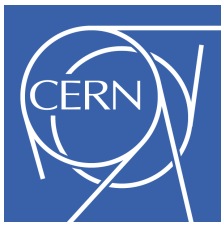
Evaluation of the Intel 4 socket Sandy Bridge-EP server processor

Sverre Jarp, Alfio Lazzaro, Andrzej Nowak, Liviu Valsan, Julien Leduc

CERN openlab, December 2012 – version 1.0



Executive Summary

In this paper we report on a set of benchmark results obtained by CERN openlab when comparing a 32-core, quad socket, "Sandy Bridge-EP" server with a 16-core, dual socket, "Sandy Bridge-EP" server and a 40-core, quad socket server using Intel's previous microarchitecture, the "Westmere-EX". The Intel marketing names for the corresponding processors are the "Xeon E5-4600 processor series", "Xeon E5-2600 processor series" and "Xeon E7-4800 processor series", respectively. All three processors are produced using a 32 nm process. Multiple benchmarks were used to get a good understanding of the performance of each processor. We used both industry-standard benchmarks, such as SPEC CPU2006, and specific High Energy Physics benchmarks, representing both simulation of physics detectors and data analysis of physics events.

Before summarizing the results we must stress the fact that benchmarking of modern processors is a very complex affair. One has to control (at least) the following features: processor frequency, overclocking via Turbo Boost Technology, the number of physical cores in use, the use of logical cores via Simultaneous Multi-Threading (SMT), the cache sizes available, the memory configuration installed, as well as the power configuration if throughput per watt is to be measured. Software must also be kept under control as a change of operating system or compiler can lead to different results. We have tried to do a good job of comparing like with like.

In summary, we found that the quad socket Sandy Bridge-EP offers the equivalent frequency-scaled performance of the quad socket Westmere-EX. We should note that the Westmere-EX processor comes with 25% more cores and 20% larger L3 caches. We also found that Turbo Boost Technology has been improved. Vectorized applications get an additional performance boost given by the new AVX instructions.

Intel has improved the thermal characteristics of the Sandy Bridge substantially and this was reflected in the measurements of idle power, but also in the measurements of a fully loaded system. Computer centers that are power constrained will, without doubt, appreciate the improvements in this important domain. By raising the bar to such a high level, Intel has set high expectations and we are keen to see whether the pace of improvement can be sustained for the 22 nm server processors, namely the Ivy Bridge microarchitecture planned for 2013 and Haswell for 2014.

Table of Contents

Introduction

Description of the servers and corresponding processors

In this paper we compare three servers equipped with Intel processors produced in 32 nm silicon process, spreading across two generations: a quad socket Xeon Westmere-EX server, a two socket Xeon Sandy Bridge-EP server and a quad socket Xeon Sandy Bridge-EP server. As Intel will not release any Sandy Bridge-EX processors a quad-socket server based on the Sandy Bridge-EP processor has been used instead. The first generation, "Westmere-EX" or "Xeon E7-4800" was introduced in Q2 2011 as a shrink of the 45 nm-based "Nehalem-EX" processor. The Westmere-EX processor contains up to 10 cores and 30 MB of L3 cache on the die. The Sandy Bridge-EP processor comes with up to 8 cores and 20 MB of L3 cache. All the processors used in this report have the same Thermal Design Power rating of 130W. The dual and quad socket Sandy Bridge-EP processors are running at 2.7 GHz whereas the Westmere-EX processor runs at 2.4 GHz. All our results are shown as frequency-scaled absolute performance, so, in our experience, this is a fair way of doing comparisons.

Both generations support Simultaneous Multithreading (SMT) that allows two threads or processes to be active on each core. Furthermore, all three processors support Turbo Boost Technology that allows the processor to increase its frequency when only a few of the cores are in use, while maintaining power within the designed envelope. As the analysed processors have different Turbo Boost Technology implementations and frequencies, this feature has been disabled while running the tests.

The major new architectural feature in the Sandy Bridge processor is undoubtedly the Advanced Vector eXtensions (AVX) that allow new Single Instruction Multiple Data (SIMD) operations to be performed on 256 bits of data. This is twice the width of Streaming SIMD Extensions (SSE) that we find in the Westmere-EX architecture and its predecessors. Consequently, this is a feature we have tested extensively in the Parallel Maximum Likelihood fitting benchmark, since it allows good usage of this SIMD feature.

Hardware configuration

The quad socket Sandy Bridge-EP processor evaluated in this paper is a Xeon E5-4650 running at 2.70GHz and the two socket Sandy Bridge-EP processor used is a Xeon E5-2680, running at the same frequency. The quad socket Sandy Bridge-EP processor has been used inside a Dell PowerEdge R820 system, fitted with 256 GB of memory (16 x 16 GB DDR3L DIMMs 1333 MHz 1.35V) and 2 x 500 GB 7200 RPM SAS hard drives configured in RAID 1 using the Dell PERC H710 RAID controller. The test system hosting the dual socket Sandy Bridge-EP processor is using an Intel S2600CP motherboard fitted with 64 GB of memory (16 x 4 GB DDR3L DIMMs 1333 MHz 1.35V), and a 1TB 7200 RPM SATA hard drive. The Westmere-EX processor was installed in a system based on the QCI QSSC-S4R motherboard, with 128 GB of

memory (32 x 4 GB DDR3 DIMMs 1066 MHz 1.5V), 5 x 160 GB Intel X25-M solid state drives and 1 x 250 GB 7200 RPM SATA hard drive.

Software configuration

All systems are running 64-bit Scientific Linux CERN 6.3 (SLC6), based on Red Hat Enterprise Linux 6 (Server). The SLC6 Linux kernel version 2.6.32-279.5.2.el6.x86_64 was used for all the measurements.

The compilers that have been used are:

- GCC 4.4.6 (the default version shipping with Scientific Linux CERN 6.3)
- ICC 13.0.1 (Intel Composer XE 2013 update 1)

Standard energy measurement procedure

Procedure and methodology

The standard energy measurement procedure is well-established in the CERN IT department for measuring the power consumption of any system that might be operated in the CERN Computing Center. Since this procedure was already thoroughly described in a previous openlab paper: "Evaluation of energy consumption and performance of Intel's Nehalem architecture", it is now included as an appendix.

