Parallel Relational Database Management Systems

Gang Cheng, Marek Podgorny

Northeast Parallel Architectures Center at Syracuse University

gcheng@npac.syr.edu
marek@npac.syr.edu

Outline

- Parallel Database Technology in Commercial Applications and Industry
- Parallel Database Technology and Theory
- Parallel Database Projects at NPAC
- Parallel Oracle7 RDBMS -- A Case Study

Motivations

- Conventional (mainframe) systems unable to support
  - time-critical queries
  - simultaneous OLTP and decision support systems
  - gigantic scientific and commercial databases
  - complex queries (approximate matching, pattern recognition ...)

Motivations

- **In commercial business world, MPP systems**
  - **can find more profitable applications than the scientific world**
  - **have better cost/performance than the mainframe system**
  - **are mature and highly available NOW**

Motivations

- **Parallel Database Technology Appeals**
  - **scalability**
  - **system expandability alleviates migration problems**
    - **business branches of different size can use the same technology**
  - **power and price/performance ratio**
  - **simultaneous OLTP and complex Decision Support queries (single enterprise image**
  - **nCUBE + Oracle capable of >100 tps at ~$2000/tps (TPC-B benchmark, compare with 425 tps for 4*VAX cluster at $16,500/tps)**
  - **High availability**
  - **Integrability**
    - **parallel Oracle RDBMS integrates as easily as conventional RDBMS**
  - **do not require re-investment in application software**
    - **Industry standard database language - SQL (Structured Query Language)**
    - **Parallelization is carried out at RDBMS server side**
    - **Migration to parallel RDBMS is painless (compare to scientific applications)**

Motivations

- Market Demand
  - Most commercial cooperations re-organize their businesses
  - to be more efficient with a smaller workforce
  - need more computing capacity to grow business and fast response to have better services
  - need better system scalability and price/performance

Application Areas: commercial, administration, scientific

- **High transaction rate applications**
  - banks, airlines, telecommunication, security brokers, retailers ...
- **Real-time applications**
  - stock trading, C3I (military, civil defense), air traffic, process control (manufacturing, accelerators, reactors/generators), real-time insurance claim analysis ...
- **Complex queries**
  - decision support (market analysis), molecular biology, chemistry/pharmacology ...
- **Massive amount of data**
  - EOS, HEP, weather, census, global environment models, oil exploration, medical history files, multimedia support ...

Applications Types

- OLTP (On Line Transaction Processing) --- consists of performing many small independent requests submitted by many clients simultaneously and operating
- DSS (Decision Support System)
- Data Warehouse --- operations mixing DSS and OLTP
- Data mining --- uncover/extract meaningful information from mass data sources

An Application Example ---
Intelligent Business systems

- **Objectives**
  - **understand customer buying habits and product preference**
  - **predict market tendency**
  - **make efficient, productive and profitable decisions**
  - **must be based on answers to questions on ALL historical business transaction data (Lots of !)**

Intelligent Business systems ---
Typical Questions

- “Give me a breakdown of all customers likely to default in the coming year”
- “Do three things: predict year-end demand, tell me which customers will fuel that demand, and tell me why”
- “Define a highly concentrated micromarket (make special offers to customer segments that meet very select criteria) within my database”
- “Create a model that really explains why some customers renew their subscriptions and others don’t”
- “Suggest ways of regrouping our customers into new market segments (for Direct-Marketing)”
- “Find some customer-preference pattern I might not be aware of”

Intelligent Business systems --- Major Technology Challenges

- Lots of Historical Data (and they are growing daily !)
- High Performance Computing System and DBMS
- Intelligent Data Analysis (Optimization)
- Not possible on a mainframe system in terms of both performance and capacity

Intelligent Business systems --- Solutions

- **MPP system**
- **Parallel Relational Database Management System**
- **Intelligent Algorithms**

