Lessons Learned in Our Benchmarking

- Data partitioning must be very carefully designed to balance I/O and CPU
- Parallel Loading is great ! --- almost linear speedup
- Backup may be important to save time (we don't have)
- The most difficult and trouble-making phase is the database preparation/loading/indexing phase which always involves storage management parameters
- you must be both a DBA and sysadm and benchmarking relies critically on system stability
- Be prepared to bring down the whole parallel system a couple of times a day
- Be prepared that the results are negative (nothing may be obtained !)
- you need at least twice the disk space of your original database size
- performance tuning is extremely difficult with Oracle 7 parallel server

Characteristics of a DSS Workloads

- Few insert/update/delete transactions, read-only queries.
- Non-selective & data-intensive; few in number, large in size
- May return a large volume of data: e.g.: mailing list
- Queries frequently scan a large percentage of the data in the database
- Often used to mine data in a recursive, drill-down fashion

Industry Benchmarking Tests

- Transaction Processing Council

  TPCD A,B,C

  TPC D (not yet finalized)
- Wisconsin Benchmark (aka Dewitt)
- Set Query
- The Neal Nelson Database Benchmark (tm)
- SPEC
- The Customer

  -- or worse >> The Consultant

- Automatic file striping and disk partitions by the operating system on nCUBE2
    - General OS and server level, less flexible, but relatively transparent to DBA
    - data skew is less possible, due to the Gigacache
- Manually file striping at database table level on IBM SP2
    - Application-specific, flexible, but difficult to handle with large number of files and impossible to determine optimal striping length
    - data skew is highly possible

Data Partitioning in Oracle 7.1.3

- Oracle 7 on nCUBE2 and IBM SP2 are actually implemented as shared-disk (vs. shared-nothing) architecture

- Shared database by sharing disks through parallel file system or parallel I/O system:

  nCUBE -- the GFS (Global File System) over a Gigacache Subcube and nCUBE2's parallel I/O system;

  IBM SP2 -- VSD (Virtual Shared Disk) parallel I/O system (over the HPS)

Example of parallel execution of a full scan (fig. , fig.)

- The shared query server is known as Query Coordinator who is responsible for parallelizing operations and interpreting the results from query servers
- Not all operators can be parallelized in Oracle 7.1.3
- Oracle Server Parallizable Operations
    - Table Scans
    - Sorts (for ORDER BY, GROUP BY, Joins, DISTINCT)
    - Joins (Sort-Merge, Nested Loops)
    - Aggregates (GROUP BY, MIN, MAX, AVG, etc.)
    - SET operations (e.g., UNION, INTERSECT or MINUS) are not parallelized

Processing Phases

1 *Parse* --- a query optimizer determines optimal execution plan, parallelize query if a full scan was included in the execution plan (note the query is parallelized dynamically at execution time, automatically adapting changes in data partition and server configuration)

2 *Execute* --- perform operations

3 *Fetch* (Queries) --- return rows from query server processes

The execution and fetch phases can be performed in parallel (pipelined parallelism) by multiple server processes

Parallel Query Processing: Overview

Data Flow:

1 A client process issues a query

2 The statement is parsed and a shared server process executes the statement

3 The shared server process determines if the statement can utilize parallel processing, and then enlists two or more query servers to execute the statement in parallel. The shared server process partitions the work of each operation in the query equally among the query servers

4 Query servers process certain operations in the query independently and return sub-results back to the shared server process

5 The shared server process reassembles the results from the query servers and returns the query's results to the client

(Fig.)

Parallel Query Option

- Parallel Query Processing
  - enables the parallel processing of lon and complex queries with parallelizable operators such as scan,join and sorts
  - speeds up query response time (vs. throughput in Parallel Server Option)
- Parallel Data Load
  - Loads data in both a direct and parallel method
  - speeds up the loading of large data loads
- Parallel Index Creation
  - speeds up the creation of large indexes
- Parallel Recovery
  - enables concurrent recovery sessions

- The PCM is handled separately from transaction lock -- when instance wish to update the same data (e.g.: Instance X and Y attempt to update Row 1), normal transaction locking is involved.

