Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 452–460

Procedia Computer Science

International Conference on Computational Science, ICCS 2011

Early performance evaluation of AVX for HPC

Pawel Gepner*, Victor Gamayunov, David L. Fraser

Intel Corporation,Pipers Way,Swindon Wiltshire SN3 1RJ, United Kingdom

Abstract

In this paper we take a look at what the new Intel instruction extensions - Intel Advance Vector Extensions (AVX) brings to high performance computing. We compare two configurations of Intel CPU based systems (one enabled with AVX and a second without AVX enabled) and present a performance evolution of these two platforms. We compare the same generation of CPU utilizing a single socket platform across a number of HPC benchmarks and macrocodes focused on different performance criteria, compare the results and discuss the implications for HPC.

Keywords: AVX; performance; HPC; Intel

1. Introduction

The 2nd generation Intel® Core™ processor family (codename: Sandy Bridge) not only symbolizes a change in processor generation but also brings some new important extensions and mechanisms which significantly improve overall performance and floating point performance characteristics. The 2nd generation Intel Core processor family is based on new micro-architecture principles and is significantly different from its predecessor. Sandy Bridge is the first Intel product with a graphics processor unit (GPU) and a central processor unit CPU integrated in a monolithic chip. The new generation Intel Core processor family is also the first implementation of AVX. The 2nd generation Intel Core processor family has been manufactured on 32nm Hi-k metal gate silicon technology with all the benefits this process brings to semiconductor products. The process technology uses a combination of Hi-k gate dielectrics and conductive materials with 32 nm lithography process to improve transistor size and properties such as reducing electrical leakage, chip size, power consumption, and manufacturing costs [1]. The first members of the 2nd generation Intel Core processor family have been designated for mobile and desktop products and the server version of Sandy Bridge will be introduced afterwards. Power consumption as well performance per watt characteristic was one of the major design criteria for this product however some of the existing technologies such as turbo mode have been improved significantly. Although the first members of the 2nd generation Intel Core processor family are focused on the desktop and mobile markets some of the technology implemented is also expected to be very applicable for the HPC segment. This study compares 2 configurations of the 2nd generation Intel Core processor family: one system with 2nd generation Intel Core I7-2600 processor with AVX enabled and the same system but with AVX disabled. We run typical HPC benchmarks, macrocodes and evaluate the performance implication of the

* Corresponding author. Tel.: +48602414128; fax: +48225708122.
E-mail address: pawel.gepner@intel.com.

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.047

AVX extension for this type of code. We will also discuss some architecture aspects of the new micro-architecture and validate their performance connotation.

2. Platform Architecture

In our study we compare two quad core based systems: first based on 2nd generation Intel Core I7-2600 processor with AVX and second with 2nd generation Intel Core I7-2600 processor with AVX disabled. Both platforms are identically configured from hardware and software perspective the only difference being that for all tests we have compiled two versions of binaries, one with AVX support, and another one without AVX. The second set of binaries has been generated with instruction set limited to SSE 4.2 using compiler switches, and linked against non-AVX version of MKL library.

2.1. Intel Advanced Vector Extensions Overview

Intel Advanced Vector Extensions expands the capabilities and programming environment that was introduced with Streaming SIMD (single-instruction, multiple-data) Extensions in 1999. Intel AVX is focused on the vector floating-point performance in scientific and engineering numerical applications, visual processing, cryptography and other application areas [2]. The primary focus of Intel AVX is to make easy, efficient implementation of applications with changeable degrees of thread parallelism, and data vector lengths. Intel AVX offers a significant increase in floating-point performance over previous generations of 128-bit SIMD instruction set extensions. Intel AVX increases the width of the 16 XMM registers from 128 bits to 256 bits. Each register can hold eight single-precision (SP) floating-point values or four double-precision (DP) floating-point values that can be operated on in parallel using SIMD instructions. Intel AVX adds three-operand syntax format (c=a+b), whereas previous instructions could only use two-operand syntax format (a=a+b). This new three-operand syntax improves also instruction programming flexibility and efficient encoding of new instruction extensions. Intel AVX also establishes a foundation for future evolution in both instruction set functionality and vector lengths by introducing an efficient instruction encoding scheme, three and four operand instruction syntax, FMA (fused multiply-add) extension and supporting load and store masking.

