Performance Engineering on CPUs and GPUs:

- CPU and Memory: Things to be Careful for Performance -

Kamer Kaya, Sabancı University

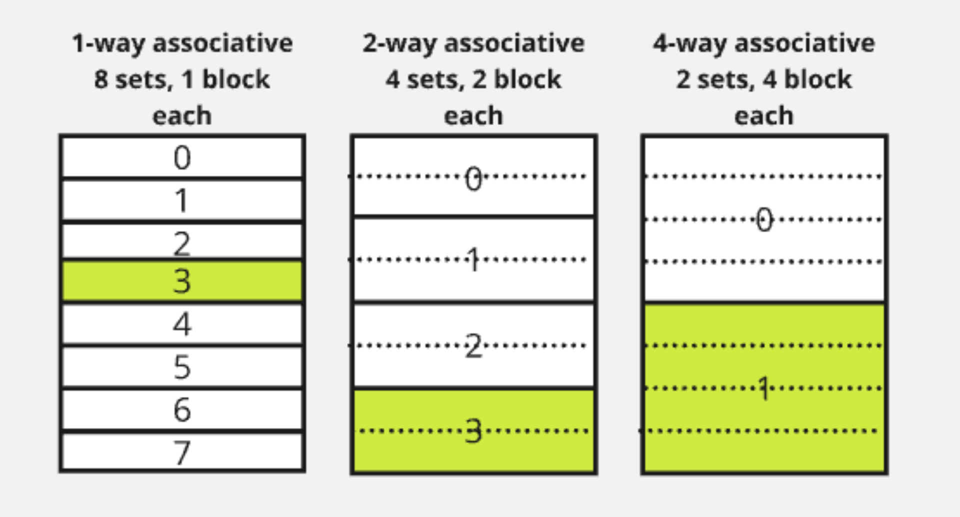
ncc@ulakbim.gov.tr

Cache: Set Associativity

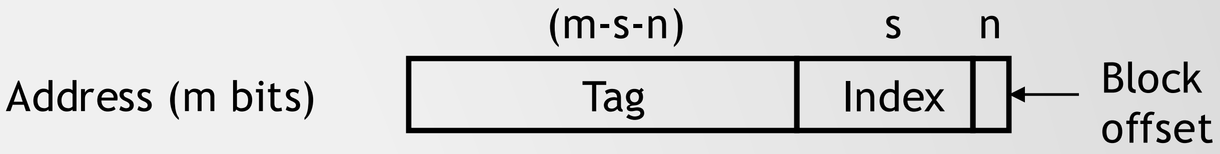
Caches we have are usually set-associative.

- The cache is divided into groups of blocks, called sets.
- Each memory address maps to exactly one set in the cache, but data may be placed in any block within that set.

If each set has 2^x blocks, the cache is 2^x-way associative cache.



- If a cache has 2^s sets and each block has 2^n bytes, the memory address can be partitioned as follows.



- Our arithmetic computations now compute a set index, to select a *set* within the cache instead of an individual block.

Block Offset = Memory Address mod 2^n
Block Address = Memory Address / 2^n
Set Index = Block Address mod 2^s

Cache: Set Associativity

Where would data from memory byte address 6195 be placed, assuming the eight-block cache designs below, with 16 bytes per block?

