COMPSCI 590_S
Emery Berger

- projects
- exams (2) midterm / final
- homework - reviews

HotCRP

by noon T/Th

emery@cs.umass.edu

emeryberger.com/teaching

SCALE "up"

more processors

"out"

more machines

- correctness
- ease-of-use (e.g. Excel)
  R, Python

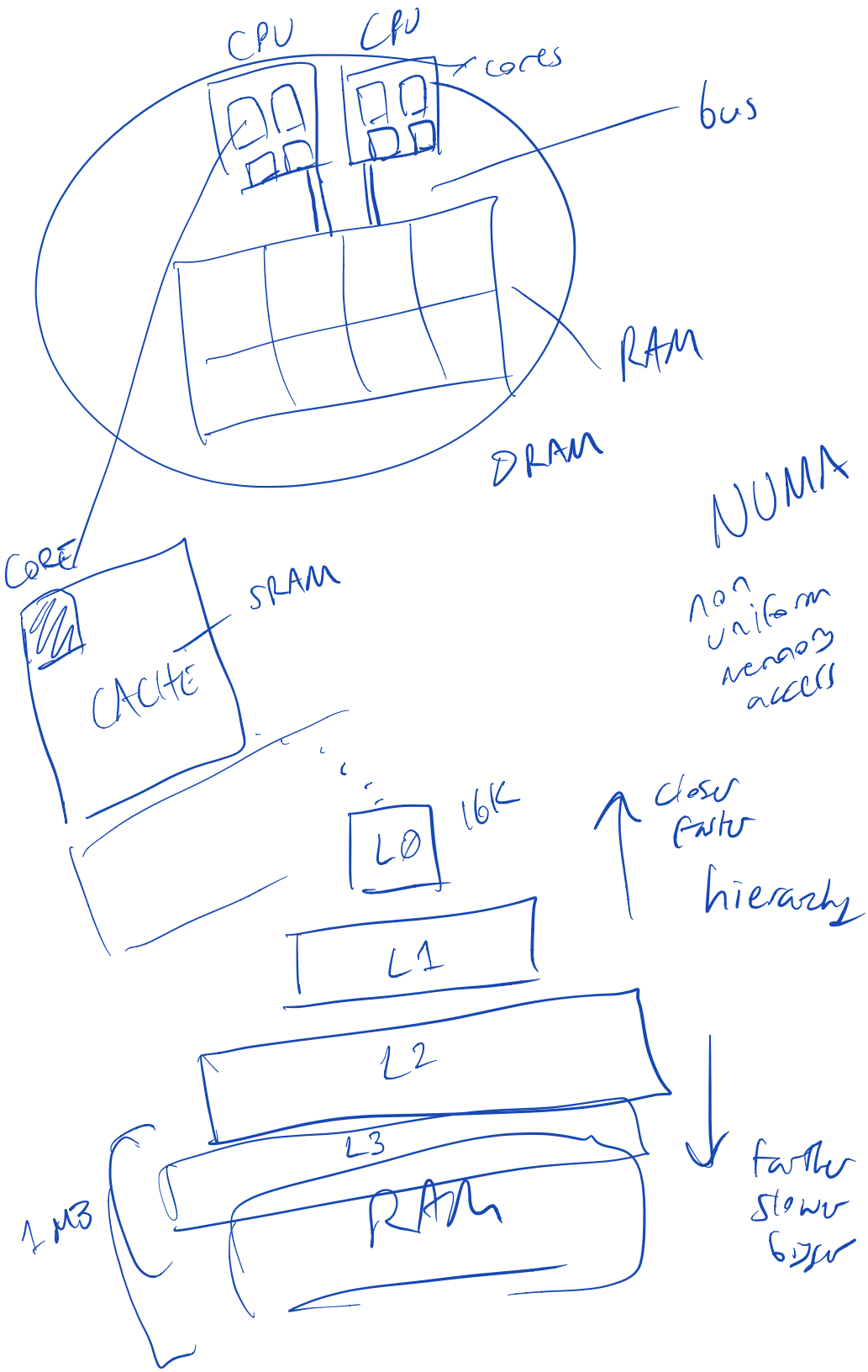
  API
  library
  abstraction

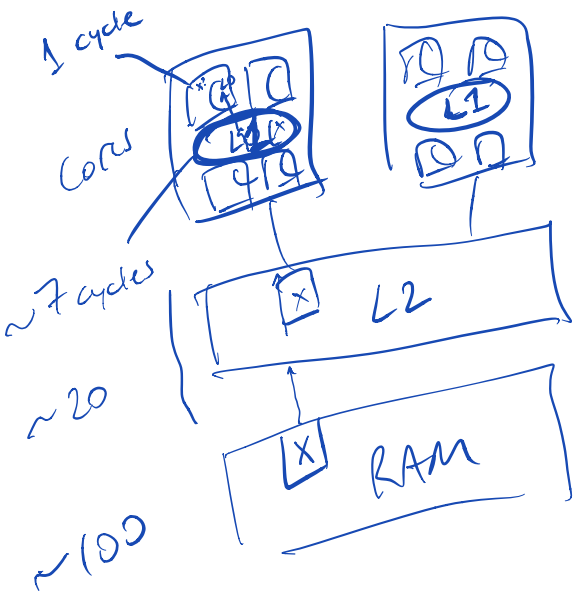
  easy to do your work

- performance — job done "quickly"
  throughput (batch)
  vs. latency (interactive)