2.2. Processor characteristic

Apart from the AVX implementation, the major processor core enhancements to the 2nd generation Intel Core processor family can be divided by three categories: improvements made on the decoding phase, enhancements to data cache structure and increased instruction level parallelism (ILP). Sandy Bridge single core has 32KB 8-way associative L1 instruction cache and during the decoding phase it is able to decode four instructions per cycle, converting four X86 instructions to micro-ops. To reduce decoding time the new L0 instruction cache for micro-ops has been added to keep 1.5K micro-ops previously decoded. All micro-ops which are already in the L0 cache do not need to be fetched, decoded, and converted to micro-ops again. Consequently, the new L0 cache reduces the power used for these complex tasks by about 80% and improves performance by decreasing decoding latency. A new feature which improves the data cache performance is the extra load/store port. By adding a second load/store port Sandy Bridge can handle two loads plus one store per cycle automatically doubling the load bandwidth. This change helps also support the AVX instructions, enabling the cache to perform one 256-bit AVX load per cycle and generally increase core internal memory bandwidth from 32 bytes per cycle (16 bytes load 16 bytes store per cycle in previous generation Core based processors) to 48 bytes per cycle (two loads plus one store per cycle). One key architecture goal was to increase the performance by finding more instruction level parallelism. To improve the Out Of Order execution segment and tune the ILP performance required increasing the depth and width of execution units, larger buffers, more data storage, more data movement, and consequently also more power and more transistors. Also implementing AVX doubles the operand size and generates a need for bigger buffers. One of the methods to achieve the bigger dataflow window is implementing Physical Register File (PRF) instead of centralized Retirement Register File. This method implements a rename register file that is much larger than the logical register file, decreases data duplication and data transfers and finally eliminates movement of data after calculation. This implementation also increases the size of scheduler and number of reorder buffer (ROB) entries by 50% and 30%,

respectively and as a result increases the size of the dataflow window globally. Figure 1 shows Sandy Bridge's single core micro-architecture block diagram [3].

