Measuring Memory Bandwidth

On the Intel® Xeon® Processor 7500 series platform

Memory bandwidth is one of many metrics customers use to determine the capabilities of a given computer platform. Often customer ask how to measure memory bandwidth and/or how can I get the same memory bandwidth score Intel has measured using an industry standard benchmarks like STREAM.

This paper shows how to reproduce memory bandwidth measurements for the Intel® Xeon® processor 7500 series platform, and why the result of the STREAM benchmark doesn't always answer the question "What is the maximum memory bandwidth of this platform?"

October 2010

Revision: 1.0

Introduction

The STREAM benchmark was created by John McCalpin while at the University of Virginia. Details about the benchmark, source code and some binaries are available here: http://www.cs.virginia.edu/stream/   It is generally accepted that the best measure of a platform's memory bandwidth is the STREAM Triad result, which performs the operation: $a(i) = b(i) + q*c(i)$ where i = the number of iterations of a matrix. The matrix is sized to be at least 2X the size of the largest cache in the system, so that the data is always read/written from main memory and not from on-die caches.   Each iteration of the STREAM Triad performs two reads from memory (for "b(i)" and "q*c(i)") and then one write (writing the result, a(i) to memory) resulting in a total of 24 bytes of data being transferred. The memory bandwidth is then calculated by how many iterations can be accomplished in a given amount of time. 1 million iterations in 1 second would result in a score of 24MB/s.

Does STREAM tell the whole story?

As noted above, the STREAM Triad performs two reads, and one write to memory. If additional data is transferred to/from memory, besides these three transactions, the additional data transferred is not counted in the memory bandwidth result calculated by the benchmark. Thus the memory bandwidth of the platform may actually be higher than what the STREAM benchmark reports.

This is important to note, since the cache coherency protocol for most processors, will not allow you to write a cache line to memory, without first reading it. The read is done before the write, to ensure no one else has a copy of that particular cache line and the processor writing the cache line has ownership of the line. When running the STREAM benchmark, if the processor must first read the cache line before writing it, you will effectively be doing three reads, and one write to memory. However, only two reads, and one write will be counted towards your STREAM bandwidth score. In such a scenario where three reads and one write must be done, 25% of the available memory bandwidth would not actually be counted by the STREAM benchmark. The extra read takes up available memory bandwidth, but is not counted by the benchmark. Thus your STREAM bandwidth score would be ~25% below what the platform is actually transferring to/from memory.

To ensure the STREAM result reflects the actual bandwidth capabilities of the platform, and the extra "read" transaction is not issued, most processors have a "non-cacheable" write transaction or something similar, which tells the processor to do the write without first doing a "read for ownership". Such a non-cacheable write transaction is only used when you don't want the processor to keep track of cache coherency, typically because the application itself is guaranteeing coherency. When running the STREAM benchmark, you typically make sure the write transaction is "non-cacheable" so all available memory bandwidth is used only for the three instructions counted by the benchmark (and the extra read for ownership transaction is not issued). While there are some cases where a non-cacheable write transaction may be useful, in most enterprise applications, you would not issue such a transaction.

Intel® Xeon® Processor 7500 Series

As noted above, to get an accurate memory bandwidth measurement for a given platform, a non-cacheable write transaction is typically used for the write transaction portion of the STREAM benchmark. However, due to the way the cache coherency protocol was designed for the Intel® Xeon® processor 7500/6500 series processors, even issuing a non-cacheable write instruction will not prevent a third read to occur when running the STREAM triad. The coherency protocol for the Intel Xeon processor 7500/6500 processors must check coherency with the IOH device before each write. IOH coherency can be checked either with a read for ownership transaction (occurs before a cacheable write) or a separate read must be made to main memory where a small coherency buffer is stored (occurs before a non-cacheable write).   So both

cacheable and non-cacheable writes will cause an "extra" read transaction to occur which is not counted by the STREAM benchmark. The net result of the coherency protocol for the Intel Xeon processor 7500/6500 series processor, is the STREAM benchmark will always return a bandwidth result which is 25% lower than what the platform is actually capable of.

Note that while the STREAM benchmark results may under call the Intel Xeon processor 7500/6500 platforms memory bandwidth capabilities, the actual bandwidth when running real applications will not be affected. Most enterprise applications will use cacheable write commands which means a write to memory is always preceded by a read for ownership transaction, and thus the memory bus transactions seen by the application that is trying to do two reads and one write, will actually result in three reads and one write, and the effective bandwidth for the applications data (not counting the extra read for ownership transaction), will be consistent with the STREAM results (e.g. 25% below the platforms capability).

