Memory Performance at Reduced CPU Clock Speeds: An Analysis of Current x86 64 Processors

Robert Schöne, Daniel Hackenberg, and Daniel Molka

Center for Information Services and High Performance Computing (ZIH)

Technische Universität Dresden, 01062 Dresden, Germany

Email: {robert.schoene, daniel.hackenberg, daniel.molka}@tu-dresden.de

Abstract-Reducing CPU frequency and voltage is a well-known approach to reduce the energy consumption of memory-bound applications. This is based on the conception that main memory performance sees little or no degradation at reduced processor clock speeds, while power consumption decreases significantly. We study this effect in detail on the latest generation of x86 64 compute nodes. Our results show that memory and last level cache bandwidths at reduced clock speeds strongly depend on the processor microarchitecture. For example, while an Intel Westmere-EP processor achieves 95 % of the peak main memory bandwidth at the lowest processor frequency, the bandwidth decreases to only $60\,\%$ on the latest Sandy Bridge-EP platform. Increased efficiency of memory-bound applications may also be achieved with concurrency throttling, i.e. reducing the number of active cores per socket. We therefore complete our study with a detailed analysis of memory bandwidth scaling at different concurrency levels on our test systems. Our resultsboth qualitative developments and absolute bandwidth numbers-are valuable for scientists in the areas of computer architecture, performance and power analysis and modeling as well as application developers seeking to optimize their codes on current x86 64 systems.

I. INTRODUCTION

Many research efforts in the area of energy efficient computing make use of dynamic voltage and frequency scaling (DVFS) to improve energy efficiency [1], [6], [8], [4]. The incentive is often that a system's main memory bandwidth is unaffected by reduced clock speeds, while the power consumption decreases significantly. Therefore, DVFS is a common optimization method when dealing with memory-bound applications.

This paper presents a study of this method on the latest generation of x86_64 systems from Intel (Sandy Bridge) and AMD (Interlagos) as well as the previous generation processors Intel Westmere/Nehalem and AMD Magny-Cours/Istanbul. We use synthetic benchmarks that are capable of maxing out the memory bandwidth for each level of the memory hierarchy. We focus on read accesses to main memory and also take the last level cache performance into account. Our analysis reveals that main memory as well as last level cache performance at reduced clock speeds strongly depends on the processor microarchitecture.

Besides DVFS, another common approach is to reduce the number of concurrent tasks in a parallel program to reduce power consumption (i.e., concurrency throttling - CT). This can be very effective, as a single thread may be able to fully utilize the memory controller bandwidth. We therefore extend our analysis to also cover CT on the latest x86_64 systems. Again, our results prove the importance of taking the individual properties of the target microarchitecture into account for any concurrency optimization.

II. RELATED WORK

DVFS related runtime and energy performance data for three generations of AMD Opteron processors has been published by LeSueur et al. [5]. They provide a small section regarding the correlation of memory frequency and processor frequency. Full memory frequency at the lowest processor frequency is only supported by their newest system, an AMD Opteron 2384 (Shanghai). While they sketch several DVFS related performance issues, we focus on a single important attribute and describe it in detail for newer architectures and also take concurrency scaling into account.

Choi et al. [1] divide the data access into external transfer and on die transfer, with the latter depending on the CPU frequency. Liang et al. [6] describe a cache miss based prediction model for energy and performance degradation, while Spiliopoulos et al. [8] describe a slack time based model. Neither of them takes the architecture dependent correlation of CPU frequency and main memory bandwidth sufficiently into account.

Curtis-Maury et al. [2] analyze how decreased concurrency can improve energy efficiency of non-scaling regions. Their main reason from an architectural perspective to do concurrency throttling is that main memory bandwidth can be saturated at reduced concurrency.

