Analyzing Memory System Performance using the AMD Opteron Performance Counters

John D. McCalpin, Ph.D.
Principal Scientist

July 16, 2007

Philosophy

AMD Smarter Choice

"I wouldn't give a fig for the simplicity on this side of complexity; I would give my right arm for the simplicity on the far side of complexity"

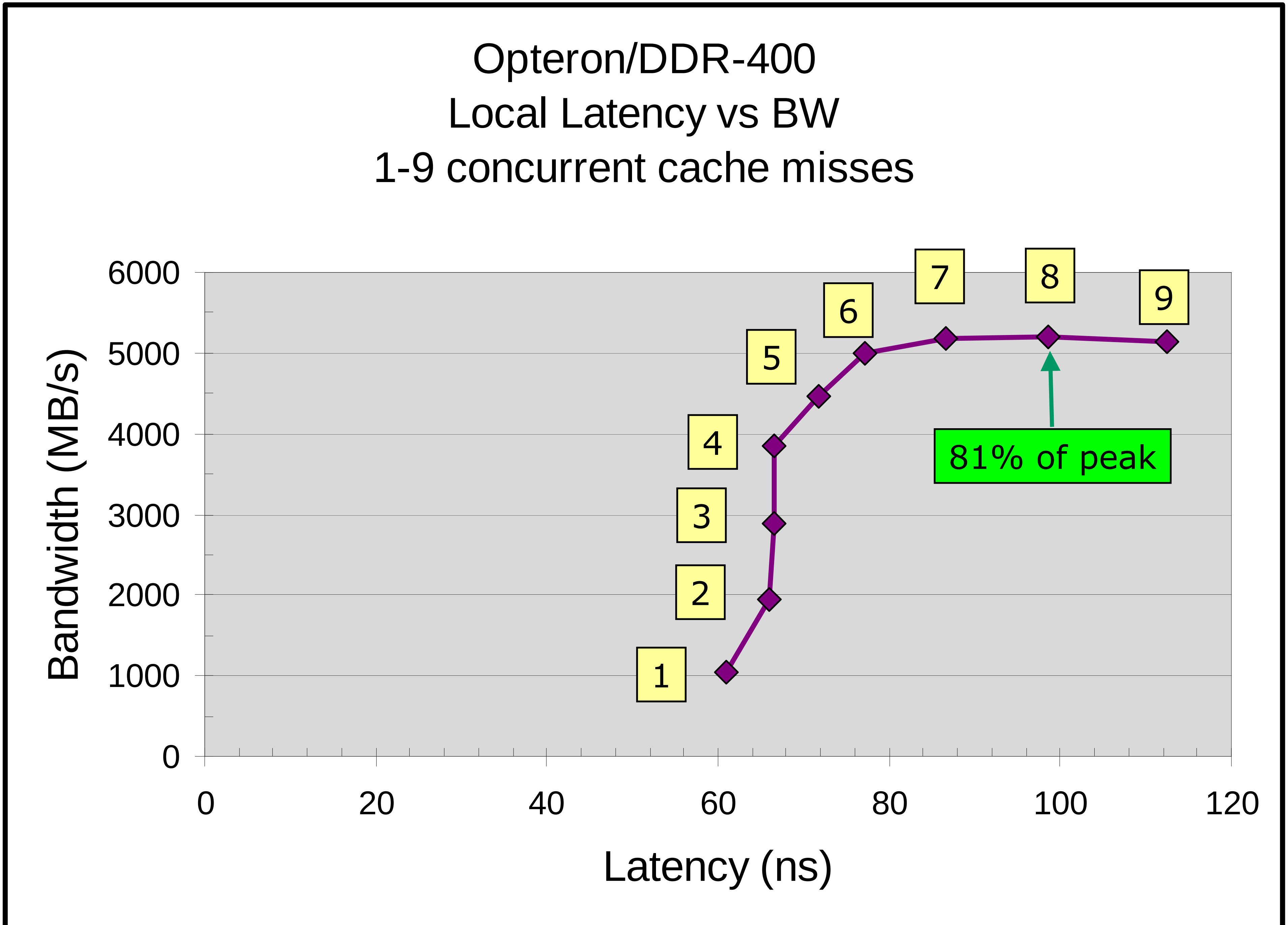
Oliver Wendell Holmes Jr. (1841 -1935)

