Many-core experience with HEP software at CERN openlab

2012 J. Phys.: Conf. Ser. 396 042043

(http://iopscience.iop.org/1742-6596/396/4/042043)

Journal of Physics: Conference Series **396** (2012) 042043 doi:10.1088/1742-6596/396/4/042043

Many-core experience with HEP software at CERN openlab

Sverre Jarp, Alfio Lazzaro, Julien Leduc, Andrzej Nowak

CERN openlab, Geneva, Switzerland

E-mail: {sverre.jarp, alfio.lazzaro, julien.leduc, andrzej.nowak}@cern.ch

Abstract. The continued progression of Moore's law has led to many-core platforms becoming easily accessible commodity equipment. New opportunities that arose from this change have also brought new challenges: harnessing the raw potential of computation of such a platform is not always a straightforward task. This paper describes practical experience coming out of the work with many-core systems at CERN openlab and the observed differences with respect to their predecessors. We provide the latest results for a set of parallelized HEP benchmarks running on several classes of many-core platforms.

1. Introduction

In this paper we report on a set of benchmark results obtained by CERN openlab [1] when comparing two groups of systems. One is the 4-socket, 40-core Intel Xeon "Westmere-EX" server (E7-4870 at 2.4GHz) with the previous generation 4-socket "Nehalem-EX" server, based on the 8-core Xeon X7560 processor at 2.27GHz. The Xeon E7-4870 processor continues the tradition of gradual change through a shrink of the silicon process, with the most notable improvement being the increased platform core count. The second group compares the 8-core "Sandy Bridge-EP" processor with an updated architecture (E5-2680 at 2.70GHz), which is compared with representatives of Intel's previous generation, "Westmere-EP". Both platforms are dual-socket servers.

Multiple benchmarks were used to get a good understanding of the performance of the new processors. We used both industry-standard benchmarks, such as SPEC2006, and specific high energy physics (HEP) benchmarks, representing both simulation of physics detectors and data analysis of physics events. This paper outlines the results of these benchmarks, while additional details can be found in our previous research [2].

Finally, we should note that benchmarking of modern processors is an increasingly complex affair. One has to control (at least) the following features:

- Processor frequency
- Overclocking via Turbo mode, which allows the processor to increase its frequency when only few of the cores are in use, while maintaining power within the designed envelope
- The number of physical cores in use
- The use of logical cores via Simultaneous Multi-Threading (SMT), which allows two hardware threads to be scheduled on each core
- The cache sizes available
- The memory configuration installed
- The power configuration if throughput per watt is to be measured.

In particular, given the improvements in process technology, the Turbo feature has recently been experiencing rapid development, as documented in this paper. We have tried to do a good job of comparing like with like.

The paper is organized as follow: section 2 describes the benchmark applications; section 3 the hardware and software setup; section 4 the standard energy measurements; section 5 summarizes the benchmark results. Conclusions are given in section 6.

2. Benchmark overview

2.1. HEPSPEC06

One of the important performance benchmarks in the IT industry is the SPEC CPU2006 benchmark from the SPEC Corporation [3]. This benchmark can be used to measure both individual CPU performance and the throughput rate of servers. It is an industry standard benchmark suite, designed to stress a system's processor, the caches and the memory subsystem. The benchmark suite is based on real user applications, and the source code is commercially available. A HEP working group demonstrated a few years ago a good correlation between the SPEC results and HEP applications when using the C++ subset of the tests from the SPEC CPU2006 benchmark suite [4]. As a result the HEP community has decided to use the C++ subset of SPEC2006, "HEPSPEC06" rather than internal benchmarks because SPEC2006 is readily available, and computer manufacturers can directly generate its results to evaluate the performance of a system aimed at running HEP applications.

Since SPEC CPU2006 benchmark execution flow consists in serially launching several single threaded applications, several independent instances have to be launched simultaneously to evaluate the system scalability. This means that the HEPSPEC06 benchmark is indeed a rate measurement.

2.2. Multi-threaded Geant4 prototype

Geant4 is one of the principal toolkits used in Large Hadron Collider (LHC) simulation [5]. 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. HEP has always been blessed with parallelism inherent in the processing model – physics events in detectors are essentially independent, which makes them a natural target for software and hardware parallelism. It is then only 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 has been updated to support multi-threading by two Northeastern University researchers: Xin Dong and Gene Cooperman [6]. 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*. For the tests relevant to the multi-threaded Geant4 benchmark the scalability is defined as throughput, according to Gustafson's Law (and not Amdahl's Law). 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.

2.3. Maximum likelihood fit prototype

The HEP community makes large use of many complex data analysis techniques for a better discrimination between interesting events with respect to the total events collected by the physics detectors. 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. The benchmark used here is based on an unbinned maximum likelihood data analysis application [7]. 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 the HEP for maximum likelihood fits [8]. The prototype makes use of an optimized algorithm with respect to the algorithm used in RooFit for the likelihood evaluation [9]. The likelihood function definition is taken from the data analysis performed at the BaBar experiment [10]. 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 [11]. 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. 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.

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 is composed by 500,000 entries per 3 observables, for a total of about 12MB. Results are stored in 29 vectors of 500,000 values, i.e. about 110MB, which are combined to get the value of the negative log-likelihood function. 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 using OpenMP. The events are organized in blocks to achieve a better overlap between computation and memory accesses. 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 4 threads on the dual-socket systems means 2 threads per CPU (instead of 4 threads on the same CPU). Note that the application takes in account Non-Uniform Memory (NUMA) effects. Considering these optimizations, the application is expected to scale close to the theoretical scalability predicted by the Amdahl's law. The fraction of the execution time spent in code that we can execute in parallel is 99.7%.

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. Secondly we look at the speed-up given by the Turbo mode, so, in this case, the speed-up is defined by the ratio between the execution time with Turbo mode off and Turbo mode on. The comparison is done when running the application in sequential (a single thread execution) and in parallel. The Turbo mode 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 mode 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 benefit from using hardware threading (HyperThreading or SMT).

3. Hardware and software setup

3.1. "Westmere-EX" system

3.1.1. Description of the processor

The codename for Intel's Xeon 7000 Family microarchitecture is "Nehalem". The initial expandable "EX" flavor of Nehalem processors used the same 45 nm manufacturing process as the previous "Dunnington" processors. According to Intel's well-known "tick-tock" model, the following microarchitecture evolution is a "tick", which corresponds to a shrink of the manufacturing process. "Westmere" is the name given to this 32 nm die shrink of Nehalem. This shrink is now offered to the expandable server market with the "Westmere-EX" coming out along with a new naming scheme: the Intel Xeon Processor E7 family.

The first new properties of the Xeon E7 family come directly from the shrink, allowing Intel to pack 25% more cores in each die (reaching 10 cores in total), and also 25% more of the shared L3 cache (reaching 30MB), compared to "Nehalem-EX" Xeon 7500 processors. Moreover, Intel managed to add these 2 cores, and the 6MB bigger L3 cache, while increasing the frequency and keeping power consumption within the same thermal design envelope. In this paper we compare the high end Westmere-EX E7-4870, which has 10 cores (20 threads) running at 2.40GHz with a Thermal Design Power (TDP) of 130W, with the corresponding "Nehalem-EX" X7560, clocked at 2.27GHz with the same 130W TDP and 8 cores (16 threads). These two processors have a fair number of features in common: they are both equipped with four Quick Path Interconnect (QPI[1]) links at 6.4 GT/s, four Scalable Memory Interconnect (SMI[2]) links and they both fit in the same LGA-1567 socket. Both support SMT and Turbo mode. It should be noted, however, that the Turbo bins for the "Westmere-EX" part have been upgraded from 2.67GHz for the "Nehalem-EX" to 2.8GHz in the latter case.

3.1.2. System configuration

As stated in the processor description, both processors fit in the same socket and have the same TDP. This allowed us to use exactly the same platform, after an extensive firmware upgrade session, for all our measurements, switching only the processors between the two series of experiments.

Our test system is a QSSC-S4R server jointly developed by Intel and Quanta. It provides four LGA-1567 sockets to connect up to four Xeon 7500/E7-4000 series processors and two Boxboro-EX IOH chipsets to handle the IO as illustrated on figure 1.

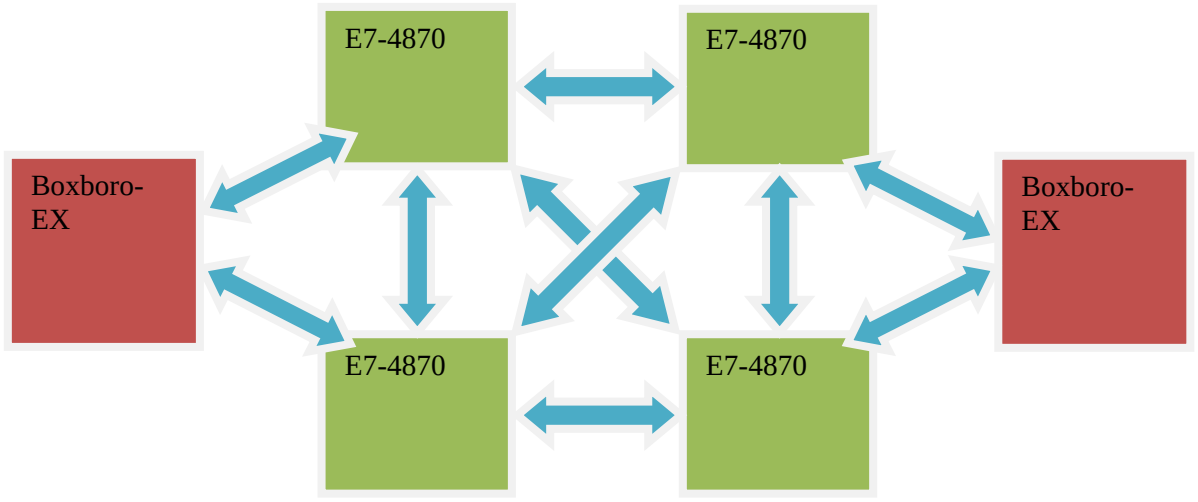

Figure 1. QPI topology of the full system.

[1] The QuickPath Interconnect protocol is a point-to-point processor interconnect developed by Intel, replacing the legacy Front Side Bus (FSB).
[2] SMI is the interface between the processor and the memory

Up to 10 PCIe expansion boards can be plugged in this 4U server, and an Intel RS2BL080 SAS/SATA RAID card for front slot hard drive connectivity occupies one of those slots. This card supervises two 146.8 GB SAS hard drives in a RAID0 configuration.

