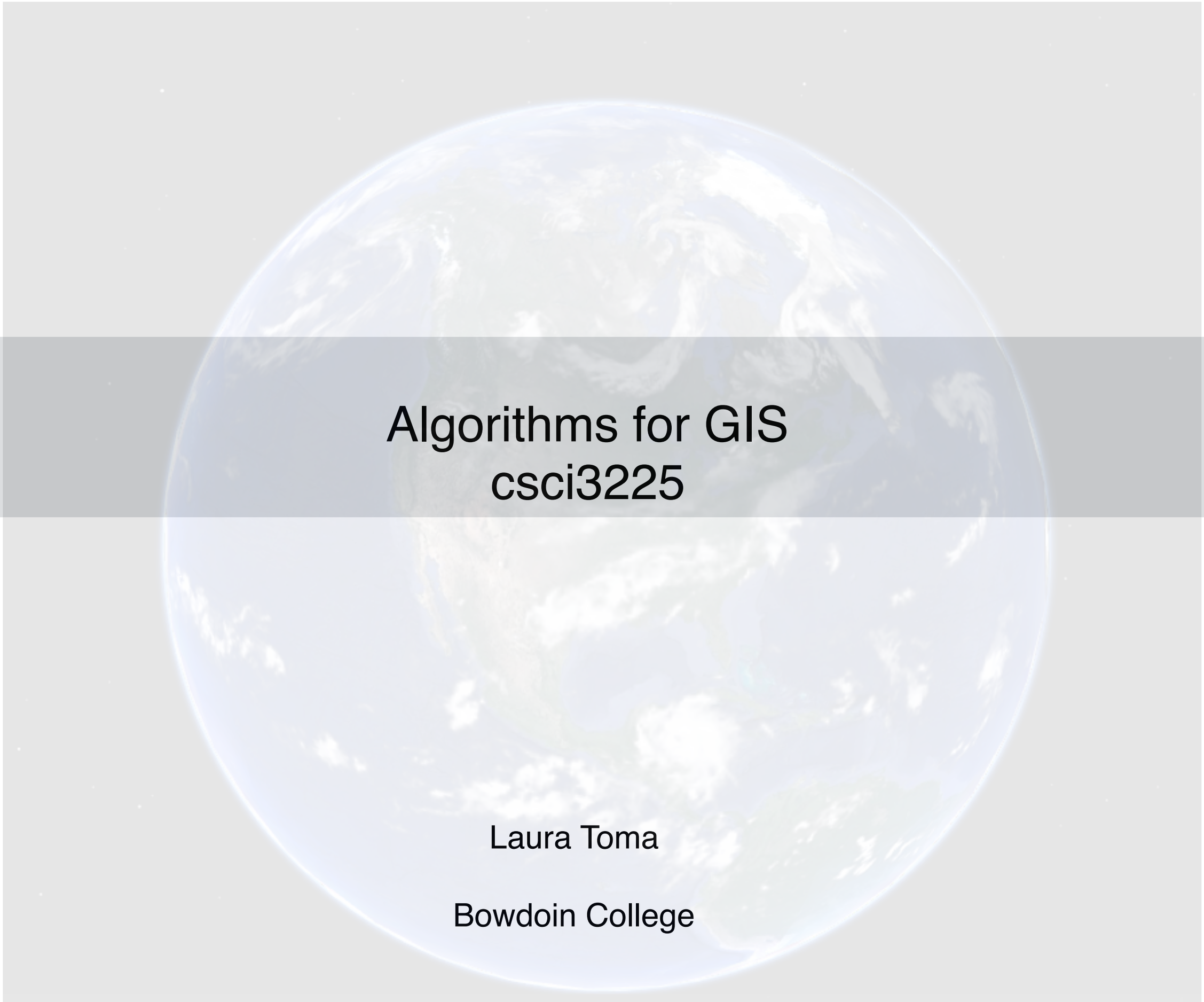


Algorithms for GIS
csci3225

Laura Toma

Bowdoin College

Limitations of the RAM model

Algorithm design and analysis

- By default, algorithms assume the RAM model

- The RAM model
  - all instructions take the same amount of time
  - all data accesses take the same amount of time
  - ..