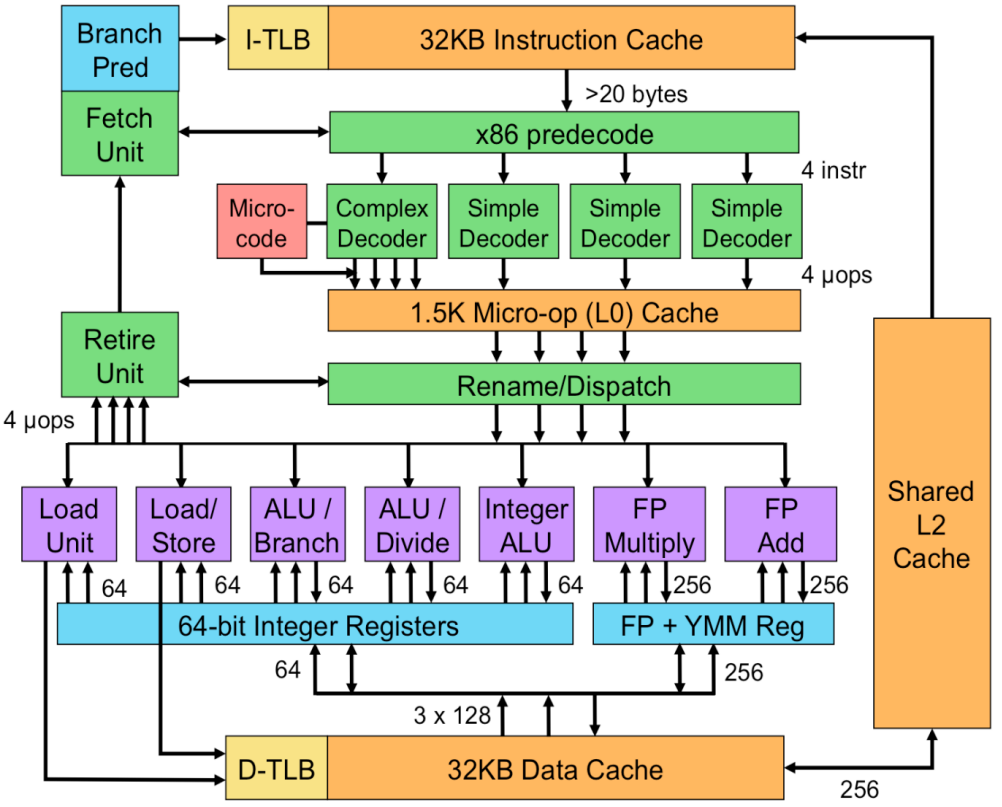

Fig.1. Sandy Bridge block diagram

All the modifications made to the Sandy Bridge cores are corresponding with changes made in the uncore part of the new CPU. Ucore part integrates last level cache (LLC), the system agent, the memory controller, PCI Express interface, embedded display port, the DMI and integrated GPU. To make this efficient realization Sandy Bridge implements a ring interconnect to connect all these parts and provide faster communication between cores, graphics, LLC and system agent area. The ring interconnect is built out of 4 separate rings – a 32 byte data ring, a request ring, an acknowledge ring and snoop ring. All these are fully pipeline and run at core frequency and voltage, so that bandwidth, latency and power scale up and down according to the core. The massive ring wires are routed over the LLC in a way that has no area impact. Access on the ring (core to cache, cache to core or cache to system agent) always takes the shortest path. The arbitration on the ring is distributed, meaning each ring stop doing its own arbitration. This is a complex architecture that has special sending rules.

Sandy Bridge divides the LLC cache into 2MB blocks, in total 8MB LLC for quad core desktop parts. The Sandy Bridge LLC is shared between cores and the graphics component. Cores and graphic component can both access all the data in the LLC, apart from of who allocated the line, after the appropriate memory range checks. The LLC contains the interface logic from core and graphics to the ring, between the cache controller and the ring. It also implements the ring logic, the local arbitration, and the cache controller itself. Each of the caches contains a full cache pipeline. The cache maintains coherency and ordering for the addresses that are mapped to it. It also maintains "core valid bits" like the previous generation of Intel Core processors, to reduce unnecessary snoops. Also LLC runs at core frequency and voltage, scaling with the ring and core.

A lot of the functionality found before in the "North Bridge" has been also integrated to the monolithic die of Sandy Bridge. This includes PCIe ports, DMI link, DDR3 memory controller; this section also contains the power control unit. Power control unit is programmable microcontroller that handles all power management and reset

functions for the entire chip. This part is also closely integrated with the ring, providing fast access and high bandwidth to memory and I/O. Coherency between I/O segment and the main cache is also handled in this section of the chip.

2.3. Platform configuration

In both cases each tested platform is based on the same preproduction Intel desktop board DP67BA with 8GB system memory (4x2GB DDR3,1333MHz) populating all available DIMM slots. We also deployed enterprise Intel SSD X25-E hard disk drive. Operating system installed is RedHat Enterprise Linux 6 kernel ver. 2.6.32-71.el6.x86_64. The new version of the Intel software tool kit has been also installed. The Intel Composer XE 2011 includes Intel Compiler 12.0.0.084 as well as the Intel Math Kernel Library (MKL) 10.3. The new compiler and libraries offer advanced vectorization support, including support for Intel AVX and include Intel Parallel Building Blocks (PBB), OpenMP, High-Performance Parallel Optimizer (HPO), Interprocedural Optimization (IPO) and Profile-Guided Optimization (PGO). All performance tools and libraries provide optimized parallel functions and data processing routines for high-performance applications and in addition contain several enhancements, including improved Intel AVX support.

3. System Performance

The main focus of this section is to present a comparison of two platforms utilizing the 2nd generation Intel Core processor and evaluate how AVX can improve performance for selected representative HPC benchmarks. In this study for evaluation of CPU performance we use a matrix to matrix multiplication micro-benchmark, LINPACK, STREAM, NAS Parallel Benchmarks (NPB) and HPC Challenge Benchmark (HPCC).