III. TEST SYSTEMS

We perform our studies on current and last generation processors from Intel and AMD. The test systems are detailed in Table I. All processors have a last level cache and an integrated memory controller on each die, being competitively shared by multiple cores. AMD

Table I HARDWARE CONFIGURATION OF TEST SYSTEMS

Vendor	Intel				AMD		
Processor	Xeon X5560	Xeon X5670	Core i7-2600K	Xeon E5-2670	Opteron 2435	Opteron 6168	Opteron 6274
Codename	Nehalem-EP	Westmere-EP	Sandy Bridge-HE	Sandy Bridge-EP	Istanbul	Magny-Cours	Interlagos
Cores	2x 4	2x 6	4	2x 8	2x 6	2x 2x 6	4x 2x 8 ¹
Base clock	2.8 GHz	2.933 GHz	3.4 GHz	2.6 GHz	2.6 GHz	1.9 GHz	2.2 GHz
max. turbo clock	3.2 GHz	3.333 GHz	3.8 GHz	3.3 GHz	-	-	2.5 GHz ²
L3 cache per die	8 MiB	12 MiB	8 MiB	20 MiB	6 MiB	6 MiB	8 MiB
IMC chan. per die	3x RDDR3	3x RDDR3	2x DDR3	4x RDDR3	2x RDDR2	2x RDDR3	2x RDDR3
DIMMs per die	3x 2 GiB	3x 2 GiB	4x 2 GiB	4x 4 GiB	4x 2 GiB	4x 4 GiB	2x 4 GiB
Memory type	PC3-10600R	PC3L-10600R	PC3-10600	PC3-12800R	PC2-5300R	PC3-10600R	PC3-12800R

processors as well as previous generation Intel processors (Nehalem-EP/Westmere-EP) use a crossbar to connect the cores with the last level cache and the memory controller. In contrast, current Intel processors based on the Sandy Bridge microarchitecture have a local last level cache slice per core and use a ring bus to connect all cores and cache slices. All processors support dynamic voltage and frequency scaling to adjust performance and energy consumption to the workload.

IV. BENCHMARKS

We use an Open Source microbenchmark [7], [3] to perform a low-level analysis of each system's cache and memory performance. Highly optimized assembler routines and time stamp counter based timers enable precise performance measurements of data accesses in different levels of the memory hierarchy of 64 Bit x86 processors [7]. The benchmark is parallelized using pthreads, the individual threads are pinned to single cores, and all threads use only local memory. We use this benchmark to determine the read bandwidth of last level cache and main memory accesses for different processor frequencies and various numbers of threads.

V. BANDWIDTH AT MAXIMUM THREAD CONCURRENCY

We use our microbenchmarks to determine the bandwidth depending on the core frequency while using all cores of each processor (maximum thread concurrency). Figure 1a shows that main memory bandwidth is frequency independent only on specific Intel architectures. While this group includes Intel Nehalem, Westmere and Sandy Bridge-HE, the second group (frequency dependent) consists of the AMD systems and Intel Sandy Bridge-EP. The consistent behavior of the AMD processors can be attributed to their similar northbridge design.³ Intel processors have undergone more significant redesigns. While the new ring-bus based uncore still provides high bandwidths at lower frequencies for the desktop processor Sandy Bridge-HE, the results for the server equivalent Sandy Bridge-EP are unexpected.

Regarding the L3 cache bandwidth depicted in Figure 1b, three categories can be distinguished: frequency independent, converging, and linearly scaling. Only on the Intel Westmere-EP system, the L3 cache bandwidth is independent of the processor clock speed. Although the uncore architecture is very similar to Nehalem-EP, the six-core design allows to utilize the L3 cache fully even at lower frequencies. With only four cores, Nehalem-EP cannot sustain the L3 performance at lower clock speeds. AMD processors show a similar con-

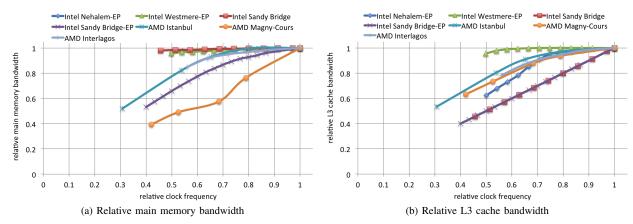


Figure 1. Scaling of shared resources with maximum thread concurrency on recent AMD and Intel processors. Main memory and L3 cache bandwidth are normalized to the peak at maximum frequency.