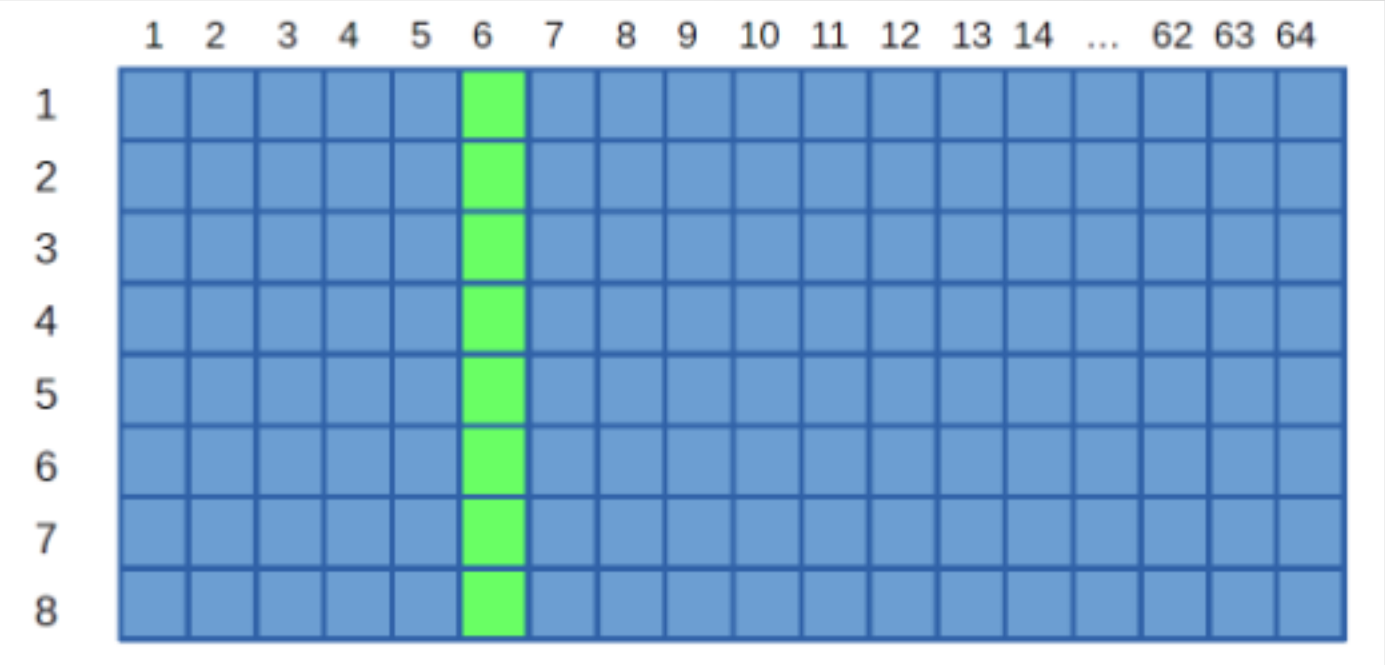
- 6195 in binary is 00...0110000 011 0011.
- Each block has 16 bytes, so the lowest 4 bits are the block offset.
- For the
  - 1-way cache, the next three bits (011) are the set index.
  - 2-way cache, the next two bits (11) are the set index.
  - 4-way cache, the next one bit (1) is the set index.
- The data may go in *any* block, shown in green, within the correct set.

Cache: Set Associativity

- The 32KB of L1 data cache in a core can therefore be envisioned as a three-dimensional box, where:
  - Depth represents the size of a cache line, e.g., 64 bytes
  - Height represents the extent of a cache set
  - Width represents the number of sets that are available
- After doing a few quick calculations, we can find the relevant properties for the L1d cache of a core, which holds 32 KB divided into 64-byte cache lines and is 8-way set associative:
  - Bytes in L1d = 32 KB * 1024 (bytes/KB) = 32768 bytes
  - Cache lines in L1d = 32768 / (line size) = 32768 / 64 = 512
  - Number of sets = 512 / 8 = 64

Cache: Set Associativity

- Recall that each square on the right represents an entire cache line (64 bytes in our case).
- When data at a particular address is requested, the congruence class of the address is computed, determining the cache set of the cache line containing the data.
- Then the entire line is fetched into one of the eight slots for that cache set.



(https://juejin.cn/post/6945477261197852703)

Cache: Set Associativity

What is the worst performance pattern for a cache like this?

Lets see the answer!

Thanks

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia