

Applying EMD/HHT analysis to power traces of applications executed on systems with Intel Xeon Phi

The International Journal of High
Performance Computing Applications
1–12

© The Author(s) 2017

Reprints and permissions:

sagepub.co.uk/journalsPermissions.nav

DOI: 10.1177/1094342017731612

journals.sagepub.com/home/hpc



Gary Lawson¹, Masha Sosonkina¹, Tal Ezer² and Yuzhong Shen¹

Abstract

Power draw is a complex physical response to the workload of a given application on the hardware, which is difficult to model, in part, due to its variability. The empirical mode decomposition and Hilbert–Huang transform (EMD/HHT) is a method commonly applied to physical systems varying with time to analyze their complex behavior. In authors' work, the EMD/HHT is considered for the first time to study power usage of high-performance applications. Here, this method is applied to the power measurement sequences (called here *power traces*) collected on three different computing platforms featuring two generations of Intel Xeon Phi, which are an attractive solution under the power budget constraints. The high-performance applications explored in this work are codesign molecular dynamics and general atomic and molecular electronic structure system—which exhibit different power draw characteristics—to showcase strengths and limitations of the EMD/HHT analysis. Specifically, EMD/HHT measures *intensity* of an execution, which shows the concentration of power draw with respect to execution time and provides insights into performance bottlenecks. This article compares intensity among executions, noting on a relationship between intensity and execution characteristics, such as computation amount and data movement. In general, this article concludes that the EMD/HHT method is a viable tool to compare application power usage and performance over the entire execution and that it has much potential in selecting most appropriate execution configurations.

Keywords

Energy, power, performance, Intel Xeon Phi, empirical mode decomposition, Hilbert–Huang transform, Sandia PowerAPI, CoMD, GAMESS

1. Introduction

Accelerators, or highly parallel processors, are a key component in reaching exascale performance due to the 20-MW power budget (Kusnezov et al., 2013). Application developers must be more conscious of power as well as performance, and accelerators are an attractive solution. One such accelerator is the Intel Xeon Phi. Although these devices require more power than the CPU, the performance increase can outweigh the energy costs.

As of 2015, a new architecture of the Xeon Phi became available; code-named “Knights Landing” (KNL), the device is available as a processor or coprocessor. The processor version is similar to a traditional node capable of hosting a full Linux OS (Sodani et al., 2016), and the coprocessor version connects to the host CPU via the Peripheral Component Interconnect (PCI) bus. The previous architecture of the Xeon Phi, “Knights Corner” (KNC), is only available as a

coprocessor. KNL yields a higher performance per watt ratio over the older KNC hardware, and since obtaining energy savings is a major goal, both architectures are investigated in this work. It is important to compare the differences of runs between various hardware systems and execution strategies to uncover performance bottlenecks and excessive power usage in an effort to minimize energy consumption.

One of the leading challenges in hardware–software codesign is understanding the interactions that occur

¹Department of Modeling, Simulation, and Visualization Engineering, Old Dominion University, Norfolk, VA, USA

²Department of Ocean, Earth, and Atmospheric Sciences, Old Dominion University, Norfolk, VA, USA

Corresponding author:

Gary Lawson, Department of Modeling, Simulation, and Visualization Engineering, Old Dominion University, 1300 Engineering and Computational Sciences Building, Norfolk, VA 23529, USA.
Email: glaws003@odu.edu

between hardware and various software applications (workloads). The goal is to maximize hardware utilization while minimizing energy consumption and time-to-solution. Power draw variability is the result of software interacting with hardware. In general, power draw fluctuations are hard to predict and analyze. By using the empirical mode decomposition and Hilbert–Huang transform (EMD/HHT) analysis method, power draw may be represented by a set of functions to shed light on hardware–software interactions that cause the fluctuations. This software method could be used to analyze power traces in real time to provide feedback for power regulating systems. In the past, the EMD/HHT technique has already proven to be useful for determining the physical interactions that occur for a given signal. For example, in oceanography, this method has been used to analyze sea level data. To authors’ knowledge, the authors are the first to use the EMD/HHT analysis method on power traces and, thereby, to demonstrate its capabilities in a new field.

This work investigates several systems containing the Xeon Phi accelerator to determine what insights might be learned from applying the EMD/HHT analysis technique to power measurements taken for different applications. On each system, an experiment has been conducted using one or both applications: code-sign molecular dynamics (CoMD) (ExMatEx, 2012) and general atomic and molecular electronic structure system (GAMESS) (Schmidt et al., 1993). The experiment explores different configuration spaces of each system depending on the usage modes available: CPU-only, CPU-KNC offload, or KNL-only. Note, two versions of the Intel Xeon Phi are investigated in this work, denoted by the architectures *KNC* and *KNL*.

Comparing executions for different applications and hardware can be difficult. One may consider time-to-solution as the only criteria to simplify the choice; however, this overlooks energy consumption and therefore provides a suboptimal choice. Another option is to compute the performance per watt; however, this is quite difficult to compute for real-world applications because the workload may include operations other than computations (e.g. data movement, Input/Output (IO)). Being able to quantify hardware component usage would be very beneficial to determination of optimal execution strategies for exascale computing.

1.1. Related work

Over the past few years, a large body of knowledge has been cultivated by researchers interested in the Intel Xeon Phi. For most of this research, the task is to determine the performance benefits and drawbacks of the hardware given the research application rather than to study energy consumption. For example, of the 32 works considered from 2013 to 2016, only 5 investigate

power along with performance (Abdurachmanov et al., 2014; Choi et al., 2014; LaKowski et al., 2015; Li et al., 2014; Wood et al., 2014).

The most common theme is comparing the relative performance for certain Xeon Phi usage modes and/or hardware for the application(s) of interest. This trend indicates that determining the optimal mapping to the Xeon Phi is critical to application developers. In a total of 19 works (Aprà et al., 2014; Bernaschi and Salvadore, 2014; Bernaschi et al., 2014; Brown et al., 2015; Heinecke et al., 2013; Höhnerbach et al., 2016; Jundt et al., 2015; Krishnaiyer et al., 2013; Lai et al., 2014; Liu et al., 2015; Lopez et al., 2015; Mathew et al., 2015; Misra et al., 2013; Newburn et al., 2013; Park et al., 2013; Saini et al., 2015; Sainz et al., 2015; Saule et al., 2014; Teodoro et al., 2014), it was found the Xeon Phi outperformed the CPU, and only two works found the CPU to be better (Li et al., 2014; Luo et al., 2013). Hence, Xeon Phi was found to be a promising accelerator. As to comparing specific ways to utilize Xeon Phi (in native, offload, or symmetric), 16 of above 19 references provided some comparisons of the former two and essentially were split 50/50 as to which one is better (i.e. 7 vs. 9, respectively) while noting that data movement over the PCI bus severely limits the offload performance. For the native mode, it was found that the performance is limited by the Xeon Phi resources: specifically the lack of memory associated with each core and total dynamic random-access memory (DRAM). Symmetric mode is limited by efficient workload balancing as found in two of all references considered, which shows that this usage mode is the least investigated.