¹4 sockets, 2 dies per socket, 8 cores per die

²C6 disabled, therefore no MAX_TURBO available

³The Magny-Cours main memory bandwidth is currently not fully understood, confirmatory measurement are underway.

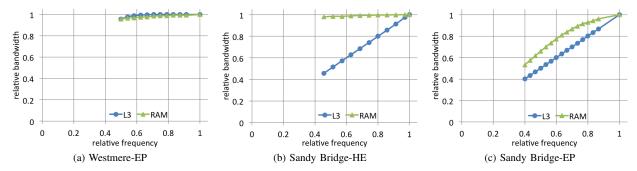


Figure 2. Intel system comparison. Main memory and L3 cache performance are normalized to peak bandwidth at maximum frequency. The two-socket Westmere-EP test system shows almost constant RAM and L3 bandwidth for all frequencies. On the one-socket Sandy Bridge system, L3 bandwidth scales linearly with the processor clock speed, while RAM bandwidth is frequency independent. The two-socket Sandy Bridge node shows the same linear L3 behavior. We additionally measure that main memory bandwidth decreases strongly for lower clock speeds.

verging L3 pattern, which is again due to the largely unchanged northbridge design of all generations. For Sandy Bridge processors, the L3 bandwidth scales perfectly linear with the processor frequency. This is not surprising, as there is no dedicated uncore frequency. Instead, the cores, the L3 cache slices, and the interconnection ring are in the same frequency domain.

Figure 2 highlights the recent development of Intel processors. It showcases that memory bandwidth at reduced processor clock speeds may not be deduced from experiences with previous systems, but instead needs to be evaluated for each new processor generation.

VI. BANDWIDTH AT VARIABLE CONCURRENCY AND FREQUENCY

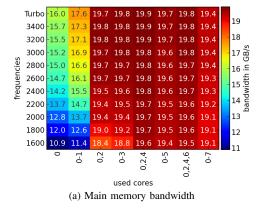
A. Intel Sandy Bridge-HE

In addition to the very good main memory bandwidth at low clock frequencies, Sandy Bridge-HE also performs exceptionally well in low concurrency scenarios. Figure 3a shows that using only one of the four cores is sufficient to achieve almost 90 percent of the maximum main memory bandwidth (HyperThreading enabled, i.e. two threads on one core). The remaining three cores

may be power gated to tap a significant power saving potential. With two or more active cores, the maximum bandwidth can be achieved even at reduced clock speeds. In terms of absolute memory bandwidth, 19.9 GB/s is well above 90% of the theoretical peak bandwidth of two DDR3-1333 memory channels.

The L3 cache of Sandy Bridge-HE scales linearly with the processor frequency (see Figure 3b). We measure a 2.1 GB/s bandwidth bonus for each frequency increase of 200 MHz. Unlike main memory bandwidth, the cache bandwidth also profits from Turbo Boost. L3 cache bandwidth also scales well with the number of cores. At base frequency, a per-core bandwidth of about 35.3 GB/s can be achieved. The Turbo Boost bonus depends on the number of cores, as does the actual Turbo Boost processor frequency. Using two threads per core (HyperThreading) results in a small but measurable bandwidth increase of about 5%.

We have repeated all measurements on a newer Ivy Bridge test system (Core i5-3470) with very similar results. Therefore, the performance numbers we report for Sandy Bridge-HE are representative for Ivy Bridge as well.