Results

The test Dell PowerEdge R820 system is equipped with two power supplies (PSU). To simplify power measurements only one PSU was used. When conducting the tests without SMT the quad-socket system appears as having 32 cores in total, so that, according to the standard energy measurement procedure, the load stress test consists of running 16 instances of *CPU Burn-in* along with 16 instances of *LINPACK* (using 16 GB of memory each).

In a second phase, now with SMT enabled, the system was considered as a 64 core server, meaning that the load stress test should be conducted by running 32 instances of *CPU Burn-in* along with 32 instances of *LINPACK* (using 8 GB of memory each). The table below shows the value of the power measurements.

Active Power		*Idle*	*Load*	*Standard measurement*
256 GB	SMT-off	149 W	534 W	457 W
	SMT-on	150 W	577 W	492 W

Table 1: Power consumption for Dell R820 (Intel Xeon E5-4650)

A first look at these power consumption measurements shows very good overall system electrical consumption values, both in the "Idle" and the "Load" states.

Benchmarks

HEP-SPEC06

One of the important performance benchmarks in the IT industry is the SPEC CPU2006 benchmark from the SPEC Corporation[1]. This benchmark can be used to measure both the individual CPU performance and the throughput rate of servers.

It is an industry standard benchmark suite, designed to stress a system's processor, caches and the memory subsystem. The benchmark suite is based on real user applications with the source code being commercially available. A High Energy Physics (HEP) working group demonstrated a few years ago a good correlation between the SPEC results and High Energy Physics (HEP) applications when using the C++ subset of the tests from the SPEC CPU2006 benchmark suite [WLCG09]. As a result the HEP community has decided to use the C++ subset of SPEC CPU2006, "HEP-SPEC06" rather than internal benchmarks because SPEC CPU2006 is readily available, and its results can be directly generated by computer manufacturers to evaluate the performance of a system aimed at running HEP applications.

In this set of benchmarks the SPEC CPU2006 suite was compiled with GCC 4.4.6 in 64-bit mode, the standard compiler available with SLC6 and the performance measurements were carried out with SMT enabled and with Turbo Boost Technology disabled.

Since the SPEC CPU2006 benchmark execution flow consists in serially launching several single threaded applications, independent instances have to be launched simultaneously to evaluate the system scalability. This also reflects the way many HEP applications are being ran. To scale out on modern multi-core systems, multiple instances of single threaded applications are being executed in parallel, often matching the core count of the host. This approach is aided by the fact that many times the events to be processed are completely independent one from each other.

HEPSPEC06 results and comparison

When comparing the three systems, frequency-scaled, the graph clearly shows an advantage to the most recent server generation, especially for the case of the quad socket platform.

In the first phase, when the systems count enough physical cores to accommodate the load, the quad socket E5-4650 exhibits almost linear scalability until reaching the total number of physical core (32).

The HEP-SPEC06 performance per core offered by the Sandy Bridge-EP E5-4650 processor is about 6% more than that offered by the Sandy Bridge-EP E5-2680 and 23% more than that of a Westmere-EX E7-4870 core. The performance per core comparison has been done at the maximum number of physical cores shared by each pair of processors: 16 in the case of the 2 servers with Sandy Bridge-EP

[1] Standard Performance Evaluation Corporation (http://www.spec.org)

processors and 32 in the case of the 2 quad socket servers (based on the Xeon E5-4650 and the Xeon E7-4870).

But when the we analyse the HEP-SPEC06 performance per core at the maximum CPU occupancy (16 cores for the 2 socket Sandy Bridge-EP E5-2680, 32 cores for 4 socket Sandy Bridge-EP E5-4650 and 40 cores for the Westmere-EX E7-4870) we discover that the two Sandy Bridge-EP processors have almost identical performance per core and they share an advantage of about 31% compared to the Westmere-EX.

