SQL Server Parallel Data Warehouse Architecture

José A. Blakeley
Partner Architect

May 24, 2012
Agenda

- PDW Fundamentals
  - Scale-out system architecture (HW and SW)
- Core functionality
  - Shell Database and Distributed Query Processing
  - Data Movement
  - Bulk Loading
- Futures
PDW Fundamentals
What is Parallel Data Warehouse?

SQL Server Data Warehousing in Appliance Model

- A SQL Server appliance for DW that is:
  - Scalable
    - Shared-nothing, MPP DBMS system
    - Scales from 10s to 100s of TB of data
    - Scales from ½ rack (4-6 nodes) to 4x rack (~60 nodes)
  - Standards based
    - Leverages commodity hardware
    - Speaks SQL Server language (T-SQL)
  - Flexible
    - Offers hardware of choice (HP/Dell)
    - Supports multiple h/w architectures
  - Cost effective
    - Low price/TB ratio
EDW Architecture

- New Product
- Increased Efficiency
- Empowered Users
- Increased Revenue
- Data Mining
- Analytics
- Collaboration
- Reporting

Data Warehouse

Aggregated Data from Various Line-of-Business Systems

LOB Systems
Customer information, claims, DDA, loans, cards, investments, and payments
Shell Database and Statement Processing
Distributed DBMS Layering

**PDW Engine in control node**

- **Compilation**
- **Global metadata**

**Shell DB SQL Server**

- SQL Server for catalogs, compilation, authorization

**Distributed execution coordination**

**Compute nodes**

- Compute node SQL Server

**Data movement**

- DMS
CREATE TABLE customer
(
c_custkey bigint,
c_name varchar(25),
c_address varchar(40),
c_nationkey int,
c_phone char(15),
c_acctbal decimal(15,2),
c_mktsegment char(10),
c_comment varchar(117))
WITH (distribution=hash(c_custkey)) ;

CREATE TABLE nation
(
n_nationkey int,
n_name varchar(25),
n_regionkey int,
n_comment varchar(117))
WITH (distribution=replicate);
1. User issues DDL statement
2. Statement runs on the Shell first
   • SQL Server (shell) performs the parsing, binding, authorization
   • The shell schema gets updated
   • PDW-specific info stored in extended properties
   • PDW resource manager manages life cycle of statement execution (e.g., Tx scope, locks)
3. Statement issued against the compute nodes
4. Results returned back
   • PDW rollback manager manages failures and clean-up
Quick Look at Query Execution

SQL Server PDW Appliance

The user connects to ‘the appliance’ like he would to a ‘normal’ SQL Server, and sends his request.

The control node handles global query execution, and generates a distributed execution plan.

The actual user data resides on control nodes, and steps of the global execution plan are executed on each compute node.

SQL Server PDW is a shared nothing MPP system, meaning user data is distributed across the nodes*. Data Movement Service is responsible for moving data around so that individual nodes can satisfy queries that need data from other nodes.

*Data Movement Service
Query Optimization

1. SQL Server parsing, access validation, query simplification and exploration
   - Query simplification (e.g. column reduction, predicates push-down, subquery unnesting)
   - Logical space exploration (e.g. join re-ordering, local/global aggregation)
   - Serializing MEMO into binary XML (logical plans)
   - De-serializing binary XML into PDW Memo

2. Optimization for distributed plan (PDW)
   - Removing unnecessary plans
   - Identifying interesting properties
   - Injecting data move operations
   - Costing different alternatives
   - Pruning and selecting lowest cost distributed plan

3. SQL Generation
   - Generating SQL Statements to be executed
Statistics

- **Local statistics (compute nodes)**
  - Standard auto-stats for user-data tables
  - Auto-stats also on temp tables created by DMS at each step

- **Global statistics (Shell DB in control node)**
  - Basis for distributed execution plan
  - No auto-stats (scoped out of AU3)
  - Manual stats: compute on each distribution, then merge to reflect global table
TPCH – AU3 Performance Results

TPCH-3TB Performance - AU2 vs. AU3

Elapsed Time (s)

Improvement Factor

Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21 Q22

AU2 AU3 Improvement
Data Movement
Data Movement Primitives

For distributed tables
- SHUFFLE_MOVE (N:N)
- BROADCAST_MOVE (N:N)
- PARTITION_MOVE (N:1)
- SHUFFLE_LOAD

For replicated tables
- MASTER_MOVE (1:N)
- TRIM_MOVE (1:1)
- REPLICATE_MOVE (1:N)
- REPLICATE_LOAD
1. Pinned pointer to DMS Memory Buffer passed to native reader routine.

2. Reader executes query and fetches rows into native-formatted buffer.

3. Return from native call signals buffer full.

4. Filled buffer retrieved by ReaderWorker.

5. ReaderWorker advances to beginning of each row and calls into Distribution Manager to Write Row To target data channel Buffer.

6. Hash distribution column and write row to correct row distribution buffer. Enqueue buffers to Sender when filled.

7. Dequeue and send buffers.
Buffer received and enqueued to target writer

WriterWorker Thread

1. Buffer received and enqueued to target writer

2. WriterWorker invokes SqlNativeBufferBulkCopy to bulk-copy contents of buffers to target table

3. When SqlNativeBufferBulkCopy requests a buffer, a buffer is dequeued from incoming writer queue and returned.

4. Pinned pointer to DMS Memory Buffer passed to native BCP routine.

5. Buffer is processed row-by-row: ODBC bcp_sendrow() used to bulk-write data to SQL Server
Bulk Loading
DWLoader Data Flow

- **DWLoader** = bulk loading utility
  - Transactional, multi-step
  - Runs on Landing Zone Node
  - One input file per destination table
- Load is performed in 3 steps:
  - Create a staging table
  - DMS movement
    - Replicated Load & Distributed Load
    - Insert-select
  - SSIS uses Adapter to load directly into DMS
- Load speed:
  - 1.2 TB/hr, 10x compute nodes
  - Target is Heap table
Other important functionality

- Backup/restore
- Fault tolerance
  - All HW components have redundancy
  - Windows Failover Cluster (WFC) for failover
  - Control, compute and management nodes have A/P
- Systems substrate
  - End-to-End setup, servicing, upgrade, replace node
  - Appliance health, monitoring, PDW SCOM Management Pack
  - PDW appliance validator
- Integration with Microsoft and 3rd party BI tools
  - SS Integration Services (ETL) has PDW as a destination
  - SS Analysis Services (OLAP) has PDW as a source
  - SS Reporting Services, Excel PowerPivot
  - SAS, Business Objects, Informatica, Microstrategy
  - Hadoop connectors (ETL)
Futures
Futures

- Column-store storage and processing
- Single-node-query optimizations
- Broader support of SQL SMP features
- Increased data load parallelism
- Hadoop integration
Summary

- PDW Fundamentals
  - Scale-out system architecture (HW and SW)
- Core functionality
  - Shell Database and Distributed Query Processing
  - Data Movement
  - Bulk Loading
- Futures
Thank you!