Consider the following scenario:

Time

1 Instance X attempts to update row 1 in (data block n)

Instance X becomes owner of PCM lock for data block n

Instance X update row 1

2 Instance Y attempts to update row 2 in (data block n)

Instance Y requests the PCM lock from instance X for data block n

Instance X releases the PCM lock

Instance Y becomes owner of PCM lock for data block n

Instance Y update row 2

3 Instance X attempts to update row 3 in (data block n)

Instance X requests the PCM lock from instance Y for data block n

Instance Y releases the PCM lock

Instance X becomes owner of PCM lock for data block n

Instance X update row 3

Using Oracle 7 Server's PCM locks:

- An instance acquires a PCM lock as a resource for update purposes
- All buffer caches must contain the same copy of a resource

PCM blocks:

- Ensure cache coherency by only allowing instances to change blcoks while holding a lock
- PCM lock work independently of transaction locks (row-level locking)
- Ensure cache coherency using minimum amount of communication
- PCM locks are allocated to files when a instance is started

- Example of concurrent control:

Time

1 *Node 1 reads block 1 from disk and stores in Node 1 memory cache, Node 2 reads block 1 from disk and stores in Node 2 memory cache.*

2 *Node 1 writes information to block 1, Node 2 writes information to block 1.*

3 *Node 1 writes block 1 to disk.*

4 *Node 3 reads block 1 from disk and stores in Node 3 memory cache.*

In this situation, Node 1 and 3 have the same information in each memory cache, but Node 2 has an incoherent cache.

Parallel Cache Management and Distributed Lock Manager

- Concurrency control of buffer caches among instances
- Maintain independent, yet consistent cache of blocks for all instances
- Allow multiple nodes to simultaneous read/write to database (disks)
- Maximizes concurrent data access and minimizes (disk) I/O activity
- Block-level PCM locks, row-level locking preserved
- A fault-tolerant DLM: if a node goes down, the DLMs will automatically recover and reconstruct the lock database on behalf of the failed DLM (fig.)

Parallel Server Option

- Multiple Instances on different nodes share a single physical database
- The data is accessed by all nodes through a parallel file subsystem
- All instances have common data & controls files and their individual log files
- All instances can simultaneously execute transactions against the single database, and each instance can have multiple users executing transactions
- Data is kept consistent using PCM Technology
- Implemented by the Distributed Lock Manager

Oracle7 Parallel Server Characteristics

- Start an Oracle Server instance on each node
- Each instance has its own set of redo log files
- All instances can concurrently execute transactions against the same database and each instance can have multiple users executing transactions

(fig.)

The Oracle Server Instance

- Every time the Oracle 7 Server is started, the System Global Area (SGA) is allocated and the Oracle 7 Server background processes are started. The combination of memory buffers and background processes is called an Oracle Server Instance. SGA (fig.)
- The SGA is a group of shared memory buffers allocated by the Oracle 7 Server for an instance
- The background processes asynchronously perform distinct tasks on behalf of all database users of the instance

What is the Oracle Parallel Server ?

- Oracle 7 Server Architecture (fig.   )
- Oracle 7 Database Files:

Data Files --- Contain all the database data; logical structure, such as tables and indexes, are physically stored in the data files

Redo Log Files --- Transaction logs that record all changes made to database information for recovery

Control Files --- File that record the physical structure of the database including all datafiles and redo log files

- All parallel implementation is transparent to users
- Key technology: Parallel Cache Management and Parallel Query Processing
- Key components: a Distributed Lock Manager and a Parallel Query Optimizer

What is the Oracle Parallel Server ?

- A high-performance RDBMS for loosely coupled and MPP systems
- Currently supported systems: IBM SP2, nCUBE2, SGI SMP, SUN SMP, HP SMP, Amdahl, Encore, KSR, CM5,
- Also known as multi-instance Oracle: Multiple instances running against a single, shared database
- A evolutionary approach on existing sequential Oracle RDBMS
  - first introduced with Oracle 6.2
  - version 6.2 = version 6.0 (sequential) + Parallel Cache Management
  - version 7.1 = version 6.2 + parallel query processing
  - newest version 7.1.3 = sequential Oracle + Parallel Server Option + Parallel Query Option
  - production release on nCUBE is just a week ago

NPAC's Experience in Parallel Database Technology

- Beta test sites with Oracle, nCUBE and IBM for their parallel RDBMS products
- Direct contacts with PRDBMS developers at Oracle and IBM
- Hands-on Expertise in large-scale (up to 100 GB db size) DSS benchmarking
- Experience in parallel computing and applications in industry, parallel system integration and data visualization