The remainder of the article has been divided into the following sections. Section 2 presents the EMD/HHT analysis method used to compare executions in this work and section 3 presents the experiment procedure used in this work (hardware, applications, and execution procedure). Section 4 presents the discussion of the EMD/HHT analysis results and section 5 concludes this work.

2. Analysis method

The EMD/HHT method (Huang et al., 1998; Wu and Huang, 2009) is used for nonparametric nonstationary time-series analysis and calculates instantaneous amplitude and frequency. It is applied to real-world systems to uncover underlying physical interactions. This method has been already successfully applied in a variety of fields, such as medicine, finance, engineering, and more recently in geosciences. The main advantage of EMD/HHT over standard spectral methods is that it detects oscillating modes with time-dependent amplitudes and frequencies, so it is useful for analyzing irregular data with unknown frequencies. On the other hand, the interpretation of the EMD/HHT results is

not straightforward since individual modes do not necessarily represent particular execution characteristics. The method has been adopted to analyze an execution as a whole as opposed to its division into *phases* based on specific resources used in each phase. Phase refers to a computation or data movement type operation, such as RAM to cache data transfers or communication on the node or over the network; the phases often overlap to optimize performance. Such a division was considered by Lawson et al. (2015) in order to model each phase differently, which has proven to be difficult in general for correlating phases with power readings. The implementation of the EMD/HHT method used here is based on the original one from Huang et al. (1998) and Wu and Huang (2009), as adapted by Ezer and Corlett (2012) and Ezer et al. (2013), and the code for EMD/HHT analysis is available in the study by Ezer (2015) in MATLAB.

EMD is used to decompose a power trace into *oscillating intrinsic mode functions* (IMFs) and a *residual trend*. An IMF is a function that satisfies two criteria (Huang et al., 1998). First, the number of extrema and number of zero crossings must be equal or differ by no more than one. Second, the mean value of the envelope defined by the local maxima and minima is 0. EMD extracts IMFs through a process called *sifting*. To sift, the minimum and maximum extrema of the time series are used to calculate the average; the difference between the average and time series is then treated as the time series for the next sift. This process continuously refines the data set until the standard deviation of the resulting time series is less than 0.2 (see Huang et al., 1998). Once this standard deviation is obtained, the resulting time series is accepted as an IMF and is subsequently removed from the original time series. This process is repeated until the residual is found from which no other IMFs may be obtained. One potential use for the residual trend is to construct a nonlinear model to relate power and time-to-solution, as proposed by Lawson et al. (2017). Note that the total number of IMFs is an output of EMD and depends on the trace characteristics. For instance, more IMF modes are found in longer traces because low-frequency oscillations are more likely to be detected.

HHT is then applied to each IMF, except the residual, to calculate instantaneous frequency: the time derivative of the oscillation phase for any time step of the signal (Huang et al., 1998). The maximum frequency that may be obtained using HHT is determined by the sampling rate r in the expression $1/(5r)$, where 5 is the minimum number of data points required to accurately define instantaneous frequency (Huang et al., 1998). In this work, two sampling rates are used, 5 and 20 ms. For 5 ms, the maximum frequency is 40 Hz, and for 20 ms, the maximum frequency is 10 Hz. As will be

shown, the lower sampling rate significantly impacts the utility of the EMD/HHT method.

Two examples of the EMD/HHT analysis procedure are provided in Figure 1; CoMD is shown in Figure 1(a) to (c) and GAMESS is shown in Figure 1(d) to (f). Power traces were collected on the KNL Xeon Phi system for each application CoMD and GAMESS, see Figure 1(a) and (d), respectively. EMD has been applied to each power trace, and the amplitudes for the resulting IMFs are shown in Figure 1(b) and (e). Note that the number of IMFs depends on the workload (cf. CoMD and GAMESS with 12 and 13 IMFs, respectively). Then, HHT is been applied to each intermediate IMF as shown in Figure 1(c) and (f). Finally, the amplitudes and instantaneous frequencies for each IMF may be accumulated in a two-dimensional (2-D) histogram of frequency versus time as introduced in Section 4.

3. Experiment procedure

An execution is defined as an application that is run according to a set of configuration parameters on a hardware platform. The configuration dictates how the application will interface the hardware, such as number of cores or what devices are to be used during execution. Before execution begins, the power measurement tool is started to collect the power trace for all hardware devices used according to the configuration. Each power trace includes a time stamp, recording the time elapsed since starting the tool, and the raw power measurement. A power trace is collected while executing an application on a hardware platform according to a specific configuration; the power trace includes idle power measurements before execution begins and after execution ends to maintain consistency between traces for EMD/HHT analysis. The procedure for the method is as follows:

- I. Collect power measurements during the execution of an application on a given hardware platform to create a power trace.
- II. Apply EMD to the power trace to decompose the time series into a set of IMFs where amplitude represents power.
- III. Apply the Hilbert transform to each IMF to calculate instantaneous frequency.
- IV. Collect instantaneous amplitude and frequency in a 2-D colored histogram to visualize the time series according to time–frequency–amplitude.

3.1. Measurements

The Sandia National Labs PowerAPI (Laros, 2016) is used to collect CPU and Xeon Phi power. CPU power