Since the memory configuration is crucial for good performance of this class of servers, geared for the enterprise rather than typical HEP tasks, the Xeon E7-4000 series is no exception having room for of 32 x 4 GB memory DIMMs. The system embeds eight memory boards; each of those boards is connected to two SMI links of a single processor. With this configuration all the SMI links of the four processors can be exploited. According to the previous processor description, each SMI link can handle four memory DIMMs through a SMB, but the available memory allowed only to accommodate two slots out of four for each SMI link. Thus the chosen last level topology balances the DIMMs on the SMBs, allowing the test system to maximize the memory bandwidth of the underlying processors configuration.

To be able to exploit all 128 GB of RAM, on a system that is fully populated with 8 memory boards, the system is powered by three Power Supply Units (PSUs), out of a maximum of four.

3.1.3. Software configuration

The system was running 64-bit Scientific Linux CERN 5.6 (SLC5), based on Red Hat Enterprise Linux 5 (Server). The default SLC5 Linux kernel (version 2.6.18-238.9.1.el5) was used for all the measurements.

3.2. "Sandy Bridge-EP" system

3.2.1. Description of the processor

We compare the two Xeon "EP" generations produced in 32 nm silicon process. The first generation, "Westmere-EP" or Xeon 5600, was produced about two years ago as a shrink of the 45 nm-based "Nehalem-EP" processor. The "Westmere-EP" contained up to 6 cores and 12 MB of L3 cache on the die. The "Sandy Bridge-EP" processor increases these features to 8 cores and 20 MB of L3 cache. Since both processors are produced in the same silicon process, the Sandy Bridge processor is necessarily quite a bit larger (435 mm^2) than the Westmere-EP processor (248 mm^2). The increase is about 75% that reflects well the L3 cache memory increase (67%), given that cache memory occupies by far the largest portion of the die. Nevertheless, the TDP for the two generations is more or less unchanged, reflecting the considerable amount of effort that Intel is putting into optimizing the power consumption. The highest TDP for "Westmere-EP" was 130W and for "Sandy Bridge-EP" 135W. In the comparison we actually use several "Westmere-EP" processors, with multiple TDP values (and frequencies lower than the peak) but all our results are either calculated as performance/Watt or shown as frequency-scaled absolute performance, so, in our experience, this is a fair way of doing comparisons.

Both generations support SMT and Turbo mode. 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-EP architecture and its predecessors. Consequently, this is a feature we have tested extensively in the Maximum Likelihood fit benchmark, since it allows good usage of this SIMD feature after a recompilation. This feature was not tested in other benchmarks.

3.2.2. System configuration

The "Sandy Bridge-EP" processor evaluated in this paper is an E5-2680 running at 2.70GHz. The motherboard installed in our test system is an Intel S2600CP motherboard that supports up to 16 DDR3 memory DIMMs. This test system is equipped with 32 GB of memory (8 x 4 GB DIMMs 1333MHz Low Voltage), and a 1TB SATA hard drive.

3.2.3. Software configuration
The system is running 64-bit Scientific Linux CERN 6.2 (SLC6), based on Red Hat Enterprise Linux 6 (Server). The default SLC6 Linux kernel (version 2.6.32-220.el6) was used for all the measurements. Some SLC5 measurements where performed and then, the system was running 64-bit Scientific Linux CERN 5.7 on the default kernel (version 2.6.18-274.3.1.el5).

4. Standard energy measurements

4.1. Power meter procedure
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. For the measurements a high precision power analyzer with several channels is required, since it must be able to measure the power consumption of any system from a simple UP system, with a single PSU to a large server equipped with 4 PSUs. To this extend a ZES-Zimmer LMG450 power analyzer 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 2.

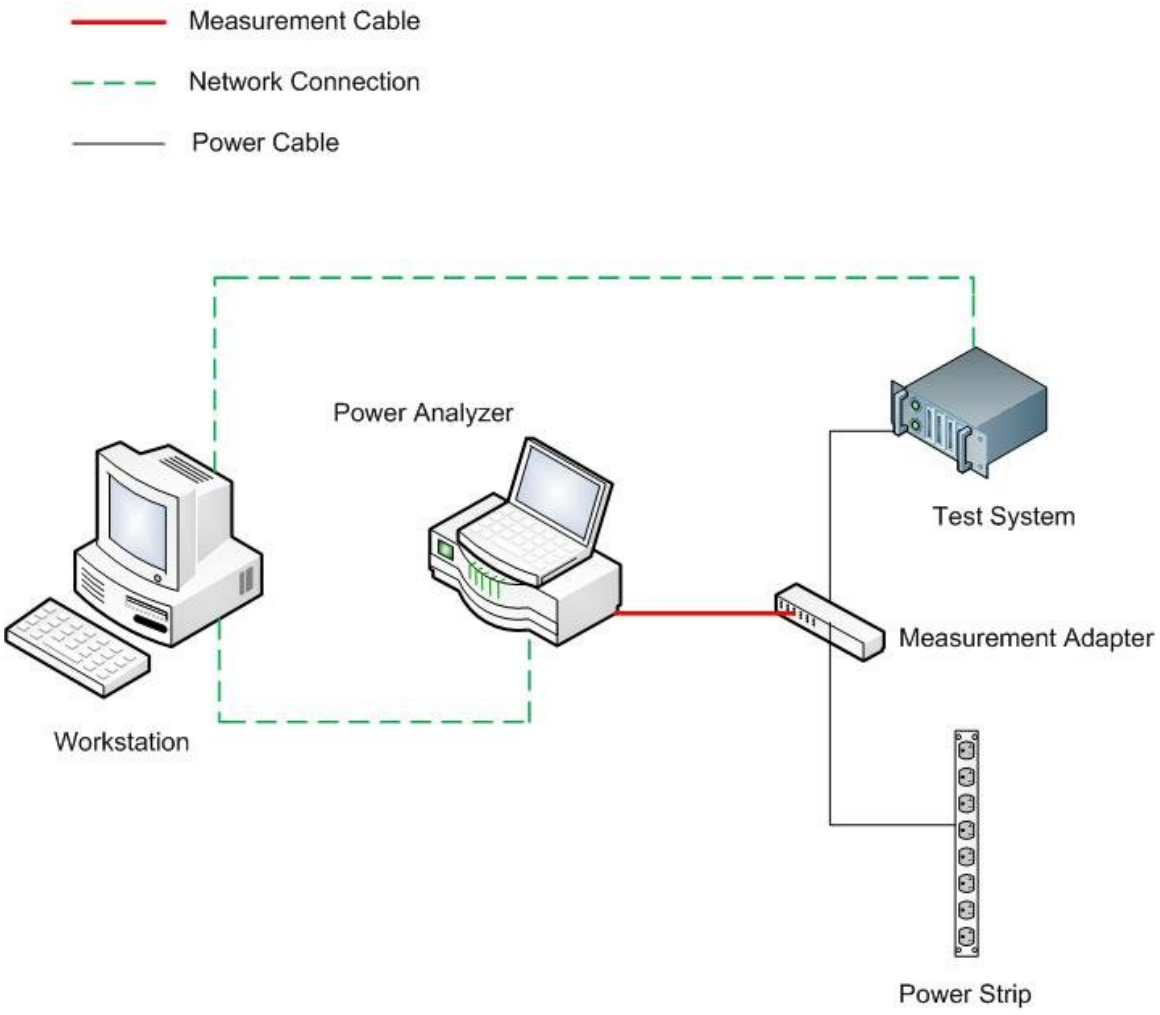

Figure 2. Power test setup.

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 = P/S*.

If the system includes several PSUs, *P* must be summed on all the channels in use to compute the total Active Power of the system.

Two stress tests are considered, providing a reproducible load for any type of server:

- *LAPACK* (Linear Algebra PACKage) was written in Fortran90 and is used to load both the memory system and the CPU. It provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The memory consumption depends on the size of the generated matrices and is easy to adapt to fit the needs.
- *CPUBurn* 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.

The standard energy measurement is a combination of the Active Power measured subjected to two different stress conditions:

- *Idle*: the system is booted with the Operating System and it does nothing.
- *Load*: the system is running CPUBurn on half of the cores, and LAPACK on all the other cores, using all the installed memory.

An example to stress a system counting 8 cores and 12 GB of memory for the Load condition, would imply to run 4 instances of CPUBurn along with 4 instances of LAPACK each consuming 3 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%.

4.2. “Westmere-EX” system

The system is equipped with three power supplies, and the resulting total power consumption is the sum of the three power consumption measurements on those PSUs.

When conducting the tests without SMT, the system exposes 40 cores in total. Thus, according to the standard energy measurement procedure, the load stress consists of running 20 instances of CPUBurn along with 20 instances of LAPACK (using 6 GB of memory each). In the second phase, now with SMT enabled, the system was considered as an 80 core server, meaning that the load stress test should be conducted by running 40 instances of CPUBurn along with 40 instances of LAPACK (using 3 GB of memory each). Results are shown in table 1. As we can observe, these power consumption measurements reach some sizeable figures, even when the server is idle. The power consumption measurements for the ”Westmere-EX” system are the same as the ones for the “Nehalem-EX” based server. This sounds reasonable considering that those two processors are rated at the same 130W TDP, and that those servers are using exactly the same hardware components except for the processor SKUs[3].

[3] A Stock-keeping Unit (SKU) is a unique code identifying a particular product item

Table 1. Total power consumption for "Westmere-EX" system using three PSUs.

SMT status	*Idle*	*Load*	*Standard measurement*
OFF	688 W	1211 W	1106 W
ON	688 W	1246 W	1134 W

4.3. "Sandy Bridge-EP" system

The system is equipped with one PSU. When conducting the tests without SMT the dual-socket system appears as having 16 cores in total, so that the load stress test consists of running 8 instances of CPUBurn along with 8 instances of LAPACK, (using 4 GB of memory each). In the second phase with SMT enabled the system was considered as a 32 core server, meaning that the load stress test should be conducted by running 16 instances of CPUBurn along with 16 instances of LAPACK, (using 2 GB of memory each). Table 2 shows the power measurements when the server is running two different versions of SLC: SLC5 and SLC6. A first look at these power consumption measurements shows an impressive difference of the overall system electrical consumption when comparing the idle and the Load state. Another interesting point is that with a more recent Linux distribution power management greatly improves, leading to 24% lower power consumption in idle mode SMT-ON. This lower idle power consumption alone is able to diminish the standard measurement by 2% which is non negligible for a data center. Using the server internal power monitoring facilities, the two CPUs consume 27% more power running SLC5 than running SLC6.

