Distributed dbms

8977571987

Centralized dbms:
Three types:
Single user
Multi user eg:mysql
Embedded eg:smart phone

Parallelism:
Coarse grained
Small no of processor cores
Fine grained
Large no of processor cores

Atomic instructions:
Test and set
Used for setting locks on memory locations
Compare and swap:
Atomically change contents of memory locations

Parallel dbms:
Handling large scale of data
Operator parellilsm
Linear scale-up

Communication topologies:
Bus
Ring
Mesh
Hypercube
Treelike

Shared memory architectures:
Shared memory:
Both disk and main memory can be shared
Shared disk:

Only disk is shared
Shared nothing(cluster architecture)
Hierarchical(combination of all i.e.group of processors shared disk and memory
Together they are connected via shared nothing)

Data partitioning strategies:
Horizontal
Vertical
Horizontal partitioning
Round Robbin
hash
range(b+tree)

Scanning distributed table:
Entire table scan
(Round Robbin is suitable)
Point queries(where clause is used hash is suitable)
Range queries(range values in where clause range partitioning is suitable)

Data skew:
Some values of attribute having many more rows of others also called as power - law or zipf distribution
Round robin is less affected with data skew(does not take into consideration perticular attribute values)
Hash and range partition can be affected
Use balanced range partitioning vector(
Use histogram or random sampling
Good for range partition only)
Virtual node mapping:
Virtual node I mapped to real node(I-1)%n+1
Round robbin
More virtual less physical node
Many virtual nodes mapped to single physical node
Virtual node mapping
Not suitable for high transactional data
Dynamic repartitioning on top of virtual nodes:
Split virtual node into two similar to b+tree node when it is full
Virtual nodes maintain partition table(in master node)
Consistent hashing
(Hash ring / distributed hash table)

Replication within data center(tree like topology)

Replication across data center

Parallel index mgmt:
Local index:
Built on tuple that store on perticular vertical or physical need not be on partition.
b+tree or hash
Global index:
Index on local indices on node
Global primary index:
Based on distribution strategy—round robin, hash or range
Clustered index
Global secondary index:
Unclustered index

Distributed file system:
Stores across machines
Google file system
Hadoop distributed file system open source implementation of gfs

Hadoop—master-slave architecture

Datanode:
A compute machine that stores distributed file blocks of its share
Blocks have associated ids for identification and retrieval.
Name node:master node to store mapping of disk blocks ids to their compute node
consistency with replication
Reads are easy but writes are non trivial so append only file system used
Distributed file system not for online transactional processing
Parallel key value store:
Same as dist hash table
Used for store of large no of small groups
Key value pair
Put and get
Berkeley db, google big table,casandra,hbase,microsoft azure table storage

Updates on distributed data:
Complex need to take care of replication and node failure
Bigtable and base built on immutable or append only file system
These are stepped merge variant of log structured merge trees(LSM tree)
LSM tree:

- separate index on tree
- stored in in memory tree when fills moves to disk tree
- several b+ tree