In both evaluated platforms each core of CPU has 3 functional execution units that are capable of generating 256 bit results per clock but for no AVX version of the binaries we will operate on 128 bit only as SSE 4.2 does not recognize 256 bit instruction size format. In this case we may assume that a single processor core with AVX enabled does 256 bit ADD instruction in fact this is four 64 bit floating-point ADD instructions and 256 bit MUL instruction equal four 64 bit floating-point MUL instructions per clock. The third 256 bit instruction (or four 64 bit) could be e.g. Shuffle or Bool but Shuffle is not taken in to account for the theoretical performance calculation. For the single processor core with AVX disabled, binaries are only using SSE 4.2, it will be automatically a half, two 64 bit floating-point ADD instructions and two 64 bit floating-point MUL instructions and of course not counted 128 bit Shuffle. Theoretical performance calculated is the product of MUL and ADD executed in each clock, multiplied by frequency. Taking this into account we have the following results. For Intel 2^{nd} generation Intel Core I7-2600 processor with AVX enabled a single core theoretical performance is; 3.4GHz x 8 operations per clock = 27.2 GFLOPS. For a single core Intel 2^{nd} generation Intel Core I7-2600 processor without AVX we will have a theoretical performance = 3.4GHz x 4 operations per clock = 13.6 GFLOPS. If we calculate the theoretical performance of all cores, we will have 108.8 GFLOPS and 54.4 GFLOPS respectively. This is the theoretical calculation without taking in to account Intel Turbo Boost Technology. The 2^{nd} generation Intel Core I7-2600 because of turbo mode allows increased frequency about 4 steps (4 x 100 MHz) for a single core and for four cores it is 100MHz frequency improvement. Theoretically with Turbo Boost Technology enabled single core performance will have 30.4 GFLOPS and 15.2 GFLOPS respectively for Intel 2^{nd} generation Intel Core I7-2600 with AVX enabled and no AVX. Consequently theoretical performance of Intel 2^{nd} generation Intel Core I7-2600 will be 3.5GHz x 8 operations per clocks x 4 cores = 112 GFLOPS for version with AVX and 56 GFLOPS for no AVX version.

This is theoretical performance only, and does not fully reflect the real life scenario. To evaluate how all the new technology enhancements are suited for HPC workload benchmarks we have selected a couple of benchmarks. For the single core performance evaluation we started with a micro-benchmark matrix to matrix multiplication (1). The size of the matrix used in our study is 2048 x 2048. The achieved performance for the single core Intel 2^{nd} generation Intel Core I7-2600 with turbo mode enabled is respectively 12.4 GFLOPS and 16.3 GFLOPS for SSE 4.2 and AVX version. The achieved performance is 81% and 53% of the theoretical performance for Intel 2nd generation Intel Core I7-2600 with AVX disabled and AVX enabled. For the four cores using OpenMP API we achieved results as follow 44.4 GFLOPS and 57.9 GFLOPS which represent 79% and 51% of theoretical

performance for SSE 4.2 and AVX version. The same code compiled with MKL does for a single core 14.2 GFLOPS and 27.2 GFLOPS which is 93% for SSE 4.2 and 89% for AVX of theoretical performance. The difference between SSE 4.2 and AVX version for a single core performance is 91%. For four cores we calculated 49.5 GFLOPS which represents 88 % theoretical peak and 89.4 GFLOPS, 80% of theoretical performance, respectively for SSE 4.2 and AVX version of binaries. For four cores AVX version shows 80% performance advantage over SSE 4.2 version. This matrix to matrix multiplication micro-benchmark shows very impressive results even without any hands-on optimizations.

```
void multiply_d(double a[][NUM], double  b[][NUM], double c[][NUM])                    (1)
{
      int i,j,k;
      for(i=0;i<NUM;i++) {
                  for(k=0;k<NUM;k++) {
            for(j=0;j<NUM;j++) {
                        c[i][j] = c[i][j] + a[i][k] * b[k][j];
                  }
            }
      }
}
```

The second aspect of platform performance we have evaluated is bandwidth. The benchmark we have used to measure the bandwidth is the STREAM benchmark. It estimates, both memory reads and memory writes. This gives us an indication of how effective the memory subsystem is. It measures the performance of four long vector operations. These operations are representative of long vector operations and the array sizes are defined in that way so that each array is larger than the cache of the processors that are going to be tested The STREAM benchmark gives only results of memory system efficiency ignoring the cache efficiencies of the tested platforms. As the memory subsystem are identical in both platforms the results we have achieved for STREAM are also almost identical the difference is less than 1% also difference between single core and four core performance is marginal. Overall bandwidth of the tested platforms with AVX enabled and no AVX was 17.8 GB/s which is 83% of theoretical bandwidth for this single socket platform.

LINPACK is benchmark used to determine the performance of world's fastest computers published at the website http://ww.top500.org/. This is a floating-point benchmark which solves a dense system of linear equations in parallel. The metric produced is Giga-FLOPS (GFLOPS) or billions of floating point operations per second. LINPACK performs operations called LU Factorization. These are highly parallel and store most of their working data set on the processors cache. It makes relatively few references to memory for the amount of computation it performs [4]. The processor operations are predominantly 64-bit floating-point vector operations and these use SSE instructions in one case and AVX in the second during our tests. Single system performance on LINPACK for Intel 2nd generation Intel Core I7-2600 processor no AVX is 50.6 GFLOPS and for platform with AVX enabled is 93.9 GFLOPS. The achieved performance is 90% and 83% of the theoretical performance for Intel 2nd generation Intel Core I7-2600 processor no AVX and Intel 2nd generation Intel Core I7-2600 processor with AVX respectively. For a single core LINPACK delivers 91% better results on AVX versus SSE 4.2 version and gives 27.62 GFLOPS.