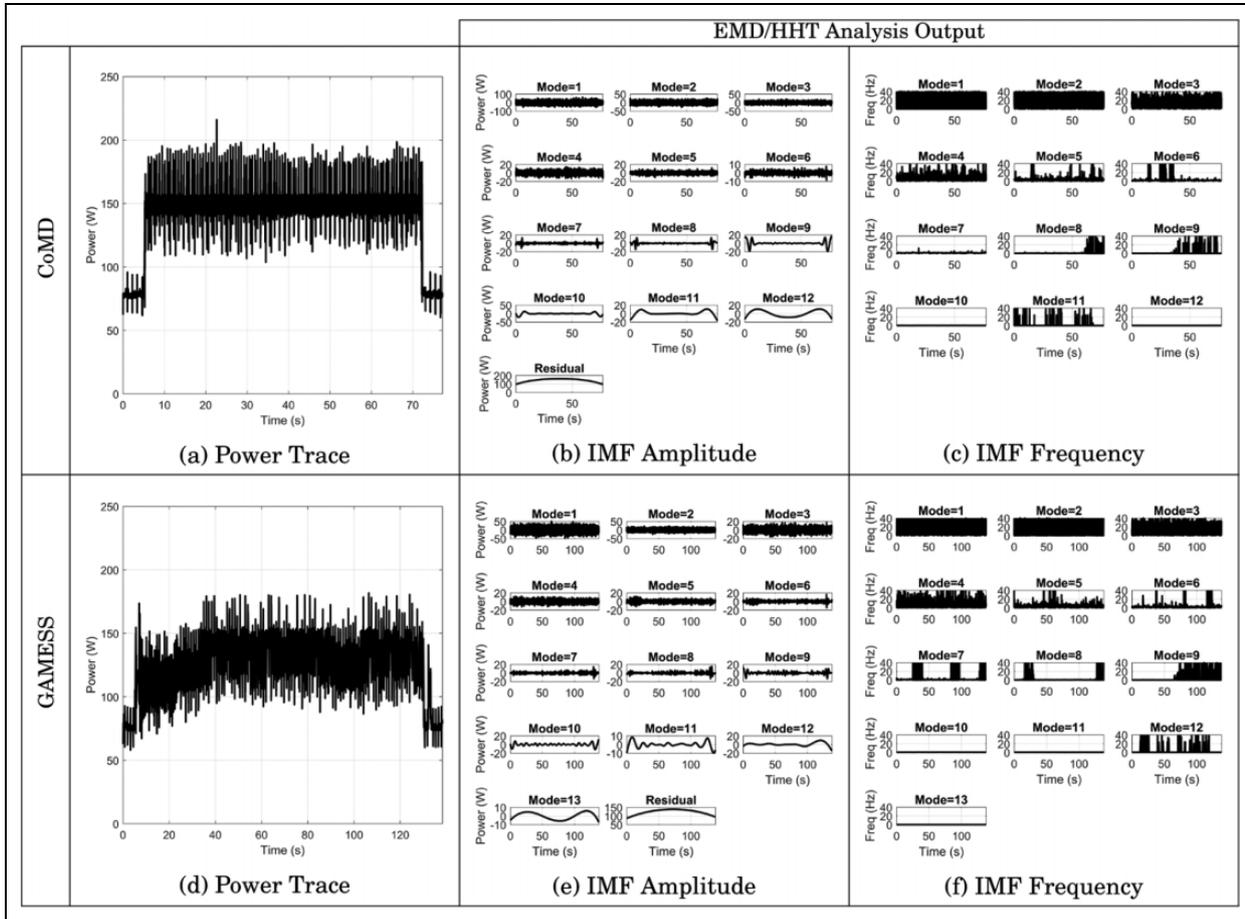


Figure 1. Illustration of EMD/HHT procedure on CoMD (top row) and GAMESS (bottom row). The original power trace ((a) and (d)) is decomposed into IMFs with respect to amplitude ((b) and (e)) and instantaneous frequency ((c) and (f)). Each trace is collected while executing CoMD or GAMESS with 59 cores at the maximum computer clock rate. EMD/HHT: empirical mode decomposition and Hilbert–Huang transform; CoMD: codesign molecular dynamics; GAMESS: general atomic and molecular electronic structure system; IMF: intrinsic mode function.

is collected using the Running Average Power Limit (RAPL) plugin (Weaver, 2011) or the Linux Power Capping Framework plugin (Linux Kernel Archives, 2016). The PowerAPI uses the hardware locality (HWLOC) API (Open MPI Project, 2016) to detect the CPU and Xeon Phi hardware. The authors have extended the functionality of the PowerAPI to identify Xeon Phi over the PCI bus and obtain power measurements. Xeon Phi power is measured using the many integrated core management (micmgmt) API released with the Intel Many Platform Software Stack (Intel, 2016); it provides access to many different measurements, such as core frequency and utilization, although only power is measured in this work.

When installing the PowerAPI library, the existence of the micmgmt API is determined since it is required for Xeon Phi power measurements. This serves as a quick check to determine whether a system may contain Xeon Phi hardware. HWLOC is then configured with the IO flag `HWLOC_TOPOLOGY_FLAG_WHOLE_IO` such that devices over the PCI bus can be detected and

is used to get the system topology. Xeon Phi is classified as a “board” in the PowerAPI. A measurement application is then created using the PowerAPI to measure CPU and Xeon Phi power at the specified sampling rate according to the measurement procedure.

A configuration is defined by the system, application, input (problem), usage mode, number of cores, clock rate, and number of nodes used for a particular run. In this work, five duplicate runs are performed for each configuration; these runs are performed one after another, with only 5 s in between. The time between executions should be longer than 1 s to provide time for the experiment scripts to properly close the previous measurement app and start the next. The measurement procedure is as follows: power measurements begin 5 s before the application is started. Upon completion of the application, an additional 5 s are allotted before the measurement is stopped. Time is allotted before and after execution to ensure the power usage of the application is completely collected and to acquire idle power measurements for all devices used.

KNC measurements are sampled at a rate of 20 ms because this is the fastest rate at which KNC power may be sampled from the host CPU. The micrgmt API requires a data handle be created for every sample, and because of this bottleneck, the sampling rate is significantly delayed. CPU measurements are sampled at a rate of 5 ms because this is the lowest rate available for all systems investigated. KNL measurements are also sampled at a rate of 5 ms because KNL uses RAPL and the Linux Power Capping Framework instead of the micrgmt API for power measurements. Ideally, a rate lower than 5 ms would have been selected, since faster sampling rates improve the output of EMD; however, modern systems only allow up to 1-ms resolution for software power measurements using RAPL.

3.2. Applications

CoMD is a proxy application developed as part of the Department of Energy codesign research effort (DOE, 2013) at the Extreme Materials at Extreme Scale (ExMatEx) center. CoMD is compute-intensive, where approximately 85–90% of the execution time is spent computing forces. In this work, the force kernel is the accurate embedded atom model (EAM) for short-range material response simulations, such as uncharged metallic materials (ExMatEx, 2012). The EAM computation consists of three compute loops and a small halo data exchange between the second and third loop which makes this an interesting kernel to investigate because computation is limited by a data transfer. Problem size is expressed as the number of copper atoms along each axis of the material, a cube in this work. For example, a problem size of 50 equates to $4 \times 50^3 = 500,000$ atoms. CoMD supports the CPU-only and KNC-native usage modes without code modifications. For KNC-native, the `-mmic` flag is required. The CPU-KNC offload usage mode was developed by

the authors, see the previous work for details on the offload usage mode (Lawson et al., 2014, 2015).

The GAMESS (Gordon and Schmidt, 2005; Schmidt et al., 1993) is a widely used quantum chemistry package capable of performing molecular structure and property calculations by a rich variety of ab initio methods finding an (approximate) solution of the Schrödinger equation for a given molecular system. The input used in this work is calculated using the second-order Møller–Plesset perturbation theory method and fragment molecular orbital approximations. The problem considered in this work is *IL2Y*, a synthetic protein tryptophan cage. GAMESS has only been tested on the KNL system in this work.