- Top-down hierarchical performance modeling/analysis has been the Holy Grail of research for the last twenty years.
- Unfortunately top-down hierarchical analysis is fundamentally inconsistent with the implementations of modern computing systems.
- The "real" performance model of the hardware has too many components, any of which can be O(1) performance limiters.

Who? What? Why?

- No PFLOPS in this presentation
- Assume that single-node performance is of interest
- Assume that there is reason to believe that memory subsystem performance may be a performance limiter
- Examples are based on AMD Opteron using publicly available microarchitecture information and publicly available performance monitor information

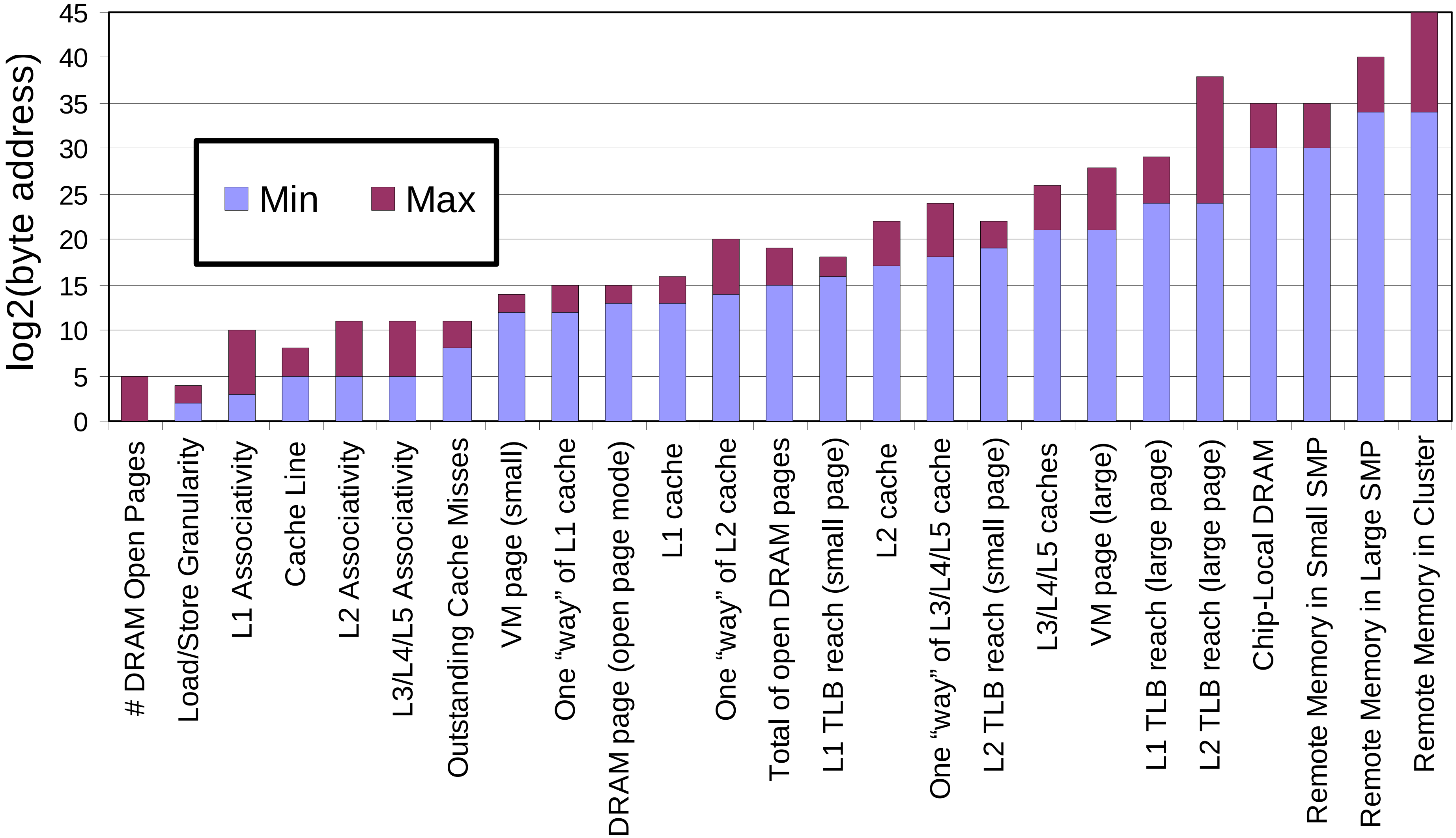
SPEC FPrate2006 on 2s/4c Opteron 2.8/667