The next code we have evaluated is the Numerical Aerodynamic Simulation (NAS) Parallel Benchmarks code. This set of benchmarks has been developed in NASA Ames Research Center for the performance evaluation of parallel supercomputers. The benchmark is based on code specifically developed by NASA computational fluid dynamics (CFD) applications and represents codes of real life applications used by NASA operational computing systems for simulating an entire aerospace vehicle system within a computing time of one to several hours [5]. NAS

Parallel Benchmarks version 1 set therefore consists of two major components five parallel kernel benchmarks and three simulated application CFD benchmarks. The simulated application benchmarks combine several computations class defined based on the problem size and memory requirements. New revisions of NPB added three more benchmarks and new versions of problem size for small memory systems. In our study we have focused on the eight tests from the first implementation of NPB as the new added tests, Unstructured Adaptive, Data Cube operator and Data Traffic are not relevant for our testing scenario where we tested only one preproduction platform with single socket. Each benchmark runs two problem sizes Class A and Class B. Same benchmarks also run Class S of the problem size but NASA does not recommend using this problem size for benchmarking purposes. For the smallest problem size (Class A) improvement in results between AVX and SSE 4.2 cross all NPB benchmarks is 7%-57% but for two benchmarks Conjugate Gradient (CG) and Integer Sort (IS) SSE 4.2 version achieve better results by 10% and 1% respectively than AVX version. For the bigger problem size (Class B) the only benchmark where we not observe results in favor of AVX is Integer Sort but the difference is marginal only 1% on the favorite to SSE 4.2. The range of the performance advantage for the problem size Class B of AVX version versus SSE 4.2 is 6%-58%. Figure 2 shows all the AVX results versus SSE 4.2 for the smallest problem size Class A.

