DATA STORAGE TECHNOLOGIES & NETWORKS

(CS C446, CS F446 & IS C446)

LECTURE 07 – STORAGE

Memory Hierarchy

- What makes the system work?
  - i.e. what makes it possible to access the fast but small memory most of the time so that Memory efficiency is high in practice?
- Locality of References:
  - Locus of memory references – at any point during execution of a typical program – is small.

Memory Hierarchy

- "Locality of References"
  - Temporal Locality
    - Same data (and therefore locations) are likely to be accessed repeatedly – over a small duration of time
      - E.g. Instructions in a loop body;
      - E.g. data locations accessed in a loop body.
      - E.g. local variables of a function/object/data structure

Memory Hierarchy

- "Locality of References"
  - Spatial Locality
    - Data (and therefore locations) in the neighborhood are likely to accessed in near future.
      - E.g. Straight line instruction sequence (no jumps)
      - E.g. array elements.
        - Special case: Sequential (Stride-1), array elements
      - E.g. streams
    - Note on the notion of "neighborhood":
      - On data: "neighborhood" is a logical notion
      - On (memory) locations: "neighborhood" is a physical notion
    - How is one mapped to the other?

Memory Hierarchy

- Recall

  $M.E = 100 / (1 + P_{miss} (R-1))$

  - Where $R = T_{h+1} / T_h$
- Performance Gap
  - What if two levels have a large R?
- Solution: Caching brings the levels closer
  - caches are usually transparent

Caching

- L1, L2, and L3 caches between CPU and RAM
  - Transparent to OS and Apps.
  - L1 is typically on (processor) chip
    - R=1 to 2
    - May be separate for data and instructions (Why?)
  - L3 is typically on-board (i.e. processor board or “motherboard”)
    - R=5 to 10

Caching - Generic

- Caching as a principle can be applied between any two levels of memory
  - e.g. Buffer Cache (part of RAM)
    - transparent to App,
    - maintained by OS,
    - between main memory and hard disk,
    - $R_{RAM,buffer} = 1$
  - e.g. Disk cache
    - between RAM and hard disk
    - typically part of disk controller
    - typically semiconductor memory
    - may be non-volatile ROM on high end disks to support power breakdowns.
    - transparent to OS and Apps