Lecture 4

Wednesday, January 15, 2025 9:26 AM

1. CPU
2. Registers
3. L1 Cache
4. L2 Cache
5. RAM
6. SDD/HDD
    - Lots of storage
    - Persistent
    - SLOW

64 bit integers -> 8 bytes
2048 byte block size

- Have to read the whole block to access a specific 64 bit int

DB systems -> min HDD/SDD Accesses

AVL Tree

- 7 nodes, each 32 bytes
- 7 x 32 bytes of memory needed
- If each node is stored in a different block...
    - Might need 7 x 2048 bytes
    - Very inefficient

Sorted array of 128 integers

- Worst case binary search is WAY FASTER than a single disk access

Shallower trees require less disk accesses

B+ Tree

- A B-tree of order m is a search tree in which each non-leaf node has up to m children. The actual elements of the collection are stored in the leaves of the tree, and the non-leaf nodes contain only keys. Each leaf stores some number of elements; the maximum number may be greater or (typically) less than m. The data structure satisfies several invariants:
    - Every path from the root to a leaf has the same length
    - If a node has n children, it contains $n-1$ keys.
    - Every node (except the root) is at least half full
    - The elements stored in a given subtree all have keys that are between the keys in the parent node on either side of the subtree pointer. (This generalizes the BST invariant.)
    - The root has at least two children if it is not a leaf.