Major Software and Hardware vendors in Parallel Database Technology

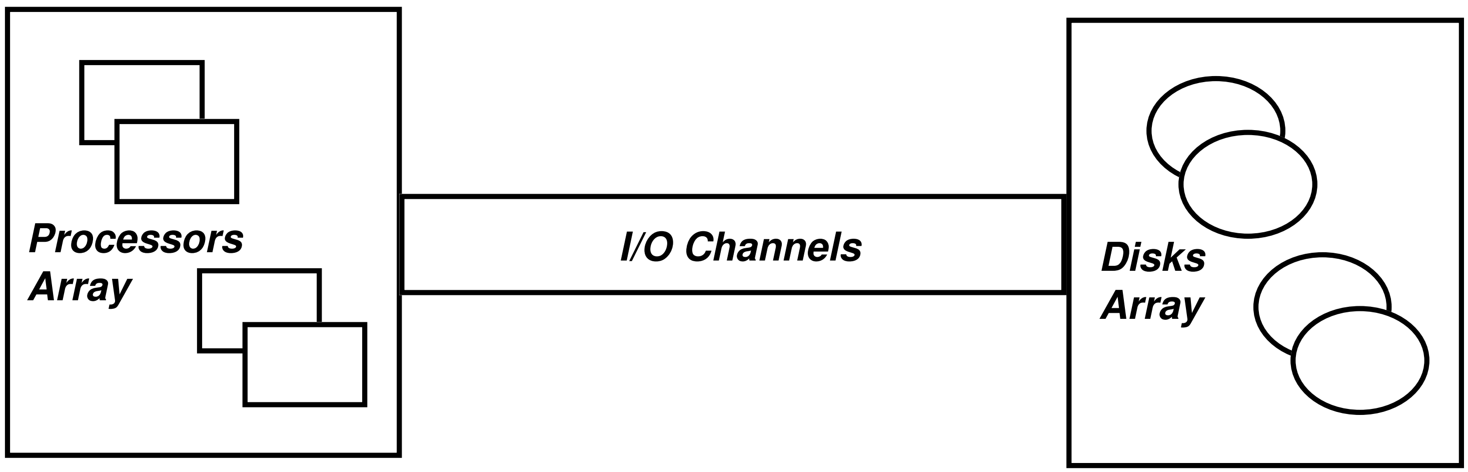
- Oracle (Oracle7 with parallel server and parallel query options)
- Sybase (Navigation Server on AT&T 3600, SUN SMPs, and IBM SP2 (planned),SQL Server 10)
- Informix (INFORMIX-OnLine Dynamic Server)
- IBM (Oracle7, DB2/6000, DB2/2, SP2, IBM 390 Parallel Sysplex)
- SGI (Oracle7 on SMP Challenge database server)
- Taradata with AT&T and NCR
- Tandem (Himalaya K10000
- Cray Research (SuperServer)
- SUN (Oracle7, DB2 on SUN's SMP)
- AT&T and NCR
- TMC (CM5, Darwin, Oracle7, Decision/SQL, Parasort)
- KSR (Query Decomposor, Oracle7)
- Amdahl (Oracle7)
- nCUBE (Oracle7, nCUBE2)
- data-CACHE Corp (SQL dataMANAGER)
- Data General Corp (Oracle7, DSO)
- DEC (POLYCENTER Manager)
- HP (Oracle7, IMAGE/SQL, HP 3000)
- Encore Computer Corp (Oracle7 on Infinity 90)

Current Major Commercial Users

- Kmart
- American Express
- AAA Airline
- Citibank
- Prudential Securities
- Bell Atlantic Corp
- BellSouth
- BT (British Telecom)
- Criminal Investigative Technology Inc.

Hardware architectures for parallel DBMS

- **Basic Considerations**



Hardware architectures for parallel DBMS

- **shared-memory (also called Symmetric Multiprocessors SMP)**
  - all processors share direct access to a common global memory and to all disk. example: Oracle7 on SGI Challenge
- **shared-disks**
  - each processor has a private memory but has direct access to all disk. example: Oracle 7 on IBM SP2
- **shared-nothing**
  - each memory and disk is owned by some processor that acts as a server for that data. Mass storage is distributed among processors by connecting one or more disk. example: Oracle 7 on nCUBE2

- shared-data model

  - combine features of the above three models, data sharing technology directly interfaces with the individual processors to enable each processor to cache shared data in its own local memory with full read/write capability (as in *shared-memory* model) Data sharing maintains data integrity by ensuring cache coherency across all processors and provide efficient cross-system locking mechanisms (as in *shared-nothing* model)All disk devices are accessible to each processor so that a workload may be scheduled to run on any available processor (as in *shared-disks* model), instead of run only on processors with data affinity (as in *shared-nothing* model)example: DB2 on IBM System/390 Parallel Sysplex.