3.3. Hardware and configuration spaces

The experiment has been conducted on two multinode, heterogeneous CPU + KNC systems: *Turing* located at Old Dominion University and *Bolt* located at Ames Laboratory of Iowa State University. Additionally, the experiment has been conducted on the Intel Xeon Phi processor system *Rulfo* also located at Old Dominion University. Table 1 provides the detailed specifications for each system. Note, thermal design power is an estimate of the amount of power consumed by the device while running applications and is provided by the vendor; it is not the peak power of the device. Up to four nodes were tested on Turing, and two nodes were tested on Bolt; however, only single-node tests are considered in this work because interpreting the EMD/HHT analysis becomes more challenging as additional nodes are included. This fact is also true when considering more than one device, and so this will be further explained when discussing the EMD/HHT analysis on offload executions. For clarification, operating frequency with respect to the hardware platform is referred to as *clock rate* in this work to avoid confusion with instantaneous frequency of EMD/HHT analysis.

Table 1. Hardware characteristics and software versions for the Xeon Phi system, Rulfo, and the heterogeneous CPU and Xeon Phi systems, Bolt, and Turing.

		Rulfo	Bolt	Turing	Xeon Phi
Hardware	Microarchitecture	KNL processor	Sandy bridge	Ivy bridge	KNC coprocessor
	Model	7210	E5-1650	E5-2670 v2	5110p
	Sockets (p node)	1	1	2	1
	Clock rate (GHz)	1.3–1.0	3.2–1.2	2.5–1.2	1.053
	P-States	N/A	16	15	N/A
	Cores (p Socket)	64	6	10	60
	LL cache (MB)	32	12	25	30
	DRAM (GB)	16	64	64	8
	TDP (Watts)	215	130	115	245
	Software	Intel compiler	2016.3	2016.1.150	2016.3
MPSS		N/A	3.4.4	3.7	

KNC: Knights Corner; TDP: thermal design power; MPSS: Many Platform Software Stack; LL: Last Level.

Turing and Bolt contain Intel Xeon processors and two Intel Xeon Phi (KNC) 5110p coprocessors per node. For the Xeon Phi, in addition to the hardware parameters listed in Table 1, the device is also capable of processing four hardware threads per core. It also contains a 512-bit vector processing unit (VPU) and a fused multiply-add operation which allows the device to obtain over one TFLOPs throughput when fully loaded in double precision (8 SIMD instructions). Rulfo is a single-node Xeon Phi (KNL) system obtained through the Intel Developer Access Program (IDAP).¹ The chosen system is a Colfax KNL Ninja Liquid Cooled Pedestal Developer Platform, as listed by IDAP. The hardware parameters are listed in Table 1. In addition, each core of KNL is capable of four hardware threads and contains two 512-bit VPUs for concurrent processing. Each core is also capable of turbo, up to 1.5 GHz. KNL now supports multichannel DRAM (MCDRAM) which is stackable DDR memory. In this work, the MCDRAM is used exclusively in flat mode, which is found to improve performance and reduce energy consumption no matter the execution configuration (Lawson et al., 2016).

Each usage mode is defined by the hardware used and the manner in which the application interfaces the hardware. For a heterogeneous system with CPU and Intel Xeon Phi, there are four available modes: *CPU-only*, *KNC-only*, *symmetric*, and *CPU-KNC offload*. In this work, the symmetric mode is not investigated because workload distribution has not been balanced between CPU and KNC. The remaining usage modes are explored in this work, although only CoMD executes using the offload execution mode. For each mode, there are three parameters consistent with each configuration space: the system, application, and input (problem size). For the CPU mode, clock rate, number of nodes, and number of cores may be varied; for the KNC native mode, only the number of cores may be varied because clock rate may not be changed directly by the user (Intel, 2015). For CPU-KNC offload, all of the parameters from the CPU and KNC modes are considered, as well as the number of Xeon Phi per node. On Rulfo, the usage mode is equivalent to CPU- and KNC-only.

4. Comparisons across configurations and applications

This section presents the EMD/HHT analysis for the experiments conducted in this work. For the hardware platforms, Bolt and Turing, only the application CoMD is considered. Its configurations investigated are as follows: problem size (40, 50, and 60), clock rate (minimum, maximum, and all evenly numbered P-states), and maximum number of cores. For Rulfo, both CoMD and GAMESS are considered and their

configurations investigated are as follows: problem size, maximum clock rate, and cores numbers (32, 40, 48, 56, and 63). For CoMD, problem sizes 80 and 100 are considered; for GAMESS, the problem considered is 1L2Y. Larger problem sizes are used for CoMD on Rulfo because up to 256 threads may be allocated, and smaller problem sizes produced short power traces (less than 30 s).

Once a power trace has been analyzed using EMD/HHT, the amplitude and instantaneous frequency are combined into a 2-D histogram. Time and frequency make up the x - and y -axes, respectively, and amplitude is collected in bins and represented as *intensity* using color from blue to red for low to high, respectively. Hence, intensity is the sum of all amplitudes for a given time/frequency bin. Intensity is used to show the concentration of power draw with respect to time and frequency. The histogram uses bin sizes of 100 ms (time) and 2 Hz (frequency). A feature of these histograms is a *band*, which is a range of frequencies having a consistent intensity throughout execution.

Figure 2 presents a power trace collected on the Bolt system while running CoMD on the CPU with maximum cores and clock rate for a problem size of 50 (500,000 atoms). The power trace (a) has been analyzed using EMD/HHT to produce IMFs ((b) and (c)), which were then combined to form the 2-D histograms ((d) to (f)) of time and frequency, where intensity is the sum of all amplitudes for a given time/frequency bin. To better understand the histogram, consider Figure 2(d) to (f). In Figure 2(d), where all the IMF modes are included,² notice the moderate-to-high intensity (in yellow) from 24 Hz to 36 Hz. In Figure 2(e), which is the same as Figure 2(d) but without mode #1, the yellow band of moderate intensity has shrunk and only encompasses 24–30 Hz. Therefore, one may conclude that the first mode contains high frequency oscillations from the original trace in Figure 2(a). One step further, in Figure 2(f), the band of moderate intensity has vanished. Comparing with the IMF data shown in Figure 2(b) and (c), it is now more apparent that the “high-frequency” modes (modes 1 and 2) contain a large portion of the total power draw for CoMD. Similarly, for GAMESS, modes 1, 2, and 3 contribute the most to total power draw (see Figure 1(e)). Hence, in this way, it is possible to quantify a significant amount of power is used by high-frequency interactions. It is also of importance to note that the highest intensity is shown at frequency close to zero (see Figure 2(d)), which can be explained by static power draw or low-frequency operations, such as data I/O.

Figure 3 presents the EMD/HHT histograms generated for power traces collected by running CoMD on different systems and for different usage modes. From left to right, the first column presents the histogram on the Bolt (Figure 3(a)) and Turing (Figure 3(e)) systems.