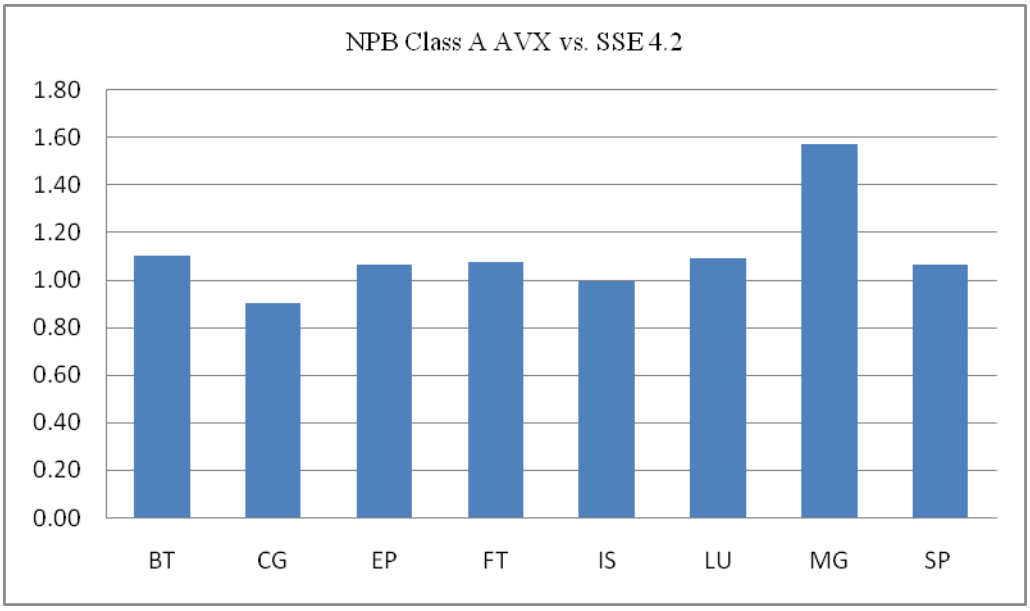

Fig.2. NPB results AVX versus SSE 4.2 for the problem size Class A

The biggest difference between AVX versus SSE 4.2 version is for Multi Grid (MG) benchmark and it is 57%. Multi Grid is a kernel benchmark which approximates the solution to a three-dimensional discrete Poisson equation using the V-cycle multigrid method. This code requires a power of two numbers of processors. Three other benchmark; Block Tridiagonal (BT), Scalar Pentadiagonal (SP) and Lower Upper (LU) solve nonlinear partial differential equations using three different algorithms. For this type of code AVX improves a code by 10%, 7%, 9% respectively. The Fast Fourier Transform (FT) and Embarrassingly Parallel (EP) benchmarks give the same level of improvement by 7% even they represent different type of algorithm, first solve a three-dimensional partial differential equation using the fast Fourier transform, and second generate random variable based on Gaussian distribution. Integer Sort (IS) benchmark which sorts small integers using the bucket sort method shows only 1 % disadvantage for AVX but CG overachieve AVX by 10%. This is interesting if we consider nature of the code. The CG estimates the smallest eigenvalue of a large sparse symmetric matrix [5].

For the big problem size Class B situation has changed and CG performs 8% better for AVX than for SSE 4.2, this is 18 % difference between the version with small Class A and bigger Class B version. For other benchmarks the differences are almost the same as it was a case with Class A. Figure 3 illustrates performance advantage AVX

versus SSE 4.2 for Class B problem. We see for BT benchmark AVX version versus SSE 4.2 version gives 10% improvement, for EP it is 6%, for FT 9%, for LU 11%, for MG 58% and for SP it is 6%.

The next benchmark suite we have tested is HPC Challenge benchmark. This is a set of tests that examines the performance of HPC architectures in a more challenging way than LINPACK, STREAM or small matrix to matrix multiplication micro- benchmark. The HPC Challenge benchmark test suite stresses not only the processors, but also the memory system and interconnects. It is a better indicator of how HPC system will perform across a spectrum of real-world applications unfortunately same of the test are ineffective in our testing scenario as HPC Challenge benchmark is optimized for complete installation of supercomputer and clusters and not perfectly scaling down to single socket platform. However a few of the tests which are a good indicator for full system implementation are also suitable to estimate the single platform performance.

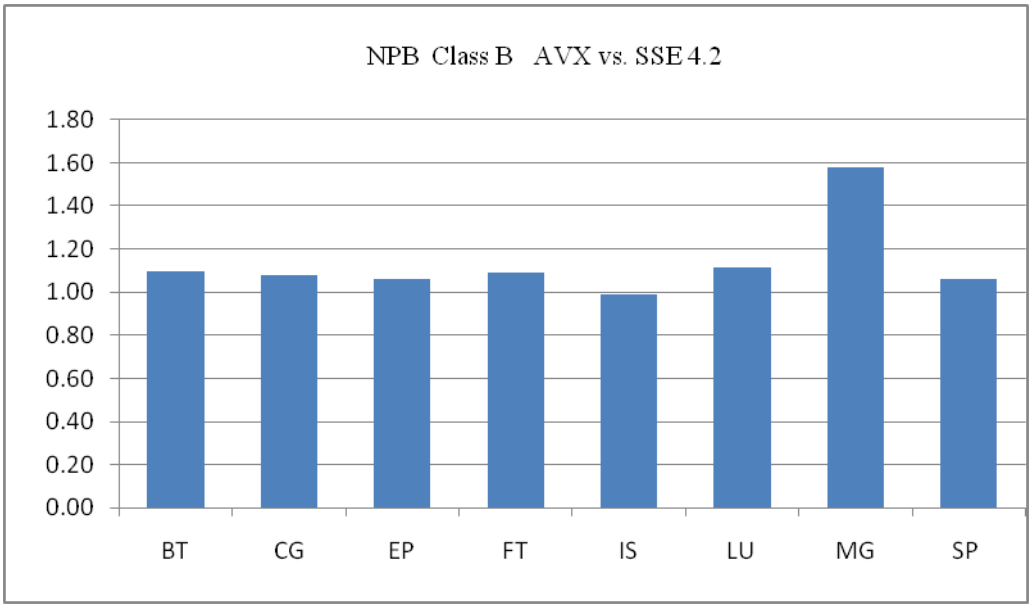

Fig.3. NPB results AVX versus SSE 4.2 for the problem size Class B

The HPC Challenge benchmark consists of basically 7 tests: HPL, DGEMM, STREAM, PTRANS, FFT, Random Order Ring Bandwidth and Random Ordered Ring Latency [6]. In this section we will evaluate single platform based on the 2nd generation Intel Core I7-2600 with AVX enabled and no AVX across all the benchmark listed above.

