Memory Performance and Cache Coherency Effects on an Intel Nehalem Multiprocessor System

Parallel Architectures and Compiler Technologies

Raleigh, September 16th 2009

Daniel Molka (daniel.molka@tu-dresden.de)
Daniel Hackenberg (daniel.hackenberg@tu-dresden.de)
Robert Schöne (robert.schoene@tu-dresden.de)
Matthias S. Müller (matthias.mueller@tu-dresden.de)
Outline

- Motivation
- Benchmark Design
- Implementation
  - Intel Xeon X5570 (Nehalem-EP) Results
    - Latency
    - Bandwidth
- Summary
- Recent Developments and Future Work
Growing complexity of memory subsystem
- Shared resources
- NUMA Systems
- Not covered by existing latency and bandwidth benchmarks

More sophisticated benchmarks required to understand behavior of parallel applications
Benchmark Design

- Data placement in arbitrary location
  - Access other cores’ caches
  - Access certain cache levels
Benchmark Design

Data placement in arbitrary location
- Access other cores’ caches
- Access certain cache levels

Coherency state control
- Enforce certain coherency states

Access each cache line only once during measurement
Implementation

Data placement
- Access data of other cores
  - One thread pinned to each core
  - Threads load data into caches of corresponding core
- Access certain cache levels
  - Optional cache flushes

Coherency state control
- Modified: write data (invalidates other copies)
- Exclusive: enforce modified state + flush caches (clflush) + read data
- Shared: enforce exclusive state + read from another core
Implementation

- Time Stamp Counter (rdtsc instr.)
  - Precise measurement of short durations
  - Required to measure without cache line reuse

- Assembler implementation of critical parts
  - Measurement routines (including timestamps)
  - Synchronization of concurrently running threads

- Memory allocation
  - NUMA aware
  - Hugetlbfs support

- BenchIT
  - Framework to develop and run microbenchmarks
  - [http://www.benchit.org](http://www.benchit.org)
Data Placement – Cache Level

Latency without cache flushes
- Mixture of effects from different cache levels
Latency without cache flushes
- Mixture of effects from different cache levels

Latency with cache flushes
- All cache levels and memory latency clearly visible
Data in local caches

- Performance for data that is not used by other cores
Data Placement – Other Cores’ Caches

- Data in local caches
  - Performance for data that is not used by other cores

- Data in other cores’ caches
  - Analyze cache coherency protocol
Coherency State Control

- Modified cache lines transferred from other core
Modified cache lines transferred from other core
Shared cache lines transferred from inclusive L3
Benchmarks

- **Latency**
  - Pointer chasing
  - One thread loads data in its cache
  - Thread on core 0 performs measurement on that data

- **Bandwidth between cores**
  - Consecutive read or write
  - One thread loads data, core 0 measures bandwidth

- **Bandwidth of concurrent accesses**
  - All threads load their data into certain cache level
  - Threads access data concurrently
  - Earliest start timestamp and latest stop timestamp used to calculate bandwidth
Test System Overview

- Dual socket Intel Xeon X5570
  - 2.93 GHz (Turbo Boost disabled)
  - Quadcore (SMT disabled)
- 32 KiB L1I, 32 KiB L1D
- 256 KiB L2
- 8 MiB shared L3
  - Inclusive of L1/L2
  - 2.66 GHz
- 6x 2 GiB DDR3-1333
  - 32 GB/s per socket
- Quick Path Interconnect (QPI)
  - 25.6 GB/s (12.8 per direction)
Core Valid Bits

- L3 keeps track which cores have a copy
  - Used to reduces core snoops

- 1 bit set
  - Line is exclusive or modified
  - L3 copy might be outdated

- 2 bits set
  - Line is shared
  - L3 copy is valid

- All bits 0
  - L3 has the only copy
Core Valid Bits – Silent Evictions

- Silent eviction of unmodified cache lines
  - Write back not required
  - Core valid bits remains unchanged

- Explicit write back of modified data
  - L3 copy needs to be updated
  - Also resets core valid bit
Latency Results – Exclusive and Modified

- Exclusive cache lines
  - L1: 4 cycles, L2 10 cycles
  - L3: 38 cycles (13 ns)
  - On-die transfer: 22 ns
  - Remote access: 65 ns
Latency Results – Exclusive and Modified

