Thoughts on Memory

Prof. Bruce Jacob
Keystone Professor & Director of Computer Engineering Program
Electrical & Computer Engineering
University of Maryland at College Park
Main problem: We don’t understand it very well
How it is represented

if (cache_miss(addr)) {
    cycle_count += DRAM_LATENCY;
}

even in simulators with “cycle accurate” memory systems—no lie
Problem: Capacity

Commodity DRAM Devices Datarate:
- Doubling Every 3 Years

New Generations of DRAM Devices (time)

Channel Capacity (GB)

DRAM Data Rate (Mbps)
*DRAM Chip Bit Density

#DIMMs/Channel

SDRAM (min burst: 1)
DDR SDRAM (min burst: 2)
DDR2 SDRAM (min burst: 4)
Problem: Capacity

JEDEC DDRx
~10W/DIMM, ~20W total

FB-DIMM
~10W/DIMM, ~300W total
Problem: Bandwidth

- Like capacity, primarily a power and heat issue: can get more BW by adding busses, but they need to be narrow & thus fast. Fast = hot.

- Required BW per core is roughly 1 GB/s, and cores per chip is increasing

- Graph: Thread-based load (SPECjbb), memory set to 52GB/s sustained
  ... cf. 32-core Sun Niagara: saturates at 25.6 GB/s
Problem: Bandwidth

Sometimes bandwidth is everything ...

Cray Black Widow memory system

32 or 64 GB Local Memory, with 8 MB shared L3 cache

Memory Daughter Card

Weaver Memory Controller

667 MHz DDR-II DRAMs

BlackWidow ASIC

0.5 MB L2 Cache

20.8 GF CPU

NIF0

NIF1

network ports
9.36 GB/s per direction

Problem: Bandwidth

Sometimes bandwidth is everything ...

Cray Black Widow memory system

32 or 64 GB Local Memory, with 8 MB shared L3 cache

Memory Daughter Card

Weaver Memory Controller

667 MHz DDR-II DRAMs

BlackWidow ASIC

0.5 MB L2 Cache

20.8 GF CPU

NIF0

NIF1

network ports
9.36 GB/s per direction
Problem: TLB Reach

- Doesn’t scale at all (still small and not upgradeable)

- Currently accounts for 20+% of system overhead

- Higher associativity (which offsets the TLB’s small size) can create a power issue

- The TLB’s “reach” is actually much worse than it looks, because of different access granularities

Maps ~1MB

~10MB
Trend: Disk, Flash, and other NV

- Flash is currently eating Disk’s lunch
- PCM is expected to eat Flash’s lunch
Obvious Conclusions I

- Want capacity without sacrificing bandwidth

- Need a new memory system architecture

- This is coming (details will change, of course)

---

Figure X. A DRAM-system organization to solve the capacity & power problems

- Fast, wide channel
- Fast, narrow channels
- Slow, wide channel
Obvious Conclusions II

- Flash/NV is inexpensive, is fast (rel. to disk), and has better capacity roadmap than DRAM

- Make it a first-class citizen in the memory hierarchy

- Access it via load/store interface, use DRAM to buffer writes, software management

- Probably reduces capacity pressure on DRAM system
Obvious Conclusions II

• Flash/NV is inexpensive, is fast (rel. to disk), and has better capacity roadmap than DRAM

• **Make it a first-class citizen in the memory hierarchy**

• Access it via load/store interface, use DRAM to buffer writes, software management

• Probably reduces capacity pressure on DRAM system
Obvious Conclusions III

- Reduce translation overhead (both in performance and in power)

- Need an OS/arch redesign

- Revisit superpages, multi-level TLBs

- Revisit SASOS concepts, *location of translation point/s*

- Probably most suited for the high-end, at least initially
Acknowledgements & Shameless Plugs

• Much of this has appeared previously in our books, papers, etc.


• Support from Intel, DoD, DOE, Sandia National Lab, Micron, Cypress Semiconductor
Questions?
(thank you for your kind indulgence)

Prof. Bruce Jacob
Keystone Professor & Director of Computer Engineering Program
Electrical & Computer Engineering
University of Maryland at College Park

blj@umd.edu
www.ece.umd.edu/~blj

… or just google “bruce jacob”