- fault tolerance ("don't lose your work")

SQL
NoSQL


CPU
CPU
cores
bus
RAM
DRAM
NUMA
non uniform memory access
CORE
SRAM
CACHE
L0
16K
closer
faster
hierarchy
L1
L2
L3
RAM
1 MB
farther
slower
bigger



L0 exclusive

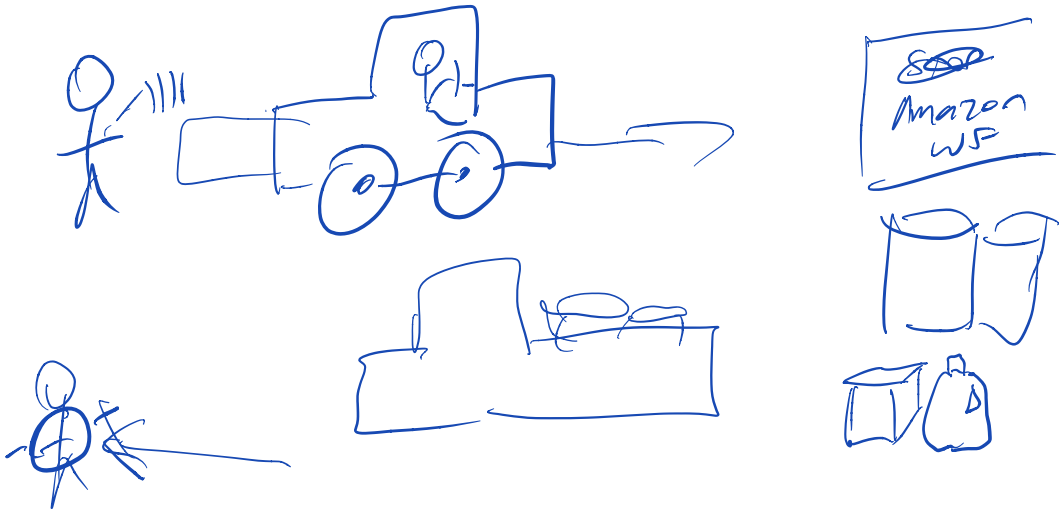
L1 shared by all cores on a chip

L2 shared by all chips

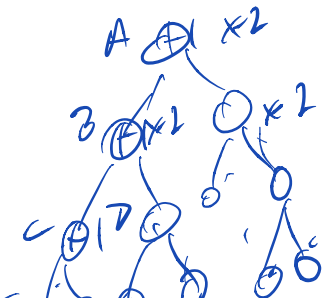
L = "level"

