Comparing Memory Systems for Chip Multiprocessors

Leverich et al.

Computer Systems Laboratory at Stanford

Presentation by Sarah Bird

Parallel is Hard

- Memory system has a big impact on
  - Performance
  - Energy consumption
  - Scalability
- Tradeoffs between different systems are not always obvious
- Performance of a the memory system is very dependent on
  - the workload
  - the programming model

Memory Models

Cache-Based Memory

- Hardware-managed
- Implicitly-addressed
- Reactive
- Advantages:
  - Best effort locality and communication
  - Dynamic
  - Easier for the programmer

Streaming Memory

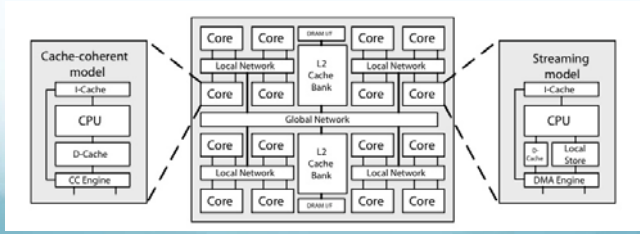
- Software-managed
- Explicitly-addressed
- Proactive
- Advantages:
  - Software can efficiently control addressing, granularity and replacement
  - Better latency hiding
  - Lower off-chip bandwidth

Questions

- How do the two models compare in terms of overall performance and energy consumption?
- How does the comparison change as we scale the number or compute throughput of the processor cores?
- How sensitive is each model to bandwidth or latency variations?

Baseline Architecture

- Tensilica Xtensa LX
  - 3-way VLIW core
    - 2 slots for FP
    - 1 slot for loads & stores
- 16-Kbyte I-Cache
  - 2-way set associative
- In-order processors similar to
  - Piranha
  - RAW
  - Ultrasparc T1
  - XBox360
- 512-Kbyte L2 Cache
  - 16-way set associative



Cache Implementation

- 32-Kbyte
- 2-way set associative
- Write-back, write allocate policy
- MESI write-invalidate protocol
- Store Buffer
- Snooping requests from other cores occupy the data cache for one cycle
  - Core stalls if trying to do a load or store in that cycle
- Hardware stream-based Prefetch engine
  - History of last 8 cache misses
  - Configurable number of lines runahead
  - 4 separate stream accesses

Streaming Implementation

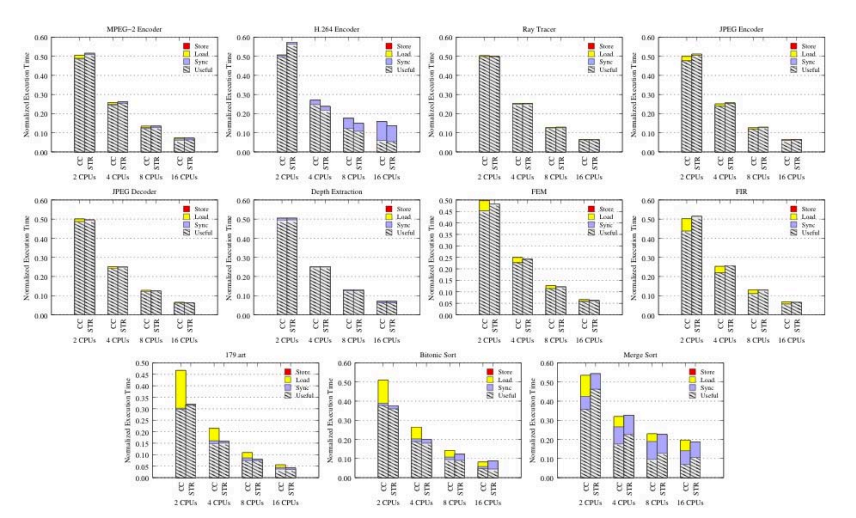
- 24-Kbyte local store
- 8-Kbyte cache
- Doesn't account for the 2-Kbytes of tag
- DMA Engine
  - Sequential
  - Strided
  - Indexed
  - Command queuing
  - 16 32-byte outstanding accesses

Simulation Methodology

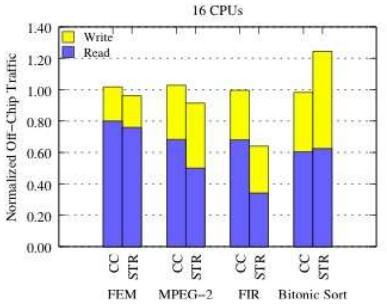
- CMP Simulator
  - Stall events & Contention events
  - Core pipeline & Memory System
- 11 applications
  - Fast-forward through initialization