Table 2. Total power consumption for "Sandy Bridge-EP" system using one PSU.

SMT status	*Idle*	*Load*	*Standard measurement*
		SLC5	
OFF	140 W	390 W	340 W
ON	149 W	425 W	370 W
		SLC6	
OFF	110 W	385 W	330 W
ON	113 W	414 W	354 W

5. Benchmark results

5.1. HEPSPEC06 - "Westmere-EX" system

5.1.1. Overall performance

In these tests the HEPSPEC06 benchmarks were compiled with GCC 4.1.2 in 64-bit mode, the standard compiler available with SLC5, and the performance measurements were carried out with SMT enabled.

The results with Turbo mode disabled are reported in table 3 for the "Nehalem-EX" system and table 4 for the "Westmere-EX" system, respectively. Figure 3 shows the comparison of the two systems, where the "Nehalem-EX" results were frequency scaled, from the initial 2.27GHz clock rate up to 2.4GHz, to match the "Westmere-EX" frequency. As previously stated in the hardware description, the "Westmere-EX" E7-4000 series processors are an evolution of the same micro architecture rather than a radical change from the previous "Nehalem-EX" 7500 series processors. This evolutionary progression is reflected on figure 3: the scalability trends of the HEPSPEC06 plots are very similar. Of course, the almost linear initial portion stops at 32 cores, when the older contender has to start using SMT, but it underlines that, at the same frequency, an E7-4870 core provides an averaged 6% performance increase over an X7560 core. If a direct comparison to the "Nehalem-EX"

based platform is considered (SMT off), the new system yields 39% more throughput, which becomes 31% frequency scaled: 6% from core performance improvements and 25% from core count increase.

Table 3. HEPSPEC06 measurements for "Nehalem-EX" system with Turbo mode disabled (not frequency scaled).

Number of Processes	*HEPSPEC06*
1	14.3
8	107
16	197
32	360
64	471

Table 4. HEPSPEC06 measurements for "Westmere-EX" system with Turbo mode disabled (not frequency scaled).

Number of Processes	*HEPSPEC06*
1	16.4
8	123
20	281
40	501
80	654

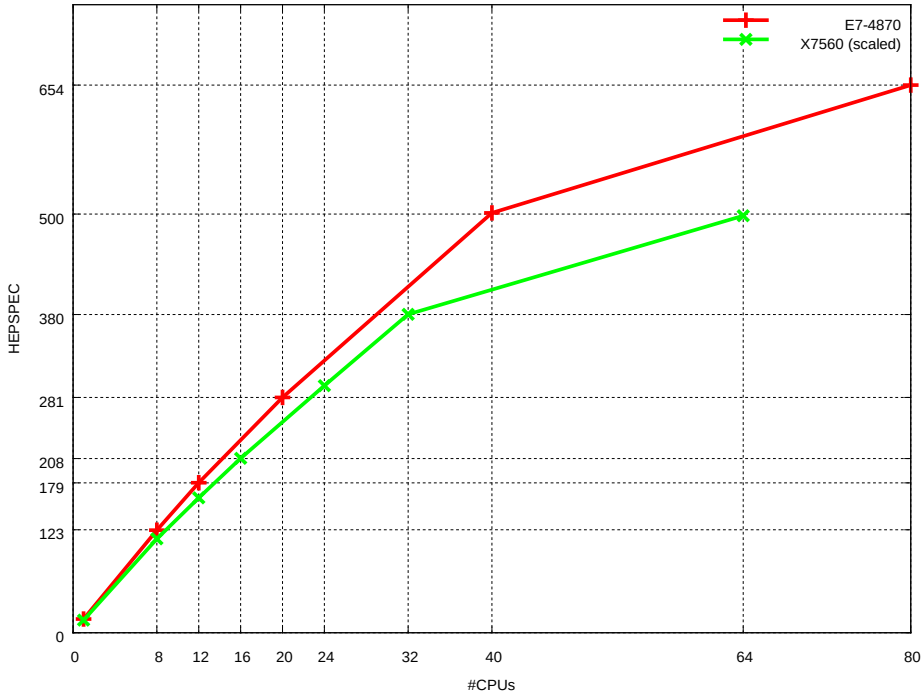

Figure 3. HEPSPEC06 performance comparison "Westmere-EX" E7-4870 vs. "Nehalem-EX" X7560 (frequency scaled). Both systems have SMT on and Turbo mode disabled.

The results with Turbo mode enabled are reported in table 5 for the "Nehalem-EX" system and table 6 for the "Westmere-EX" system, respectively. The frequency scaled comparison is shown in figure 4. The evolutionary progression is similar as in the Turbo disabled comparison. But here, surprisingly, from 1 to 16 cores, the two platforms offer approximately the same performance, and the "Westmere-EX" continues close to linear progression until 40 cores, when all its physical cores are fully occupied. For the "Nehalem-EX" processor, after 16 cores, the HEPSPEC06 measurements are increasing at a slower pace, showing that the Turbo effect is not able to sustain higher core occupancies. As previously stated the newer core offers about 6% more performance than an "Nehalem-EX" X7560 core, therefore from 1 to 16 cores the Turbo gain is higher for the "Nehalem-EX", allowing it to close the performance gap with its younger peer.

Table 5. HEPSPEC06 measurements for "Nehalem-EX" system with Turbo mode enabled (not frequency scaled).

Number of Processes	*HEPSPEC06*	*HS/proc*
1	16.2	16.2
8	120	15.0
16	227	14.2
32	372	11.6
64	478	7.46

Table 6. HEPSPEC06 measurements for "Westmere-EX" system with Turbo mode enabled (not frequency scaled).

Number of Processes	*HEPSPEC06*	*HS/proc*
1	17.9	17.9
8	130	16.3
24	345	14.4
40	521	13.0
80	654	8.2

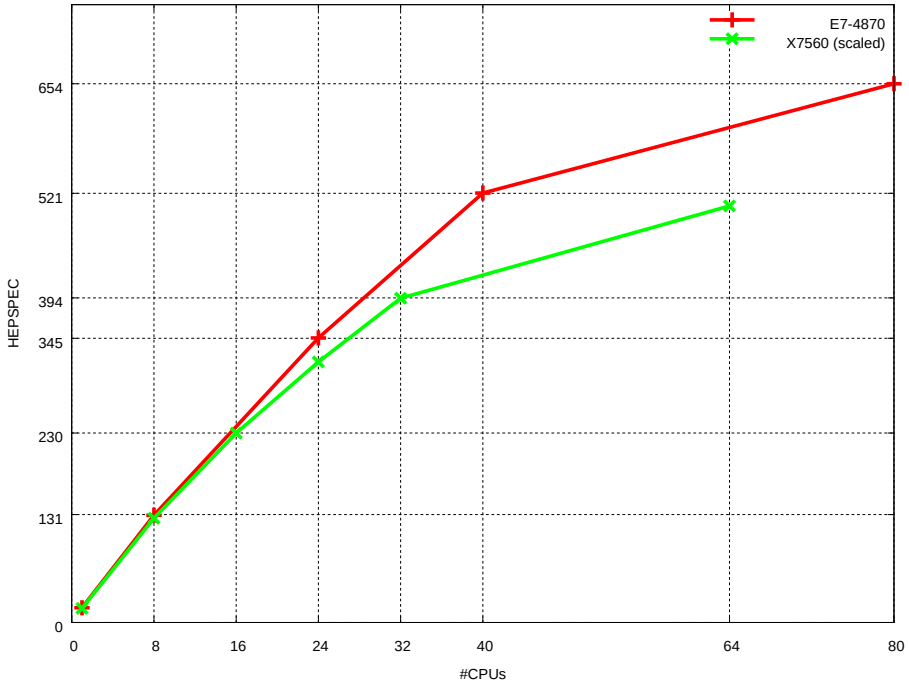

Figure 4. HEPSPEC06 performance comparison "Westmere-EX" E7-4870 vs. "Nehalem-EX" X7560 (frequency scaled). Both systems have SMT on and Turbo mode enabled.

5.1.2. Turbo mode advantage

The comparisons of the HEPSPEC06 performance with Turbo mode enabled and disabled are reported in table 7 for the "Nehalem-EX" system and table 8 for the "Westmere-EX" system, respectively. Using more than 4 cores per processor, the Turbo performance gain drops for the "Nehalem-EX" processor, while "Westmere-EX" shows a really linear decrease of the Turbo gain when inversely increasing the processor occupancies. The low values for the Turbo gain with low core occupancies indicate that our "Westmere-EX" family CPUs may suffer from a firmware issue (in early test phases, several BIOS releases are often required to reach nominal performance). When performing a direct measurement of the core frequency stressing one core on a single socket, the system reports 2513MHz while on the processor specifications page, Intel indicates that the E7-4870 is able to reach 2800MHz.

This Intel provided maximum frequency is more in line with our expectations. Indeed with 2.8GHz, the observed Turbo boost should be close to 17% for HEPSPEC06 runs on a single core.

If a direct comparison to the "Nehalem-EX" is considered, the new system allows for 37% more throughput with Turbo enabled. Frequency scaled the "Westmere-EX" yields 30% more throughput.

Table 7. HEPSPEC06 Turbo gains for "Nehalem-EX" system.

Number of Processes	*HEPSPEC06 Turbo gain*
1	+13%
8	+13%
16	+10%
32	+3%

Table 8. HEPSPEC06 Turbo gains for "Westmere-EX" system.

Number of Processes	*HEPSPEC06 Turbo gain*
1	+9%
8	+6%
24	+5%
40	+4%

5.1.3. SMT advantage

The gain produced by SMT can be deduced by comparing the HEPSPEC06 results for 40 and 80 processes for the "Westmere-EX", and for 32 and 64 processes for the "Nehalem-EX": in the case of both processors the SMT gain is the same at 26%. This additional gain shows the remarkable scalability potential of the four socket "Westmere-EX" system, increasing significantly its performance up to its maximum 80 SMT cores. Additionally, the consistent SMT boost provided by the "Westmere-EX" system shows that the common Boxboro-EX platform provides enough bandwidth for the CPU upgrade, with respect to HEPSPEC06 multiprocessor performance.

