PERFORMANCE TO POWER AND PERFORMANCE TO COST RATIOS WITH INTEL XEON PHI COPROCESSORS (AND WHY 1X ACCELERATION MAY BE ENOUGH)

Andrey Vladimirov
Colfax International

January 27, 2015

Abstract

The paper studies two performance metrics of systems enabled with Intel Xeon Phi coprocessors: the ratio of performance to consumed electrical power and the ratio of performance to purchasing system cost, both under the assumption of linear parallel scalability of the application.

Performance to power values are measured for three workloads: a compute-bound workload (DGEMM), a memory bandwidth-bound workload (STREAM), and a latency-limited workload (small matrix LU decomposition). Performance to cost ratios are computed, using system configurations and prices available at Colfax International, as functions of the acceleration factor and of the number of coprocessors per system. That study considers hypothetical applications with acceleration factor from 0.35x to 2x.

In all studies, systems with Intel Xeon Phi coprocessors yield better metrics than systems with only Intel Xeon processors. That applies even with acceleration factor of 1x, as long as the application can be distributed between the CPU and the coprocessor.

Table of Contents

1. ACCELERATED COMPUTING WITH INTEL XEON PHI COPROCESSORS

Intel Xeon Phi coprocessors are the first generation of Intel Many Integrated Core (MIC) architecture devices. They support the same programming languages and parallel frameworks as multi-core Intel Xeon processors, and, in many cases, optimized code for coprocessors is also optimal for CPUs, and vice-versa.

The current generation of Intel Xeon Phi coprocessors based on the Knights Corner (KNC) chip dates to Q4 2012 and Q1-Q2 of 2013. While there was no update to the coprocessor lineup in 2014, the next generation, based on the Knights Landing (KNL) chip, is expected in 2015.

This paper sets out to benchmark the efficiency of coprocessors against alternative CPU-only solutions. In this comparison, the following concepts and terms are used:

1. **Acceleration factor** – the ratio of performance of an application on a single Intel Xeon Phi coprocessor to its performance on an Intel Xeon multi-core processor. This ratio, of course, is subject to clarification, because different processor and coprocessor models can be used for comparison, and also this factor varies from one application to another. In this paper, the acceleration factor is defined using an Intel Xeon Phi coprocessor 7120P (highest performance model) and a dual-socket Intel Xeon E5-2697 V2 processor (top of the line of the Ivy Bridge architecture Xeons).
2. **Performance to power ratio (PPR)** – the ratio of performance of an application on a heterogeneous system to the *total* power consumed by this system. Note that in this paper, we do not estimate the consumed power using the thermal design power (TDP) metric; all reported power values are actual measurements taken with AC power meters.
3. **Performance to cost ratio (PCR)** – the ratio of performance of an application to the commercial cost of that system. In this study, costs are based on actual price estimates on solutions offered by Colfax International.

2. PERFORMANCE TO POWER RATIO (PPR)

The total electrical power consumed by a computing system under load depends on what components the application stresses: processor cores, processor caches, main memory, PCIe bus, interconnects, hard drives, etc. In order to provide useful metrics for HPC uses, this work benchmarks three applications stressing the most important components for HPC workloads: cores (specifically, vector processing units (VPUs) on coprocessors or arithmetic and logic units (ALUs) on processors), caches, and memory. These applications are listed in Table 1 and discussed in Sections 2.1–2.3.

2.1. DGEMM

DGEMM is a LAPACK double-precision matrix-matrix multiplication routine known to be compute-bound, i.e., it stresses VPUs and ALUs. This work used the multi-threaded DGEMM implementation in the Intel Math Kernel Library (MKL) [2].

Each device in the benchmark (a CPU or a coprocessor) performed DGEMM on its own matrix, and only one matrix multiplication at a time per device was computed. Such approach does not necessarily estimate DGEMM performance in a distributed system; rather, it represents hypothetical perfectly scalable applications with negligible communication.