- Algorithm complexity in the RAM model:
  - the number of operations executed
  - asymptotic analysis: O(), Theta(), Omega()

- Goal: design algorithms that optimize the RAM-complexity

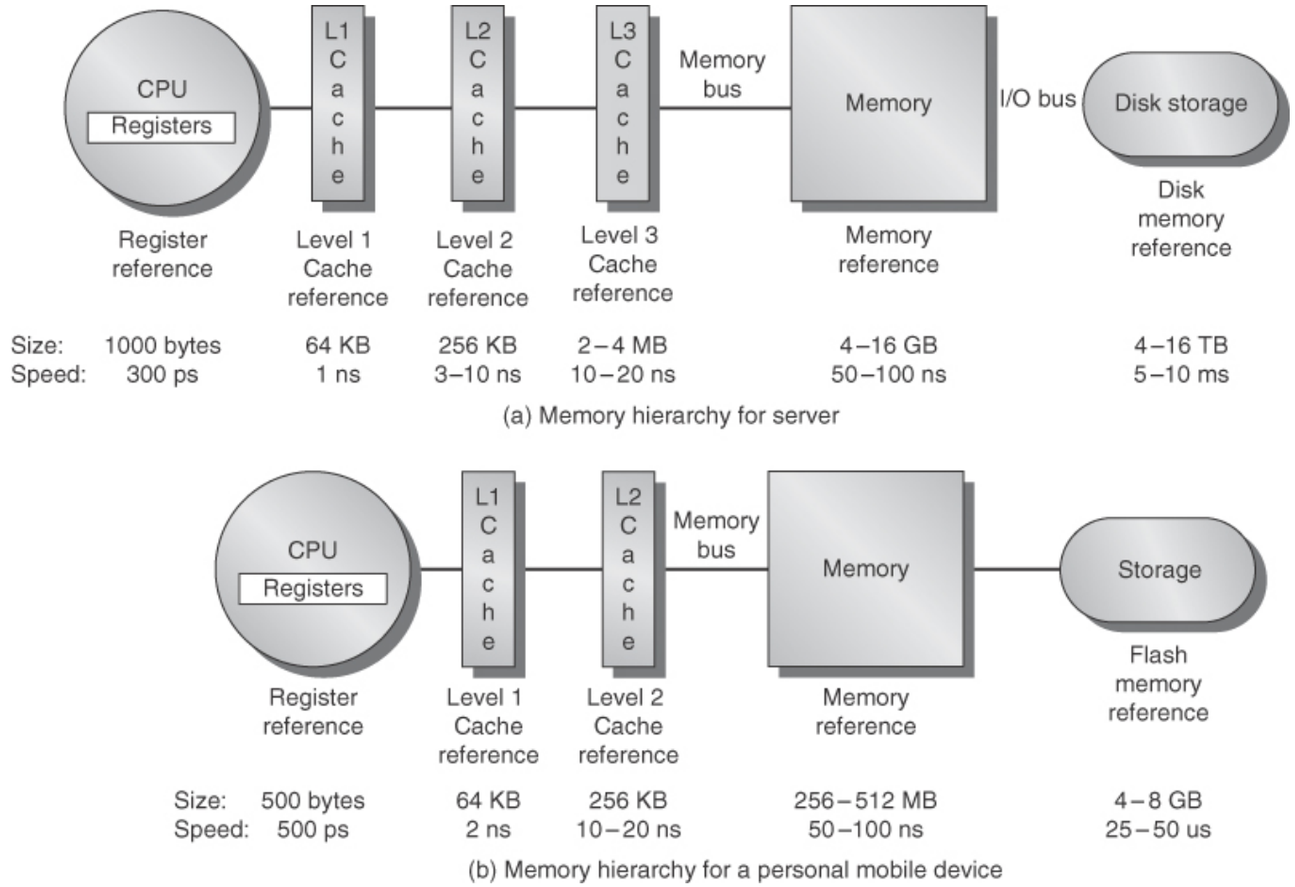


Figure 2.1 The levels in a typical memory hierarchy in a server computer shown on top (a) and in a personal mobile device (PMD) on the bottom (b). As we move farther away from the processor, the memory in the level below becomes slower and larger. Note that the time units change by a factor of 109—from picoseconds to milliseconds—and that the size units change by a factor of 10^{12}—from bytes to terabytes. The PMD has a slower clock rate and smaller caches and main memory. A key difference is that servers and desktops use disk storage as the lowest level in the hierarchy while PMDs use Flash, which is built from EEPROM technology.