- **Exclusive cache lines**
  - L1: 4 cycles, L2 10 cycles
  - L3: 38 cycles (13 ns)
  - On-die transfer: 22 ns
  - Remote access: 65 ns

- **Modified cache lines**
  - Identical for local access
  - On-die transfer:
    - L1: 28 ns, L2: 26 ns
    - L3: 13 ns
  - Remote access:
    - >100 ns (write backs to memory)
Latency Results – Shared and Main Memory

- Shared Cache lines
  - On-die transfer
    - Faster than exclusive
    - Equal to local L3 latency

![Graph showing latency results for Intel Xeon X5570 with data set size in bytes on the x-axis and latency in ns on the y-axis, comparing local, core 1 exclusive, and core 1 shared transfers.]

Daniel Molka
Latency Results – Shared and Main Memory

- Shared Cache lines
  - On-die transfer
    - Faster than exclusive
    - Equal to local L3 latency
  - Silently evicted as well
  - 2 core valid bits set
  - Cores not snooped

![Diagram of cache lines and evict from cores]
Latency Results – Shared and Main Memory

- **Shared Cache lines**
  - On-die transfer
    - Faster than exclusive
    - Equal to local L3 latency
  - Silently evicted as well
    - 2 core valid bits set
    - Cores not snooped

- **Main memory**
  - Local: 65 ns, remote: 106 ns
  - 41 ns for access via QPI
Bandwidth of Transfers Between Cores (and Processors)

- Exclusive cache lines
  - L1: 45.6, L2: 31.1, L3: 26.2 GB/s
  - On-die transfer: 19.7 GB/s
  - Remote: 9.2 GB/s (limited by QPI)
Bandwidth of Transfers Between Cores (and Processors)

- **Exclusive cache lines**
  - L1: 45.6, L2: 31.1, L3: 26.2 GB/s
  - On-die transfer: 19.7 GB/s
  - Remote: 9.2 GB/s (limited by QPI)

- **Modified cache lines**
  - Faster on-die transfer from L3
  - Rather slow from other cores
  - Remote 5.6 GB/s (write backs)
Bandwidth of Transfers Between Cores (and Processors)

- **Exclusive cache lines**
  - L1: 45.6, L2: 31.1, L3: 26.2 GB/s
  - On-die transfer: 19.7 GB/s
  - Remote: 9.2 GB/s (limited by QPI)

- **Modified cache lines**
  - Faster on-die transfer from L3
  - Rather slow from other cores
  - Remote 5.6 GB/s (write backs)

- **Main memory**
  - Local: 10.1 GB/s
  - Remote: 6.3 GB/s (below QPI limit)
Read bandwidth (exclusive data)

- L1/L2 scale well
- L3 limit at 85 GB/s per socket
- Main memory
  - Max. 23 GB/s per socket
  - 72% of theoretical peak
Bandwidth of Concurrent Accesses

- **Read bandwidth (exclusive data)**
  - L1/L2 scale well
  - L3 limit at 85 GB/s per socket
  - Main memory
    - Max. 23 GB/s per socket
    - 72% of theoretical peak

- **Write bandwidth (modified data)**
  - L1/L2 scale well
  - L3 limit at 26 GB/s per socket
  - Main memory
    - Max. 12 GB/s per socket
    - Write allocate
Coherency state control

- Exclusive: silently evict cache lines
Bandwidth of Concurrent Accesses – Coherency Overhead

- Coherency state control
  - Exclusive: silently evict cache lines
  - Modified: write back of higher level caches required
Summary

**Benchmarks**
- Unveil important undocumented performance data
- Measure properties that are not covered by existing benchmarks
- Data placement
  - Analyze performance of communication between cores
- Coherency state control
  - Analyze coherency protocol implementation

**Nehalem Performance**
- Inclusive L3 cache handles all coherency issues between cores on die
- Core valid bits filter most unnecessary snoops
- Limited L3 write bandwidth
Recent Developments

- Experimental support for Owned and Forward state
- Performance counter support (PAPI)
- Experimental uncore performance counter support (perfmon2)

Future Work

- Support other architectures
- Analyze larger shared memory HPC systems
- Measure impact of AMD’s HT Assits
Thanks for your Attention

- Benchmarks and BenchIT Framework available as open source
  - BenchIT available at http://www.benchit.org
  - Find x86 benchmarks at http://www.benchit.org/wiki/index.php/X86membench