For benchmarks, square matrices of size 23424 were used. This size yields good performance because it is a multiple of the number of threads (24 on the host and 244 on the coprocessor), and the per-thread counts of columns, $23424/24 = 976$ and $23424/244 = 96$, are multiples of 8, which amounts to the size of the 64 byte cache line.

For optimum performance, the host used OpenMP affinity of type `KMP_AFFINITY=scatter` and the coprocessor used `compact`. The coprocessor version was run in the native mode (i.e., without offload) using an SSH session.

To translate the wall clock time of DGEMM into performance measured in GFLOP/s, a conversion factor was used which assumes that every DGEMM call performs $2N^3$ operations, where N is the matrix size.

Application	Source	Description	Type	Target
DGEMM	Intel MKL	Double precision general matrix-matrix multiplication.	Compute-bound	VPU/ALU
STREAM	Public	Benchmark of streaming memory bandwidth.	Bandwidth-bound	RAM
LU decomp.	Original	Double precision LU factorization of small square matrices.	Latency-limited	Caches

Table 1: Applications used in benchmarks of PPR.

2.2. STREAM

STREAM is a memory bandwidth benchmark which reads and writes large contiguous arrays, performing four tests: copy, scale, add, and triad. The source code of STREAM developed by J. D. McCalpin is publicly available [3]. Again, to run STREAM on multiple devices, on each device its own independent executable was launched.

For the host, STREAM was compiled with the Intel C compiler with the following arguments: `-qopenmp -O3 -DSTREAM_ARRAY_SIZE=64000000 -DNTIMES=100`. To compile the coprocessor executable, following [4], additional arguments were added:

```
-ffreestanding -opt-prefetch-distance=128
-opt-streaming-cache-evict=1
-opt-streaming-stores always.
```

The benchmark was run on the host with 24 threads and affinity of type "scatter". On the coprocessor, 60 threads and the default affinity (also "scatter") was used.

2.3. LU Decomposition

LU decomposition expresses a matrix A as a product of a unit lower triangular matrix L and an upper triangular matrix U. For small single precision matrices of size 128×128, with each thread decomposing an independent matrix, the performance of LU decomposition appears to be limited by cache latency [5].

The most recent Intel MKL implementation of LU decomposition (`sgetrf`) is not well optimized for Intel Xeon Phi coprocessors in the small matrix regime, so instead of the MKL implementation, a custom C++ code presented in [5] was used.

On the host, the code was run with 48 threads and affinity "scatter", and on the coprocessor, it was run with 244 threads and default affinity (also "scatter").

2.4. System Configuration

All of the benchmarks presented in this section were taken on a Colfax ProEdge™ SXP8600 workstation based on a two-way Intel Xeon E5-2697 V2 processor (12 cores per socket, 24 cores total) with 128 GB of RAM at 1600 MHz. The system contains up to four Intel Xeon Phi 7120P coprocessors. For connectivity, the system was also equipped with two Intel True Scale QLE7340 interconnects. The code was compiled using the Intel C++ compiler version 15.0.1.133 and run under MPSS 3.4.1 on a CentOS 7.0 Linux OS.

2.5. Single-Device Performance

The results of the benchmark on a single dual-socket CPU and on a single coprocessor are shown in Table 2.

Application	Perform. on CPU	Perform. on Coprocessor	Acceleration Factor
DGEMM	485 GF/s	984 GF/s	2.03x
STREAM	85.0 GB/s	176 GB/s	2.07x
LU decomp.	249 GF/s	259 GF/s	1.04x

Table 2: Performance of benchmark applications on a single host and on a single coprocessor.

Notably, compute-bound and bandwidth-bound workloads are a good fit for the Intel MIC architecture, so DGEMM and STREAM clocked an acceleration factor of 2x. In contrast, the LU decomposition, which has a dependence on cache performance, has an acceleration factor marginally over 1x.