Opteron/DDR-400
Local Latency vs BW
1-9 concurrent cache misses
Bandwidth (MB/s)
Latency (ns)
0
1000
2000
3000
4000
5000
6000
0
20
40
60
80
100
120
1
2
3
4
5
6
7
8
9
81% of peak

Locality – not just for caches

- Locality is ubiquitous in modern architectures
  - Instruction & Data Caches
  - Address Translation (multi-level TLBs)
  - Hardware prefetch functions
  - Memory controller open page table(s)
  - DRAM banks
- More cores/threads?
  - Great for increasing concurrency to tolerate latency
  - Bad for exploiting locality – at all levels of hierarchy

Memory Hierarchy Locality Domains



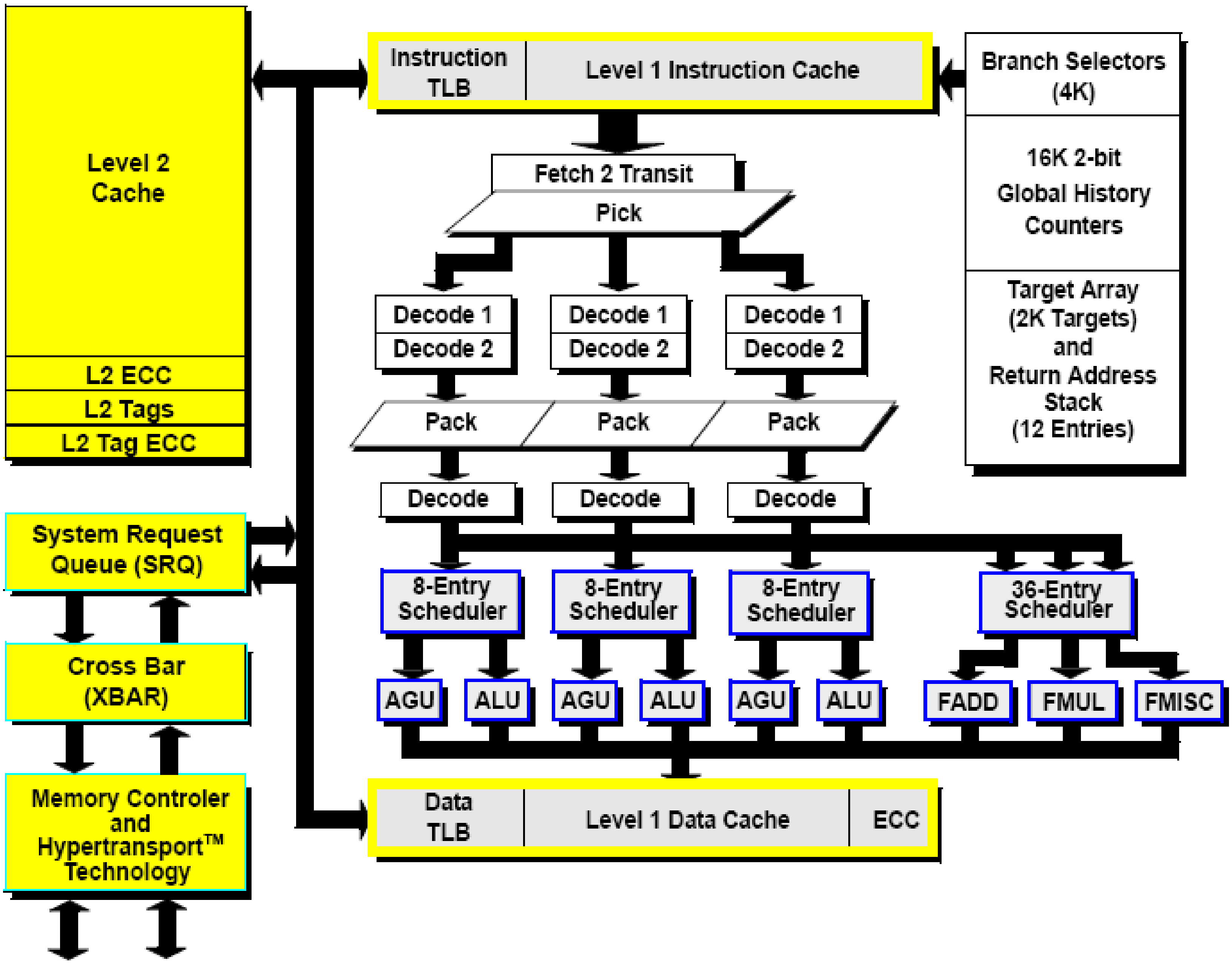
25 different kinds of Locality!