Topological view of MP machines (with “Scalability” in mind)

- **Shared-Memory and Shared-Disk**
  - interference limits scalability
  - complicated and relatively expensive
- **Shared Nothing**
  - minimal interference between processors
  - moves only questions and answers over the network
  - can scale up to hundreds (maybe thousands) of processors
  - composed of cheap, stock parts and a fast interconnecting network
  - achieve near-linear improvements
  - good for system scalability and portability

Dataflow perspective

- Pipelined: “assembly line”
  - average query does not have that many steps
  - some operational stages cannot be pipelined (e.g. Sort)
  - one operator is longer than the other (a kind of skew)
- Partitioned: “divide and conquer”
  - fairly straight forward to implement
  - “tough” operations are divided, parts run in parallel
- Combination: “best of both world”
  - several partitions running parallel
  - each partition is, where possible, a short pipeline

Parallelisms in parallel database systems

- pipelined parallelism (inter-operator parallelism)
- query parallelism
  - inter-query parallelism
  - intra-operator parallelism
- partitioned (declustered) parallelism (I/O parallelism)
- transaction parallelism

Data Partitioning --- How to divide data among multiple disks ?

- a common practice on large number of disks and mass storage systems (file striping)
- high I/O bandwidth of multiple disks and parallel I/O
- no need for specialized hardware
- important to query load balance on a shared-nothing system

Major Approaches in Data Partitioning

- Round Robin
  - data scattered
  - cannot locate specific records
- Hashing
  - scatters the data
  - can locate specific records
  - danger of data skew
- Range Partitioning
  - data not scattered
  - can locate specific records
  - danger of data skew
  - “related” data can be clustered

Pitfalls in data partitioning

- operations/application dependent
- danger of data skew arises on shared-nothing with data partitioning
- data skew is the focus of a number of research projects
- automatic OS file striping vs manual table striping

Performance Matrics In Parallel Database Systems

- speedup (degree of parallelism)
- scaleup (scalability) -- transaction scaleup, batch scaleup, processor scaleup, I/O scaleup
- sizeup (both)

Performance barriers

- startup (latency)
- interference (communication, locking)
- load balance (execution skew - all the execution occurs in one partition)
- data-partition (data skew - all the data is placed in one partition)
- imbalance of system resources (CPUs and I/O channels)
- CPU-intensive or I/O intensive queries

Some basic terminology for relational model

- **Data Structure**
  - relations (files, tables)
  - tuples (records, rows)
  - attributes (fields, columns)
- **Relation operators**
  - scan (select-project) (a relation, a predicate, and an attribute list
  - sort (reorder)
  - aggregate operators (SUM,AVG,MAX,MIN,...)
  - insert/delete/update
  - set operators (union, intersection, difference)
  - join, merge and division
- **embedded operators**
  - uniformity of the data and operators
  - source of data-flow execution model

Major RDBMS functionality

- Data Access --- SQL, Transactions, PL/SQL, Data Integrity
- Data Concurrency and Consistency --- Concurrency, Locking
- Data Security --- Users and Schemes, Roles and Privileges, Profiles and Auditing
- Database Backup and Recovery --- Redo log, Rolling Forward/Back
- Distributed Processing --- Client-Server environment

Structure Query Language (SQL)

- built on relational model, simple and powerful
- non-procedure language, only specify “what-to-do”, but “how-to-do” (4GL)
- Industry Accepted Standard
- data-independent --- two-phase database schema (logical and physical database)
- extended to object-oriented programming interface

Parallel Database Software Architecture

- **Distributed Lock Manager**
  - Implementation of Parallel Cache Management
  - Support transaction parallelism of multiple OLTP
  - A simple and effective approach to port sequential RDBMS to MPP and loosely coupled systems with shared-nothing architecture
  - support parallel loading, parallel indexing, parallel insert/update and parallel recovery
  - Major functionality
    - keep track of the current “ownship” of a resource
    - accepts requests for resources from application processes
    - notifies the requesting process when a resource is available
    - get exclusive access to a resource for a resource

Parallel Database Software Architecture

- **Parallel Query Optimizer**
  - Parallel Query Optimizer builds a query execution plan employing all system resources on the basis of data-partition knowledge, it can easily employ few tens of processors to execution a single query with high degree of parallelism. Both data-parallel and data-flow paradigms are used to achieve this goal
  - Support complex CPU-intensive or I/O-intensive queries for DSS
  - near-linear or even super-linear speedup can be achieved
  - Major focus of research area --- optimized parallel algorithms needed
  - carried out by a query coordinator on the RDBMS server (the query coordinator generates the optimal parallel plan rather than having the best serial plan simply executed in parallel. This optimization is performed by the coordinator without the need for any external information (DB2 and Oracle7 use different approaches)
  - most relational operators are parallizable, including full-table scan, sort, aggregate operators, join, merge
  - System/Server tuning plays an important role in achieving optimized performance in a parallel RDBMS --- benchmarking is an effective means