5.1.4. Platform efficiency

Given that the platform power consumption and HEPSPEC06 measurements are available, the power efficiency in terms of HEPSPEC06 per Watt can be deduced.

Efficiency can be evaluated for two cases:

1. SMT off: taking the standard power measurement without SMT and the HEPSPEC06 measurement for 40 cores
2. SMT on: taking the standard power consumption with SMT and the HEPSPEC06 measurement for 80 cores.

The efficiency comparison between the "Westmere-EX" and the "Nehalem-EX" based systems is straightforward, as the power consumption is the same for the two servers, the only difference is the HEPSPEC06 performance measurement. Therefore, the same improvement as in HEPSPEC06 performance is observed on the efficiency side, the transition to the E7 family CPU allowing a 31% efficiency boost over the previous generation (frequency scaled and Turbo mode disabled).

Another interesting point can be underlined when plotting the normalized platform efficiency histogram, using 2 GB of memory per core (figure 5). According to openlab previous evaluation of the "Westmere-EP" X5670 system [2], a quad socket "Westmere-EX" E7-4870 server with 3 PSUs offers similar efficiency as a dual socket "Westmere-EP" X5670 server with a single PSU. The quad socket server is 2.4% less efficient that the dual socket server SMT-off, but 3.1% more efficient when SMT is enabled.

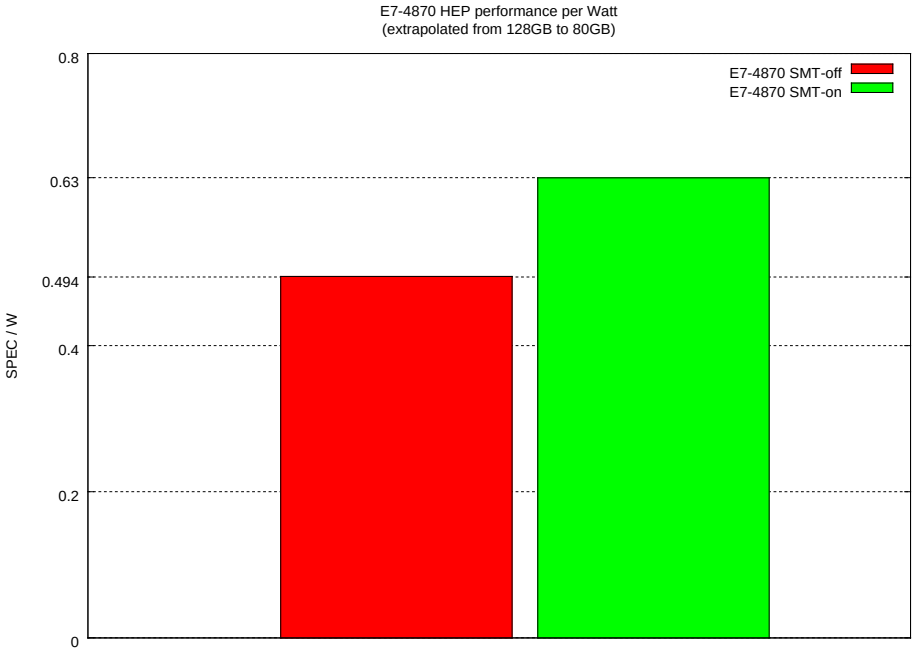

Figure 5. Efficiency of the "Westmere-EX" E7-4870 system with 3 PSUs.

5.2. HEPSPEC06 - "Sandy Bridge-EP" system

5.2.1. Overall performance

The comparison of the "Sandy Bridge-EP" and "Westmere-EP" HEPSPEC06 results is shown in figure 6, frequency-scaled to the "Sandy Bridge-EP" frequency. The graph clearly shows an advantage to the most recent server generation. In the first phase, when the two systems count enough physical cores to accommodate the load, the "Sandy Bridge-EP" core is offering at least 13% more HEPSPEC06 performance than the "Westmere-EP" core. But when the CPU occupancy approaches its maximum: 12 cores for the "Westmere-EP" system and 16 cores for the "Sandy Bridge-EP", this advantage reaches a peak at 20% additional HEPSPEC06 performance per core for the E5 core. This can be explained by the fact that the E5 scalability is better than the 5600 series CPU: according to our previous Nehalem/Westmere HEPSPEC06 comparison [2], an inflexion was observed in "Westmere-EP" HEPSPEC06 scalability between 8 and 12 cores, and this is not the case for the Sandy Bridge.

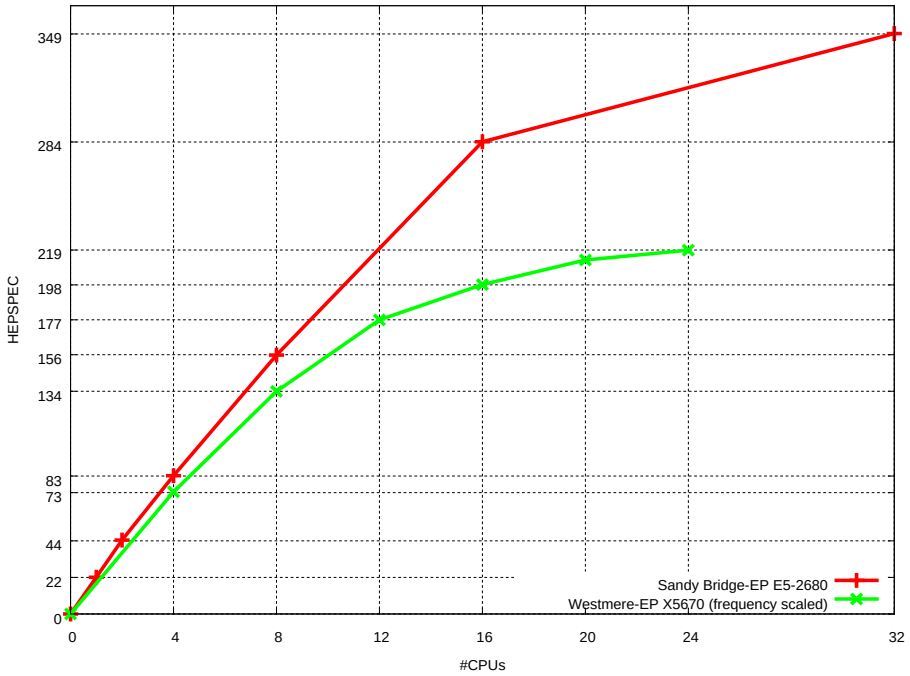

Figure 6. HEPSPEC06 performance comparison "Sandy Bridge-EP" E5-2680 vs. "Westmere-EP" X5670 (frequency scaled). Both systems have SMT on and Turbo mode disabled.

5.2.2. SMT advantage

The gain produced by SMT can be computed by comparing the HEPSPEC06 results for 16 and 32 cores for the "Sandy Bridge-EP", and for 12 and 24 cores for the "Westmere-EP". In the case of the "Westmere-EP", the gain is 24% and for the "Sandy Bridge-EP", the SMT gain is 23%. This shows that SMT is a well-established technology that steadily delivers around 23% of additional HEPSPEC06 performance on Intel Xeon CPUs.

5.2.3. Platform efficiency

We report the efficiency of the "Sandy Bridge-EP" system, as described in section 5.1.4. The evaluation is done when running on SLC5 (figure 7) and SLC6 (figure 8), respectively. For the former SMT offers an 11% boost in terms of efficiency with Turbo mode enabled. SMT additional power consumption is around 10% when Turbo mode is enabled, and this additional power reduces some of the SMT gain when looking at power efficiency. We highlight the fact that the "Sandy Bridge-EP" E5-2680 is able to cross the 1 HEPSPEC06/W line that can be considered particularly impressive. When running SLC6, the efficiency gets an additional 6% boost with SMT on, reaching 1.107 HEPSPEC06/W. Some of the performance is due to the compiler change: going from GCC 4.1.2 for SLC5 to GCC 4.4.6 for SLC6: this change provides 19% increase of HEPSPEC06 SMT-off and 8% SMT-on.

Comparing the results when using SLC5, we observe that the microarchitecture change from the X5600 series to the E5-2600 series allowed a massive improvement in terms of platform efficiency: going from 0.611 HEPSPEC06/W to 1.039 HEPSPEC06/W, which is a 70% increase for efficiency. This is, for instance, twice the efficiency improvement from the previous microarchitecture change that occurred between Harpertown and Nehalem.

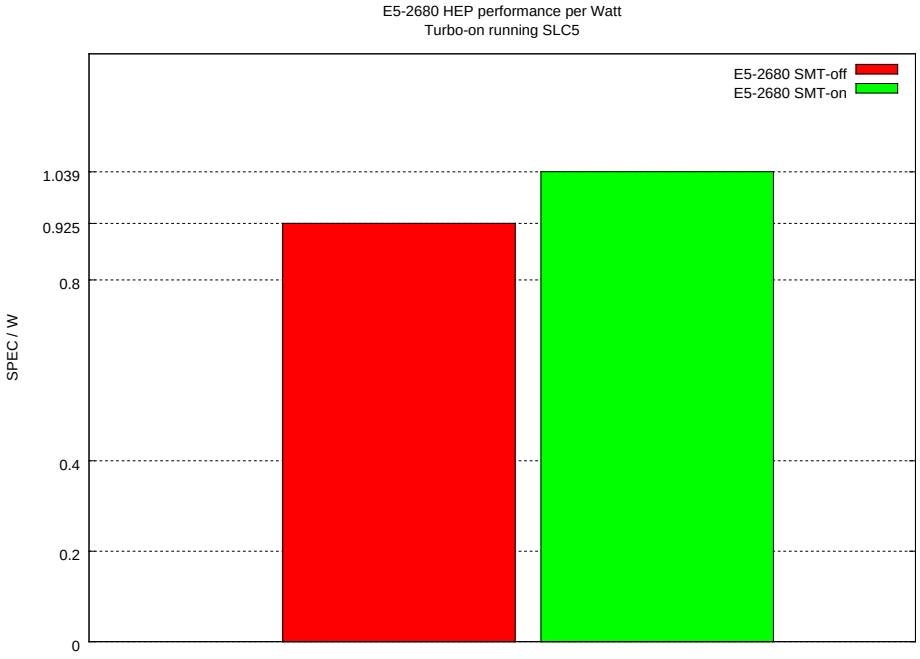

Figure 7. Efficiency of the "Sandy Bridge-EP" E5-2680 system with Turbo mode enabled, under SLC5.