2.6. Power Benchmarks

In order to measure the performance to power ratio, the application were run on the benchmark system one by one, and consumed power was measured. There were three classes of tests:

a) CPU-only calculations. These establish the baseline for the Xeon Phi-accelerated runs. For power measurement in CPU-only calculations, all coprocessors were removed from the system.

b) Coprocessor-only calculations. These leave the host CPU idle, but load all available Intel Xeon Phi coprocessors with the benchmarked application. For power measurements with fewer than 4 coprocessors, the inactive coprocessors were physically removed from the system, so that their idle power consumption is not factored into the measurement.

c) Heterogeneous calculations. These run the same application on the CPU and on each of the available coprocessors. Again, for power measurements with fewer than 4 coprocessors, inactive ones were removed.

To measure power consumption, the system was plugged into the electrical circuit via two AC power meters, one for each of the two power supplies (Figure 1).

A significant factor determining power consumption is the settings of the cooling system. The workstation was cooled by room temperature air. Onboard fans were set in the "Optimal" mode, which means that the fans cooling the CPU and RAM part of the board were controlled by the CPU temperature, while the fans cooling the coprocessor part of the board were controlled by the temperatures of the coprocessors' chips.

Table 3 reports raw performance and power measurements as well as the PPR metric. The power measurements are the sum of the readings of the two power meters taken at least 2 minutes after the start of the calculation (the delay was needed for the temperature and fans to settle to a steady state). Even though the standard deviation of the readings was not formally calculated, it can be estimated at well under 10 W because in most cases, the readings of the power meters fluctuated by less than a few Watts. Performance is calculated as the sum of the performances for all compute devices using Table 2. The standard deviation of that number is of order 1%. Finally, the PPR ratio is computed in relation to the baseline PPR, which is that of the CPU-only calculations.

Information shown numerically in Table 3 is also plotted in Figure 2.

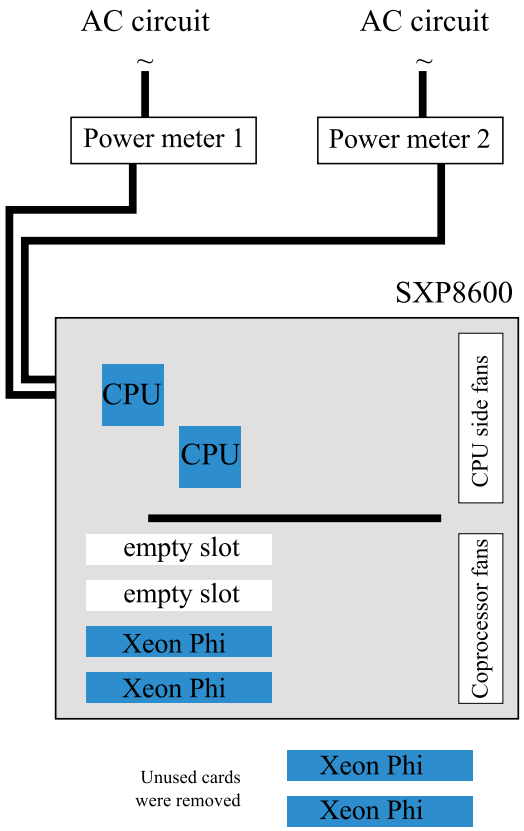

Figure 1: Power measurement setup illustration.

Setup	DGEMM			STREAM			LU		
	Power, W	Perform. GFLOP/s	PPR	Power, W	Perform., GB/s	PPR	Power, W	Perform., GFLOP/s	PPR
Xeon Only	346	485	1.00	331	85	1.00	335	249	1.00
1 Xeon Phi	385	984	1.82	356	176	1.93	399	259	0.87
1 Xeon Phi + Xeon	592	1469	1.77	551	261	1.84	596	508	1.15
2 Xeon Phi	604	1968	2.32	557	352	2.46	637	518	1.09
2 Xeon Phi + Xeon	813	2453	2.15	753	437	2.26	836	767	1.23
4 Xeon Phi	1066	3936	2.63	978	704	2.80	1131	1036	1.23
4 Xeon Phi + Xeon	1290	4421	2.44	1182	789	2.60	1335	1285	1.29

Table 3: Performance of benchmark applications with multiple coprocessors.