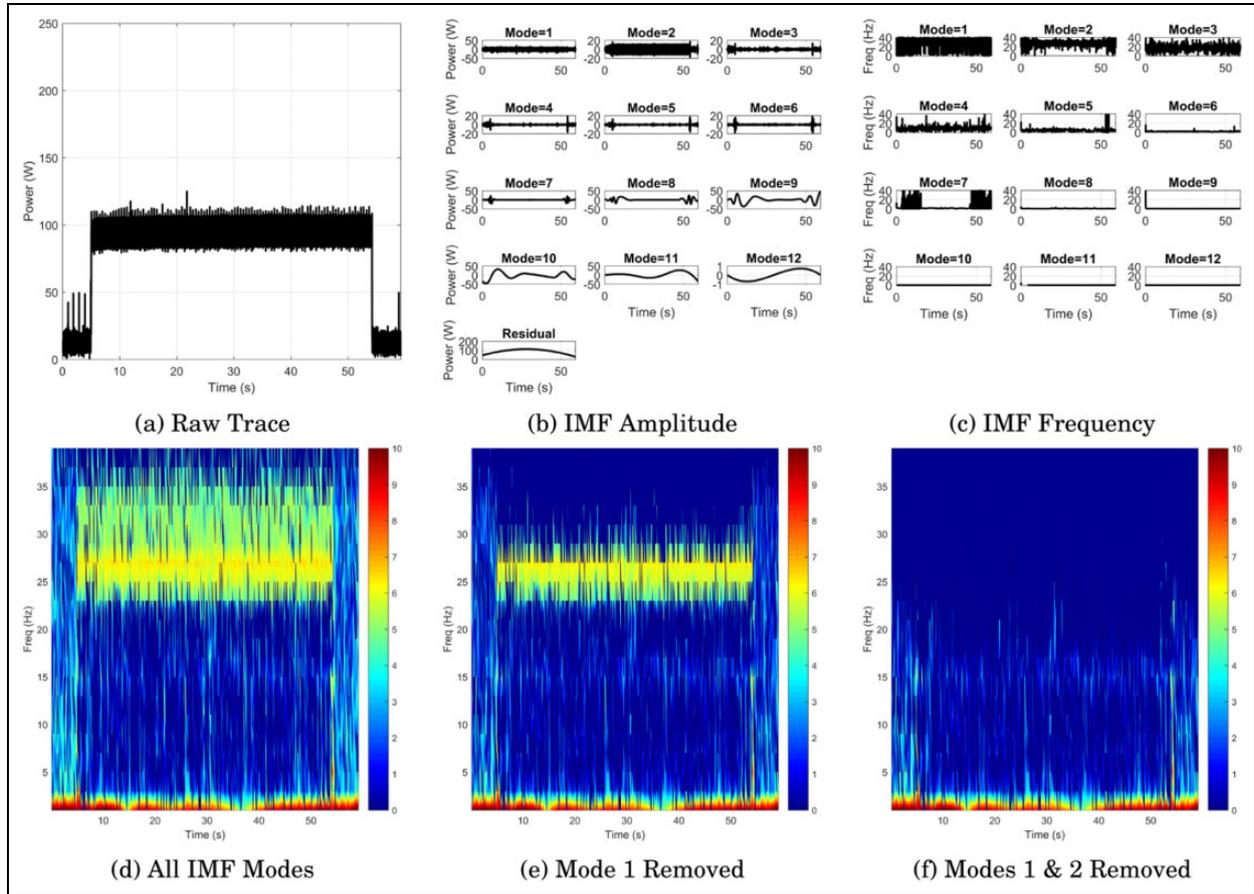


Figure 2. Illustration of EMD/HHT histograms generated using a power trace (a) collected on Bolt-CPU running CoMD-50 with maximum cores and clock rate. The EMD/HHT analysis produced IMFs shown as amplitudes (b) and frequencies (c), which were then used to generate histograms ((c) to (e)); Histogram (c) was created with all available IMF modes, (d) all modes minus mode 1, and (e) all modes minus modes 1 and 2. EMD/HHT: empirical mode decomposition and Hilbert–Huang transform; CoMD: codesign molecular dynamics; IMF: intrinsic mode function.

The following two columns present the offload histograms, with the CPU output on the left and KNC output on the right for Bolt (Figure 3(b) and (c)) and Turing (Figure 3(f) and (g)). The final column presents the histograms for the two Xeon Phi systems, KNL on Rulfo (Figure 3(d)) and KNC on Bolt (Figure 3(h)).

Comparisons of the histograms in Figure 3(a) and (e) provide insights on how the different hardware platforms respond to a similar workload—CoMD on the CPU with maximum cores and clock rate for a problem size of 50. The histogram for Bolt (Figure 3(a)) shows a concentrated band of moderate-to-high intensity (yellow) above 24 Hz, suggesting that the hardware is approaching performance bottlenecks. Specifically, an operation that occurs at 28 Hz causes high intensity throughout the Bolt execution and may be indicative of a performance bottleneck. Turing (Figure 3(e)), on the other hand, shows a moderate-to-low intensity (cyan) throughout execution, and this intensity band spans the entire spectrum from 0 Hz to 40 Hz. From such comparisons, it may be deduced that a more consistent

intensity over frequency and time suggests the application performs more optimally. Indeed, Turing is able to solve the problem almost twice as fast as Bolt thanks to having increased parallelism of 20 cores versus 6 cores on Bolt. By comparing Bolt and Turing, and the CPU and offload usage modes, the following findings may be observed. Comparing CPU executions (Figure 3(a) vs. (b) and Figure 3(e) vs. (f)), data transfer over the PCI bus can be observed. This is the critical difference between CPU-only and offload usage modes, since data must be shared between the host CPU and KNC devices. In particular, an increase in intensity is found for frequencies below 10 Hz throughout execution. Data transfer over the PCI bus is a form of I/O, which is considered low-frequency because data are often transferred in large chunks that experience varying degrees of performance. High-frequency data transfers include RAM and cache memory because these subsystems operate more frequently than PCI bus transfers. The KNC histograms (Figure 3(c) and (g)) also provide insights with the frequency limit of 10 Hz. The low