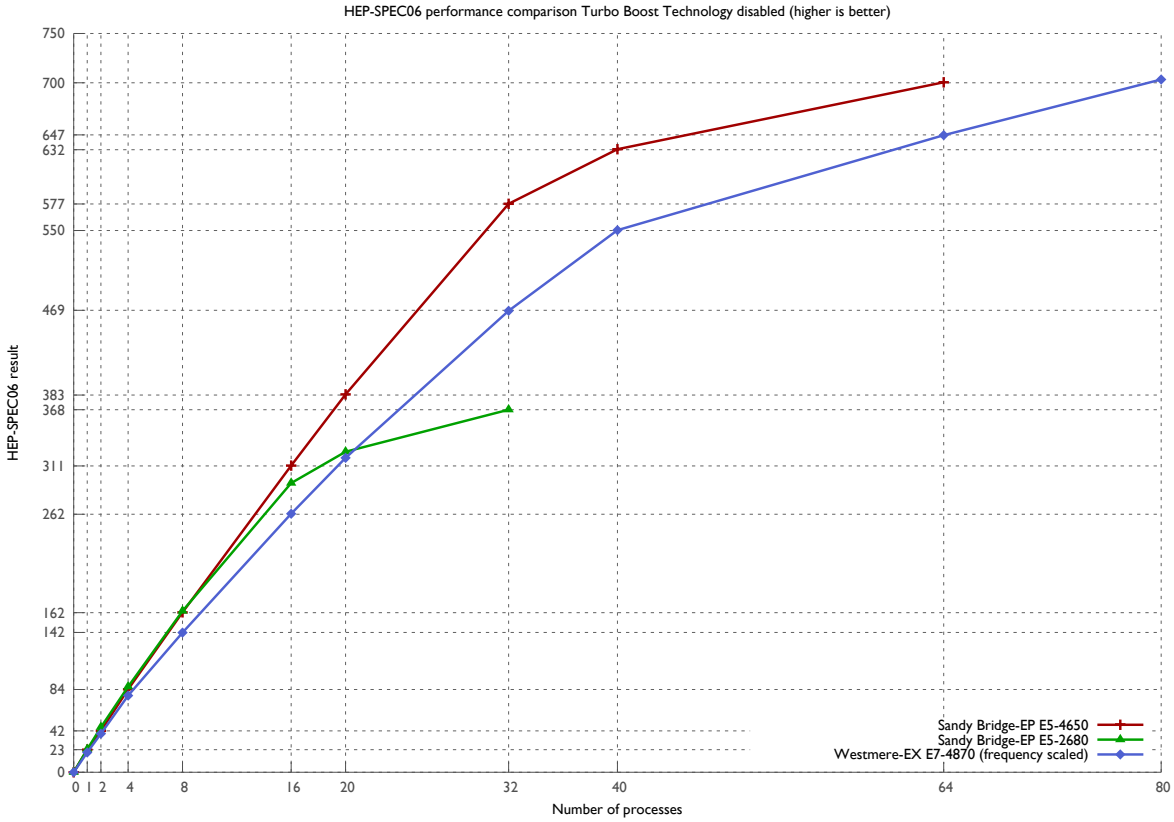


Figure 1: HEP-SPEC06 performance comparison Turbo Boost Technology disabled (higher is better)

This can be explained by the fact that the Sandy Bridge-EP scalability is better than the Westmere generation of CPUs in this initial phase: according to our previous Nehalem/Westmere HEP-SPEC06 comparison [OPL10], an inflexion was observed in Westmere-EP HEP-SPEC06 scalability between 8 and 12 cores, and this is not the case for the Sandy Bridge-EP. These results are also in line with our previous Westmere-EP/Sandy Bridge-EP HEP-SPEC06 comparison [OPL12], with the remark that with the increase in core count from the Westmere-EP to the Westmere-EX we see a slight decrease of performance per core in the case of the Westmere-EX.

SMT advantage

The gain produced by SMT can be computed by comparing the HEP-SPEC06 results for 32 and 64 cores in the case of the 2 socket Sandy Bridge-EP processor, for 16 and 32 cores in the case of the 2 socket Sandy Bridge-EP processor and for 40 and 80 cores for the Westmere-EX. In the case of the Westmere-EX the gain is 27.7%, for the dual socket Sandy Bridge-EP the gain is 25.3% and for the quad socket Sandy Bridge-EP the SMT gain is 21.3%. This shows that SMT is a well-established technology that steadily delivers around 25% of additional HEP-SPEC06 performance on Intel Xeon CPUs.

Overall performance

If we analyse the overall performance, with the systems fully loaded by matching the number of HEP-SPEC06 processes to the number of available hardware threads (physical cores + SMT) we discover that the performance of the quad socket Sandy Bridge-EP system roughly matches that of the quad socket Westmere-EX system, when frequency scaled. That's particularly impressive given the fact that the Westmere-EX packs 25% more cores than the Sandy Bridge-EP while also boosting 20% more L3 cache per core. That proves just how well the Sandy Bridge-EP microarchitecture has been improved compared to the Westmere-EX.

Multi-threaded Geant4 prototype

Geant4 is one of the main toolkits used in Large Hadron Collider (LHC) simulations. Its primary purpose is to simulate the passage of particles through matter. This type of simulation is a CPU-intensive part of a bigger overall framework used to process the events coming from the detectors. It is representative to an extent of real life workloads and can constitute a substantial portion of the CPU time of the Worldwide LHC Computing Grid. Since HEP has always been blessed with parallelism inherent in the processing model, it is natural to try to utilize modern multi-core systems by converging towards multi-threaded event processing. The Geant4 prototype discussed here is one of the key steps in that direction.

Based around Geant4, this suite was updated to support multi-threading by two Northeastern University researchers: Xin Dong and Gene Cooperman. The example used in this case is "ParFullCMSmt", a parallelized version of the "ParFullCMS" program, which represents a simulation close in properties to what the CERN CMS experiment is using in production. Thus, this is an example of a real-world application in use at CERN.

One of the key metrics of a parallel application and the underlying parallel platform is scalability. The tests described in this chapter focus on the scalability of the multi-threaded Geant4 prototype, which is defined as throughput. In principle, a single-threaded process has a specified average time it takes to process 100 events. Thus we measure the influence of the number of processes (and implicitly the number of processing units engaged) on the processing time of 100 events. In an ideal case, as more processes with more work are added, one would expect the throughput to grow proportionally to the added resources, and so the processing time of 100 events

would remain unchanged (per thread). Another key metric considered in this case is "efficiency", which is defined as the scaling of the software relative to the single thread runtime of the parallelized code, confronted with ideal scaling determined by the product of the core count and the single thread throughput. In cases where multiple hardware threads are being used, perfect scaling is defined by the maximum core count of the system (32).

Technical test setup

The threads were pinned to the cores running them, and the throughput defining factor was the average time it takes to process one 300 GeV pi- event in a predefined geometry based on the real world Compact Muon Solenoid (CMS) detector. Although the Turbo Boost Technology remained disabled, the system was SMT-enabled which means that the hardware threading feature was activated and used during the tests. Thus, if there were no more physical cores available, the jobs would be pinned to hardware threads, still maximizing the amount of physical cores used. The tested framework was based on Geant4 4.9.2p01, CLHEP 2.0.4.2 and XercesC 2.8.0, and was compiled using the GCC 4.4.6 compiler.

Scalability testing

The tests showed stable efficiency figures up to the full physical core count. When running with 32 software threads on 32 cores, the scaling factor was 15.72x, which means that the setup was 98% efficient. The figure matches good results observed on previous Intel processors, especially from the "Westmere" family. It also matches the results obtained on the two socket Sandy Bridge-EP. Detailed scalability data for intermediate points reveals good scaling, as expected, all the way between 1 and 32 processes.

The graph below (Figure 2) shows the total simulation time curve in green and the efficiency (scalability) curve in blue. Towards its end, the efficiency curve surpasses 100%, since for thread counts higher than 32 expected scalability is fixed to that obtained by using 32 threads. Thus a final value of 125% indicates that the system loaded with 64 threads of the benchmark yields 25% more throughput than in the case of running on 32 physical cores.