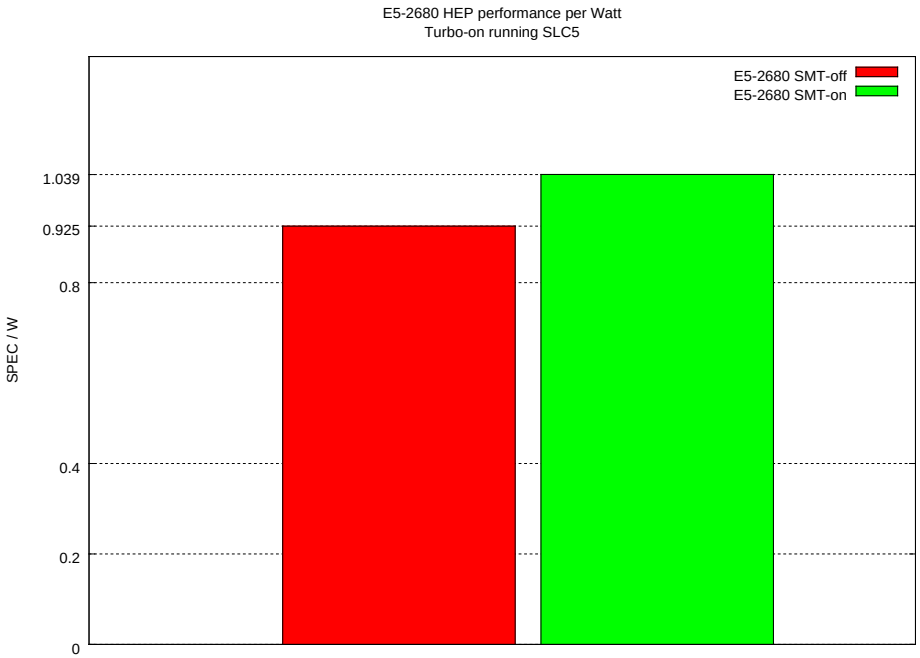

Figure 8. Efficiency of the "Sandy Bridge-EP" E5-2680 system with Turbo mode enabled, under SLC6.

5.3. Multi-threaded Geant4 prototype - “Westmere-EX” system

5.3.1. 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 π^- event in a predefined geometry. 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. In addition, the pinning system minimized the amount of sockets engaged. Turbo mode was off during all tests. 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 4.3.3 compiler.

Based on samples obtained from this and previous measurements, the measurement error for this benchmark is considered to be approximately ±0.5%.

5.3.2. Scalability testing

The application tested very well up to 40 physical cores. The efficiency under full physical core load was 100%, which corresponds to a perfect scaling factor of 40x. Detailed scalability data does not show efficiency penalties in scaling between 1 and 40 cores. Key intermediate scaling data points:

- 2x for 2 processes (101% efficiency)
- 8x for 8 processes (101% efficiency)
- 20x for 20 processes (101% efficiency)
- 32x for 32 processes (101% efficiency)

This data shows clearly that this benchmark exhibits excellent scalability on the tested system. One reason for the unusually high efficiency rating can possibly be the fact that some common data remains in the large caches of the CPU as different threads are trying to access it. Figure 9 demonstrates the gathered data while running on physical cores only, without the use of SMT. The simulation time is scaled on the left (blue) x-axis, while the efficiency is scaled on the right (green) x-axis.

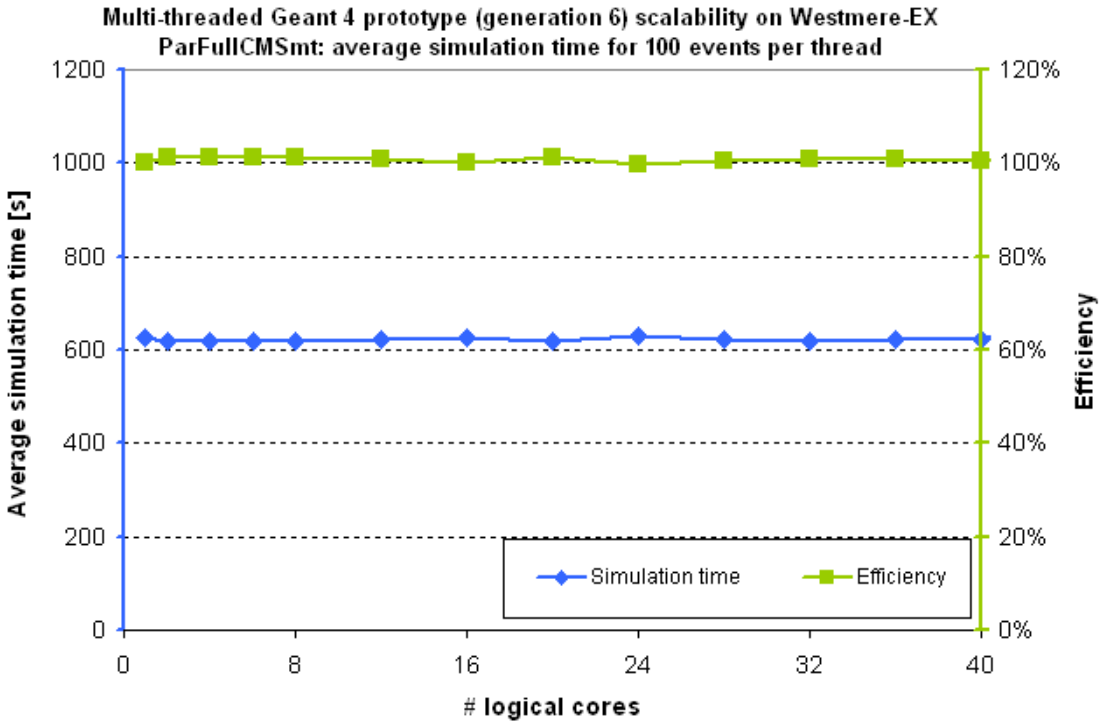

Figure 9. ParFullCMSmt scalability on "Westmere-EX" system (without SMT).

In contrast, the graph in figure 10 shows the data for points between 1 and 80 threads, i.e. with SMT. The efficiency curve grows past 40 cores to surpass 100%, since for thread counts higher than 40, expected scalability is fixed to 40x. Thus a final value of 123% indicates that the system loaded with 40 threads of the benchmark yields 23% more throughput than a perfectly scaled serial version on 40 physical cores.

5.3.3. SMT advantage

The advantage of running with SMT was:

- 4% with 56 hardware threads,
- 10% with 64 hardware threads,
- 17% with 72 hardware threads,
- and finally 23% with all hardware threads engaged.

One should note that this extra performance is traded in for a penalty in memory usage, as the number of software threads is twice that of the case of 40 cores. Nevertheless, this advantage is in line with the figures that CERN openlab has seen for many other processors since the first Nehalem family (about 25%) [2].

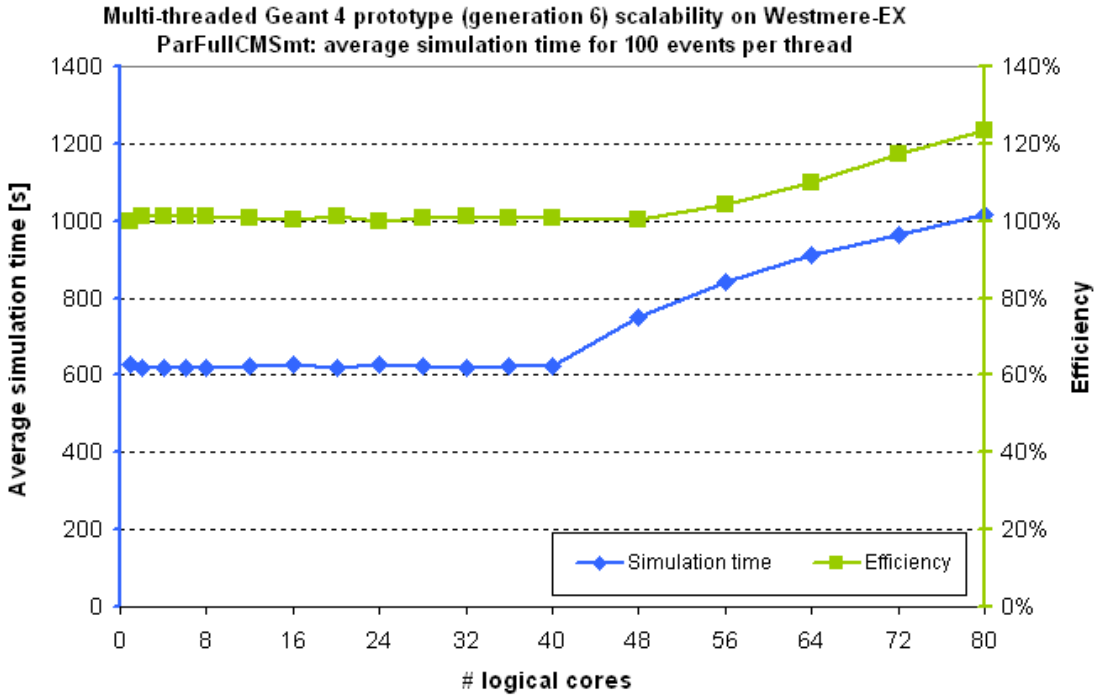

Figure 10. ParFullCMSmt scalability on "Westmere-EX" system (with SMT).

5.3.4. X7560 based "Nehalem-EX" comparison

For this benchmark, the frequency scaled figures for the single cores X7560 and E7-4870 are the same or within a very small percentage of each other. This indicates that no noticeable advances beneficial to this benchmark were made on the microarchitectural level. However, a more pragmatic comparison will confront the two entire systems: the new "top of the line" 4-socket system "Westmere-EX" (4 x E7-4870) and the previous corresponding 4-socket system "Nehalem-EX" (4 x X7560), respectively. In this comparison, the new "Westmere-EX" system wins significantly.

Let us first consider the additional theoretical performance that is to be obtained when replacing the old system with the new one. There are 25% more cores on each "Westmere-EX" chip, so that should give an ideal scaling factor of 1.25x. In addition, the new "EX" parts are clocked at a higher frequency than their predecessors: 2.4 GHz instead of 2.27 GHz, within the same thermal envelope. That introduces an ideal scaling factor of 1.057x. Multiplying the two we obtain 32% of additional expected throughput, which is matched perfectly by the measured 32% increase in the throughput of the whole platform. These figures certify that the "Westmere-EX" E7-4870 delivers expected scalability both on the frequency and on the core count fronts.

