Summer Term 2020

Advanced Topics in Databases

Exercise 1

1. Compare disk-based DBMSs and main-memory DBMSs! What advantages and disadvantages
Emerge using main memory as primary storage?

Disk based DBMS-

More space in disk and data is stored in disk primarily
High latency between main memory and disk data transfer
Less used main memory space while much more is available in new systems
Unused CPU processing power
Only current used data is buffered in main memory

Main Memory DBMS-

More space in main memory and data is stored in main memory and backup data is stored in disk
Latency is between main memory and CPUs
CPUS are most of the time sitting idle because of less efficient use

Advantages because of main memory-

No latency between disk and main memory
Replicated data storage in disk
Efficient use of available main memory

Disadvantages because of main memory-

Memory wall, increasing GAP between CPU and memory speed
Idle CPU most of the time

2. What is the reason for the main-memory access bottleneck? Why is the main-memory? Access bottleneck a problem in main-memory DBMSs, but not that important in traditional Disk-based DBMSs? How can the main-memory access bottleneck be addressed?

Bottleneck is called the memory wall, which is the speed difference between memory speed and available CPU speed. CPUS most of the time sit idle waiting for the memory.

It is a problem because in case of processing a query CPU shall finish processing its part and wait for data to be retrieved from main memory to process further hence wasting a lot of time. Sitting basically doing nothing, which could have turned the processing faster.

In older system there is another huge bottle neck which is between disk and memory which is relatively bigger than between main memory and CPUs.

We can address this issue by cache awareness and there are lot of methods available in this area.

3. Which of these database algorithms benefit from spatial and/or temporal locality?

• Block Nested Loop Join

Spatial since loops

• Merge Join on sorted Tables

Spatial since data is already sored and just needs to be merged, placed closed

• Full-Table Scan

Spatial since each rows is accessed in a sequential manner which is closer

• Range Selection using a B+-Tree

Temporal since range of rows shall be selected re using the data and B+ tree shall keep re suing it after sorting

• Hash-based Aggregation

Since it needs grouping which might be far apart based on conditions hence Temporal

4. What is the memory hierarchy? How is it used during memory access? Describe the Basic principle behind caches?

From the top we have registers, then cache then memory and then disk typically.

From the disk data is retrieved from main memory then in CPU it's accessed between cache and registers. There can be level of cache and then in the end it's stored and processed into registers.

Caches keeps and process the data into small spaces temporarily which is being used frequently hence supporting the current operations.

5. Name and explain three approaches for block placement within caches. What are? Their advantages and disadvantages?
Fully associative caches- Block can be loaded into any cache line
Adv. Offer freedom for block replacement strategy
Disadv. Doesn't not scale to large cache

Direct Mapped Cache-a block has only 1 place it can appear in cache
Adv. fast and simpler to implement, known location
Disadv. Chances of conflicts increases.

Set Associative Cache- group cache lines into sets , each memory block maps to one set and can be placed anywhere in the set
Adv. Adv. of freedom of replacement within set
Disadv. Sets can have conflicts but manageable

6. For each of the following scenarios, name the best block placement strategy. Explain Why it is the best.
• Minimal amount of cache line evictions
Direct mapped since only 1 can get and hence only 1 shall be evicted.
• Minimal search cost for a cache line
Direct Mapped cache because address is known in advance or may be fully associative also because whichever is free first
• Very small caches
Fully associative since can fit anywhere whichever is free
• Accessed memory addresses are evenly distributed
Direct Mapped since it can amp t to only one and since evenly distributed less chances of conflict
• Accessed memory addresses are skewed, but minimal search cost is needed
Set associative since know sets hence less cost and skewed which can be within sets

7. Explain the terms poor data locality and poor code locality with respect to the traditional Processing model (tuple-at-time processing) using the following query plan.

Poor code quality here is not selecting the title earlier on from the table it's placed in, it's a long process to get the selected column when all we need is a title column. It could have been selected earlier.

For poor data quality data is being stored in different tables which are far placed, author and borrowing date. For each row all these functions has to ben re run.