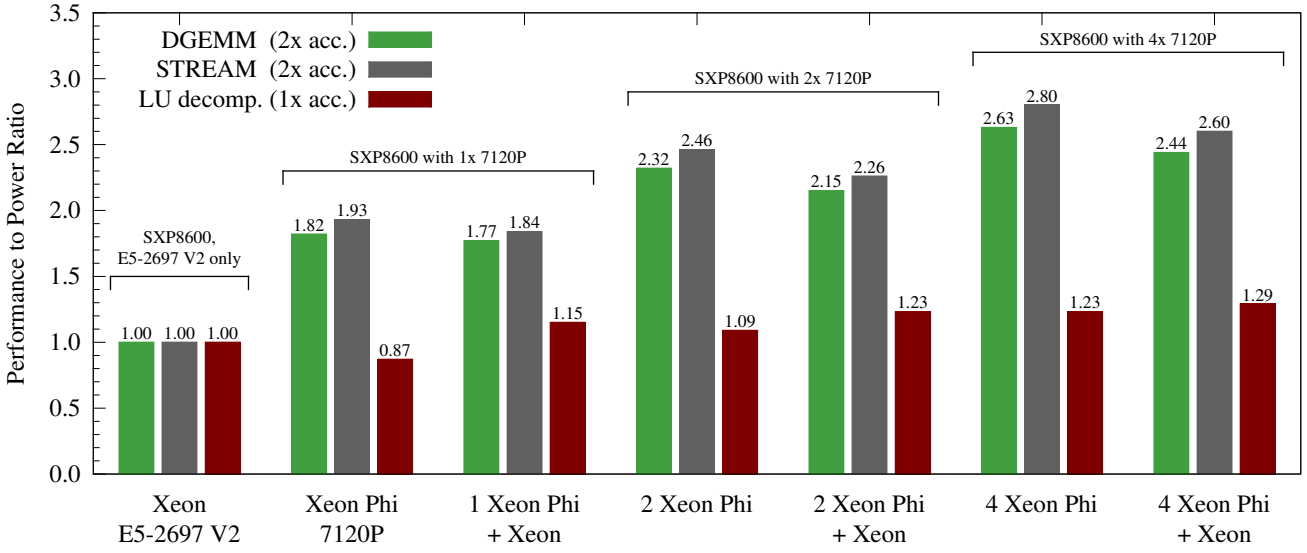

Figure 2: Performance to power ratio (PPR).

3. PERFORMANCE TO COST RATIO (PCR)

My goal for this paper was to estimate the performance to cost ratio as a function of two parameters: the number of Intel Xeon Phi coprocessors in the system, and the the acceleration factor provided by each of the coprocessors. It is, of course, a less well-defined metric than performance to power ratio because of the numerous base platforms available with support for different wayness of CPUs and different numbers of coprocessors[1].

For that reason, the numbers presented in this paper should not be interpreted as purchasing advice; they are merely here to show the trend of performance to cost ratio with the use of coprocessors.

The methodology of PCR estimate was this:

1. The base platform (SXP8600) in the configuration described in Section 2.4, without any coprocessors, has an estimated cost of B=\$10,000, of which around \$5,600 is contributed by the Intel Xeon E5-2697 V2 processor;
2. each additional Intel Xeon Phi coprocessor adds X=\$3,500 to the cost, i.e., with N coprocessors, the system cost is $B + NX$. See Table 4 for specific costs.
3. Consider a hypothetical application with the acceleration factor of A and perfect scalability across multiple devices. If N coprocessors are used, then the relative performance of this application is AN (if the CPU is not used) or $1 + AN$ (if the CPU is used).
4. The PCR relative to the CPU-only baseline can now be estimated as

$$\mathrm{PCR} = \frac{AN}{B + NX} \times B \qquad (1)$$

if the CPU is used, or

$$\mathrm{PCR} = \frac{1 + AN}{B + NX} \times B \qquad (2)$$

if the CPU is not used.