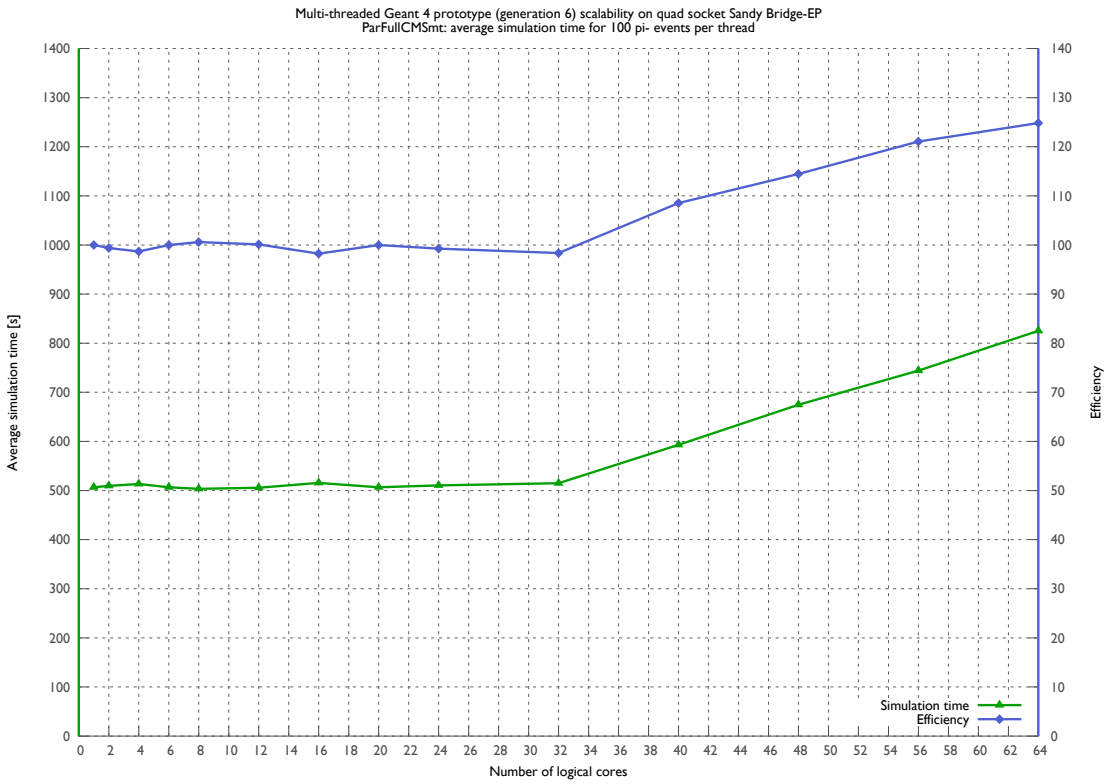


Figure 2: ParFullCMSmt scalability on quad socket Sandy Bridge-EP E5-4650

Hyper-threading advantage

The advantage of running with SMT was already 9% for 40 hardware threads, suggesting that the run with 32 threads (full physical core count) might be slightly suboptimal. The SMT gains later were 14% with 48 hardware threads, 21% with 56 hardware threads and finally 25% with all hardware threads engaged. This advantage matches the previously tested Sandy Bridge-EP dual-socket systems. One should note that this extra performance is traded in for a penalty in memory usage, as the number of software threads is twice the one in the case of 32 threads.

E7-4870 based "Westmere-EX" comparison

When compared to a Xeon E7-4870 based Westmere-EX quad socket platform tested earlier, the probed quad socket Sandy Bridge-EP system performs better in terms of frequency scaled performance, despite the fact that the Sandy Bridge-EP comes with 25% less cores and 20% less L3 cache per core. The overall advantage of the quad socket Sandy Bridge-EP solution over the quad socket Westmere-EX platform was established to be 2% at full load, which is when using 64 threads for the Sandy Bridge-EP, and 80 threads for the Westmere-EX.

A 4% average frequency-scaled advantage was noticed when running with up to 32 jobs to match the physical core count of the Sandy Bridge-EP system. This is attributed to the many innovations in the Sandy Bridge microarchitecture. The tested quad socket Sandy Bridge-EP solution continues the tradition of highly efficient quad

socket systems based on the Intel microarchitecture, delivering scalable and predictable performance.

Parallel Maximum Likelihood fit

The HEP community makes large use of many complex data analysis techniques, like maximum likelihood fits, neural networks, and boosted decision trees [STAT01]. These techniques are employed for a better discrimination between interesting events with respect to the total events collected by the physics detectors, in order to discover possible new physics phenomena [PHYS08]. The increase of the sample sizes and use of complex data analysis models require high CPU performance for the execution of the data analysis software applications. The execution can be speeded up by having recourse to parallel implementations of the data analysis algorithms, i.e. strong scaling of the parallel applications. Traditionally all software for data analysis developed in HEP do not use parallelism. However, with the introduction of multi-core CPUs, an effort for parallelization has been started in recent years. The benchmark used here is based on an unbinned maximum likelihood data analysis application. The code has been developed by CERN openlab, and it represents a prototype of the RooFit package (package inside the ROOT software framework developed at CERN), generally used in HEP for maximum likelihood fits [ROF06]. The prototype makes use of an optimized algorithm with respect to the algorithm used in RooFit for the likelihood evaluation [JAR11a]. The implementation of the algorithm has been parallelized using a number of different techniques: OpenMP, Intel Threading Building Blocks (TBB), Intel Cilk Plus and MPI [JAR12]. Because of the design of the algorithm, the initial implementation was subject to a high rate of last-level cache (LLC) load misses, i.e. shared L3 cache, and a significant OpenMP overhead that limited the scalability. These bottlenecks have been mitigated by an improved version of the code, described in Ref. [JAR11b]. In this new case, the OpenMP overhead becomes negligible (<1% when running with high number of threads). Then the high rate of LLC misses is reduced by means of splitting the data in blocks, achieving a better overlap between computation and memory accesses. Clearly the application will benefit from systems with a bigger L3 shared cache size. Furthermore, using a scattered affinity topology maximizes the cache memory available per thread, i.e. threads are bound to cores of CPUs on different sockets before filling the cores of a given CPU. For example, running with 8 threads on a quad-socket systems means 2 threads per CPU (instead of 8 threads on the same CPU). Note that the application takes in account NUMA effects. Considering these optimizations, the application is expected to scale close to the theoretical scalability predicted by the Amdahl's law.