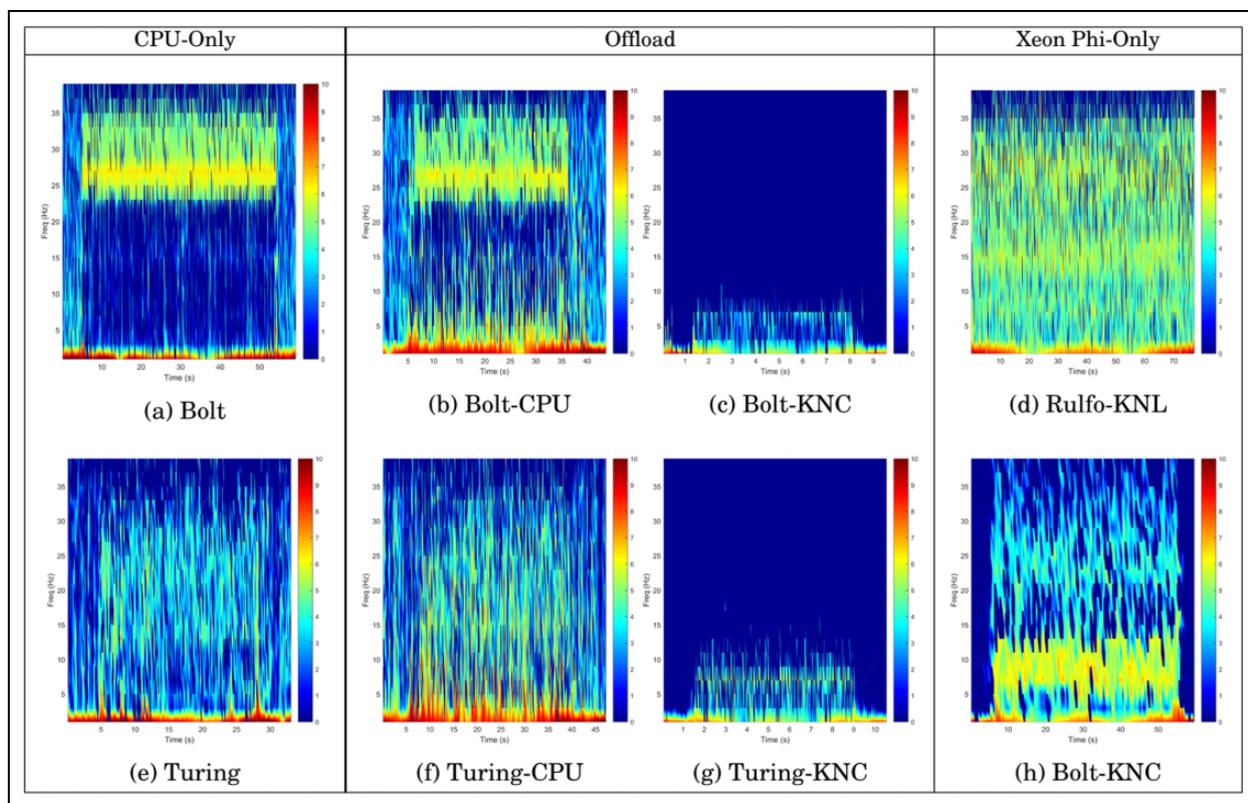


Figure 3. Comparison of EMD/HHT histograms generated for power traces collected by running CoMD on different systems and for different usage modes. From left to right, the first column presents the histogram on the Bolt (a) and Turing (e) systems. The following two columns present the offload histograms, with the CPU output on the left and KNC output on the right for Bolt ((b) and (f)) and Turing ((c) and (g)). The final column presents the histograms for the two Xeon Phi systems, KNL on Rulfo (d) and KNC on Bolt (h). EMD/HHT: empirical mode decomposition and Hilbert–Huang transform; CoMD: codesign molecular dynamics; KNC: Knights Corner.

intensity on Bolt suggests the KNC device was prone to latency due to load balance problems between the CPU and the KNC. Bolt suffers from a lack of parallelism, whereas Turing can achieve better load balancing due to the increased parallelism. Note that obtaining frequencies above 10 Hz, as in Figure 3(h), for a sampling rate of 20 ms suggests that the sampling resolution is not sufficient for EMD/HHT analysis. Figure 3(d) presents another example of an optimal execution performance, as is explained further using Figure 4.

Figure 4 presents the EMD/HHT histograms generated for power traces collected by running CoMD and GAMESS on different systems while varying the number of cores or clock rate. From left to right, the first two columns present the histograms on the Rulfo for CoMD and GAMESS with 63 cores (Figure 4(a) and (b)) and with 32 cores (Figure 4(d) and (e)). The final column presents the histograms for the Turing system with maximum clock rate (Figure 4(c)) and minimum clock rate (Figure 4(f)). Consider two numbers of cores, 63 and 32, as shown for CoMD in Figure 4(a) and (d), and for GAMESS, in Figure 4(b) and (e), respectively. For the smaller number of cores, the intensity of the trace decreased over the entire time–frequency domain.

Although this is an expected behavior, the histograms are telling because they show that the processor power draw impacts at all frequencies. In particular, CoMD is a compute-intensive application that achieves optimal performance with the maximum number of cores. The intensity for the maximum number of cores is moderate, and for the minimum number of cores, the intensity is moderate-to-low; factoring time-to-solution with this difference, it is apparent that a moderate intensity coincides with the more optimal execution. It has been observed earlier (Lawson et al., 2017) that GAMESS is a memory-intensive application that achieves optimal performance with half of the maximum number of cores on Rulfo because of the limited L2 cache size (32 MB for 64 cores). Indeed, for GAMESS, a moderate intensity is seen in the 32-core trace (Figure 4(e)) while the plot for 63-core trace exhibits high intensity, notably between 50 and 200 s (Figure 4(b)). Such a high intensity for larger frequencies suggests that performance bottlenecks have been encountered by the execution.

Figure 4(c) and (f) presents comparisons of the maximum and minimum clock rate (P-state) for Turing. Similarly to decreasing the number of cores, smaller

