VISION CHAIN

Taming the VLDBeast

Leslie Tierstein
Vision Chain, Inc.

Overview - Environment

- Data Warehouse
- Supported on various Oracle platforms
  - 8i and 9i
  - AIX, Sun Solaris, HP
- Subject Area: Retail Sales/CPG
- Size
  - Main fact table grows by 300M-500M rows per week
  - Starts with 2 years of history
  - May be daily or weekly data

Overview – VLDB Issues

- Sizing and Architecture
  - Server and SAN
  - Database
- Write performance – ETL processing
  - Proprietary tool (“Data Manager”) written in perl with extensive PL/SQL
- Read performance – Reports and queries
  - 3rd Party Reporting Tools
    - MicroStrategy
    - Business Objects
- Maintenance, Administration, Backups

SAN/Server

- BAARF
  - Battle Against Any Raid F
  - http://www.miracleas.dk/BAARF/BAARF2.html
- RAID 0+1 – not RAID 5
- SAME – Stripe and Mirror Everything
- Reasons:
  - No write performance penalty
  - Cache benefit is negligible for VLDB full table scans
- Fight this battle up front – it affects cost

SAN/Server

- BAARF – References
  - Old (1996) but still valuable
  - Milsap: Is RAID 5 Really a Bargain?
  - Milsap: Configuring an Oracle Server for VLDB
    - Also available from HOTSOS

Server Disk Sizing

- Disk Size =
  - Non-Database +
  - Database
- Non-Database =
  - Operating System (negligible)+
  - Software (negligible) +
  - Staging Area (Input Files for ETL process)
    - File format?
    - How much history to maintain/archive?

Non-Database Disk Sizing

- Staging Files - Format
  - Compressed (zip files)
  - EDI formats
    - AS/2 is becoming standard in CPG, especially with pressure from Wal-Mart
    - Must be delivered non-compressed
  - XML – very large text files

Database Sizing

- With RAID 0+1, you need 2x your estimated database size
- But how do you estimate that size to begin with?
- Formulas for computing database size
  - Published with Oracle 8i
  - But not with Oracle 9i (“subject to too much variation”)

Database Sizing

- Database Size =
  - Code (negligible) +
  - Tables +
  - Indexes
- Object Size =
  - Data (average size, not maximum size) +
  - Row Overhead +
  - Block Overhead +
  - Object Overhead
- How to know “average size” beforehand?

Database Sizing

- See sample Vision Chain spreadsheet
- Factors
  - Functional
    - Number of rows
    - Average row size
    - Number of indexes on a table
  - Oracle object attributes
    - DB Block Size
    - Table/Index Pct Free

Database Sizing

- PCT FREE
  - Amount of space to leave free in a block when inserting rows, to accommodate growth in row size due to subsequent updates
  - How many updates do you do to a fact table in a data warehouse?
  - Set to a low, non-zero number

Database Sizing

- Block Size
  - Larger block size means better performance
    - More data retrieved per block read
    - Especially for FTS
  - Larger block size also means more data per block, since fewer block headers
    - Minimal – 1 gig saved in a 200 gig DB
  - Possible objections from OLTP DBAs or old-timers
    - Vary block size by tablespace? (9i and above)

Database Sizing

- Oracle Metalink, "NOTES ON CHOOSING AN OPTIMAL DB BLOCK SIZE"
  - "Use db_block_size of larger values than 8KB for systems in which your undo generation is not a meaningful part of your workload. Data warehouses fit this profile. With bigger blocks, you reduce total system I/O setup costs dramatically, yet you incur none of the disadvantages that you would incur in an OLTP system because people generally are not executing transactions (inserts, updates, deletes, and selects for update)."

Database Sizing

- Demonstration
  - Vision Chain Database Sizing spreadsheet

Write Performance - Load

- Use Direct Path SQL*Loader to load staging tables
  - Lightening fast (really)
  - Job went from 10 minutes to 2+ hours
- Use INSERT /*+APPEND */ to add to fact tables
- Use NOLOGGING as appropriate
- Implications for backup?