Of note, on the Westmere EX processor, which is the next generation Intel processor following the Intel Xeon processor 7500, the IOH coherency buffer, noted above, has been moved into the processor. The result of this micro-architecture change means non-cacheable write transactions will *not* cause the extra read transaction to occur, and the bandwidth measured by STREAM will in fact be the full bandwidth the platform is capable of (e.g. only 2 reads and 1 write transaction will occur when running STREAM).

What does this mean to a typical user running real enterprise applications? The Intel Xeon processor 7500/6500 will have a 25% lower STREAM score than the next generation Westmere EX processor, however the effective memory bandwidth seen by the application will not be appreciably different (~74GB/s effective bandwidth for the application data) since both platforms will issue the additional read for ownership transaction, prior to a write transactions. For any platform which can successfully issue the two reads and one write transaction required by the STREAM benchmark (without needing to issue an additional read transaction), this ~25% reduction in effective bandwidth vs. the peak STREAM score, is typical.

Measuring top STREAM results on the Intel® Xeon® processor 7500 platform

The above discussed why the STREAM benchmark under calls the memory bandwidth of the Intel® Xeon® processor 7500/6500 platform. However, even to get the best results possible, the platform and software must be configured properly. Below are is a list of how to configure the Intel 4-socket platform, to get the best STREAM result (assuming 4 processors installed)

1) Install 64 Quad Rank 1066 MHz DMMS (64 x 4GB is sufficient, although larger DIMMS can be installed without affecting the results)
    - Quad Rank DIMMS provide ~2% higher bandwidth than Dual Rank and Dual Rank provide ~9% higher bandwidth than Single Rank DIMMs
    - Populating all memory channels is a must, but 2 DIMMS per channel (64-DIMMs) will increase memory bandwidth by ~6% vs 1 DIMM per channel (32-DIMMs)
2) Use the following BIOS Settings:
    - Intel® Hyper-Threading technology: Disabled
    - NUMA: Enabled
    - Memory Interleave: 2-way
    - Hardware Prefetchers: Enabled
    - EIST: Enabled
    - Turbo: Disabled
3) Operating System:
    - Novell* SLES 11 SP1 (although Red Hat* RHEL 5.3 (2.6.18 – 128) can also be used and get similar performance)
4) Use the following source code, source changes and compiler options when compiling the STREAM benchmark:
    - Source: stream_omp.c (V5.9) located here:
            http://www.cs.virginia.edu/stream/FTP/Code/Versions/
    - Changes to source:
        o # define N   60,000,000
        o # define NTIMES   100
    - Open_MP settings:
        o OMP_NUM_THREADS=64
        o KMP_AFFINITY=scatter
    - Intel Compiler Version 11.1   Build 20091012
        o Build Instructions: icc –O3 –ip –xSSE4.2 –openmp -static

Measured Intel STREAM triad results: (measuring only 2 Reads & 1 Write transaction)
    71,771 MB/s   (see Note 1) (Measured)

Effective (Actual) Memory Bandwidth (accounting for all 3 Reads and 1 Write transaction)
    95,695 MB/s   (See Note 1) (Estimated)

Conclusions

Like all benchmarks, you really need to understand the benchmark and the system being tested to make appropriate conclusions about the results, and how to compare those results to other systems. The STREAM benchmark is no different – system architecture and processor micro-architecture can affect the benchmark result, but may or may not affect real application performance.

Real application performance is the best measure of a platform's capabilities. Synthetic benchmarks, like STREAM, are helpful to understand how a particular sub-system of a computer platform works, however the results of such benchmarks should not be used in isolation to determine which platform is better at running real applications. Most users are purchasing their computers to run real applications, not synthetic benchmarks Overall platform performance is based on a combination of how each individual sub-system works, how all of the sub-systems work together and ultimately how a particular application runs on that platform.

Notes:

1. STREAM benchmark tested by Intel Corporation TR#1173 as of June 2010. Configuration details: Four Intel Xeon processors X7560 (24M cache, 2.26GHz, 6.40 GT/s Intel® QPI) on Intel 7500 Chipset-based internal software development platform, 64x 4GB DDR3-1066 Quad-Rank DIMMs, Novell* SUSE LINUX Enterprise 11 OS.

Author

Scott Huck is a Performance Architect in Intel's Digital Enterprise Group