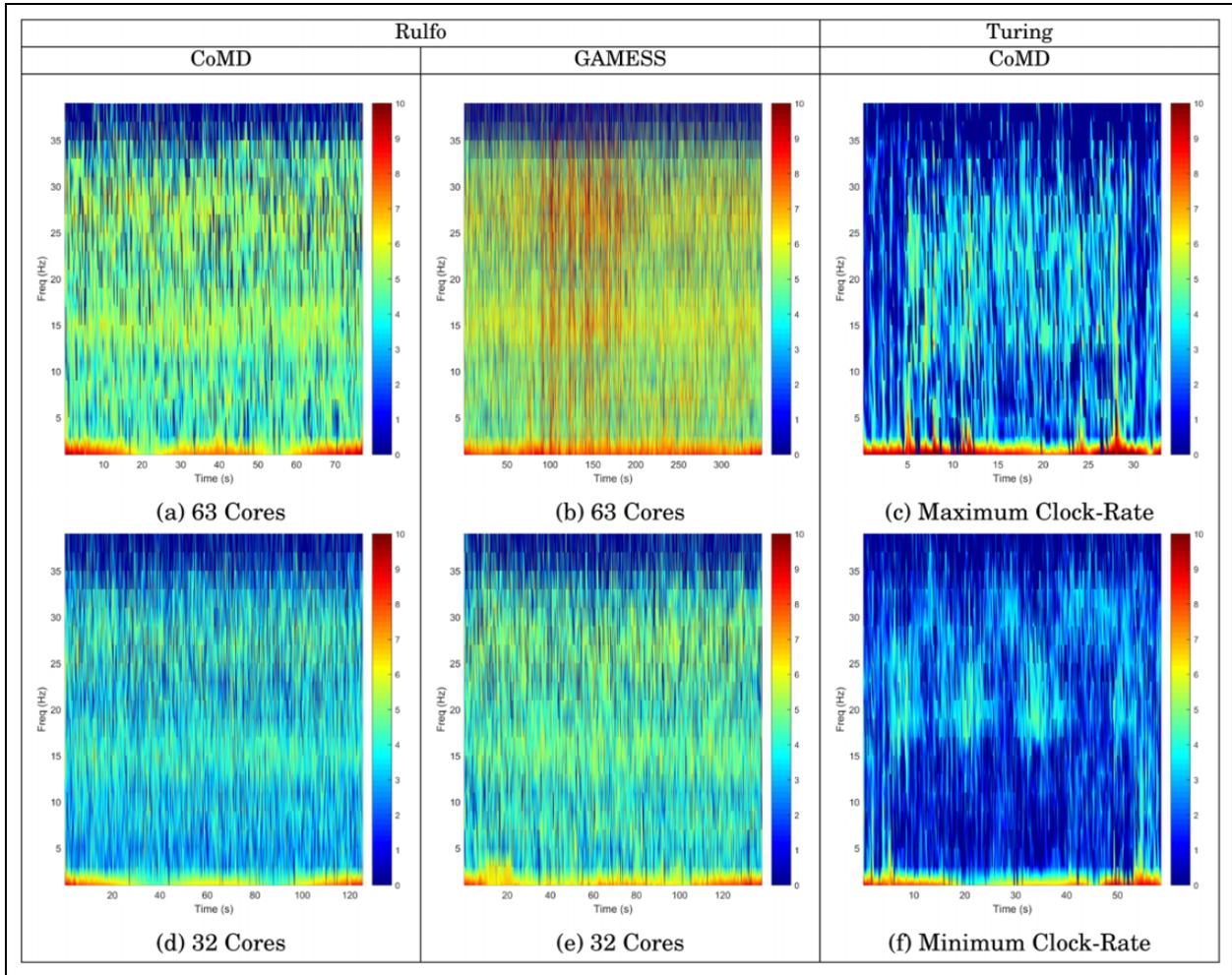


Figure 4. Comparison of EMD/HHT histograms generated for power traces collected by running CoMD and GAMESS on different systems while varying the number of cores or clock rate. From left to right, the first two columns present the histograms on the Rulfo for CoMD and GAMESS with 63 cores ((a) and (b)) and with 32 cores ((d) and (e)). The final column presents the histograms for the Turing system with maximum clock rate (c) and minimum clock rate (f). EMD/HHT: empirical mode decomposition and Hilbert–Huang transform; CoMD: codesign molecular dynamics; GAMESS: general atomic and molecular electronic structure system.

clock rate reduces the intensity across the entire time–frequency domain. A smaller clock rate, however, did not impact the bands found in the Turing trace for frequencies from 18 Hz to 34 Hz.

5. Conclusions

The EMD/HHT analysis method has been applied to the power traces collected on several hardware platforms, featuring KNC and KNL accelerators, and two different realistic applications. The applications considered are the compute-intensive CoMD and memory-intensive GAMESS. Power traces were collected using the PowerAPI, which has been extended by the authors to measure Xeon Phi power.

Using EMD/HHT analysis, hardware utilization can be broadly classified based on the overall intensity of the resulting histogram. Intensity can be roughly

interpreted as a representation of operations performed by the hardware, where computations and data movement may have a measurable relationship to instantaneous frequency. It was shown that varying clock rate or the number of cores impacts the entire time–frequency domain of the EMD/HHT analysis. Bands, a feature of EMD/HHT histograms, may represent memory subsystems, although further investigation is required to clarify this result.

At this point, it is clear that intensity and instantaneous frequency are related to workload characteristics, such as computation, data movement, I/O, and communication. However, this relationship is highly dependent on the underlying hardware. Selecting an optimal execution configuration based on the EMD/HHT analysis procedure outlined in this article is left as future work. Other future work directions include reducing the amount of trace information required for

the EMD/HHT analysis and investigating predictive capabilities of the method based on the final residual representation of the trace.

Acknowledgement

The authors would like to thank the editor and reviewers for their time, consideration, and feedback on this work.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the Air Force Office of Scientific Research under the AFOSR award FA9550-12-1-0476, by the National Science Foundation grants 0904782, 1047772, and 1516096; by the U.S. Department of Energy, Office of Advanced Scientific Computing Research, through the Ames Laboratory, operated by Iowa State University under contract no DE-AC02-07CH11358; and by the U.S. Department of Defense High Performance Computing Modernization Program, through an HASI grant. This work was also supported by the high-performance computing system at Ames Laboratory of Iowa State University (Bolt) and the system at Old Dominion University (Turing).

Notes

1. <https://software.intel.com/en-us/articles/developer-access-program-for-intel-xeon-phi-processor-codenamed-knights-landing>.
2. The residual trend is not considered an IMF.

References

Abdurachmanov D, Bockelman B, Elmer P, et al. (2014) Heterogeneous high throughput scientific computing with APM X-Gene and Intel Xeon Phi. *CoRR* abs/1410.3441 (2014). Available at: <http://arxiv.org/abs/1410.3441> (accessed 7 August 2016).

Aprà E, Klemm M and Kowalski K (2014) Efficient implementation of many-body quantum chemical methods on the intel® Xeon Phi™ coprocessor. In: *Proceedings of the international conference for high performance computing, networking, storage and analysis (SC '14)*, New Orleans, LA, USA, 16–21 November 2014, pp. 674–684. IEEE Press, Piscataway, NJ, USA. DOI: 10.1109/SC.2014.60.

Bernaschi M and Salvadore F (2014) Multi-Kepler GPU vs. multi-intel MIC: a two test case performance study. In: *2014 International conference on high performance computing simulation (HPCS)*, Bologna, Italy, pp. 1–8. IEEE DOI: 10.1109/HPCSim. 2014.6903662.

Bernaschi M, Bisson M and Salvadore F (2014) Multi-Kepler GPU vs. multi-Intel MIC for spin systems simulations. *Computer Physics Communications* 185(10): 2495–2503. DOI: 10.1016/j.cpc.2014.05.026

Brown W, Carrillo J, Gavhane N, et al. (2015) Optimizing legacy molecular dynamics software with directive-based offload. *Computer Physics Communications* 195: 95–101. DOI: 10.1016/j.cpc.2015.05.004

Choi J, Mukhan M, Liu X, et al. (2014) Algorithmic time, energy, and power on candidate HPC compute building blocks. In: *2014 IEEE 28th international symposium on parallel distributed processing (IPDPS)*. Phoenix, Arizona, USA. IEEE.

DOE (2013) Co-Design. (2013). Available at: <http://science.energy.gov/ascr/research/scidac/co-design/>. (accessed 7 August 2016).