Opteron Memory Access Flow (abridged)

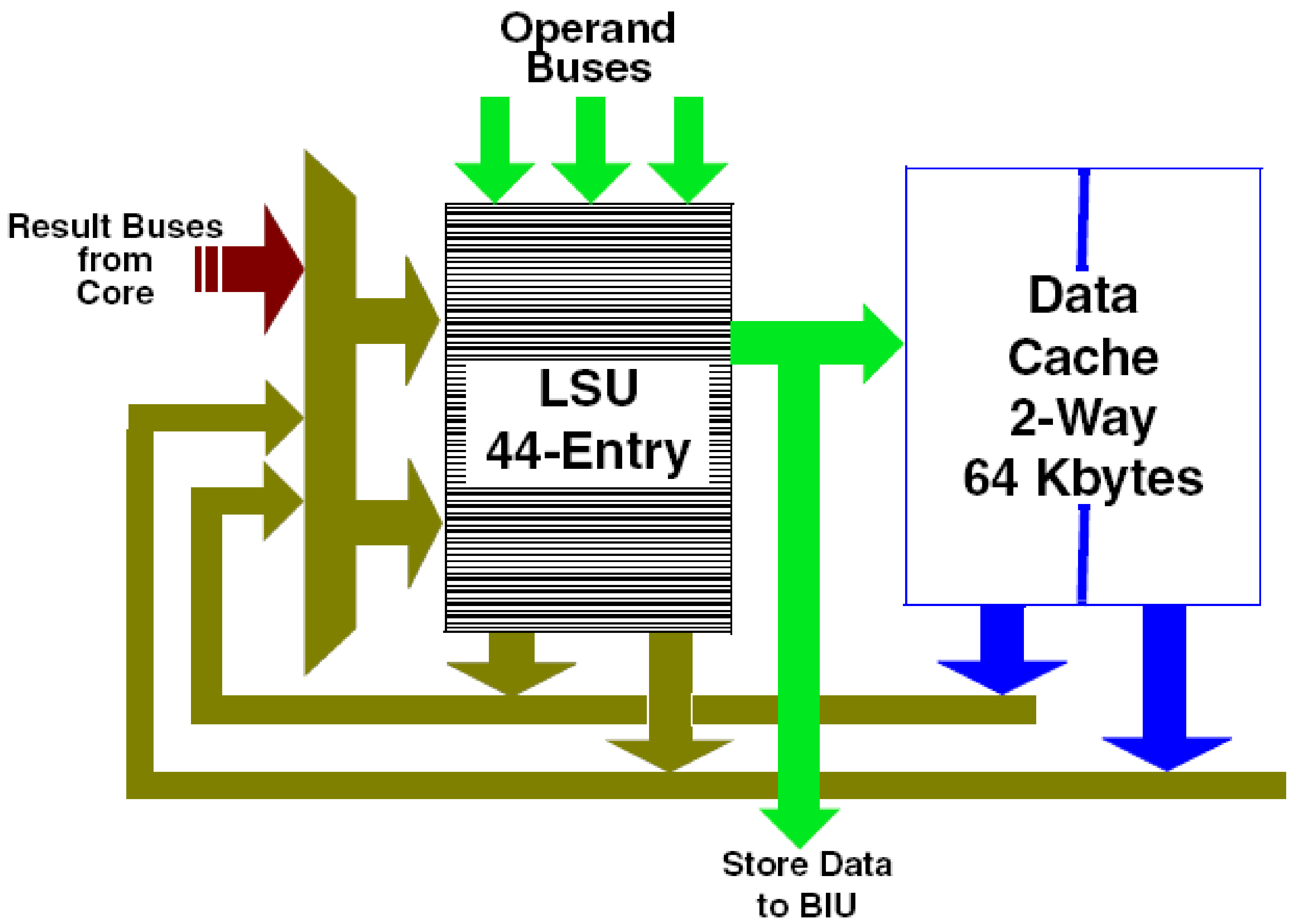
Fetch Instruction
Issue Instruction
Load Request sent from ALU to LS (Load/Store unit)
Insert Request into LS1
Probe D$ (miss)
Move Request to LS2
Allocate Miss Address Buffer
Probe L2 (miss)
Send Request to Memory Controller at “home” Opteron chip
Memory Controller Issues Probes
Memory Controller Checks internal buffers (miss)
Memory Controller sends request to DRAM controller
DRAM Controller checks Open Pages (miss)
DRAM Controller checks for available Open Page buffer (miss)
DRAM Controller closes a page
DRAM Controller Activates target page
DRAM Controller Reads target line
DRAM Controller returns data to Memory Controller
Memory Controller returns data to Requesting CPU
CPU Receives data
CPU Waits for all Probe Responses
CPU Sends SourceDone to Memory Controller
Memory Controller can deallocate buffer

Each of these steps has unique characteristics of concurrency, latency, occupancy, address conflict detection, ordering, etc

Fortunately only a few of these are ***typically*** found to be performance limiters.



Level 2 Cache
L2 ECC
L2 Tags
L2 Tag ECC
System Request Queue (SRQ)
Cross Bar (XBAR)
Memory Controler and Hypertransport™ Technology
Instruction TLB
Level 1 Instruction Cache
Branch Selectors (4K)
16K 2-bit Global History Counters
Target Array (2K Targets) and Return Address Stack (12 Entries)
Fetch 2 Transit
Pick
Decode 1
Decode 2
Decode 1
Decode 2
Decode 1
Decode 2
Pack
Pack
Pack
Decode
Decode
Decode
8-Entry Scheduler
8-Entry Scheduler
8-Entry Scheduler
36-Entry Scheduler
AGU
ALU
AGU
ALU
AGU
ALU
FADD
FMUL
FMISC
Data TLB
Level 1 Data Cache
ECC



Operand Buses
Result Buses from Core
LSU
44-Entry
Data
Cache
2-Way
64 Kbytes
Store Data to BIU