5.4. Multi-threaded Geant4 prototype - "Sandy Bridge-EP" system

5.4.1. Scalability testing

The technical test setup was identical as in the case of the "Westmere-EX" processor (section 5.3.1). The tests showed stable efficiency figures up to the full physical core count. When running with 16 software threads on 16 cores, the scaling factor was 15.8x, which means that the setup was 98% efficient. The figure matches good results observed on previous Intel processors, especially from the "Westmere" family. Detailed scalability data for intermediate points reveals good scaling, as expected, all the way between 1 and 16 processes:

- 2x for 2 processes (100% efficiency)

- 4x for 4 processes (100% efficiency)
- 8x for 8 processes (100% efficiency)
- 12x for 12 processes (100% efficiency)

There was no noticeable efficiency drop.

The graph in figure 11 shows the total simulation time curve in blue and the efficiency (scalability) curve in green, with the use of SMT. The simulation time is scaled on the left (blue) x-axis, while the efficiency is scaled on the right (green) x-axis. Towards its end, the efficiency curve surpasses 100%, since for thread counts higher than 16 expected scalability is fixed to 16x. Thus a final value of 125% indicates that the system loaded with 32 threads of the benchmark yields 25% more throughput than a perfectly scaled serial version on 16 physical cores.

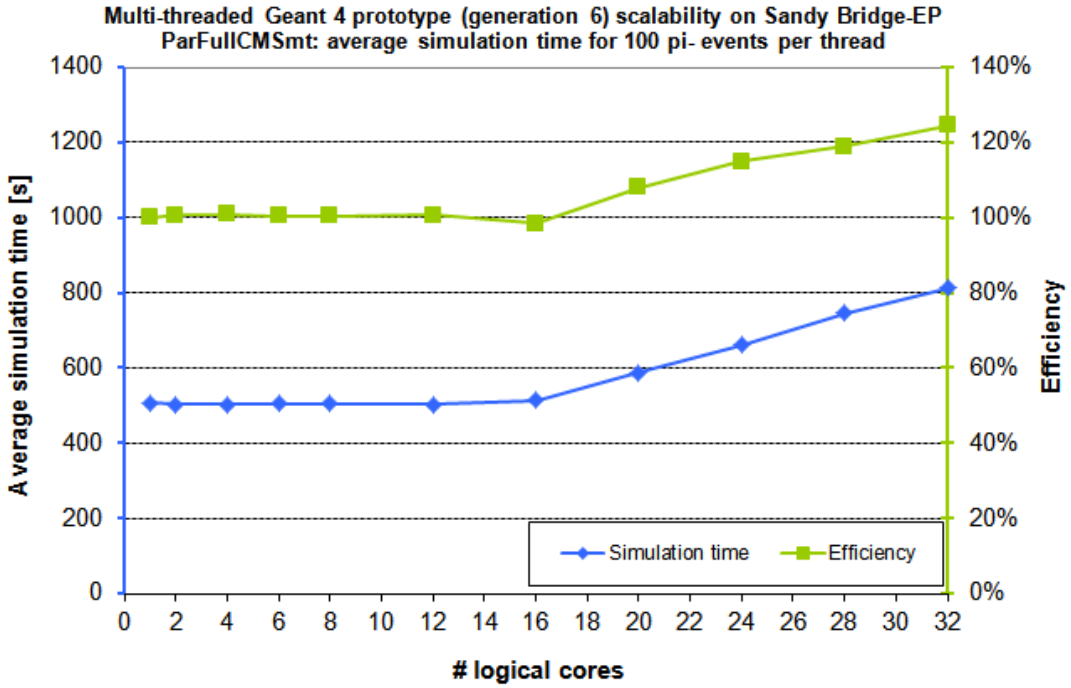

Figure 11. ParFullCMSmt scalability on "Sandy Bridge-EP" system (with SMT).

5.4.2. SMT advantage

The advantage of running with SMT was already 8% for 20 hardware threads, suggesting that the run with 16 threads (full core count) might be slightly suboptimal. The SMT gains later were 15% with 24 hardware threads and finally 25% with all hardware threads engaged. 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 16 cores.

This advantage is comparable to previously tested dual-socket systems, even if slightly higher [2]. It could testify to the slightly increased efficiency of SMT, but also to a potential minor sub optimality of the benchmark runs with all physical cores loaded (16 threads).

5.4.3. L5640 based "Westmere-EP" comparison

When compared to a Xeon L5640 based "Westmere-EP" platform, the probed "Sandy Bridge-EP" system performs better in terms of frequency scaled performance, and also delivers a substantial 33% increase in core count. The overall advantage of the "Sandy Bridge-EP" solution over the comparable

"Westmere-EP" platform was established to be 46% at full load, which is when using 32 threads for the "Sandy Bridge-EP" and 24 threads for the "Westmere-EP".

An 8-9% average frequency-scaled advantage was noticed when running with up to 12 jobs to match the physical core count of the "Westmere-EP" system. This is attributed to the on-die innovations in the Sandy Bridge microarchitecture.

The product of the core count increase (1.333) and the average microarchitectural advantage (1.09) yields a 45% expected improvement over the previous platform at full load, and the measured 46% improvement is within 0.5% of that goal (1.46 / 1.45 = 1.005). This means that the tested "Sandy Bridge-EP" solution continues the tradition of highly efficient dual socket systems based on the Intel microarchitecture, delivering scalable and predictable performance.

5.5. Maximum likelihood fit prototype - "Westmere-EX" system

5.5.1. Technical test setup

The threads were pinned to the cores running them. The systems 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 by requiring 2 threads per core. In addition, the pinning system maximized the amount of sockets engaged. RooFit prototype v4.0 and MINUIT from ROOT v5.28 were used in the tests. Code was compiled with the Intel C++ compiler version 12.0.2, supplying the following options:

```
-O2 -m64 -fPIC -funroll-loops -finline-functions -msse3 -ip -openmp.
```

Auto-vectorization (SSE) was enabled during all tests.

Timings were obtained from the average over three consecutive runs of the application (errors are <0.5%). Running the sequential application on the "Westmere-EX" with Turbo mode off took about 311 seconds.

5.5.2. Turbo mode advantage

The results of the comparison of the execution times obtained when running with Turbo mode off and Turbo mode on are shown in figure 12. It is possible to see that on the "Westmere-EX" Turbo mode had less impact on the performance (around 3-6% for 1 to 40 threads) with respect to the "Nehelem-EX" results (18% for 1 thread, with a constant decrease with the number of threads, up to 3% with 32 threads). A similar behavior is observed also with HEPSPEC06 benchmarks (see section 5.1.2). In particular, Turbo mode had a constant effect on the "Westmere-EX" when varying the number of threads, while on the "Nehalen-EX" the speed-up went rapidly down (as expected). There was no effect from Turbo mode when using SMT threads for both systems.

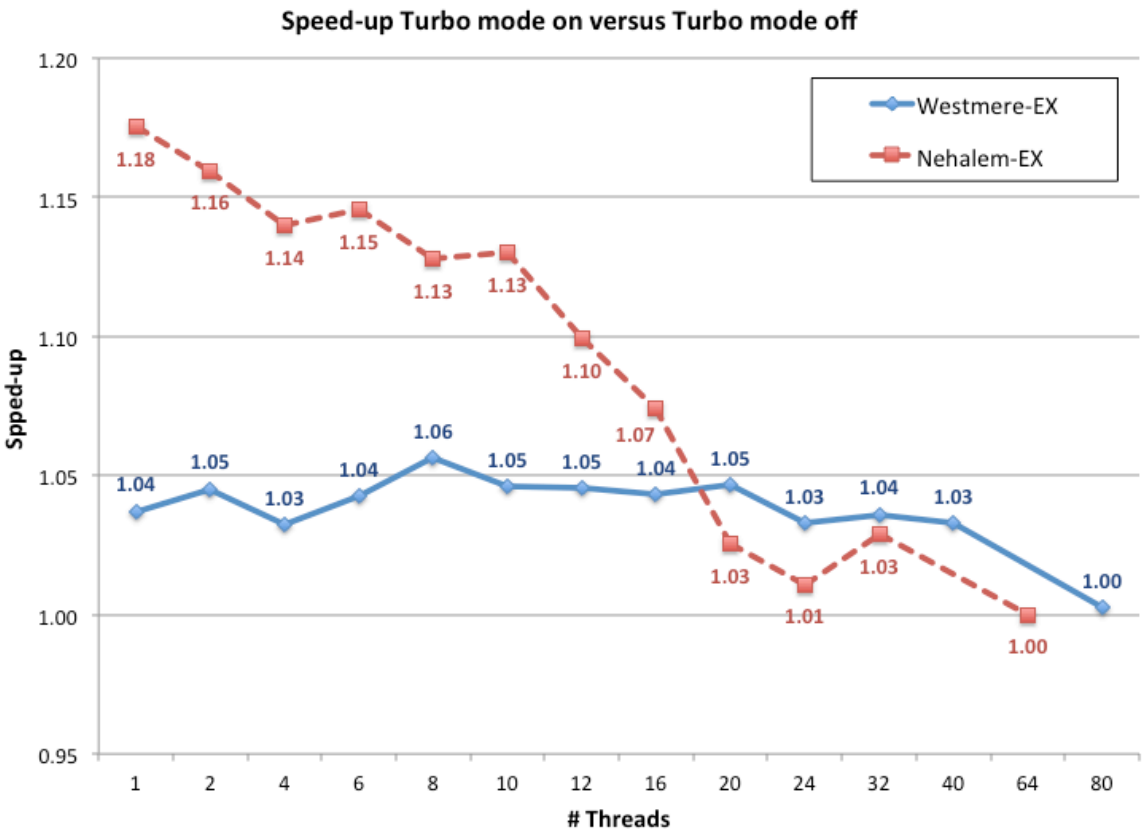

Figure 12. Speed-up ratios obtained from the comparison of the execution times with Turbo mode off and Turbo mode on.

