Ferhat Elmas 214805

Advanced Databases Hw-4

1 - a) For key in mappers, I use join key (customer-id) and an identifier for the tables since we have one-to-many relation. Mappers sort their inputs in terms of join key and Customer table is processed before Order table. In our dataset, it seems they are already ready to read tuples one by one and apply above key conversion. Moreover, I remove customer-id from value part since it is already in key and to save bandwidth.

In mapper 1: <Adrian, C> → 5
<Baker, C> → 3
<Adrian, O> → 100
<Adrian, O> → 200
<Adam, O> → 50
<Adam, O> → 300

In mapper 2: <Adam, C> → 10
<Bridge, C> → 8
<Baker, O> → 450
<Baker, O> → 600
<Bridge, O> → 1000
<Bridge, O> → 360

Then, we need a partitioner since keys are composite but we want to reduce according to join key means that same join key must go to the same reducer.

Reducer only needs keep one tuple that is the new tuple comes from Customer table and one booking variable to sum because of above explanation and the structure of tables we are sure tuples with same join key will come to same reducer and first Customer tuples, then Order tuples. When reducer encoutered new key keeps it and sums following values and again new key encoutered, kept value is emitted.

In reducer 1: <Adrian, C> → 5
<Adrian, O> → 100
<Adrian, O> → 200
<Adam, C> → 10
<Adam, O> → 50
<Adam, O> → 300

In reducer 2: <Baker, C> → 3
<Baker, O> → 450
<Baker, O> → 600
<Bridge, C> → 8
<Bridge, O> → 1000
<Bridge, O> → 360

2 - Speed up: The time taken by the operations decreases proportionally to increased number of CPUs and disks. It means more hardware can perform the same task faster. For example, if we currently perform an operation in 100 ms and we double the hardware and system speeds up, we will be able to perform it in 50 ms.

Scale up: The performance of the system is maintained if CPUs and disks are increased proportionally to the amount of the data. For example, if the system process the data in 100 ms, then we double the data and CPUs and disks, it will be able to process in 100 ms again.

Speed up enables us to solve problems faster while scale up enables us to solve bigger problems.

3 - Synchronous replication of updates: A technique that ensures that transactions get the same value from any copy of the object they access. It mainly has two different methods, voting and write all. Transaction must write majority of the copies to update an object and read at least enough copies to make sure that one of the read is up-to-date. This has disadvantages of reading multiple times for an object. Generally, objects are read more than written therefore, efficient reading is needed for the performance. Write all method requires to write all replications on a update but object can be read from any of the replicas. However, this makes writing so costly. This kind of replication ensures atomic property (good +) but uses multiple reads and writes by a commit protocol which degrades performance (bad -).

Asynchronous replication of updates: Copies of an object are updated periodically. Therefore, multiple transactions can read or write the same object. Main advantage of this technique is elimination locking that is necessary for synchronous scheme so that means better performance. However, it comes with its own trade-offs because it violates the principle of distributed data independence; users must be aware of which copy they are accessing, recognize that copies are brought up-to-date only periodically , and live with this reduced level of data consistency.

4 - Requirements of a LAN are different than those of the distributed database so it is difficult. Each LAN has its own policy: different system admins, different access and charging algorithms and different constraints on servicing remote requests. Typical optimizers calculate the cost and do processing accordingly but in WAN case, other constraints must be included in cost function such as access or time of day. Therefore, we need to update / improve cost function, due to this reason we don’t expect to see distributed database in WAN. However, main concern of the parallel databases is to improve the performance by parallel executions of some operations. Data can be kept in distributed fashion or not. Main goal is using multiple machines and processors to run queries in parallel. Therefore, parallel databases have less conflicting requirements with WANs which makes seeing parallel databases in WAN more reasonable than distributed databases.