### ↳ Comparison with other competitor (TimesTen)

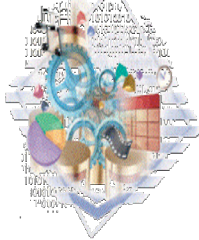
- ◆ To show that DBCache works better than other competitor, it is necessary to compare the throughput and response time with the competitor.
- ◆ TimesTen is in-memory caching approach, whereas DBCache requires DBMS.

October 2, 2002



DB2 UDB

## Performance Issues - Cont.



### Cache whole table

- ◆ May be impractical to cache a large table (e.g. 2GB)
  - ◆ DBCache will work even harder to accommodate the table scan query.
- ◆ Should not cache volatile table. (Many inserts/deletes)
- ◆ No mention of increasing the bufferpool (memory) in DBCache to ease performance gain is a shortsightedness in the paper.

October 2, 2002



DB2 UDB

## Experimental Results



### Bad Assumptions :

- ◆ read-only simulation
- ◆ 3.5GB of data

### Overhead of Adding a Front End Cache

- ◆ Most work is bottleneck in processing the 3.5GB

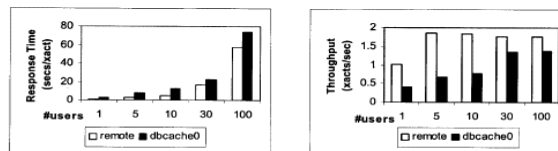


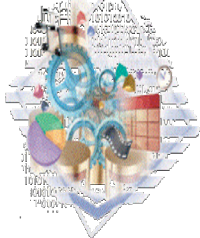
Figure 7: Overhead of Adding a Front End Cache with a 0% Cache Hit Rate

October 2, 2002



DB2 UDB

### Experimental Results - Cont.



#### Cache Effect with Varying Server Workload

- I don't think the result is statistically sounds. There is only one data that shows DBCache may improve performance, I think, is insufficient.

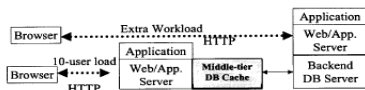


Figure 8: Setup for Varying Server Workload

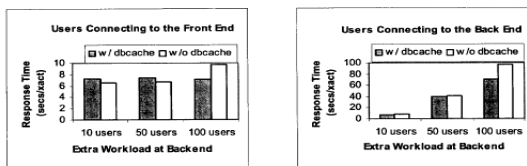


Figure 9: Caching Effect with Varying Server Workload

October 2, 2002



DB2 UDB

### Experimental Results - Cont.



#### Update Propagation Cost

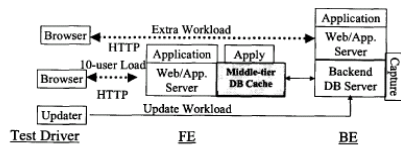


Figure 10: Setup of DPropR with Varying Server Workload

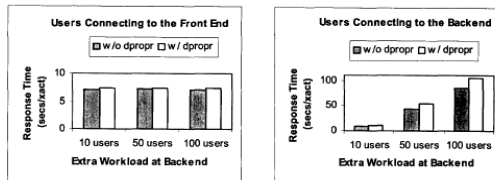
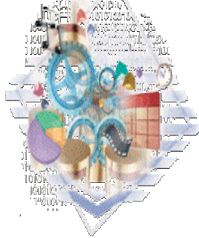


Figure 11: Update Propagation Cost with Varying Server Workload

October 2, 2002



DB2 UDB



## Experimental Results - Cont.

### ↪ Update Propagation Cost - Cont.

- ◆ It is a known fact that running apply and capture will introduce overhead. However, the fact that the overhead is insignificant does not prove DBCache will improve performance.
- ◆ If the experiment is to simulate the browser doing read and write operation, the data may be more interesting to analyze.

October 2, 2002



DB2 UDB



## Misc. Comments

### ↪ **The limitation of only able to simulate 100 users is bad. If they don't use 3.5GB of database for the simulation, I believe that they can simulate >100 users and can produce some concrete results.**

- ◆ DBServer is working hard on table scan.
- ◆ Why not consider using a 500MB database?

### ↪ **Adaptivity**

- ◆ Not adaptive to schema changes
- ◆ Not adaptive to query changes

October 2, 2002





DB2 UDB



## Tips & Tricks

- ↗ **IUD support for nickname in V8**
  - ◆ This may reduce some work for implementing the auto-passthru feature.
- ↗ **Replication : Partitioning Key Change (PKC) - YES**
  - ◆ If the PK is updated, w/o PKC set to YES, duplicate rows in the cached table will happen.
    - ✦ Replication will update the PK with new value. Since the old PK is gone, the new PK will be inserted into the cached table.
  - ◆ If the PK is updated and PKC is YES, replication will split the update operation into DELETE/INSERT pair.
    - ✦ It deletes the old PK row, then inserts the new PK row into the table.

October 2, 2002



DB2 UDB



## Bibliography

- ↗ IBM. DB2 UDB V7 Replication Guide and Reference.
- ↗ IBM. DB2 UDB V7 SQL Reference.
- ↗ Qiong Luo, Sailesh Krishnamurthy, C. Mohan, Hamid Pirahesh, Honguk Woo, Bruce G. Lindsay, Jeffrey F. Naughton. Middle-tier Database Caching for e-Business. SIGMOD Conference 2002.
- ↗ TimesTen. TimesTen Front-Tier. <http://www.timesten.com>

October 2, 2002