IBM SolidDB Rebuttal

Krishna Aluru

“The solidDB solution is to provide a single hybrid database server…”

“

- The main reason to use a hybrid database is to support applications that use a large amount of data by storing tables on disk and caching them in memory as required.
- The application connects to the In Memory database but all requests that are made of tables not in memory are transparently passed through to the On Disk database.
- The hybrid allows cached tables to be accessed quickly, but also allows highly transactional tables to be passed through and avoid the limitations of the in memory database (data loss because additional memory is not persisted)

“In Memory Databases - Tony Bain.” Accessed February 2, 2015.

“The solidDB solution is to provide a single hybrid database server…”

“

- But the tables that are most suitable for caching are not usually the tables that are contributing to the bulk of the performance overhead. The tables causing the bulk of performance overhead are typically the large and highly transactional tables that do not lend themselves to an In Memory model due to their data volatility and strict persistence requirements. While the smaller, and less transactional tables can be cached, often queries issued against the “database” involve both sets of tables (reference data, transactional data) meaning the overhead is still occurring on the On Disk database.

- And the main reason to use an in memory database is to reduce the performance overhead so having a hybrid that does not adequately optimize for highly transactional tables being stored in memory while ensuring data persistence is counter productive to a large extent.

“In Memory Databases - Tony Bain.” Accessed February 2, 2015.

“The solidDB solution is to provide a single hybrid database server…”

“

- While RAM sizes are growing, database sizes are growing just as fast, if not faster, therefore it is unlikely we will reach a stage where RAM sizes will be adequate to cache all mainstream databases anytime soon.

- So, for now, in-memory databases should not be used at all for applications that have huge data sets until there is more innovation to solve the issues currently being faced by the hybrids.

“In Memory Databases - Tony Bain.” Accessed February 2, 2015. ____

“The solidDB solution is to provide a single hybrid database server…”

- Further a DBA or developer has to take extra care designing how data is distributed across solidDB servers so that an application calls on the data co-located on the same server.

- This is because SolidDB’s sweet spot is to optimize results by not having to go across the network or even the across a process in the same server.

- So, if data was not carefully co-located, the performance results would not be very pleasing even if better than a pure on disk database.

http://download.microsoft.com/.../Gigaom_Research_SQL_Server_2014_InMemory_OLTP_White_paper.pdf

“IBM solidDB executes a snapshot-consistent checkpoint…“

- While row shadows allow currently executing checkpoints to continue, they probably cannot process new transactions because of updates caused in the previous transactions, which would need to be reflected in the shadows.

- Row shadows also add a performance overhead because their transmission to storage takes time.

“High Availability Through the Use of Hot-Standby Replication”

“

- Hot Standby is the term used to describe the ability to connect to the server and run read-only queries while the server is in archive recovery or standby mode. This is useful both for replication purposes and for restoring a backup to a desired state with great precision.

- It also refers to the ability of the server to move from recovery through to normal operation while users continue running queries and/or keep their connections open.

- In general, IBM SolidDB implements Hot-Standby Replication very well and also uses a High Availability Controller to automatically detect errors in HSB and recover from them.

“PostgreSQL: Documentation: 9.0: Hot Standby.” Accessed February 2, 2015. ____

“High Availability Through the Use of Hot-Standby Replication”

“

- There is just one limitation with HSB that I wanted to point out but this can probably be fixed and might already be in SolidDB but they do not mention how they deal with it.

- Transactions that use large numbers of sub-transactions will delay the start of read only connections until the completion of the longest running write transaction
full knowledge of transactions is needed before the nodes HSB connects to the database can process new transactions.

“PostgreSQL: Documentation: 9.0: Hot Standby.” Accessed February 2, 2015. ____________________

Benchmarks

- SolidDB is compared with a regular on-disk database (unspecified) in terms of response time and it is clear that it is much better.
- Then it is run through a TATP benchmark, which is very relevant because it has over a million users placing calls and each call needs a lookup or modification to the register.
- It would have been nice to compare the TATP results with those of the same on-disk counterpart to show how well it compares in terms of load scalability.
- It would have also been nice to see comparisons of response times and TATP benchmarking between SolidDB and other in memory databases like Oracle TimeTen and SAP HANA. Especially because SolidDB is a hybrid whereas as the other two are not.

Thank you :)