2–6 October, 2006

Hilton Vienna

Vienna, Austria

IDUG® 2006 Europe

GoFurther

G13

Big DB2 databases, moving data, big problems

Walter Guerrero
CA

04 October 2006 • 17:30 p.m. – 18:30 p.m.

Platform: Tools and Utilities

Abstract

With DB2 databases growing larger each day, you can never seem to extract data fast enough to make anyone happy. And when someone wants a subset of the data each day to load into another application, their request alone can turn into a full time job. This session will explore the methods and best practices available when moving mainframe and distributed DB2 data and get it into other databases and applications.

Objectives

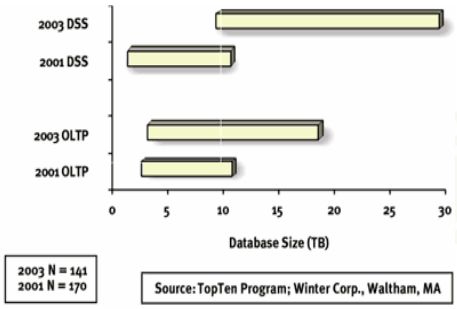
- Unloading data
- Transferring data
- Techniques to move data
- Administering data movements
- Lessons learned

Today's Problems

- Data sets growing exponentially
- Systems must run 24/7/365
- Maintenance window reduced
- Increasingly complex datasets with the addition of unstructured data

Data Growth

- Structured and unstructured growth
- Heterogeneous data (Binary and XML data as part of a dataset)
- Data storage now in terabytes
- Data subsets needed to maintain other systems
- More sophisticated business intelligence needs



Data Organization Issues

- Structured attribute selection
- Unstructured attribute selection
- Type of table entity relationships
- Date/Time formats selected
- National character set
- Data partitions

Transfer Methodologies

- Mainframe based
- Distributed systems based (Linux, UNIX, Windows)
- Unit of transfer
  - Multiple small files
  - Single large file
  - Partitions

Transfer Methodologies - Mainframe

- Local datasets
- Remote datasets
- FTP services
- HTTP services
- Tape services

Transfer Methodologies -- Distributed Systems

- Local file systems
- Remote files systems
  - NFS mounting points
  - NetBIOS mounting points
- FTP services
- HTTP services

Types of Data Movements

- Single File
  - Small File
  - Large File
- Multiple Files
  - Small Files
  - Large Files
- Consolidation of data partitions into coordinating node

Data Files

- A series of small files improve unload performance
- Single file can be used to unload small datasets
- Transfer of large files will create bottleneck and system accommodation
- EBCDIC/ASCII translations and limitations
- National character set need to be considered

Effects of Data Movement

- Unloading process can affect system performance
- Available storage will be impacted as datasets are unloaded
- Network traffic will be impacted if unloading to a remote file system

Extraction Methods

- Using DB2's unload utilities
- Using the export facility
- "Select into" statement
- Commercial unload utilities
- User created unload utility

Extraction Formats

- Delimited
- PC/IXF
- WSF
- Fixed-width
- FixedDB2

Universal Delimited Format

- Universal format
- Can be used to transport data between different database engines
- Delimiting format can be controlled
- Watch out for national character set and date-time format inconsistencies
- Resulting data files are large
- Issues with unstructured data (binary)

Other Unload Formats

- Fixed-width
- FixedDB2
- IXF (Integrated Exchange Format)
- Worksheet Format (WSF)

Extracted data sets

- Text files created
- Based on extraction format selected
- BLOBs unloaded as hex files
- XML data unloaded as part of text files for XMLVarchar
- CLOBs unloaded as text files
- Only data in files
- Format definition files are separate

SQL Queries to Control Data Extraction

- SQL queries can be used to control unload criteria
- Sub-set of SQL language allowed
- SQL queries should only work on one table
- Do not use table joins
- SQL statement design still important, since it affects performance

Administration of Data Movements

- Unload jobs can be run interactively
- Advanced scheduling of Unload jobs
  - z/OS
  - LUW (Linux, UNIX, Windows)
- Third-party scheduling sub-systems
- Built-in scheduling facilities of commercial unload facilities

Administration Do's/Don'ts

- Scheduled unload jobs for off-peak hours at times that will cause the least impact
- Tables will be locked during the read phase of the unload
- Write SQL queries against a single table.

Performance Considerations

- Be mindful of SQL statements
- Unload files size
- Retrieving data from a partitioned database into a coordinating node
- Available storage to hold all the unload data files
- Created unload file in file system other than that storing the database files

Hardware Considerations

- Number of CPUs
- Available memory
- Available storage facilities
- Number of threads that can be dedicated to the unload process
- Disk I/O (if reading/writing to same disk)

Database Performance

- Available CPUs for the DB2 instance
- Correct DB2 parameters setup
- DB2 instance has been sized correctly
- Proper selection of tablespace parameters
- Proper selection of buffer pools
- Correct implementation of partitioned databases
- Pruning of database log files

Table Tuning

- Table statistics gathered
- Table reorganized
- Correct tablespace selection for given table
- Correct type of table lock has been selected
- Separate tablespaces for table, index, and/or LOB object
- Select the proper percentage of free space left after a load or reorganization

Tools – Custom Made

- Highly tuned to present conditions
- Targets a single/multiple table(s)
- Maintained in-house
- Can interact with OS scheduling systems or third-party
- Application will have to keep up with table DDL updates
- Mostly single thread applications

Commercial Tools

- Can handle unloading multiple tables concurrently
- Multi-threaded applications
- Own scheduler or can use system/third party scheduler
- Graphical interfaces (GUI or BUI)
- Can unload large amounts of data quickly
- Can handle single or multiple output files
- Can use different output formats

Extraction Tips

- Determine the subset to be unloaded
- Determine the correct output file size to use
- Single/multiple output file selection
- How many system threads to use?
- Is a SQL statement important for the unloading?
- Schedule the unload for a time when the system has a light load and the maximum number of CPUs will be available

Character Set

- Single Byte Character Sets
- Double Byte Character Sets
- EBCDIC
- ASCII
- Unicode

Unstructured Data

- LOB
- XML

Date/Time Attributes Considerations

- Select the correct date/time output format for the database engine that will be receiving the unload file
- A custom time stamp might be required
- Select the correct code page

Handling Large Terabytes data sets

- Check for available storage for extracted dataset
- Break the dataset into smaller files to improve its performance
- Verify that enough CPUs and memory are available to execute this unload
- Check the I/O performance of the storage sub-system

Lessons Learned

- Consider availability of storage
- Efficiency of using the maximum number of available CPUs
- The DB2 database tuning
- Optimize SQL statements to extract the dataset to lighten system load
- Reorganize tables performance
- Determine the required number of unload files based on the size of the data being unloaded

Summary

- Unloading of data sets needs to be planned
- Data extraction and movement is very important
- Review available hardware prior to starting the data extraction and movement
- Review the tools available from IBM and third-parties to determine what will work best

Questions?

G13

Big DB2 Databases,
moving data, big problems

Walter Guerrero

CA

Walter.guerrero@ca.com