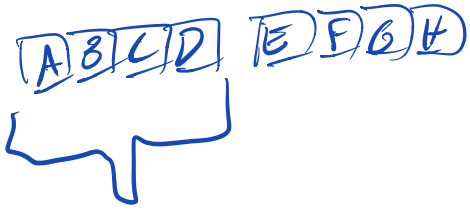
hit

miss

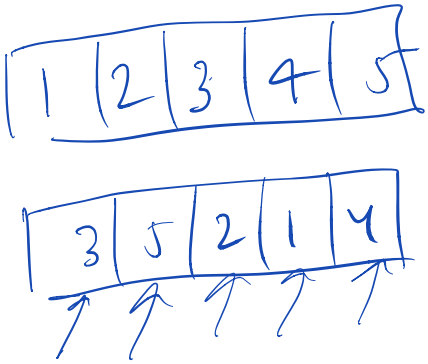


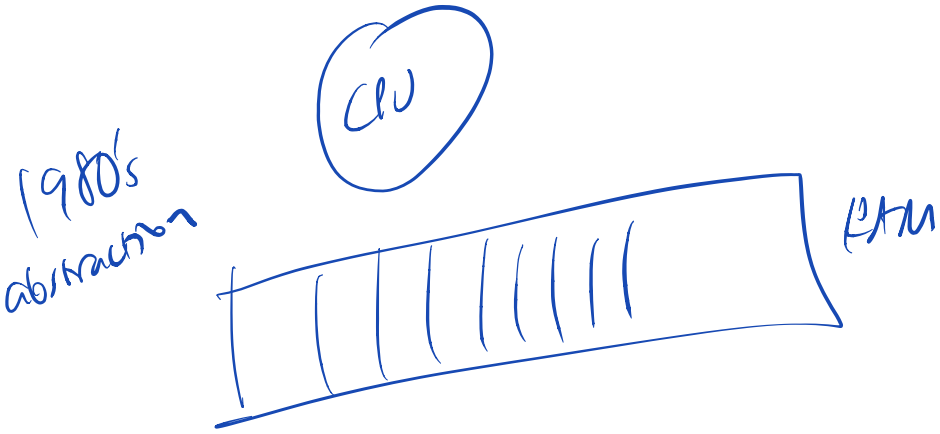
temporal locality

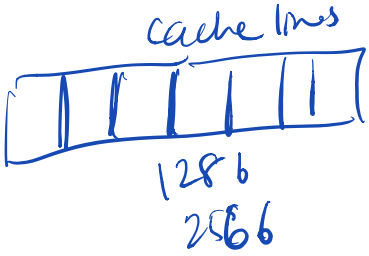


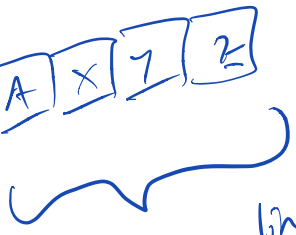


spatial locality

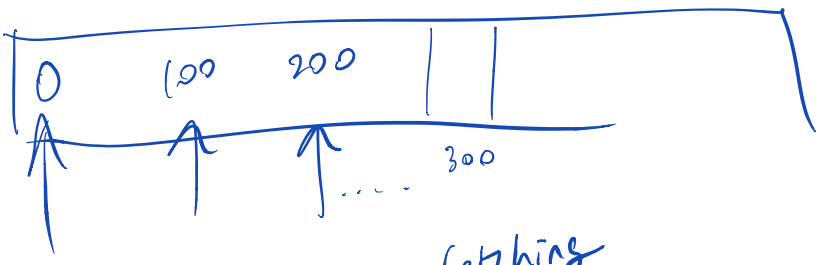




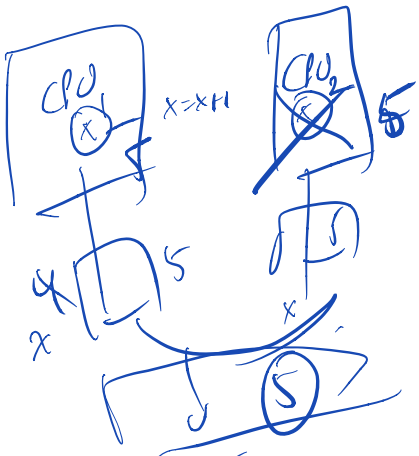




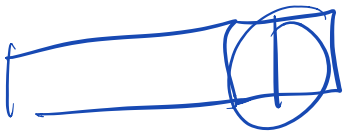
cache line
128 bytes
256 bytes



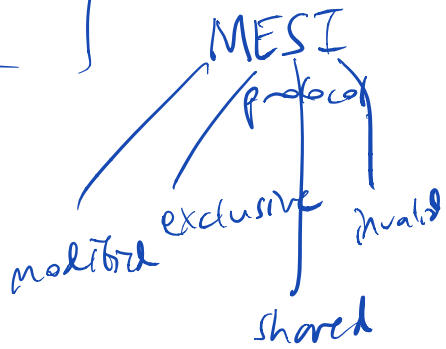
prefetching



Cache coherence



line



example

$w(L0) \rightarrow w(L0)$

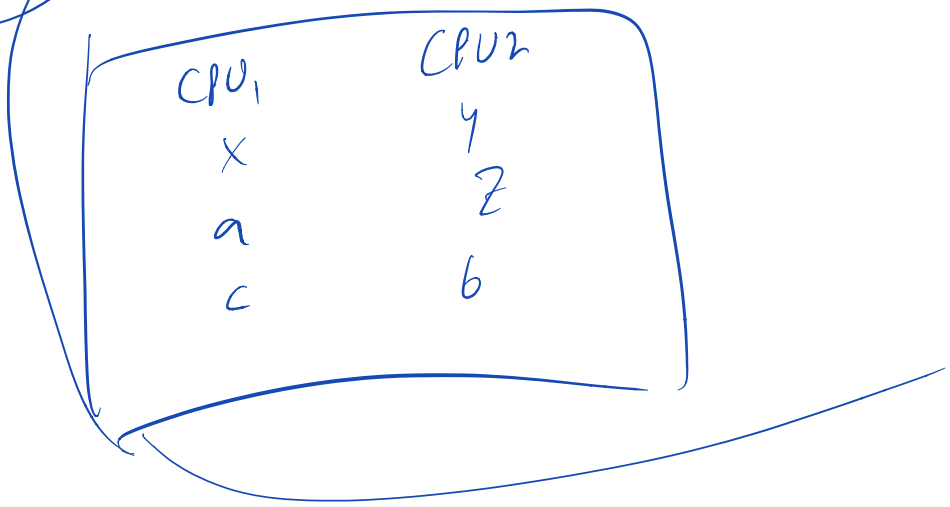
write-back

write-through $w(L0) \rightarrow w(L0), w(L1), w(L2) \ldots$

these are details!

Snoopy cache

directory-based cache coherence protocol



False sharing

logically unshared but reside in same unit of coherence