5.5.3. Scalability testing

Scalability tests were executed with Turbo mode off. We should underline that because Turbo mode 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 ran the tests with Turbo mode off. We show the scalability results in figure 13. We can clearly see that the scalability is very close to the theoretical expectation. With 32 threads, “Westmere-EX” system reached a speed-up of 28.9x, which became 35.0x for 40 and 43.3x for 80 threads, respectively. In the meantime, “Nehalem-EX” system had 28.5x for 32 threads and 37.3x for 64 threads. At this point it should be reiterated that one of the main limitations to the scalability of the application is accessing data in memory. The fact that the “Westmere-EX” system has more L3 cache (30 MB) with respect to the “Nehalem-EX” system (24 MB) improved the performance when using the same number of threads on both systems, i.e. up to 32 threads.

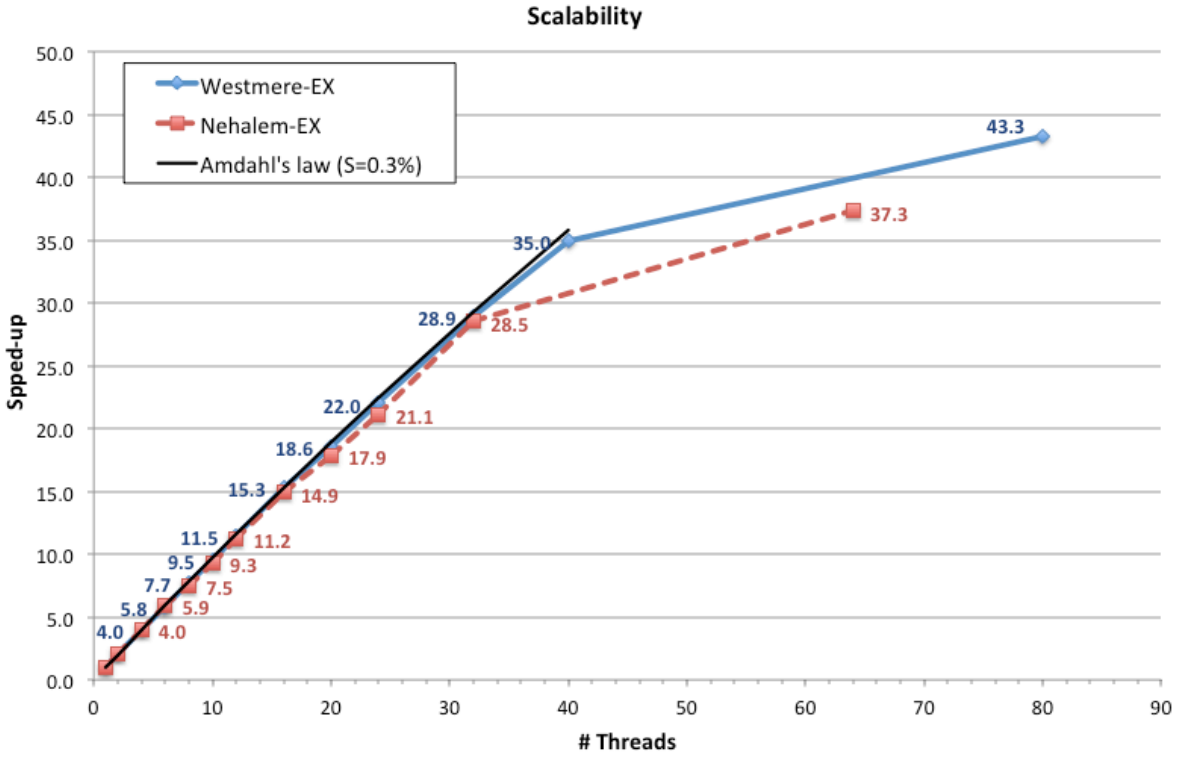

Figure 13. Scalability results. Data labels on the left (right) of the markers are for the "Westmere-EX" ("Nehalem-EX") system.

5.5.4. SMT advantage

SMT benefit can be extracted from the scalability results: +24% for the "Westmere-EX" system (80 versus 40 threads) and +31% for the "Nehalem-EX" system (64 versus 32 threads). We should consider that running more threads to fully load the "Westmere-EX" system introduces more OpenMP overheads then loading the Nehalem-EX system with fewer threads.

5.5.5. X7560 based "Nehalem-EX" comparison

Table 9 reports the comparison between the "Westmere-EX" E7-4870 and the "Nehalem-EX" X7560 performance results when using the same number of threads on both systems.

Table 9. Comparison of the "Nehalem-EX" and the "Westmere-EX" execution times. A positive (negative) number refers to faster (slower) "Westmere-EX" execution.

Number of threads	*Turbo mode OFF*	*Turbo mode ON*
1	+10%	−3%
2	+6%	−5%
4	+11%	+1%
8	+14%	+7%
12	+13%	+8%
16	+13%	+10%
32	+12%	+13%

For a single thread execution the "Westmere-EX" system was 10% faster with Turbo mode off (4% frequency scaled), but it was 3% slower with Turbo mode on. This means that a single core on the

"Westmere-EX" system gave better performance with Turbo mode off, while it was the opposite for Turbo mode on. With higher number of threads and Turbo mode on the "Westmere-EX" system became faster than "Nehalem-EX" system because of the decrement in the effect of Turbo mode.

Finally, table 10 reports a comparison of the performance of the two systems when they were fully loaded. The results do not depend on Turbo mode. Two main conclusions can be drawn from these numbers:

- Without using SMT, the "Westmere-EX" system had an excellent performance with respect to "Nehalem-EX" system. Assuming +25% due to more available cores and +5.7% from the higher frequency, the rest can be reasonably attributed to the bigger L3 cache size.
- With SMT the consideration of the previous point is still valid, but then we are limited by the OpenMP overheads due to running 16 more threads. This is a clear limitation of the application for the specific case under consideration when running on such a large number of threads.

Table 10. Comparison of the "Nehalem-EX" and the "Westmere-EX" execution times. The positive numbers refers to faster "Westmere-EX" execution. Same results are obtained for Turbo mode on and off.

Number of threads "Nehalem-EX" vs. "Westmere-EX"	
32 vs. 40	+36%
64 vs. 80	+28%

5.6. Maximum likelihood fit prototype - "Sandy Bridge-EP" system

5.6.1. Technical test setup

The technical test setup was similar to the case of the "Westmere-EX" tests (section 5.5.1). In the current set of tests the code was compiled with the Intel C++ compiler version 12.1.0. Three different flags for the auto-vectorization were considered during the compilation: `-no-vec`, `-msse3`, `-mavx`.

The reference "Westmere-EP" system was a dual-socket system with Intel Xeon CPU X5650 at 2.67GHz (12 physical cores in total, 12MB L3 cache per CPU). The sequential execution time of the application was 417 seconds when running on the "Sandy Bridge-EP" system without vectorization and Turbo mode off (the slowest execution).

5.6.2. Vectorization advantage

The average speed-up results of the application compiled with the different vectorization flags are shown in table 11. The results were obtained from the ratio between execution times for the configurations in the columns and the corresponding configurations in the rows. We found that the speed-up results did not significantly depend on the number of threads and the Turbo mode (standard deviation is 0.03x). The "Westmere-EP" system had the same speed-up of the "Sandy Bridge-EP" system when using the `-msse3` configuration.

Table 11. Average speed-up results for different vectorization configurations (given by the indicated compiler flags) on the "Sandy Bridge-EP" system.

	`–no-vec`	`–msse3`
`–msse3`	1.78x	
`–mavx`	1.99x	1.12x

A detailed analysis of speed-up result value between the `-msse3` and `-mavx` configurations (1.12x) showed that the vectorized loops that contain exponential function operations were faster executed in the `-mavx` configuration. For these loops the Intel compiler used the functions `_svml_exp2.R` for SSE and `_svml_exp4.R` for AVX configurations, respectively. Although the total time exclusively spent in the executions of the AVX function was 1.61x faster than the SSE case, the divide and square root operations and the and loads/stores of data, which were executed inside the same loops with the exponential function, limited the speed-up of the loops to values between 1.14x and 1.17x (vectorized AVX versions of divide and square root operations take the same time of the corresponding SSE versions). Surprisingly, also the execution of loops with evaluation of pure polynomials inside did not go faster with AVX, as it would be expected, because of the loads/stores. All these effects explain the smaller value of the speed-up for `-msse3` versus `-mavx` with respect to `-no-vec` versus `-msse3` configurations.

5.6.3. Turbo mode advantage

Speed-up results obtained from the comparison of the execution times with Turbo mode off and on are shown in figure 14.

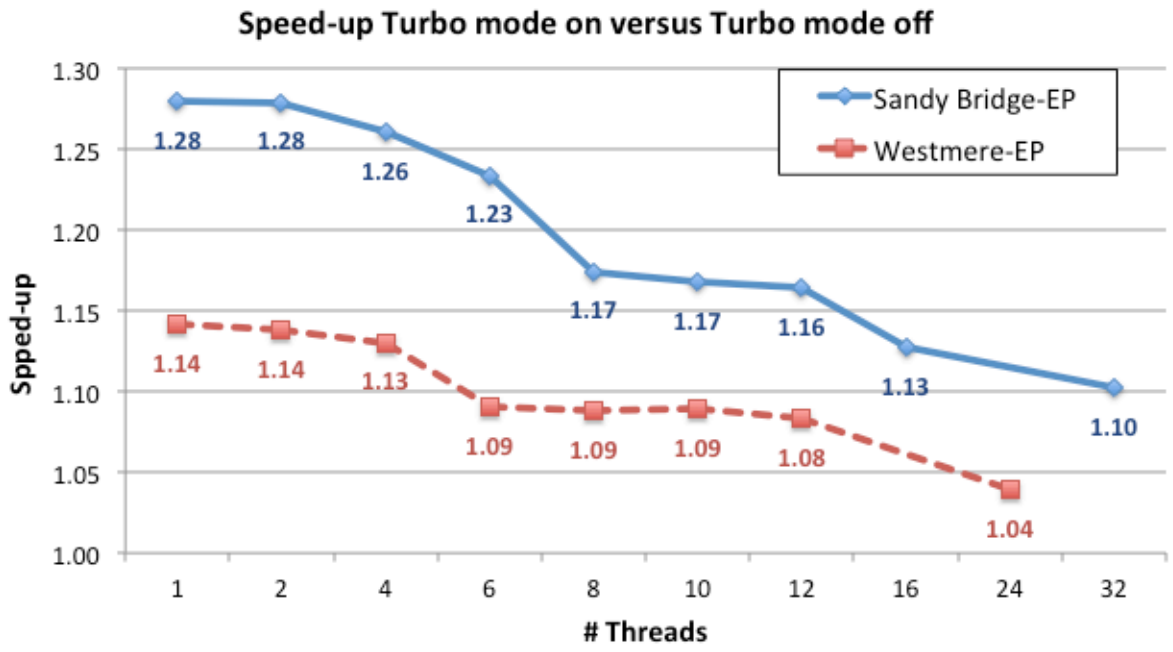

Figure 14. Speed-up ratios obtained from the comparison of the execution times with Turbo mode off and Turbo mode on.

