Lecture Summary

Date: 27th September, 2011

Prepared by –

1. **Abhinandan Nath 09010101**
2. **Abhinav Sonker 09010102**

TOPIC: Introduction to physical optimisation

IMPORTANT POINTS:

1. **Capability of a database depends only on the logical model followed. Physical storage pattern only affects performance.**

2. **Memory Hierarchy :**

    a. **Cache :**

        Size of cache - typically of the order of MBs
        Speed – order of GBps

    b. **Main Memory :**

        Slower than cache
        Size: order of GBs

    c. **Flash Memory :**

        Cheap and fast but cannot replace main memory
        Why? Because it has fixed no. of reads and writes and is block based

    d. **Magnetic Disk :**

        Organised into physical and logical parts
        Physical parts: Platter, read-write head, tracks
        Logical parts: Sectors
        Large data in one read-write is preferable over smaller data in many read-writes : because of physical adjustments of read-write head

    e. **Optical Disk :**
    f. **Magnetic Tapes :**

        Very secure and cheap, used for storing important information like records, logs etc. and in banks.
        As it supports only sequential access, it is quite slow. It is generally used for storing data that do not change very frequently.

3. **Optimization techniques :**
    a. **Buffering :**

        Exploits locality of data in space and time. Provides faster access.
        Used in programming languages, for example in reading and writing from interfaces and files.
        Disadvantages : consistency problems.
        If data size is comparable to buffer size, then random access is not a problem. But if data is too large than buffer, then frequent buffering is required.
        If locality is not used, then no gain from buffering.
        Performance depends on parameters like buffer size, data size and disk size.

b. Pre-fetching:

Using history of data accesses like query patterns, guess data for next requirement and fetch it earlier than requested.
Performance gain depends on access history and algorithm correctness.
Aggressive vs. Defensive pre-fetching : If guess is wrong, then in aggressive fetching, fetch more and more data – probability of correctness increases. In defensive, fetch lesser data.

Difference b/w buffering and prefetch : In buffering, data is fetched on request, in prefetching data is fetched before request by guessing.

c. Scheduling :

Scheduling a bunch of requests to reduce fetch time.
Disadvantage : Computation is done "on the fly"

4. FILE STRUCTURE :

2 types: fixed length vs. variable length records

Locating and modifying (insertion, deletion) is easier in fixed length records than in variable length records.

For Ex;- Expanding a record in variable length structure creates problem and if we use shifting then for n records it takes O(n) time. We need to use other techniques to solve such problems in fixed length records.

Storage space is used more efficiently in variable length records than in fixed length records.