The likelihood function definition is taken from the data analysis performed at the *BaBar* experiment [BBR09]. Thus, this is an example of a real-world application in use in the HEP community. The maximization of the likelihood function, or the equivalent minimization of the negative logarithm of the likelihood function (negative log-likelihood), is performed by using the MINUIT package [MIN72]. The main algorithm in this package, MIGRAD, searches for the minimum of a function using the gradient information [NUM07]. Several evaluations of the negative log-likelihood are

needed before reaching the minimum of the function, so it becomes important to speed up the evaluations. The implementation guarantees that the number of evaluations and the results do not depend on the number of executed parallel threads, i.e. the workload is fixed between the parallel executions.

All calculations are performed in double precision. Input data and results are organized in vectors, which are shared across the threads so that there is a small increase of the memory footprint with the number of threads. Input data was composed of 500,000 entries per 3 observables, for a total of about 12MB. Results are stored in 21 arrays of 500,000 values, *i.e.* about 110MB. Operations are performed on all elements by means of loops, so there are 500,000 iterations in total per loop. Loops are auto-vectorized by the Intel compiler and parallelized. The events are organized in blocks for the reason described before. A heuristic approach is followed to decide the dimensions of the blocks, which depend on the number of parallel threads. They can be put in relation with the cache size available per thread, so the dimensions decrease accordingly with the number of threads. In particular the block dimensions allow having the fastest execution for a given parallel execution. Table 2 reports the block sizes used in the tests.

Block size	*# Threads*		
(# events)	*E5-2680*	*E5-4650*	*E7-4870*
10,000	1 - 2	1 - 4	1 - 4
5,000	4	6 - 10	6 - 10
3,000	6	12	12
2,000	8 - 12	16 - 24	16 - 24
1,000	16 - 32	32 - 64	32 - 80

Table 2 Block dimensions. Note that: for the dual socket Sandy Bridge-EP E5-2680, 24 and 32 threads use SMT; for the quad socket Sandy Bridge-EP E5-4650, 32, 40 and 64 threads use SMT; for the quad socket Westmere-EX E7-4870, 64 and 80 threads use SMT.

First of all we look at the speed-up given by the vectorization. In this case the speed-up is defined as the ratio between the execution times of the non-vectorized and vectorized code. Note that Sandy Bridge microarchitecture introduces a new set of SIMD instructions, namely Advanced Vector Extensions (AVX) instructions, which enables 256-bit vector registers. These instructions extend the SSE ones, which use 128-bit vector registers.

Secondly we look at the speed-up given by the Turbo Boost Technology, so, in this case, the speed-up is defined by the ratio between the execution time with Turbo Boost Technology disabled and Turbo Boost Technology enabled. The comparison is done when running the application in sequential (a single thread execution) and in parallel. Turbo Boost Technology is expected to give the maximum contribution when the system is loaded with a low number of threads per CPU. However, a small overall benefit is expected also when running with fully loaded parallel threads, just because the Turbo Boost Technology speeds up the application during its sequential portion. Finally we discuss the performance of the sequential and parallel executions and we show the scalability results, including the SMT benefit. All results are compared with

the Westmere-EX system, which is frequency-scaled (nominal frequency) to allow a direct comparison of the performances between the systems.

Test setup

The systems are SMT-enabled, which means that the hardware threading feature was activated and used during the tests. Thus, if there are no more physical cores available, the jobs are pinned to hardware threads by requiring 2 threads per core.

Timings are obtained from the average over three consecutive runs of the application (errors are <0.5%). The sequential execution time of the application is 379 seconds when running on the quad socket Sandy Bridge-EP without vectorization and Turbo Boost Technology disabled (the slowest execution). It is worth to underline that the application is floating-point intensive; in particular the execution of the exponential function takes about 60% of the total execution time.

Vectorization results

We compile the code in 3 different configurations by using the ICC flags:

1. `-no-vec`
2. `-msse3`
3. `-xAVX`

We found that the speed-up results do not significantly depend on the number of threads and the enabling or disabling of the Turbo Boost Technology. The average speed-up results are shown in Table 3. The Westmere-EX system has a slightly lower speed-up of 1.88x when using the `-msse3` configuration.

	`-no-vec`	`-msse3`
`-msse3`	1.90x	
`-xAVX`	2.20x	1.15x

Table 3 Average speed-up results for different vectorization configurations (given by the indicated compiler flags) on the Sandy Bridge-EP system. The results are obtained from the ratio between execution times for the configurations in the columns and the corresponding configurations in the rows.

Turbo Boost Technology results

Speed-up results obtained from the comparison of the execution times with Turbo Boost Technology enabled and disabled are shown in Figure 3.