Application	Input Dataset	L1 D-Miss Rate	L2 D-Miss Rate	Instr. per L1 D-Miss	Cycles per L2 D-Miss	Off-chip B/W
MPEG-2 Encoder [28]	10 CIF frames (Foreman)	0.58%	85.3%	324.8	135.4	292.4 MB/s
H.264 Encoder	10 CIF frames (Foreman)	0.06%	30.8%	3705.5	4225.9	10.8 MB/s
KD-tree Raytracer	128x128, 16371 triangles	1.06%	98.9%	256.3	654.6	45.1 MB/s
JPEG Encoder [21]	128 PPMs of various sizes	0.40%	72.9%	577.1	84.2	402.2 MB/s
JPEG Decoder [21]	128 JPGs of various sizes	0.58%	76.2%	352.9	44.9	1059.2 MB/s
Stereo Depth Extraction	3 CIF image pairs	0.03%	46.1%	8662.5	3995.3	11.4 MB/s
2D FEM	5006 cell mesh, 7663 edges	0.60%	86.2%	368.8	55.5	587.9 MB/s
FIR filter	2^{20} 32-bit samples	0.63%	99.8%	214.6	20.4	1839.1 MB/s
179.art	SPEC reference dataset	1.79%	7.4%	150.1	230.9	227.7 MB/s
Bitonic Sort	2^{19} 32-bit keys (2 MB)	2.22%	98.2%	140.9	26.1	1594.2 MB/s
Merge Sort	2^{19} 32-bit keys (2 MB)	3.98%	99.7%	71.1	33.7	1167.8 MB/s

Performance Comparison



Off-Chip Traffic

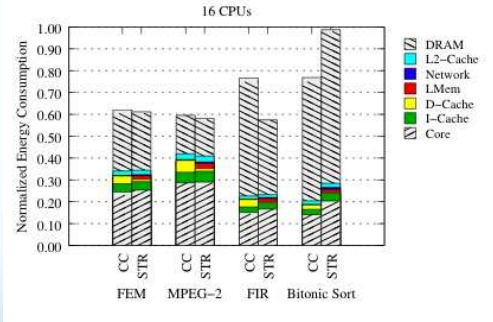


- Streaming has less memory traffic in most cases because it avoids superfluous refills for output only data
- Bitonic Sort writes back unmodified data in the streaming case

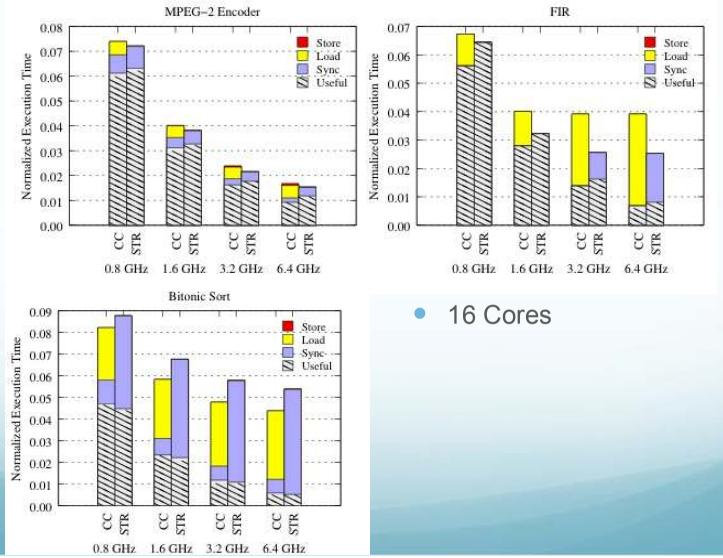
Energy Consumption

Energy Model

- 90 nm CMOS process
- 1.0 V power supply
- Usage statistics
- Energy data from layout of 600MHz design
- Memory Model use CACTI 4.1
- Interconnect uses a model and usages statistics
- DRAM uses DRAMsim