Write Performance - Load

- Using external tables is tempting, if
  - You need only one pass through the tables
  - You don’t need to update/cleanse the data before inserting

Write Performance - Load

- Use Global Temporary tables
  - INSERT/SELECT into Global Temporary table is faster than an UPDATE
  - Table goes away when the connection ends

Write Performance - Load

- We don’t need no stinkin’ index
  - Drop index on the staging tables before Load
  - Recreate after load

Database Structure

- Partitions
  - Enhance read performance by enabling partition elimination
  - Enhance write performance by allowing truncating partitions, rather than deleting rows
    - Need to delete/insert because POS data is restated

Database Structure

- Partitions/Sub-Partitions
  - Partition Key must be in Table Primary Key
    - May need to include extra columns

Logical Table Primary Key Constraint	**Partition Key**	**Actual Table Primary Key Constraint**
DAY_DT STORE_ID ITEM_ID	DAY_DT **COUNTRY_ID**	DAY_DT STORE_ID ITEM_ID **COUNTRY_ID**

Database Structure

- Partitions/Sub-Partitions
  - Use non-prefixed primary key
    - Partition elimination uses partition key
    - Primary key index uses primary key constraint

Logical Table Primary Key Constraint	**Partition Key**	**Actual Table Primary Key Constraint**
DAY_DT **STORE_ID** ITEM_ID	DAY_DT COUNTRY_ID	**STORE_ID** **DAY_DT** ITEM_ID COUNTRY_ID

Database Structure

- Partitions/Sub-Partitions
  - Create day-range partitions dynamically
    - Use a partition template
  - Use local indexes
    - Truncating the table can truncate the index
  - Analyze just new partitions or sub-partitions, as they are created/populated

Database Structure

- Tablespaces
  - OFA (Optimal Flexible Architecture) still applies
    - Is there a practical maximum size for a datafile?
  - Place date-based partitions into different tablespaces
    - Make a tablespace read-only
    - Drop tablespace after archiving (hah!)

Database Structure

- Tables/Tablespaces
  - Number of extents doesn't matter
    - No effect on performance
  - Keep extent sizes the same in the same tablespace
    - More usable space, less fragmentation
  - These principles are not limited to VLDBs

Database Structure

- Materialized Views
  - Summary/Aggregation tables almost always need Materialized View Logs
  - Need to create indexes on Materialized View?
  - What is refresh strategy?
    - Scheduled as part of batch stream
  - Verify execution plans

Database Structure

- Bitmap Indexes
  - Great for queries that need to select a very small subset of a very large set of rows
    - Be sure all columns are indexed
  - Not so good if FTS is a viable approach
  - Verify execution plans
    - Convert bitmap to ROWID

Execution Plans

- No hints!
  - Is the optimizer smarter than you? Or vice versa?
  - What if data distribution changes? Or is not identical between environments?
    - Example: Items with no Sales in the Past Week, by Store
- SQL generated from 3rd party tools
  - Can't insert hints
  - Can try stored outlines – if cursors can be shared (use bind variables not literals)

Execution Plans

- Database Statistics
  - When to gather statistics in a production system:
    - Always (Scalzo) ... Never (Ensor)
    - Sometimes ... when data distribution changes
  - DBMS_STATS vs ANALYZE
    - ANALYZE is deprecated in 9i
    - Enhanced features in DBMS_STATS, especially analyzing partitions, sub-partitions, and doing histograms

Execution Plans

- DBMS_STATS for sub-partitions

```
SELECT 'DBMS_STATS.GATHER_TABLE_STATS(
 'ownname => ' ||CHR(39)|| 'VC_REP' ||CHR(39)|| ', ' ||
 ' tabname => ' ||CHR(39)|| table_name ||CHR(39)|| ', '
||  CHR(10) ||
 ' partname => ' ||CHR(39)|| partition_name ||CHR(39)||
', ' ||
 ' estimate_percent => 5,
 ' method_opt => ' ||CHR(39)|| 'FOR ALL COLUMNS ' ||
 CHR(39)|| ', ' || CHR(10) ||
 ' granularity => ' ||CHR(39)|| 'SUBPARTITION' ||
CHR(39) || ', cascade => TRUE);'
  FROM all_tab_partitions
where subpartition_count > 0
and table_name IN (<list of tables>)
and last_analyzed IS NULL
```

Database (init.ora) Parameters

- See initdw.ora as a starting point
  - Select small, medium or large
- Vision Chain examples
  - Shared pool size: DW vs OLTP concerns
  - Sort area size: FTS and index builds
  - Parallel threads per CPU

Conclusion

- Any questions?

About the Author

- Leslie Tierstein is a Senior Consultant with Vision Chain, Inc, a privately held company which provides data warehouses to Consumer Packaged Goods (CPG) companies
- She has extensive experience in OLTP and data warehouse development and project management
- For more information:
  - Web site: http://www.visionchain.com
  - Email: leslie.tierstein@visionchain.com