PayPal

Global Platform & Infrastructure

OLTP Meets Bigdata, Challenges, Options, and Future

Saibabu Devabhaktuni

NoCOUG • November 20, 2015

Agenda

- Database trends for the past 10 years
- Era of Big Data and Cloud
- Challenges and Options
- Upcoming database trends
- Q&A

Scope

- OLTP Very Large Databases (VLDB)
- Databases running custom applications at web scale
- Oracle database focused

Database trends for the past 10 years

- Database sizes are now around 1 to 5 TB
- SGA memory size are typically 10G to 50GB
- Redo generation is reaching rates of 1MB to 5MB per second
- Executions per second trending from 5K to 20K
- DML insert rate per second from 100 to 500

Growth pattern of key DB load profile metrics

- Database size doubling every 2 years
- System memory doubling every 2 to 3 years
- Faster and more cores per processor every year
- Hard disk and flash capacity doubling every 2 years
- Executions per second doubling every 2 years

Era of Big Data and Cloud

- Velocity, volume, and variety of data led to Bigdata and distributed database systems (NoSQL and NewSQL)
- Cloud properties like linear scaling with commodity hardware and 99.99+ availability with self service capabilities propped up NoSQL and NewSQL innovations
- Simplified and append only schema tightly coupled with application
- Security, data management, and governance also tightly coupled with application

Challenges – Data size

- Data lifecycle management (DLM) as best effort option
- Consolidated application use cases need longer data retention, leads to less accountability
- Trade off in performance vs DLM in defining table partition strategy
- Archiving is not transparent to application
- Poor compression ratio for row format data
- Poor DML scalability for column format compressed data

Options – Data size

- Define and enforce DLM policies
- Isolate use cases for the same data, rely on event based processing
- Implement batched delete jobs when partitioning is not an option
- “Instead-Of” trigger based views and DB links for transparent archiving of selective tables
- With all DDL commands being online, reorganize table/index periodically for better compression
- Transparent compression also available at storage layer

Challenges – Memory

- Most data is not in memory, memory to disk access ratio over 1000
- Longer cache warmup time after instance restart
- Bigger SGA means longer startup time
- Multiple copies of same data blocks in buffer cache lead to less hit ratio
- Any full memory scan operations take longer, i.e. datafile offline, tablespace read only, table read only operations, etc.
- Database system level diagnostic operations take longer, i.e. system state dump almost impossible to get on large memory, busy databases

Options – Memory

- Performance tuning and effective data model is still important
- Cache at storage layer to compensate for DB instance cache warmup
- Avoid transaction commit delays to reduce multiple copies of data block
- Test and plan effectively for datafile and tablespace level operations
- Limit dump file sizes and use faster diagnostic operations like short_stack

Challenges – Redo

- Higher redo rate means higher possibility of intermittent slowdowns
- Higher redo rate means longer crash recovery times
- Full supplemental logging for Goldengate generates more redo
- Online object reorganzation generates more redo, interfering with OLTP transactions
- Limited scalability of logical replication
- NOLOGGING operations require significant manual intervention

Options – Redo

- Optimize DML's and indexes to reduce redo generation
- Define recovery target parameters for predictable crash recovery times
- Limit full supplemental logging tables
- Determine redo impact before online reorg of large objects
- Test and determine latency impact on logical replication prior to bulk DML/DDL operations
- Isolate and limit NOLOGGING operations

Challenges - Executions

- Reparse storm after any cursor invalidations due to DDL
- Possible mutex contention due to higher executions per cursor
- Index contention due to higher DML execution rate
- LOB contention due to higher DML execution rate
- Reaching limit on number of sessions per instance
- Unable to perform DDL on busy tables because of high execution rates

Options - Executions

- Avoid any retry and limit number of sessions for reducing reparse storm
- Create multiple versions of the same SQL (add a “/* <n> */”) or spread it across the cluster
- Rely on hash and other partitioning options to avoid index contention
- Rely on partitioning and optimize LOB DML’s to reduce LOB contention
- Investigate Shared Server (MTS), DRCP, or custom connection pool to multiplex query executions across processes

Challenges - IOPS

- Design for lower IOPS is tradeoff with DML scalability, i.e. global indexes
- RAID amplifies write overhead, i.e. RAID5 convert one write to five writes
- Predictable response time
- Optimizing crash recovery time is tradeoff with write IOPS
- Chatty neighbor (bulk loads on new DB) can impact production DB

Options - IOPS

- Limit global indexes and reverse key indexes to lower IOPS
- Use RAID 10 for critical databases and RAID5 for archive and FRA
- Tweak storage features using techniques such as dynamic tiering for consistent performance
- Test for IOPS impact and arrive at acceptable crash recovery times
- Test and determine accept level of bulk loads for sharing workloads on SAN

Future (5 to 10 years)

- Database sizes will reach over 1PB
- Flash will replace hard disk for all OLTP workloads
- With Intel 3D cross point, memory to processor ratio to go over 10TB
- Dual format (row and column) for all data in one system
- Transparent archival of older data with no changes to application
- Sharding and cloud compatibility for all database systems
- Real time analytics and transactions in one database system (e.g. “In Memory” Oracle Database 12c option)
- Over 75% convergence across NoSQL, NewSQL, and big RDBMS vendors

Summary

- Brace for continued tremendous innovation in database space
- Consolidation of NoSQL and NewSQL vendors within 3 years
- Multi database vendor (RDBMS and NoSQL/NewSQL) here to stay
- Big RDBMS vendors continue to lead in security, maturity and integration
- NoSSQL and NewSQL vendors will continue to lead in agility, economics at scale, and tighter integration with application