Example: STREAM Benchmark

- STREAM is supposed to be an "easy" benchmark for showing the maximum sustainable memory bandwidth of a system
  - Copy: c[i] = a[i]
  - Scale: b[i] = scalar * c[i]
  - Add: c[i] = a[i] + b[i]
  - Triad: a[i] = b[i] + scalar * c[i]
- Given the complexity of current memory systems, STREAM is not "easy" any more! Single-thread on DDR2/667
  - Standard Optimizations: 3.4 GB/s (32%/48%)
  - Aggressive Optimizations: 6.5 GB/s (61%/61%)
  - Hand-Tuned Assembly: 7.2 GB/s (68%/68%)

More Detail

- “Standard” Optimizations
  - PathScale: pathcc –m64 –O2
  - Similar to “gcc –O2”
  - Normal loads and stores generated
- “Aggressive” Optimizations
  - PathScale: pathcc –m64 –O3
  - Generates non-temporal stores (MOVNT)
  - Eliminates write-allocate traffic
  - Eliminates dirty data in caches
- “Hand-Tuned” Assembly
  - Block non-temporal prefetches
  - Packed Double SSE arithmetic

Analysis: "Standard" optimizations

- 3.4 GB/s (5.1 GB/s including write allocates) is poor even when allocates are included
- Components
  - 10.66 GB/s Peak
  - 5.1 GB/s used: 3.4 GB/s useful + 1.7 GB/s "wasted"
  - 10.6 – 5.1 GB/s used → 5.5 GB/s "missing"
  - DAXPY runs at 5.4 GB/s, suggesting that stalling on store misses is a modest problem (~0.3 GB/s of the 5.5 GB/s "missing")
  - This leaves 5.2 GB/s "missing"
- What do we do now? Try Performance Counters!

Analysis: Performance Counters

- Opteron Processor (core) Performance Counters
  - Dcache and L2 miss rates are trivially calculated for STREAM
  - Translation miss rates are trivially calculated for STREAM
  - Performance counters confirm expected values
- Result: Core counters are uninformative for STREAM
  - I.e., performance monitors measure consequences of slow bandwidth rather than causes of slow bandwidth
- Next Step: Opteron Memory Controller Performance Counters....

Analysis, continued

- Useful Opteron Memory Controller Perf Counters
  - DRAM Access: Page Open, Page Closed, Page Conflict
  - “Chip select” delays
  - “R-W turnaround” and “W-R turnaround” delays
- To understand what these mean, we need to review more details of how the memory controller and DRAMs work
  - DDR2 currently, DDR and DDR3 are similar
  - “Graphics DDR3” is *not* JEDEC DDR3
- SDR/DDR/DDR2/DDR3 are all based on the same Dynamic RAM technology
  - Primary difference is level of prefetch (1/2/4/8) and associated pipelining
  - Some differences in number of banks, etc

DDR/DDR2/DDR3 DRAM Technology

- Each DRAM chip has a 2-D array of memory
  - 256 Mbit (old), 512 Mbit (production), 1024 Mbit (production), 2048 Mbit
- Each DRAM chip provides 4 or 8 bits of the 64-bit output of a DIMM
- A set of DRAMs that are activated to provide a single data burst is called a “Rank”
- DRAM transfers are in “bursts”
  - DDR/DDR2: 4-bit, 8-bit
  - DDR3: 8-bit
- DRAM chips have internal buffers called “banks”
  - 512 Mbit: 4 banks
  - 1024 Mbit: 8 banks

DRAM Technology (cont'd)

- A "bank" is a wide buffer that holds the contents of a "row" of the 2-D array of memory
  - This is also referred to as a "DRAM Page"
  - Banks are similar to caches – another type of locality
- "Typical" DDR2 DIMMs provide 8kB to 16kB "DRAM Page" size (aggregated across all DRAMs in a "rank")
  - Note that a single transfer is typically 32 Bytes or 64 Bytes
  - So a "DRAM Page" holds data for ~256 independent transfers
- Memory Access is a multi-step process
  - Close an open page if needed (PRECHARGE)
  - Open the target page if needed (ACTIVE)
  - Read the desired line (READ)