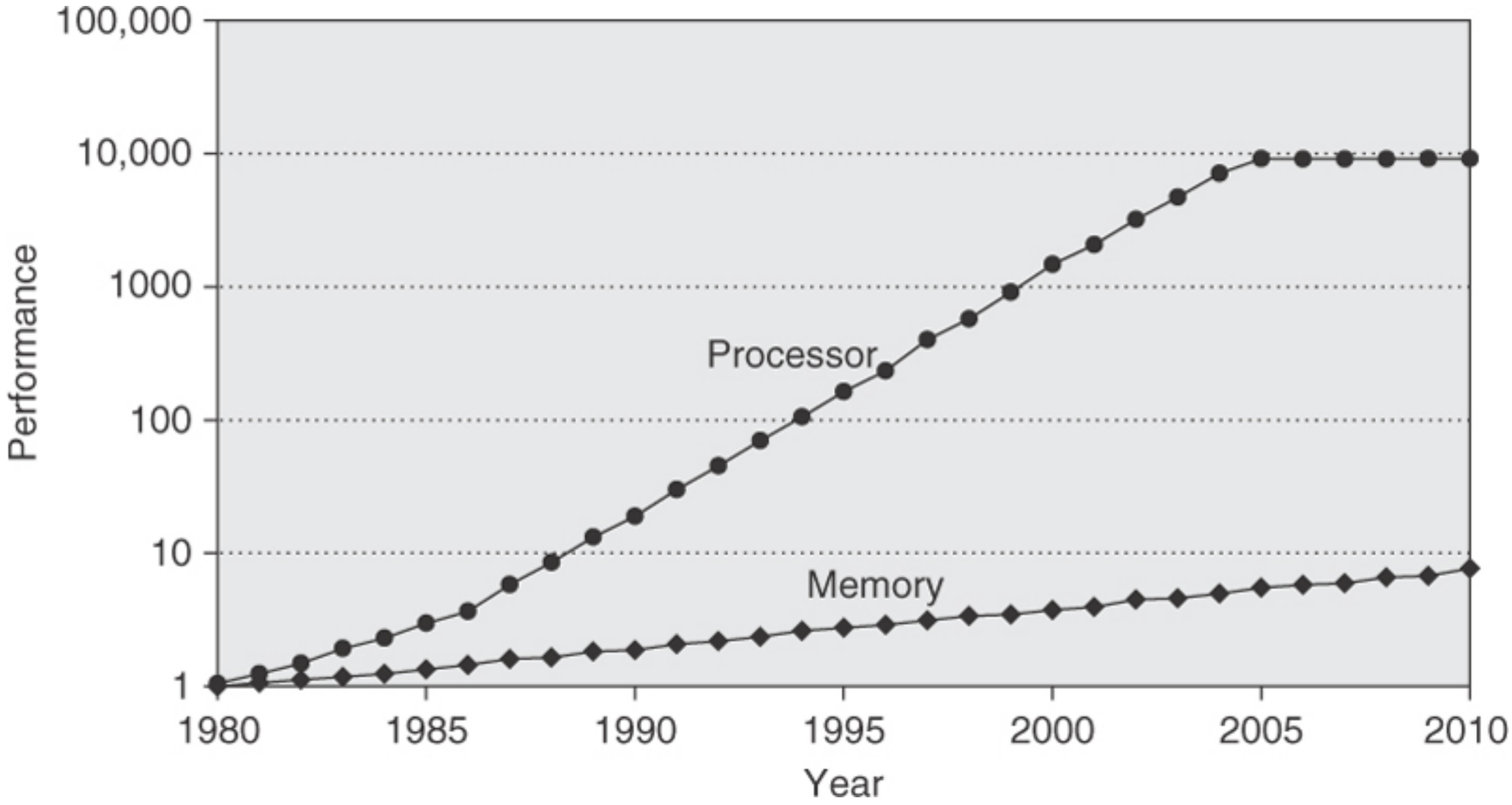
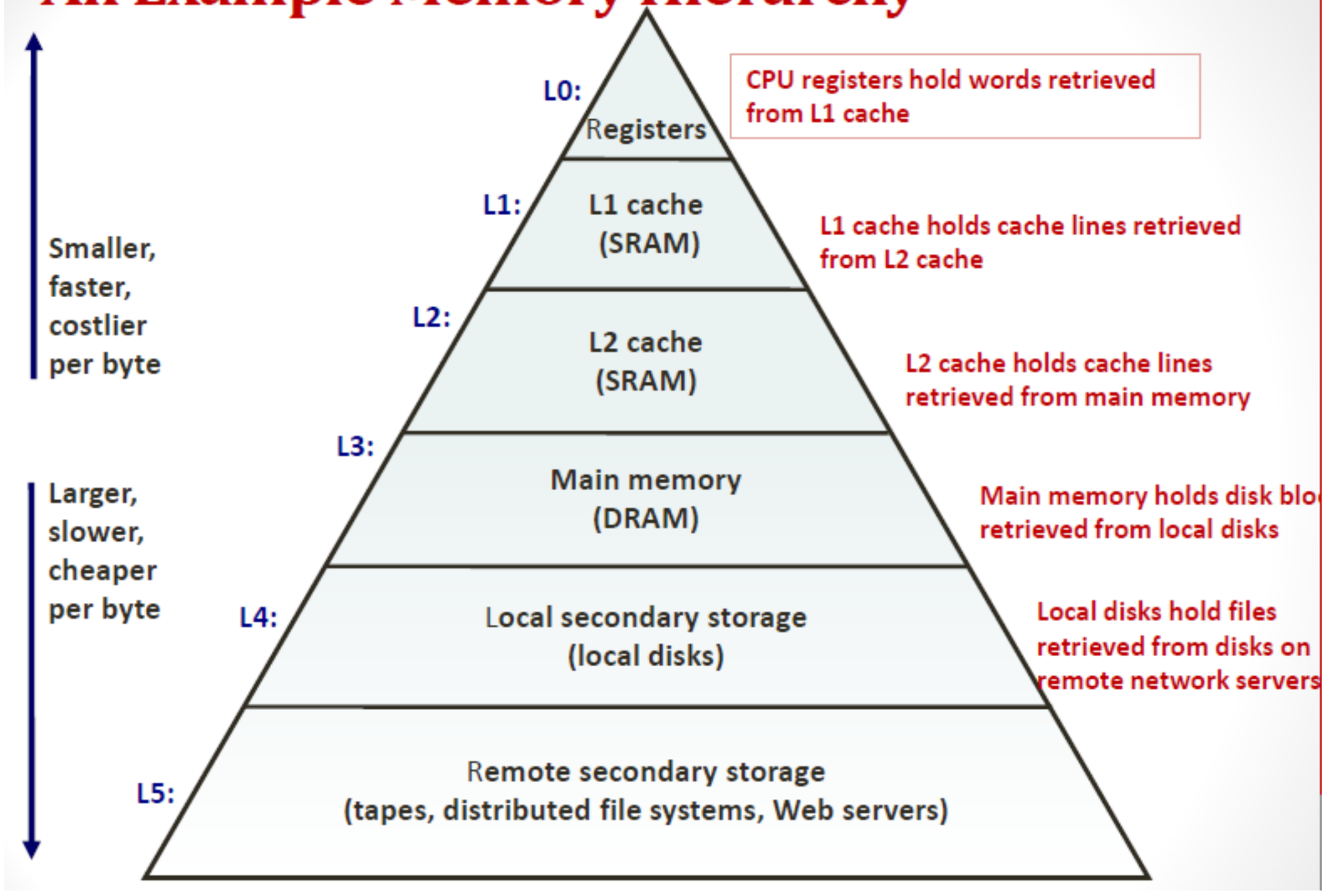