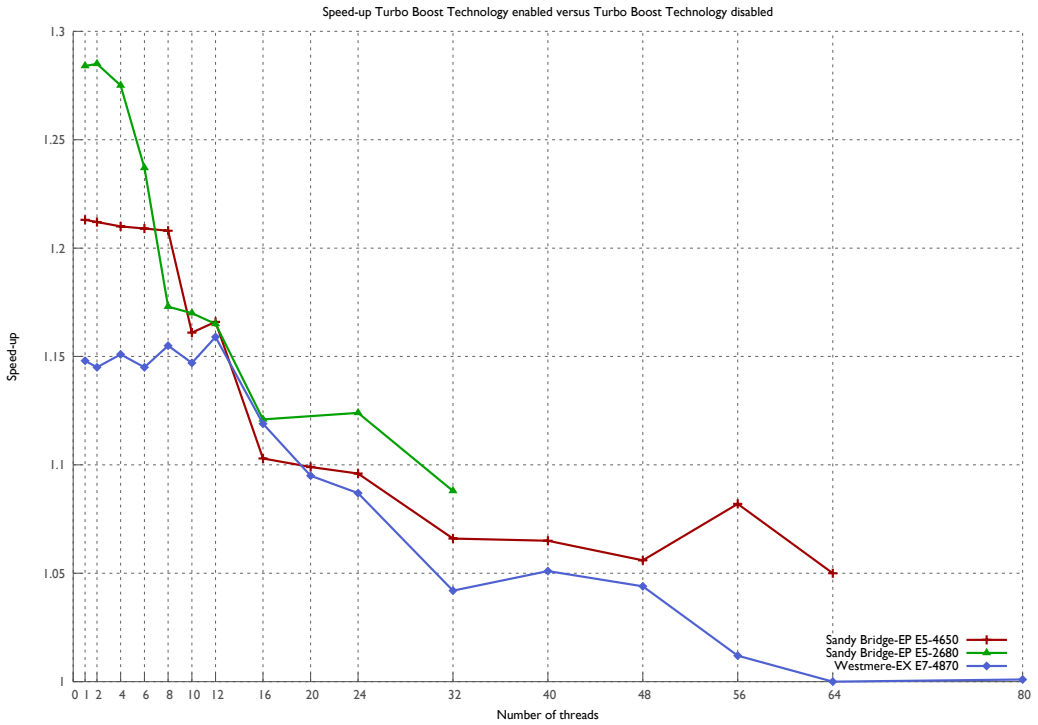


Figure 3 Speed-up ratios obtained from the comparison of the execution times with Turbo Boost Technology disabled and Turbo Boost Technology enabled.

We found that the effect of the Turbo Boost Technology does not depend on the vectorization mode. When using one thread per CPU (4 threads in total for the quad socket systems and 2 threads in total for the dual socket server) the speed-up of the Turbo Boost Technology is compatible with the increase in frequency from the nominal value:

- 2.70 GHz to 3.46 GHz for Sandy Bridge-EP E5-2680 (1.28x)
- 2.70 GHz to 3.30 GHz for Sandy Bridge-EP E5-4650 (1.22x)
- 2.40 GHz to 2.80 GHz for Westmere-EX E7-4870 (1.17x)

For the Westmere-EX system the effect of Turbo Boost Technology is significant less given the much smaller increase from the nominal frequency to the Turbo Boost frequency. The benefit of the Turbo Boost Technology decreases when more parallel threads are executed. With 16 threads on the dual socket Sandy Bridge-EP the speed-up drops to 1.12x, with a variation of 0.16x with respect to the single thread case. With 32 threads on the dual socket Sandy Bridge-EP the speed-up drops to 1.09x, with a variation of 0.19x with respect to the single thread case. For the corresponding case (40 threads) for the Westmere-EX the variation is 0.10x. When also SMT is used (fully loaded system), Turbo Boost Technology still has a contribution for the dual socket Sandy Bridge-EP system (1.09x) and the quad socket Sandy Bridge-EP system (1.05x), but virtually no effect in the case of the quad core Westmere-EX. To conclude, the Turbo Boost Technology behaves much better on the Sandy Bridge-EP system with respect to the Westmere-EX, reaching higher speed-up in most cases.

Performance comparison and scalability results

We compare the execution time of the application compiled with AVX vectorization on the Sandy Bridge-EP systems with respect to the execution time when running with SSE vectorization on the Westmere-EX. All three systems have Turbo Boost Technology enabled. Comparing the corresponding runs executed with the same number of threads on the quad socket systems, i.e. between 1 and 32 threads, we obtain that the Sandy Bridge-EP system is on average 1.48x faster than the Westmere-EX system. Removing the speed-up due to the AVX vectorization and Turbo Boost Technology enabled, we obtain that the Sandy Bridge-EP single core performs 1.23x faster than the Westmere-EX one. The same factor can be simply obtained from the comparison of the execution time of the runs with SSE vectorization and Turbo Boost Technology disabled on both systems. Then we compare the fully loaded quad socket Westmere-EX and Sandy-Bridge-EP systems with and without SMT, i.e. 32 versus 40 and 64 versus 80 threads, respectively. We obtain that the overall gain in performance between the two systems is 1.18x, without SMT, and 1.19x, with SMT. Note that the very small increment in the comparison of the overall performance is due to the higher number of threads running on the Westmere-EX system. Also in this case we can compute the speed-up eliminating the effects of AVX vectorization and Turbo Boost Technology, which is about 0.96x. We obtain that this number is in agreement with the equivalent single core speed-up value (1.23x) scaled by the different core counts of the systems. If we also consider SMT the speed-up is 1.02x, matching the results obtained with the HEP-SPEC06 benchmark.

We perform scalability tests using the AVX vectorization configuration and Turbo Boost Technology disabled. In these tests the speed-up is defined as the ratio between the execution times spent by the application running with a single thread and with a certain number of threads in parallel. The fraction of the execution time spent in code that we can execute in parallel is 99.7%. We should underline that because Turbo Boost Technology contributes mainly when running with a low number of threads per CPU, it worsens the scalability for a high number of threads. This is the reason why we run the tests with Turbo Boost Technology disabled. We show the scalability results in Figure 4. We can clearly see that the scalability is very close to the theoretical expectation.

Finally we determine the contribution given by using SMT. This contribution is obtained from the ratio of the execution times of the application when running with maximum number of threads without and with SMT, i.e. 32 and 64 threads. The results do not depend on the vectorization, while Turbo Boost Technology gives a small impact: 1.17x and 1.14x for Turbo Boost Technology disabled and enabled, respectively. The speed-up for the non vectorized version is 1.35x whereas for the vectorized one is 1.17x. Corresponding values for Westmere-EX are worse. This can be attributed to the higher numbers of threads involved in the Westmere-EX case.

