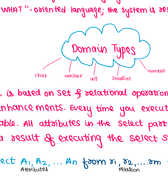
Lecture 2

Gamma is a highly parallel high performance data management system. Gamma hides parallelism using SQL. The user writing the query is aware that they are using parallel database management system.

SQL - Data Definition language

A "what"-oriented language, the system is responsible for "How"



SQL is based on set of relational operations with certain modifications & enhancements. Every time you execute select statement it produces a table. All attributes in the select part specify the schema produced as a result of executing the select statement.

Select $A_1, A_2, \ldots$ from $r_1, r_2, \ldots r_m$ where P
Attributes, Relation, predicate

- A selection clause is a comparison between an indexed tuple variable and a constant $(<, \leq, >, \geq, \neq, =)$
- A join clause is a comparison between 2 indexed tuple variable $(<, \leq, >, \geq, \neq, =)$

to eliminate duplicates from the result, use the keyword distinct in the target list.

for eg: Select distinct e.salary from Emp e where e.salary > 50000

Conceptualize the execution of a select query as one of:
- Evaluate qualification to locate qualifying tuples.
- project out columns of these tuples using the target-list.
- make the resulting tuples into a relation & eliminate duplicates if necessary.

Relational Algebra + Query tree

The user is writing SQL statements; & the system underneath takes that SQL statement & convert it to relational algebra; this relational algebra representation is represented as a query tree. This query tree is executed by the execution engine of the database management system.

Relational Algebra is a procedural query language i.e user must define both "HOW" & "WHAT" to retrieve, used to implement SQL.

Equality join connects tuples from 2 relations that match on a certain attribute

Natural join connects tuples from 2 relations that match on the specified common attributes.

Query tree is nothing more than just unpacking of the algebraic expressions inside out.

Disk Input/output

DISK will construct a copy of the disk page in memory if it does not not exists. But if it's not inside the main memory (DRAM); it must read it from the disk into the DRAM & then access it. That's because DISK is extremely slow & CPU is fast & ∴ there is a mismatch in Speed. & DRAM is fast enough to keep up with CPU.

A write operation will update the copy of the disk page in memory & marks it as dirty; to indicate that the copy inside the DRAM is now different than the copy on the disk. one may write the page at different times, may write it a thousand times & it's dirty, but you can flush it only once, & sync with the copy in Disk.

Virtual nodes

Each node may consist of one or several processors, & one or several disk drives.

Why use multiple node?
- Speed up: perform a complex query/update faster.
- Scale up: Maintain the same speed for a complex query/update as the database grows by increasing the number of nodes.

GAMMA

Gamma executes query trees, so even though it's SQL & reduces it to relational algebra, & then subsequently it executes the algebraic expression also represented as a query tree.

Architecture
- Each node stores its fragment on its local disk.
- Each node may build a B+tree (clustered/non-clustered) & hash index on its fragment of a table.
- Each node has its own concurrency control & crash recovery protocol.

Divide & conquer the data; (fragments / pieces of the original table) horizontally partition the table.

Horizontal clustering How?
- Random
  take the records & randomly distribute them across the nodes.
- Round-Robin
  take the first row, put it on the first node; second row -- second node.

components of the system.

The 1st thing it does is to partition the data across nodes where each node is represented as a disk; showing fragments scattered across your disk drives. we have a scheduler process, crash recovery, concurrency control deadlock detection mechanism & at the very top is the catalog manager, that acts as a central repository of all conceptual & internal schema information for each database. Catalog manager also stores meta-data, used for query optimization.

Query Manager is associated with each user that is outside the system & is responsible for caching schema information locally & provides an interface for ad-hoc queries, query parsing, optimization & compilation by using the catalog managers meta-data.

Scheduler process, while executing a complex query that requires participation of 2 or more nodes, scheduler process is involved. It tries to stay autonomous so that to minimize its overhead.

Operator process partly much implements different algebraic operators — select, project, join operators. These operators are pointed at system initialization time, to avoid the overhead.

Software Architecture, the process is executing on one node & shared memory, so they're very similar to today's threads.

Hash-Join operator consists of 2 distinct steps:
- Build: constructs a hash table on the join attribute. (inequality joins — executed one after another)
- Probe: look up the hash table for a joining record.

How does an operator know where to send its stream of records?

Scheduler informs them the records themselves never go to scheduler (contents & co-ordinating)

It is just starting these processes & handing off port numbers.

Software Modularity split tables.

By using hashing; where the scheduler provides telephone numbers, port numbers & essentially the data is going to use a hash function to route the records to appropriate process for staging & for probing of the hash tables.

Node Failures.

To maintain the data available in the presence of node failures, by constructing a primary & a back-up replica of fragments.

How does it do that?

→ Interleaved declustering

Failure of Node1 distributes its load on Node 0, 2 & 3 & by increasing the cluster size, the probability of the data becoming unavailable goes up.

→ chained Declustering (supersede than interleaved)

In presence of failure, push a fraction of the load of a failed node to its adjacent node, balancing the workload across all nodes.