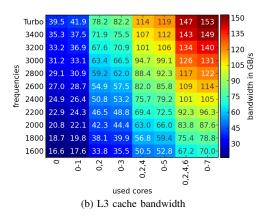


Figure 3. Sandy Bridge-HE main memory and last level cache bandwidth at different processor clock speeds and core counts. (0-1) refers to two threads on one core (HyperThreading) while (0,2) refers to two cores with one thread each (no HyperThreading).

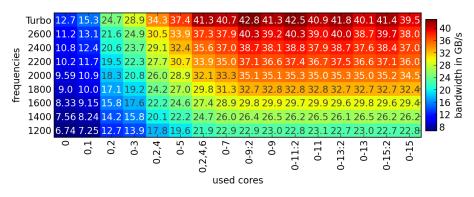


Figure 4. Main memory bandwidth on Sandy Bridge-EP

B. Intel Sandy Bridge-EP

The pattern for main memory bandwidth depicted in Figure 4 differs noticeably from the Sandy Bridge-HE platform. Five (instead of two) active cores are required for full main memory bandwidth. Fewer cores can not fully utilize the bandwidth because of their limited number of open requests. While for a smaller number of active cores HyperThreading provides a performance benefit, it degrades the bandwidth for higher core counts. Furthermore, main memory bandwidth strongly depends on the processor frequency, thus Turbo is needed for maximum performance. If the data is not available in the local L3 cache, a request to the remote socket via OPI is performed to check for updated data in its caches. This coherence mechanism likely scales with the core/ring frequency and therefore limits the memory request rate at reduced frequency.

The resemblance of Figure 3b and Figure 5 shows that the L3 cache bandwidth on Sandy Bridge-EP scales similarly to the Sandy Bridge-HE processor. The additional performance due to HyperThreading is about 10%, compared to 5% for Sandy Bridge-HE. However, the bandwidth per core is approximately 10% lower on Sandy Bridge-EP. This could be due to the longer ring bus in the uncore and the increased number of L3 slices, that results in a higher percentage of non-local accesses.

C. AMD Interlagos

The results obtained from the Interlagos platform have to be considered with AMDs northbridge architecture in mind. All cores are connected to the L3 cache via the System Request Interface. This interface also connects to the crossbar, which itself connects further to the memory controller. These and other components form the northbridge. It is clocked with an individual frequency that corresponds to the northbridge p-State. On our test system, this frequency is fixed at 2 GHz.

Figure 6a depicts the main memory performance of a single AMD Interlagos processor die. The bandwidth scales to three active modules. For lower module counts using two cores per modules increases the achieved bandwidth per module. This can be explained by the per core Load-Store units which enable more open requests to cover the access latency when using two cores. For three and four active modules, two active cores can reduce the maximal bandwidth due to ressource contention caused by the concurrent accesses.

Figure 6b shows the L3 cache bandwidth on Interlagos. Although a core-frequency independent monolithic block of L3 cache should be able to provide a high bandwidth to a single core even at low frequencies, this is clearly not the case. We see two bottlenecks that limit the L3 cache bandwidth. The first one depends

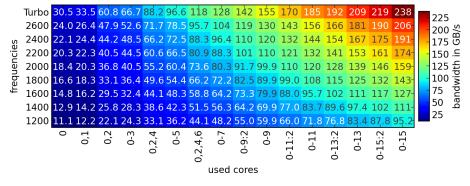


Figure 5. L3 cache bandwidth on Sandy Bridge-EP

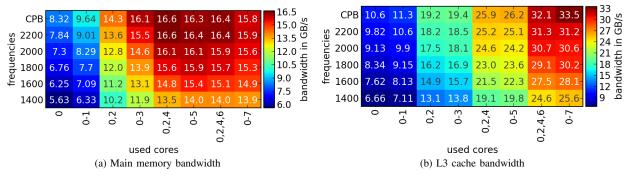


Figure 6. AMD Interlagos main memory and last level cache bandwidth (single die) at different processor clock speeds and core counts. (0-1) refers to two active cores per module while (0,2) refers to two modules with one active core each.

on the core frequency and is therefore located neither in the L3 cache nor the SRI. This bottleneck can not be explained with per core Load-Store buffers, as using two cores per module does not increase the bandwidth. Apparently the number of requests per module to the SRI is limited. The second bottleneck is located in the SRI/L3 construct, which limits the number of requests a single module can send to the L3 cache. Therefore, the bandwidth also scales with the number of accessing modules.