We found that the effect of the Turbo mode did not depend on the vectorization mode. With two threads (one thread per CPU) the speed-up of the Turbo mode was compatible with the increase in frequency from the nominal value of 2.70GHz to 3.46GHz. For the "Westmere-EP" system the effect on Turbo mode was significant less, increasing from the nominal 2.67GHz to 3.04GHz. The benefit of the Turbo mode decreased when more parallel threads were executed. With 16 threads on the "Sandy Bridge-EP" system the speed-up drop to 1.13x, with a variation of 0.15x with respect to the single thread case. For the corresponding case (12 threads) for the "Westmere-EP" the variation was 0.06x. When also SMT was used (fully loaded system), Turbo mode gives still a contribution for the "Sandy Bridge-EP" system (1.10x), which was higher than the corresponding case of the "Westmere-EP"

system (1.04x). To conclude, the Turbo mode behaves much better on the "Sandy Bridge-EP" system with respect to the "Westmere-EP", reaching higher speed-up for any number of loaded threads.

5.6.4. Scalability testing

We performed scalability tests using the AVX vectorization on the "Sandy Bridge-EP" system and SSE vectorization on the "Westmere-EP" system. Turbo mode was off on both systems for the same reason explained in section 5.5.3. 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. We show the scalability results in figure 15. We can clearly see that the scalability is very close to the theoretical expectation. The application scales slightly worse on the "Westmere-EP" system. Analysis of the difference showed that this was a consequence of the smaller L3 cache size.

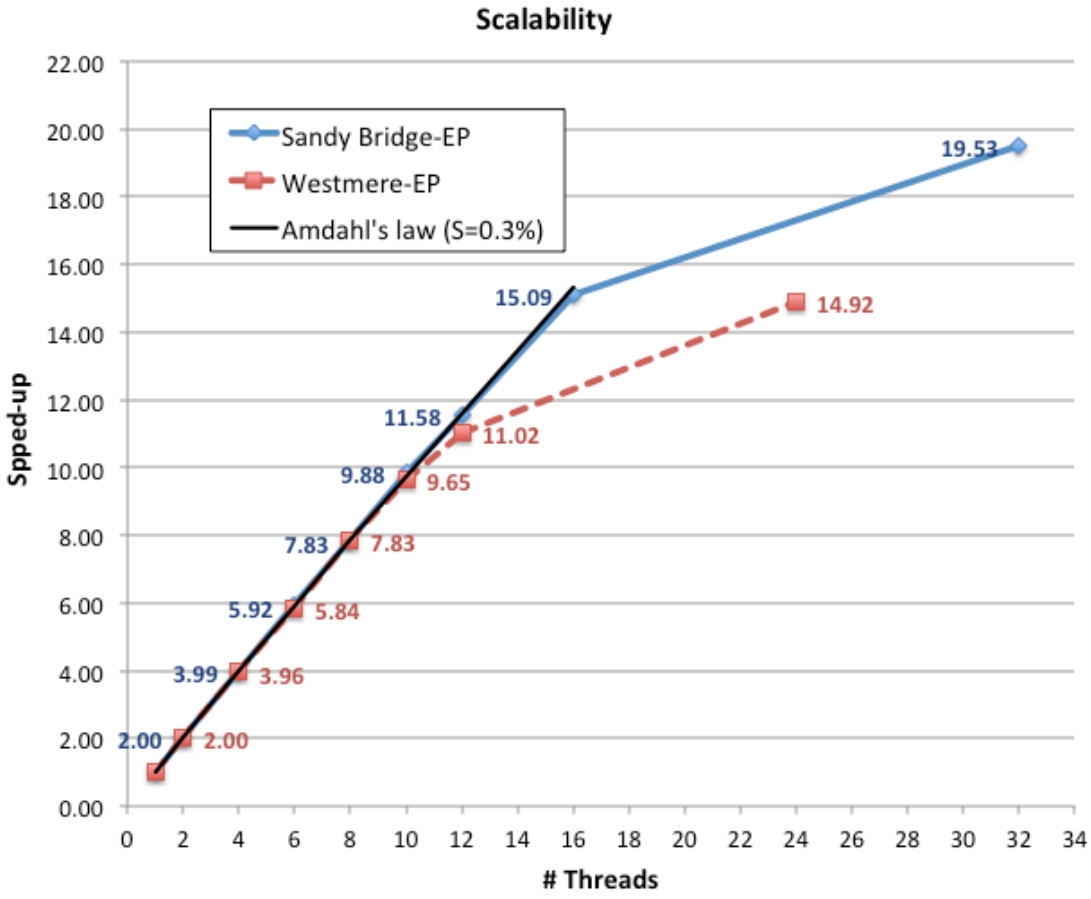

Figure 15. Scalability results. Data labels on the left (right) of the markers are for the "Sandy Bridge-EP" ("Westmere-EP") system.

5.6.5. SMT advantage

The contribution given by using SMT was obtained from the ratio of the execution times of the application when running with maximum number of threads without and with SMT, i.e. 16 and 32 threads. The results do not depend on the vectorization, while Turbo mode gives a small impact: 1.29x and 1.26x for Turbo mode off and on, respectively. Corresponding values for "Westmere-EP" are

slightly higher: 1.35x and 1.30x. This can be attributed to the higher numbers of threads involved in the "Sandy Bridge-EP" case.

5.6.6. X5650 based "Westmere-EP" comparison

We compare the execution time of the application compiled with AVX vectorization on the "Sandy Bridge-EP" system with respect to the execution time when running with SSE vectorization on the "Westmere-EP". Both systems had Turbo mode on. Comparing the corresponding runs executed with the same number of threads on both systems, i.e. between 1 and 12 threads, we obtain that the "Sandy Bridge-EP" was in average 1.49x faster (standard deviation is 0.03x). Removing the speed-up due to the AVX vectorization and Turbo mode on, we obtain that the "Sandy Bridge-EP" single core performed 1.19x faster than the "Westemere-EP" one. The same factor can be simply obtained from the comparison of the execution time of the runs with SSE vectorization and Turbo mode off on both systems. Analyzing this comparison we derive that the speed-up was mainly due to a faster execution of exponential functions (1.16x) and square root operations (1.54x). A small benefit came also from the larger L3 cache available on the "Sandy Bridge-EP" system. Then we compare the fully loaded "Westmere-EP" and "Sandy-Bridge" systems with and without SMT, i.e. 12 versus 16 and 24 versus 32 threads, respectively. We obtain that the overall gain in performance between the two systems was 1.91x, without SMT, and 1.86x, with SMT. Note that the small decrement in the comparison of the overall performance was due to the higher number of threads running on the "Sandy Bridge-EP" system that limited the scalability. Also in this case we can compute the speed-up eliminating the effects of AVX vectorization and Turbo mode, which was about 1.60x. We obtain that this number is in agreement with the equivalent single core speed-up value (1.19x) scaled by the different core counts of the systems (1.33x).

6. Conclusions

In this paper we have reported on a set of benchmark results when comparing two groups of systems:

- quad-socket: 40-core Intel Xeon "Westmere-EX" server (E7-4870 at 2.4GHz) with the previous generation 32-core Intel Xeon "Nehalem-EX" server (Xeon X7560 at 2.27GHz). Both systems are "top of the line".
- dual-socket: 16-core Intel Xeon "Sandy Bridge-EP" server (E5-2680 at 2.70GHz) compared with representatives of Intel's previous generation "Westmere-EP" servers.

The "Westmere-EX" platform provided a 39% advantage over the "Nehalem-EX" one for the HEPSPEC06 benchmark. In this comparison the power consumption remained constant, yielding an appreciable 39% of throughput per Watt improvement. Other benchmarks had similar scores, between 28% and 36%, depending on the configuration of SMT and Turbo mode. The benefits of SMT remained constant at around 24%. In addition, Turbo mode efficiency was explored and compared in depth for the first time. While before Turbo mode was found to provide large boosts for low active core counts and no boosts for high active core counts, the situation with "Westmere-EX" was the opposite. Turbo mode seemed to provide small but consistent improvements across all active core counts.

In the comparison of the "Sandy Bridge-EP" system with respect to the "Westmere-EP" system we obtained a performance improvement of 9-20% per core and 46-60% improvement across all cores available (all results frequency scaled). We also found that Turbo mode has been improved. Vectorized applications get an additional performance boost given by the new AVX instructions. We were particularly impressed by the performance/Watt measurements where we obtained a 70% improvement. Intel has improved the thermal characteristics of the Sandy Bridge processor 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 processors, namely the Ivy Bridge processor planned for 2013 and the Haswell for 2014.

References

[1] CERN openlab http://openlab.web.cern.ch/
[2] Jarp S, Lazzaro A, Leduc J and Nowak A 2011 *J. Phys.: Conf. Series* **331** 052009
[3] Standard Performance Evaluation Corporation http://www.spec.org/
[4] Merino G *et al.* 2009 *Transition to a new CPU benchmarking unit for the WLCG*
[5] Agostinelli S *et al.* 2004 *Nucl. Instrum. Methods Phys. Res.*, Sect. A **506** 250
[6] Apostolakis J, Cooperman G and Dong X 2010 *Multithreaded Geant4: Semi-Automatic Transformation into Scalable Thread-Parallel Software* Europar 2010
[7] Cowan G 1998 *Statistical Data Analysis* (Oxford: Clarendon Press)
[8] Verkerke W and Kirkby D 2006 *Proc. PHYSTAT05* (London: Imperial College Press) The RooFit Toolkit for Data Modeling
[9] Jarp S, Lazzaro A, Leduc J, Nowak A and Sneen-Lindal Y 2011 *Proc. Int. Conf. Parallel Computing* (Ghent, Belgium) Parallel Likelihood Function Evaluation on Heterogeneous Many-core Systems (*Preprint* `CERN-IT-2011-012`)
[10] Aubert B *et al.* (Babar Collaboration) 2009 *Phys. Rev. D* **80** 112002
[11] James F 1972 *MINUIT - Function Minimization and Error Analysis* CERN Program Library Long Writeup D506