The HPL benchmark is the portable version of LINPACK for distributed memory systems but for single socket platform it does not make a difference. We have been already evaluated LINPACK in the section above and we will not analyze this again. The results completed with HPL are the same as we have achieved with single node version of LINPACK. The same situation we have also with STREAM and for DGEMM where our matrix to matrix multiplication micro benchmark does exactly the same type of operation as DGEMM benchmark form HPCC.

The PTRANS benchmark performance depends on the network and on memory bandwidth. As the two systems have same memory bandwidth and we do not have any interconnect element in our study archived results are identical.

The Random Access benchmark uses SANDIA_OPT2 algorithm as Giga Updates per second (GUP/s). Both systems have equal memory bandwidth as well it is no interconnecting element associated to this platform so results on both platforms are also the same.

For FFT benchmark on both systems we used Intel Compiler 12.0.0.084 as well as the Intel Math Kernel Library 10.3. The benchmark performs as mixture of flops, memory, and network bandwidth. Both systems have the same clock and the same memory subsystem DDR3-1333MHz and difference is only 8% better results for AVX version versus SSE 4.2.

The random-ordered ring bandwidth (RORB) benchmark measures conflict in the network and reports bandwidth achieved per core in a ring communication model. The ROR bandwidth measures the accumulated bandwidth of the communication network of parallel computing systems. The algorithm uses an average, short and long messages transferred with different bandwidth values. Because two systems have no interconnect element difference in the topology of communication network does not exist consequently the results are very comparable and difference is only 4% better results for AVX version versus SSE 4.2.

The random-ordered ring latency (RORL) measures the latency of the communication network of parallel and, or distributed HPC systems. Several message sizes, communication patterns and methods are used. The algorithm uses an average to take into account that short and long messages are transferred with different bandwidth values in real applications and this has consequences in latency as well [7]. As we have only single system the relevance of this benchmark is minimal. It also does not reflect nature of big system build based on the tested platform. The result of the system with AVX has lover latency vs. system with SSE 4.2 but difference is marginal and is 4 %.

All the results as the difference between AVX and SSE 4.2 are presented on the figure 4 and as we can see difference is almost marginal for all those the benchmark which are more I/O or interconnect or memory bandwidth related and are very significant where we have a benchmark which is very floating point intensive in this case difference is 88% and 83% like for HPL and DGEMM.