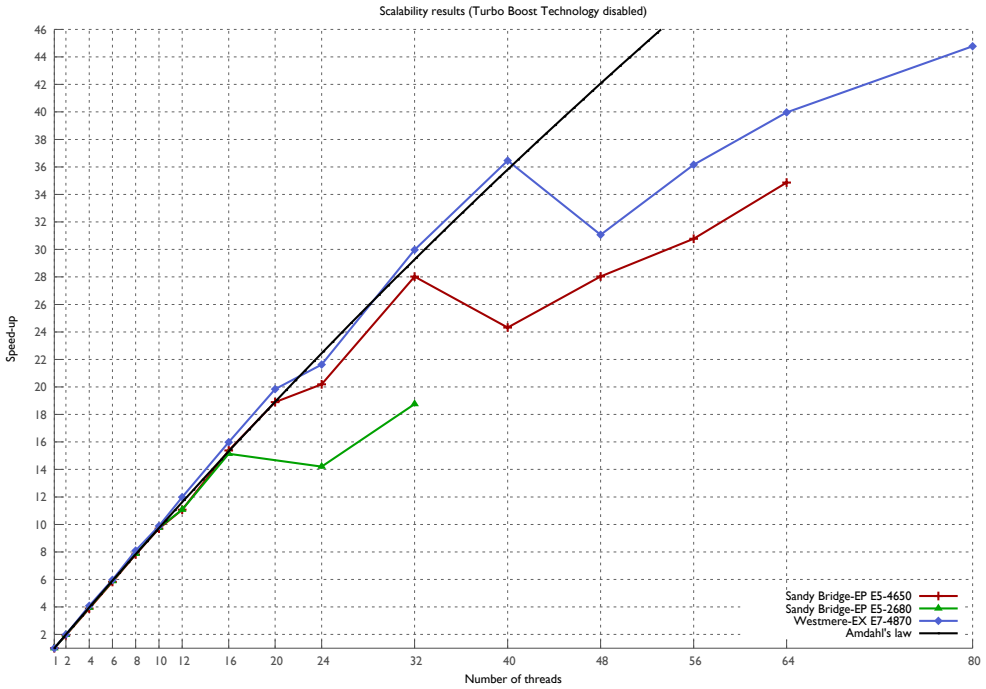


Figure 4 Scalability results. The solid black line is the theoretical speed-up obtained by Amdahl's law with a parallel fraction of 99.7%.

Conclusions and summary

The performance of the quad socket Sandy Bridge-EP E5-4650 processor evaluated in this report closely matches that of the two socket Sandy Bridge-EP E5-2680, which was to be expected as they share most characteristics, the Xeon E5-4650 being the quad socket version of the Xeon E5-2680.

What's been most interesting was the way the quad socket Sandy Bridge-EP system compared with the Westmere-EX server. Even with 25% less cores and 20% less L3 cache it still managed to outrun the Westmere-EX in most of the benchmarks that we've performed. The reason for the additional performance can therefore be found in the combination of improved thermal management inside each core and a more efficient microarchitecture.

In the HEP-SPEC06 tests the quad socket Sandy Bridge-EP system performed at the same level as the quad socket Westmere-EX when frequency scaled (within 0.5% one from each other). Given the extra 25% cores of the Westmere-EX that means that we have a 25% performance increase per core when both servers were fully loaded. If we consider the performance per core gains using a number of threads matching the number of physical cores on each system the figure increases to 31%. That's due to the fact that the gain with SMT is 27.7% for the Westmere-EX and 21.3% for the quad socket Sandy Bridge-EP.

When we tested weak scaling using the Multithreaded Geant4 benchmark we found that performance increased by 2% when using all SMT cores on both quad socket

servers. This is very similar to the results obtained using HEP-SPEC06. The SMT benefit was 24.8%.

When running the Parallel Maximum Likelihood fitting benchmark, which has a fixed problem size enforcing a strong scaling behaviour, we obtained a 23% performance increase per core when using SSE on both generation of processors. This corresponds to about equivalent performance when considering all available cores – very similar to the HEP-SPEC06 results. The pleasant surprise with this benchmark was the fact that the Intel compiler allows a further 15% gain when using autovectorization with AVX. This is quite a bit below the theoretical gain of 100% but it has to be remembered that mathematical functions, such as divide and square root, have roughly the same throughput as on Westmere-EX.

In conclusion we confirm that the quad socket Sandy Bridge EP processor is a significant improvement in terms of performance per core when compared to the previous Westmere-EX generation which is Intel's flagship platform when it comes to expandable architectures. We also found that the Turbo Boost Technology has been improved. With this new generation Intel has allowed expectations to be set at a high level and we are keen to see whether the pace of improvement can be sustained for the 22 nm processors, namely the Ivy Bridge-EP and Ivy Bridge-EX processors planned for 2013 and Haswell for 2014.

References

WLCG09	Multiple authors: *Transition to a new CPU benchmarking unit for the WLCG* (2009)
OLP10	S. Jarp *et al.*: *Evaluation of the Intel Westmere-EP server processor*, CERN openlab (2010). EPRINT: CERN-IT-Note-2011-004
OLP12	S. Jarp *et al.*: *Evaluation of the Intel Sandy Bridge-EP server processor*, CERN openlab (2012). EPRINT: CERN-IT-Note-2012-005
STAT01	J. Friedman, T. Hastie and R. Tibshirani: *The Elements of Statistical Learning*, Springer (2001)
PHYS08	G. Kane, A. Pierce: *Perspectives on LHC Physics*, World Scientific (2008)
ROF09	W. Verkerke and D. Kirkby: *The RooFit Toolkit for Data Modeling*, proceedings of PHYSTAT05, Imperial College Press (2006)
JAR11a	S. Jarp *et al.*: *Evaluation of Likelihood Functions for Data Analysis on Graphics Processing Units*, ipdpsw, pp. 1349--1358, 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum (2011). EPRINT: CERN-IT-2011-010
JAR11b	S. Jarp *et al.*: *Parallel Likelihood Function Evaluation on Heterogeneous Many-core Systems*, proceeding of International Conference on Parallel Computing, Ghent, Belgium (2011). EPRINT: CERN-IT-2011-012

JAR12	S. Jarp *et al.*: *Comparison of Software Technologies for Vectorization and Parallelization*, CERN openlab (2012)
BBR09	B. Aubert *et al.*: B *meson decays to charmless meson pairs containing η or η' mesons*, Phys. Rev. D80, 112002 (2009)
MIN72	F. James: *MINUIT - Function Minimization and Error Analysis*, CERN Program Library Long Writeup D506 (1972)
NUM07	W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery: *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press (2007)

Appendix A - standard energy measurement procedure

Measurement equipment

A high precision power analyser with several channels is required for the measurements, since it must be able to measure the power consumption of any system from a simple uni-processor system, with a single power supply unit (PSU) to a large server equipped with 4 PSUs.