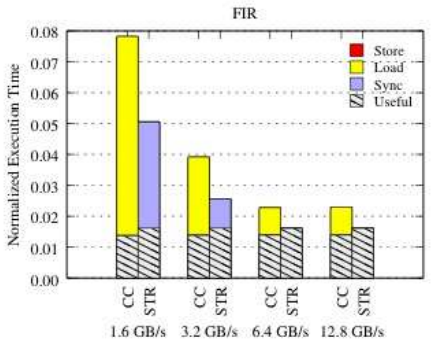


Increased Computation



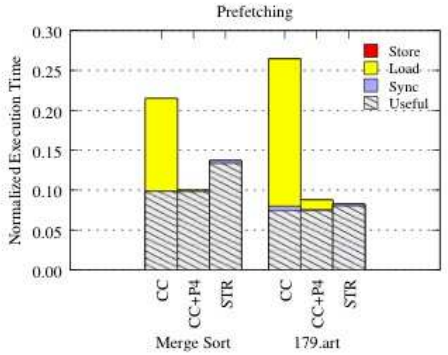
- 16 Cores

Increased Off-Chip Bandwidth



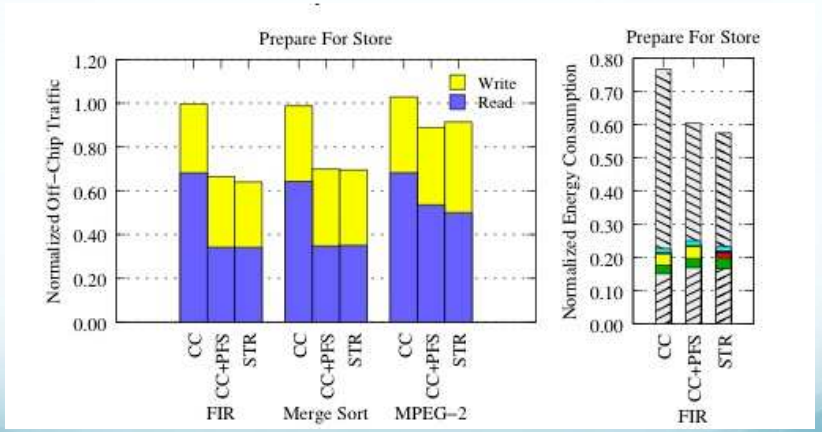
- 3.2 GHz
- 16 Cores
- Doesn't close the energy gap
- Needs non-allocating writes

Hardware Prefetching

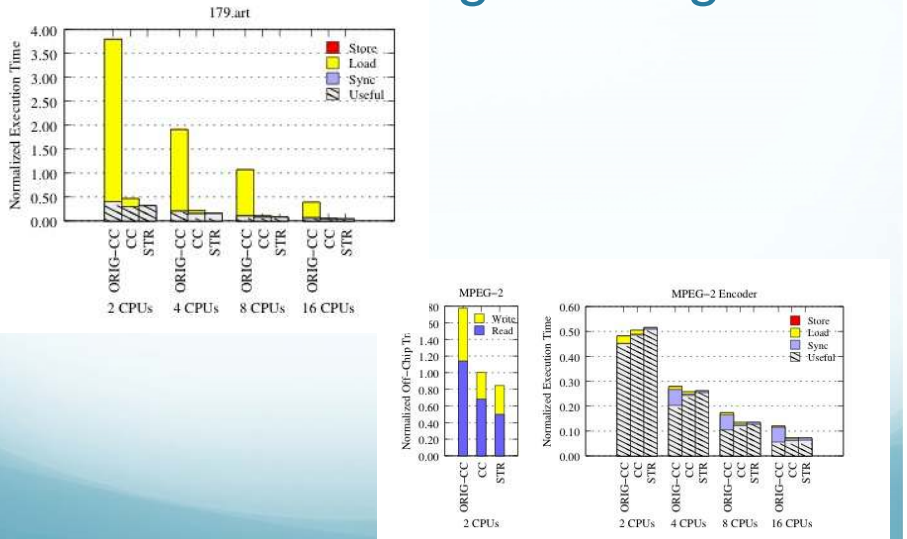


- Prefetch Depth of 4
- 2 Cores
- 3.2 Ghz
- 12.8 GB/s memory channel bandwidth

Prepare for Store



Stream Programming



Results

- Data-Parallel Applications with high data reuse
  - Cache and local store perform/scale equality
  - Streaming is 10-25% more energy efficient than write-allocate caching
- Applications without much data reuse
  - Double-buffering gives streaming a performance advantage as the number/speed of the cores scales up
  - Prefetching helps caching for latency bound apps
  - Caching out performs streaming in some cases by eliminating redundant copies

Conclusions

- Streaming Memory System has little advantage over cache system with prefetching and non-allocating writes
- Streaming Programming Model performs well even with caching since it forces the programmer to think about their application's working set