VII. CONCLUSION AND FUTURE WORK

In this paper we study the correlation of processor frequency and main memory as well as last level cache performance in detail. Our results show that this correlation strongly depends on the processor microarchitecture. Relevant qualitative differences have been discovered between AMD and Intel x86_64 processors, between consecutive generations of Intel processors, and even between the server and desktop parts of one processor generation. Some systems provide full bandwidth at the lowest CPU frequency. However, decreasing CPU frequencies also degrades memory performance in many cases. Similarly, there is also no general answer regarding the effect of concurrency throttling.

Based on our findings we make the case that all DVFS/CT based performance or energy efficiency efforts need to take the fundamental properties of the targeted microarchitecture into account. For example, results that have been obtained on a Westmere platform are likely not applicable to a Sandy Bridge system.

The results presented in this paper can be used to save energy on under-provisioned systems. Scheduling multiple task per NUMA node before populating the next one allows other nodes to use deeper sleep states. This could be implemented similar to sched_mc_power_savings in Linux. However, our study focuses on the performance effect of frequency scaling and concurrency throttling, leaving further measurements of power and energy for future work.

Acknowledgment This work has been funded by the Bundesministerium für Bildung und Forschung via the research projects CoolSilicon (BMBF 13N10186) and eeClust (BMBF 01IH08008C), and by the Deutsche Forschungsgemeinschaft DFG via the SFB HAEC (SFB 921/1 2011).

REFERENCES

- [1] K. Choi, R. Soma, and M. Pedram. Dynamic voltage and frequency scaling based on workload decomposition. In *Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004 International Symposium on*, pages 174 –179, aug. 2004.
- [2] M. Curtis-Maury, F. Blagojevic, C. D. Antonopoulos, and D. S. Nikolopoulos. Prediction-based power-performance adaptation of multithreaded scientific codes. *IEEE Trans. Parallel Distrib. Syst.*, 19(10):1396–1410, October 2008.
- [3] D. Hackenberg, D. Molka, and W. E. Nagel. Comparing cache architectures and coherency protocols on x86-64 multicore SMP systems. In MICRO 42: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, pages 413–422, New York, NY, USA, 2009. ACM.
- [4] C. Hsu and W. Feng. A power-aware run-time system for high-performance computing. In *Proceedings of the 2005* ACM/IEEE conference on Supercomputing, SC '05. IEEE Computer Society, nov. 2005.
- [5] E. Le Sueur and G. Heiser. Dynamic voltage and frequency scaling: the laws of diminishing returns. In Proceedings of the 2010 international conference on Power aware computing and systems, HotPower'10, pages 1–8, Berkeley, CA, USA. USENIX Association.
- [6] W. Liang, S. Chen, Y. Chang, and J. Fang. Memory-aware dynamic voltage and frequency prediction for portable devices. In *Embedded and Real-Time Computing Systems* and Applications, 2008. RTCSA '08. 14th IEEE International Conference on, pages 229 –236, aug. 2008.
- [7] D. Molka, D. Hackenberg, R. Schöne, and M. S. Müller. Memory performance and cache coherency effects on an Intel Nehalem multiprocessor system. In PACT '09: Proceedings of the 2009 18th International Conference on Parallel Architectures and Compilation Techniques, pages 261–270, Washington, DC, USA, 2009. IEEE Computer Society.
- [8] V. Spiliopoulos, S. Kaxiras, and G. Keramidas. Green governors: A framework for continuously adaptive dvfs. In *Green Computing Conference and Workshops (IGCC)*, 2011 International, pages 1 –8, july 2011.