Figure 2.2 Starting with 1980 performance as a baseline, the gap in performance, measured as the difference in the time between processor memory requests (for a single processor or core) and the latency of a DRAM access, is plotted over time. Note that the vertical axis must be on a logarithmic scale to record the size of the processor–DRAM performance gap. The memory baseline is 64 KB DRAM in 1980, with a 1.07 per year performance improvement in latency (see Figure 2.13 on page 99). The processor line assumes a 1.25 improvement per year until 1986, a 1.52 improvement until 2000, a 1.20 improvement between 2000 and 2005, and no change in processor performance (on a per-core basis) between 2005 and 2010; see Figure 1.1 in Chapter 1.

An Example Memory Hierarchy



http://classconnection.s3.amazonaws.com/149/flashcards/3088149/png/memory_hierarchy1367201501848.png

More realistic models

- But…the RAM model ignores the memory hierarchy (among other things)
  - Two algorithms that have the same big-Oh can differ **a lot** in performance depending on their cache and IO efficiency

- To analyze and fine tune the algorithm we need to look at the performance across the memory hierarchy

- CPU-efficient algorithms
  - CPU-complexity = standard complexity in the RAM model
  - Goal: optimize CPU-complexity

- Cache-efficient algorithms:
  - cache-complexity = number of cache misses
  - Goal: optimize cache-complexity
- IO-efficient algorithms:
  - IO-complexity = number of page faults
  - Goal: optimize IO-complexity

- Ideally, we'd of course want to optimize everything (not always possible)