Configuration #	# of Coprocessors	Price*
1	0	\$10,000
2	1	\$13,500
3	2	\$17,000
4	4	\$24,000

Table 4: Retail price of the SXP8600 workstation with 0 to 4 Intel Xeon Phi 7120P coprocessors in the configuration used in this work.
* Quoted price is based on a sample system configuration in Q1 2015. These provided numbers are informational estimates and do not serve as a commercial offer.

Figure 3 plots the relative PCR with 0, 1, 2 and 4 coprocessors for hypothetical applications with different acceleration factors A:

1. $A = 0.35$ is the case of very poor acceleration; it provides a "break-even" case in which using the whole system with any number of coprocessors yields the same PCR as using a Xeon-only system;
2. $A = 1.00$ is a case similar to the LU decomposition application discussed above. This is a moderate acceleration factor.
3. $A = 2.00$ is a case of excellent acceleration, as represented by the DGEMM and STREAM applications.

4. DISCUSSION

The object of interest in this publication was the power and cost efficiency of systems with Intel Xeon processors and Intel Xeon Phi coprocessors. For a best-case estimate, the benchmarks needed to be taken with the most efficient solutions. With power efficiency being a priority (and the same, to some degree, applies to cost efficiency), one must consider not only the efficiency of the compute devices (i.e., processors), but also the overhead introduced by the server board, memory, interconnects, drives and the cooling system. In that sense, intuitively, the most efficient solution is the most computationally dense one. This is why this work chose to benchmark the highest performing processors and coprocessors, and a system that can support 4 coprocessors.

[1] See the Colfax International Web site for information about workstations and servers supporting Intel Xeon Phi coprocessors

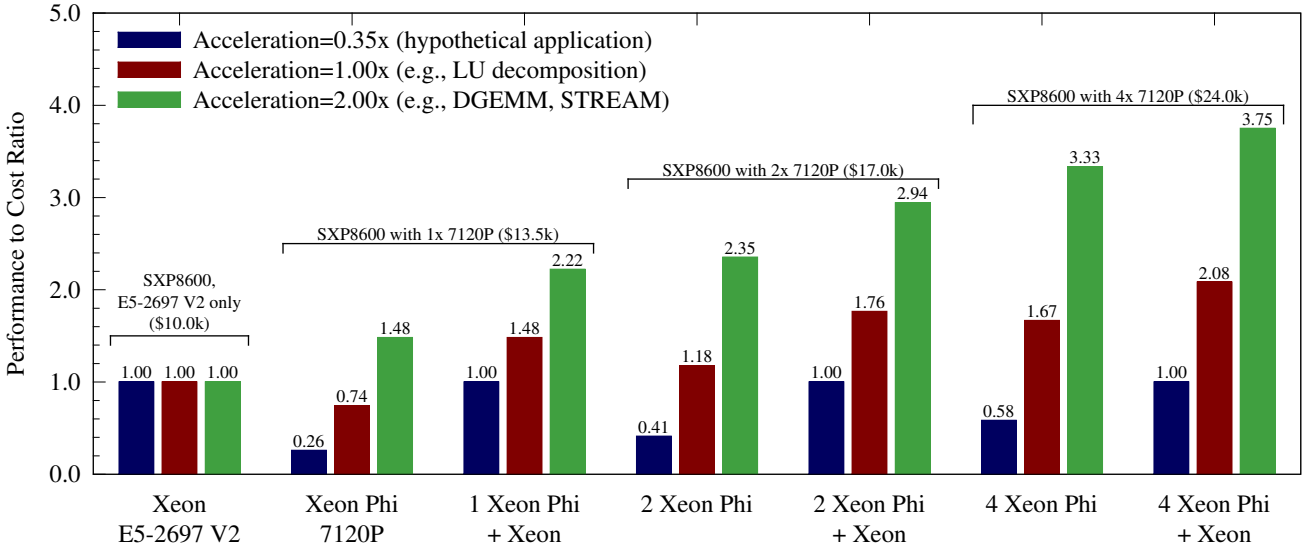

Figure 3: Performance to cost ratio (PCR).