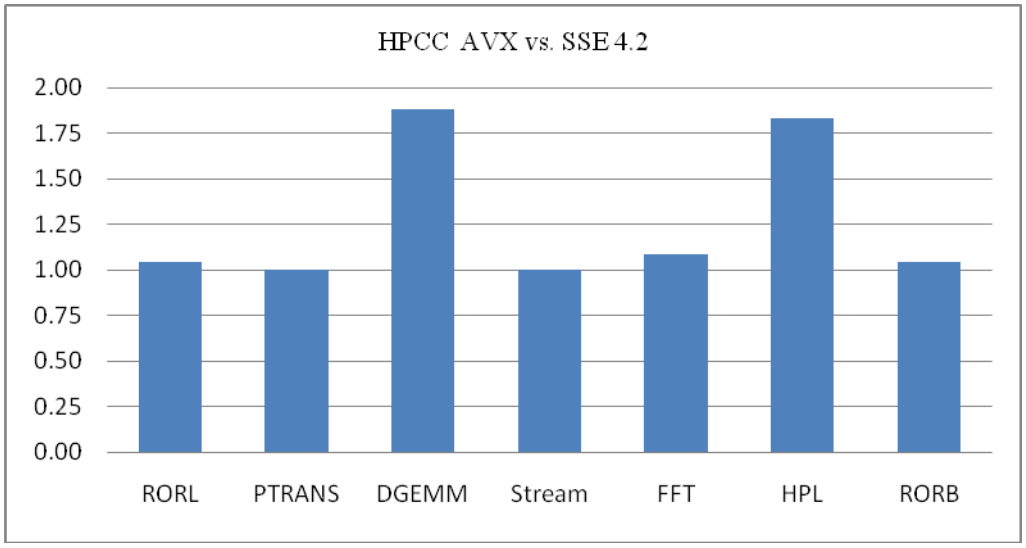

Fig.4. HPC Challenge results ratio AVX versus SSE 4.2

HPC Challenge is optimized for evaluating the complete system installation and as already mentioned it doesn't precisely estimate the performance of a single socket platform. In some of the evaluated benchmarks from the HPC Challenge suite the performance of AVX and SSE 4.2 version were identical. The benchmarks measure the communication aspect of the system and testing a single node only does not make any difference on achieved results. Random Order Ring Bandwidth, Random Ordered Ring Latency and PTRANS are focused on the interconnect aspects of the platforms under test and results are almost identical. This is exactly the portion of the system performance where AVX does not bring performance improvements as this is not floating point intensive but I/O and interconnect focused type of code. However the significant performance improvement for floating point intensive calculation like DGEMM and HPL is only possible when we use MKL. If the code is compiled without MKL the achieved results for DGEMM are only 31% better for AVX version and for HPL we observed only 27% better results for AVX versus SSE 4.2. The same benchmarks compiled with MKL show 88% performance advantage of AVX versus SSE 4.2 for DGEMM and 83% better results for HPL. This very clearly indicates that to get maximum performance advantage of the AVX instructions requires the AVX to be enabled, an optimized compiler and requires the use of well optimized mathematical library like Intel Math Kernel Library 10.3.

4. Conclusion

This paper examined the performance characteristics of AVX instructions set extension using standard HPC benchmarks; LINPACK, STREAM, HPCC and NPB. We utilized preproduction system with the first member of AVX enabled CPU, the new 2nd generation Intel Core I7-2600 processor. We have found that evaluated platform shows significant performance improvement versus exactly the same configured platform but without AVX support. The performance improvement we have been able to observe behaves as we have been expecting taking in to account theoretical performance of both CPUs. Intel AVX delivers significant performance improvements to compute-intensive codes and for same test we have observed 88% difference between AVX version and SSE 4.2 version. The impact is significantly smaller when the application is memory-bound as both platforms have the same configured memory subsystem. As might be expected, for larger systems the interconnect bandwidth and topology have an increasing impact but in our testing scenario limited to the single platform all such tests, do not show any advantage or disadvantage for AVX platform as in fact they are irrelevant for a single socket platform validation. The performance advantage of compute-intensive benchmarks on AVX enabled platform is between 1.58 and 1.88 over the version limited to SSE 4.2. All not compute-intensive benchmarks show around 8% to 10% performance improvement. Intel Turbo Boost Technology was always enabled and delivers additional frequency improvement and consequently improvement in performance.

The performance improvement is only possible with AVX optimized compiler and library. This is important to emphasize that difference between results obtained from compiler only version versus results linked with AVX version of MKL library is significant. For matrix to matrix multiplication micro-benchmark for a single core with SSE 4.2 version versus AVX version we obtained 12.4 GFLOPS versus 16.3 GFLOPS when we compile both versions without MKL. This is only 30% performance advantage for AVX version. For the same benchmark but with MKL support we have 14.2 GFLPOS and 27.2 GFLOPS respectively for SSE 4.2 and AVX, so 91% performance advantage for AVX version. This is also valid for four cores configuration as well for other benchmarks e.g. LINPACK.

In summary, we can state that AVX even in the early phase of the platform implementation not optimized for HPC brings lot of performance improvement for compute intensive applications and become a compelling choice for many of the new HPC installations.

Acknowledgements

We gratefully acknowledge the help and support provided by Jamie Wilcox from Intel EMEA Technical Marketing HPC Lab.

References

1. P.Gepner, D. Fraser, L. Kowalik, K. Wackowski. Early performance evaluation of new Six-Core Intel® Xeon® 5600 family processors for HPC, Proceedings of 9th International Symposium on Parallel and Distributed Computing, Istanbul, 2010.
2. Intel® Advanced Vector Extensions Programming Reference, July, 2009
3. L. Gwennap. Sandy Bridge spans generations, Microprocessor Report, September, 2010.
4. J. Dongarra., P. Luszczek,A. Petitet,. Linpack Benchmark: Past, Present, and Future, http://www.cs.utk.edu/~luszczek/articles/hplpaper.pdf
5. D. Bailey, E, Barszcz, J.Barton, D. Browning, R.Carter, L.Dagum, R. Fatoohi,S. Fineberg, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon,V. Venkatakrishnan,S. Weeratungal, The NAS Parallel Benchamrks, 1994.
6. HPC Challenge Benchmarks, http://icl.cs.utk.edu/hpcc/
7. S. Saini, D. Talcott, D. Jespersen, J. Djomehri, H. Jin, and R.Biswas, Scientific Application-based Performance Comparison of SGI Altix 4700, IBM Power5+, and SGI ICE 8200 Supercomputers, Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, Austin, Texas, 2008.