DRAM Technology (cont'd)

- Increasing pipelining requires increasing concurrency
  - DRAM Cell Recharge time is still ~60 ns
  - DDR2/800 4-bit burst requires 5 ns
  - If open pages cannot be reused, this means >= 12 banks needed to fill pipe
    - Note that 1024 Mbit DRAM with 8 banks can only "roll" 4 banks at a time
  - A reasonable configuration might be 2 dual-rank DIMMs per channel
    - 2 DIMMs * 2 ranks/DIMM * 4 banks/rank = 16 banks
    - Still requires lots of re-ordering in memory controller to use well
- Notes:
  - DDR3/1600 with 8-bit burst will have same ratios
  - "Rolling" banks draws more power than open page

Memory Counter Perf Analysis: cont'd

- Page Access
  - What is a "chip select" delay?
  - Reads to different banks in the same DRAM can be pipelined effectively
  - Reads to different "ranks" (different sets of DRAM chips) incur a two-cycle delay (i.e., ½ of the 4-cycle data transfer time)
- What are "R-W delay" and "W-R delay"?
  - DDR DRAMs use tristate busses that must "settle" between reads and writes
  - Specific numbers vary across technologies, but "R-W" and "W-R" delays are coupled, so the average value is OK

STREAM memory counter analysis

- "Standard" STREAM results show that the ~5.2 GB/s of "lost" bandwidth is approximately evenly divided between "chip select" and "R-W" + "W-R" delays

- This is not surprising:
  - Kernels have 3 or 4 streams – often go to different ranks even when generally hitting open pages
  - Each kernel has one store stream, so R-W and W-R turnarounds are common

- Can we "fix" these problems?

"Aggressive" Optimization

- PathScale: pathcc –m64 –O3
  - Unrolls loops
  - Generates non-temporal stores
    - Stores bypass cache – eliminates backpressure due to store misses
    - No dirty data in cache – no L2 to memory writebacks
  - 6.5 GB/s = 61% of peak
  - Missing 4.1 GB/s
    - Still dominated by "chip select" and "R-W + W-R" turnarounds

"Hand-Tuned" Assembly Optimization

- Begin with PathScale –m64 –O3 code

- Manual changes:
  - Prefetch a 16kB block of source data (1x16kB or 2x8kB)
  - Uses PrefetchNTA to prevent victimization to L2
  - Uses packed double SSE instructions for arithmetic
  - No change to MOVNT stores

- Result: 7.2 GB/s = 68% of peak
  - Latency*BW = 58 ns * 10.66 GB/s = 618 Bytes = >9 lines
  - Only 7 MABs available – read concurrency is marginal
  - 2 threads on 1 chip: >7.6 GB/s → extra MABs help
- This code gives >90% peak on DDR/400 machine
  - Overheads are 67% larger with DDR2/667 than DDR/400

Summary

- Memory Systems have gotten much more complex and are continuing to become more complex
  - Many levels of buffers/queues
  - Deep pipelining
  - Many types of locality
- Block optimizations can increase bandwidth for contiguous accesses by 2x or more
  - Larger gains possible when HW prefetch is not helping the "naïve" code
  - DRAM power is significantly reduced
  - Number of DRAM ranks required is reduced
- Simple models come from deep understanding, not from "a priori" simplification

To Do

- For high-BW workloads, technology is moving toward:
  - CPU is very cheap
  - Bandwidth is cheap
  - Untolerated Latency is the primary performance limiter
- Example:
  - Prefetchable: 7.2 GB/s = 0.975 ns per 64-bit load
  - Not Prefetchable: 58 ns per load = 60x more expensive!

- More cores = more streams = harder to effectively manage memory system locality
- Address trace analysis should consider prefetchability as important as cache miss rate

AMD
Smarter Choice

Trademark Attribution

Backup