To this extend a ZES-Zimmer LMG450 power analyser is used. It allows the measurement of common power electronics. It has an accuracy of 0.1% and allows the measurement of four channels at the same time, and thanks to its convenient RS232 port, it can be linked to a PC to sample the values on the 4 channels, as shown on Figure 5.

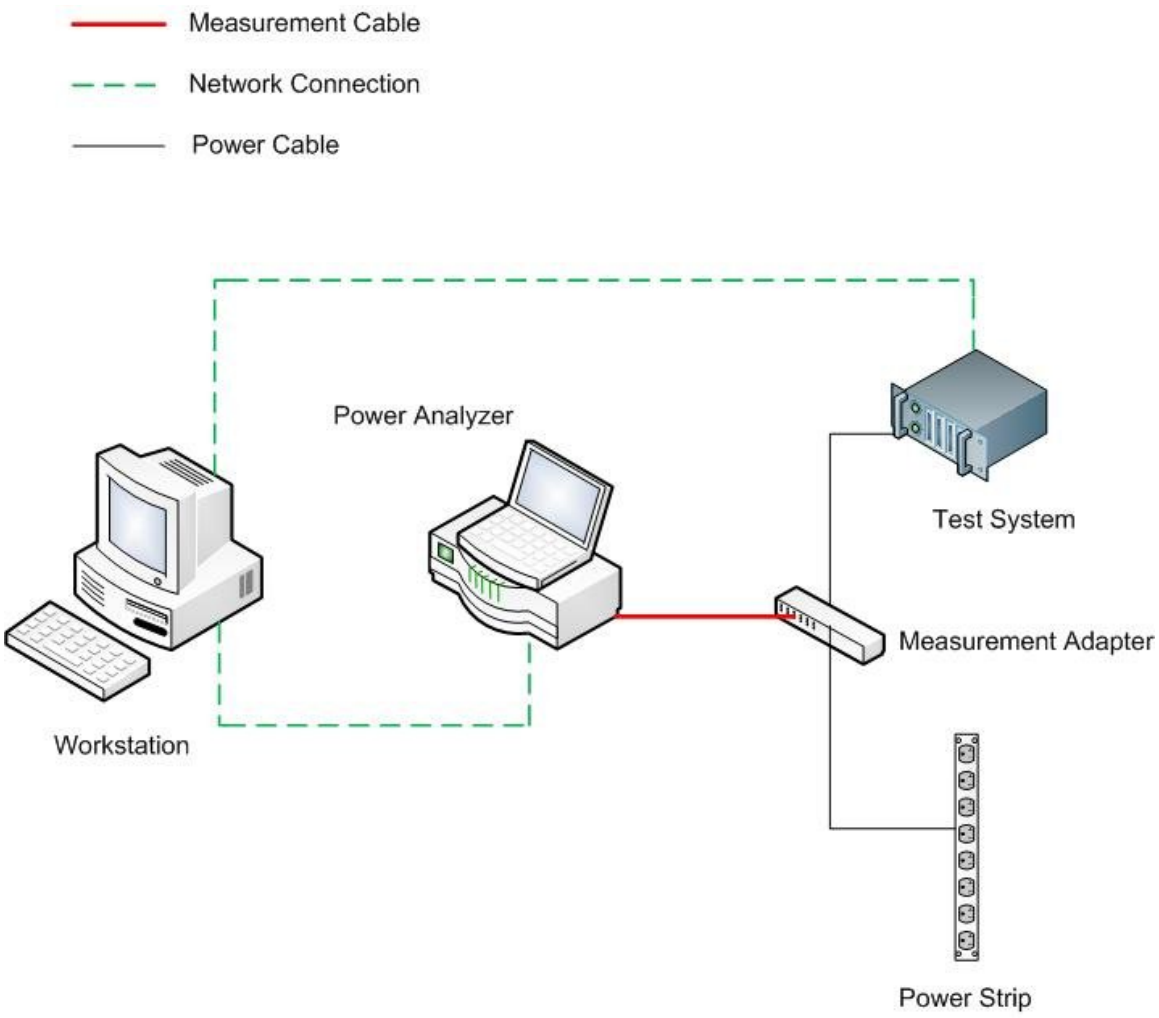


Figure 5: Power test setup

Three units are measured for each channel:

- *Active Power (P)*: The active power is also often called "real" power and is measured in Watts (W). If the active power is measured over time the energy in kilowatt-hours is determined.
- *Apparent Power (S)*: Apparent power is the product of voltage (in volts) and current (in amperes) in the loop. This part describes the consumption from the electrical circuit. It is measured in VA.
- *Power Factor:* In our case, the power factor means the efficiency of the power supply. The closer the power factor is to one, the better is the efficiency: power factor = active power/apparent power

If the system includes several PSUs the Active Power must be summed on all the channels in use to compute the total Active Power of the system, for the two stress conditions.

LINPACK/CPU Burn-in

Those two tools are used to stress the evaluated systems, providing a reproducible load for any type of server:

1. *The Intel Optimized LINPACK Benchmark* is a generalization of the LINPACK 1000 benchmark. It solves a dense (real*8) system of linear equations (Ax=b) and tests the results for accuracy. The generalization is in the number of equations (N) it can solve, which is not limited to 1000. It uses partial pivoting to assure the accuracy of the results. It is used to load both the memory system and the CPU. The memory consumption depends on the size of the generated matrices and is easy to adapt to fit the needs.
2. *CPU Burn-in* was originally written as a tool for overclockers, so that they can stress the overclocked CPUs, and check if they are stable. It can report if an error occurs while the benchmark is running. It runs Floating Point Unit (FPU) intensive operations to get the CPUs under full load, allowing the highest power consumption to be measured from the CPU.

Standard energy measurement

The standard energy measurement is a combination of the Active power measured under two different stress conditions:

1. *Idle*: the system is booted within the Operating System and is left idle.
2. *Load*: the system is running CPU Burn-in on half of the cores, and LINPACK on all the other cores, using all the installed memory.

An example to stress a system counting 32 cores and 64 GB of memory for the Load condition, would imply to run 16 instances of CPU Burn-in along with 16 instances of LINPACK, each consuming 4 GB of memory.

According to that, the standard energy measurement is a mix of the active power under the Idle condition, accounting for 20%, and the active power under Load condition, accounting for 80%.