With a very powerful CPU used as a baseline, the quotes the acceleration factors are not as high as in the case of mid-range or low-end CPUs. The highly optimized DGEMM and STREAM workloads had an acceleration factor of 2x, and the small matrix LU decomposition a factor of 1x. Nevertheless, even with these low acceleration factors, the contribution of Intel Xeon Phi coprocessors to power and cost efficiency is positive:

1) For DGEMM, STREAM (acceleration factor of 2x),
   - adding only one coprocessor can increase the PPR by 80-90% and the PCR by 120%;
   - adding four coprocessors boosts PPR by 160-180% and PCR by almost 300%.
2) For LU decomposition (acceleration factor of 1x), as long as the coprocessor(s) are used in tandem with the CPU,
   - adding one coprocessor can increase the PPR by 15% and PCR by 50%,
   - adding four coprocessors can increase the PPR by 30% and PCR by 110%.
3) Speaking of PCR, a hypothetical acceleration factor of 0.35x is the "break-even" point where adding coprocessors to the system does not change the PCR.

Gaining more performance per watt and performance per dollar of set-up costs with computing accelerators is, of course, not news: it is the selling point of these products. However, where Intel Xeon Phi coprocessors stand out is applications with low acceleration factors. That is because, as Figures 2 and 3 show, achieving increased PPR or PCR with acceleration factor of 1x is dependent upon using a heterogeneous approach where the CPU is used together with the coprocessor to process the problem in parallel. This approach is only possible if equally well optimized CPU and accelerator code of the application is available. With Intel Xeon Phi coprocessors, *the same performance-critical code* may be used for the host and for the coprocessor (it has been shown in numerous case studies – see, e.g., [6, 7, 8, 9, 10, 11]), therefore the heterogeneous approach does not require increased development effort.

REFERENCES

[1] Landing page for this paper, "Performance to Power and Performance to Cost Ratios..". http://research.colfaxinternational.com/post/2015/01/27/1x.aspx.

[2] Intel Math Kernel Library. http://software.intel.com/en-us/intel-mkl.

[3] John D. McCalpin. STREAM: Sustainable Memory Bandwidth in High Performance Computers. http://www.cs.virginia.edu/stream/.

[4] Karthik Raman. Optimizing Memory Bandwidth on Stream Triad, Feb 2013. http://software.intel.com/en-us/articles/optimizing-memory-bandwidth-on-stream-triad.

[5] Fine-Tuning Vectorization and Memory Traffic on Intel Xeon Phi Coprocessors: LU Decomposition of Small Matrices. http://research.colfaxinternational.com/post/2015/01/27/LU.aspx.

[6] Primer on Computing with Intel Xeon Phi Coprocessors. Slides from a presentation, with links to additional resources. http://research.colfaxinternational.com/post/2014/03/06/Geant4-Tutorial.aspx.

[7] Andrey Vladimirov and Vadim Karpusenko. Test-driving Intel Xeon Phi coprocessors with a basic N-body simulation. http://research.colfaxinternational.com/post/2013/01/07/Nbody-Xeon-Phi.aspx.

[8] Accelerated Simulations of Cosmic Dust Heating Using the Intel Many Integrated Core Architecture. http://research.colfaxinternational.com/post/2013/06/07/HEATCODE.aspx.

[9] Andrey Vladimirov and Vadim Karpusenko. Hetetogeneous Clustering with Homogeneous Code. http://research.colfaxinternational.com/post/2013/10/17/Heterogeneous-Clustering.aspx.

[10] Andrey Vladimirov. Multithreaded Transposition of Square Matrices with Common Code for Intel Xeon Processors and Intel Xeon Phi Coprocessors. http://research.colfaxinternational.com/post/2013/08/12/Trans-7110.aspx.

[11] Andrey Vladimirov and Cliff Addison. Cluster-Level Tuning of a Shallow Water Equation Solver on the Intel MIC Architecture. http://research.colfaxinternational.com/post/2014/05/12/Shallow-Water.aspx.