ExMatEx (2012) CoMD Proxy Application. (2012). Available at: <http://www.exmatex.org/comd.html>. (accessed 7 August 2016).

Ezer T (2015) EEMD/HHT. (2015). Available at: <http://www.ccpo.odu.edu/tezer/HHT/>. (accessed 7 August 2016).

Ezer T and Corlett W (2012) Is sea level rise accelerating in the Chesapeake Bay? A demonstration of a novel new approach for analyzing sea level data. *Geophysical Research Letters* 39(19). DOI: 10.1029/2012GL053435.

Ezer T, Atkinson LP, Corlett WB, et al. (2013) Gulf Stream's induced sea level rise and variability along the U.S. mid-Atlantic coast. *Journal of Geophysical Research: Oceans* 118(2): 685–697. DOI: 10.1002/jgrc.20091

Gordon MS and Schmidt MW (2005) Advances in Electronic Structure Theory: GAMESS a Decade Later. In: Dykstra CE, Frenking G, Kim KS and Scuseria GE (ed) *Theory and Applications of Computational Chemistry: the first forty years*, pp. 1167–1189.

Heinecke A, Vaidyanathan K, Smelyanskiy M, et al. (2013) Design and implementation of the linpack benchmark for single and multi-node systems based on intel xeon phi coprocessor. In: *2013 IEEE 27th international symposium on parallel distributed processing (IPDPS)*, Boston, MA, USA, pp. 126–137. IEEE.

Höhnerbach M, Ismail AE and Bientinesi P (2016) The Vectorization of the Tersoff Multi-Body Potential: An Exercise in Performance Portability. *ArXiv e-prints* (July 2016). abs/1607.02904.

Huang N, Shen Z, Long S, et al. (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc R Soc A: Math Phys Eng Sci* 454(1971): 903–995. DOI: 10.1098/rspa.1998.0193

Intel (2015) Intel Xeon Phi Coprocessor: Datasheet. Available at: <http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-coprocessor-datasheet.html>. (accessed 7 August 2016).

Intel (2016) Intel Many Platform Software Stack (Intel MPSS). Available at: <https://software.intel.com/en-us/articles/intel-manycore-platform-software-stack-mpss>. (accessed 7 August 2016).

Jundt A, Tiwari A, Ward W Jr, et al. (2015) Optimizing Codes on the Xeon Phi: A Case-study with LAMMPS. In: *Proceedings of the 2015 XSEDE conference: scientific*

- advancements enabled by enhanced cyberinfrastructure (XSEDE '15)*. ACM, New York, NY, USA, Article 28, p. 2. DOI: 10.1145/2792745.2792773.
- Krishnaiyer R, Kultursay E, Chawla P, et al. (2013) Compiler-based data prefetching and streaming non-temporal store generation for the Intel Xeon Phi coprocessor. In: *2013 IEEE 27th International Symposium on Parallel Distributed Processing (IPDPS)*, Boston, MA, USA, pp. 1575–1586. IEEE.
- Kusnezov D, Binkley S, Harrod B, et al. (2013) DOE Exascale Initiative. Available at: <http://www.industry-academia.org/download/20130913-SEAB-DOE-Exascale-Initiative.pdf>. (accessed 7 August 2016).
- Lai C, Hao Z, Huang M, et al. (2014) Comparison of parallel programming models on intel mic computer cluster. In: *2014 IEEE international parallel distributed processing symposium workshops (IPDPSW)*, Phoenix, AZ, USA, pp. 925–932. DOI: 10.1109/IPDPSW.2014.105. IEEE.
- LaKomski D, Zong Z, Jin T, et al. (2015) Optimal balance between energy and performance in hybrid computing applications. In: *2015 Sixth international green computing conference and sustainable computing conference (IGSC)*, Las Vegas, NV, USA, p. 1–8. DOI: 10.1109/IGCC.2015.7393697.
- Laros J (2016) Sandia national laboratories high performance computing power application programming interface (API) specification. (2016). Available at: <http://powerapi.sandia.gov/>. (accessed 7 August 2016).
- Lawson G, Sosonkina M and Shen Y (2014) Performance and energy evaluation of CoMD on Intel Xeon Phi coprocessors. In: *Proceedings of the 1st international workshop on hardware-software co-design for high performance computing (Co-HPC '14)*, New Orleans, LA, USA, Piscataway, NJ, USA: pp. 49–54. DOI: 10.1109/Co-HPC.2014.12.
- Lawson G, Sosonkina M, Ezer T, et al. (2017) Empirical mode decomposition for modeling of parallel applications on intel xeon phi processors. In: *2nd International Workshop on Theoretical Approaches to Performance Evaluation, Modeling and Simulation (TAPEMS)*. Madrid, Spain, 18 September 2017, pp. 1000–1008. IEEE.
- Lawson G, Sundriyal V, Sosonkina M, et al. (2015) Modeling performance and energy for applications offloaded to Intel Xeon Phi. In: *Proceedings of the 2nd international workshop on hardware-software co-design for high performance computing (Co-HPC '15)*. Austin, TX, USA, ACM, New York, NY, USA, Article 7, 7:1–7:8. DOI: 10.1145/2834899.2834903.
- Lawson G, Sundriyal V, Sosonkina M, et al. (2016) Runtime power limiting of parallel applications on Intel Xeon Phi processors. In: *Proceedings of the 4th International Workshop on Energy Efficient Supercomputing (E2SC '16)*, Salt Lake City, UT, USA, pp. 39–45. IEEE Press, Piscataway, NJ, USA. DOI: 10.1109/E2SC.2016.9.
- Li B, Chang HC, Song S, et al. (2014) The power-performance tradeoffs of the Intel Xeon Phi on HPC applications. In: *2014 IEEE international parallel distributed processing symposium workshops (IPDPSW)*, Phoenix, AZ, USA, pp. 1448–1456. DOI: 10.1109/IPDPSW.2014.162. ACM.
- Linux Kernel Archives (2016). Linux Power Capping Framework. (2016). Available at: <https://www.kernel.org/doc/Documentation/power/powercap/powercap.txt>. (accessed 7 August 2016)
- Liu X, Peng S, Yang C, et al. (2015) mAMBER: Accelerating Explicit Solvent Molecular Dynamic with Intel Xeon Phi Many-Integrated Core Coprocessors. In: *2015 15th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid)*, Shenzhen, China, pp. 729–732. DOI: 10.1109/CCGrid.2015.66. IEEE.
- Lopez MG, Young J, Meredith JS, et al. (2015) Examining recent many-core architectures and programming models using SHOC. In: *Proceedings of the 6th international workshop on performance modeling, benchmarking, and simulation of high performance computing systems (PMBS '15)*. Austin, TX, USA, ACM, New York, NY, USA, Article 3, p. 12. DOI: 10.1145/2832087.2832090.
- Luo M, Li M, Venkatesh A, et al. (2013) UPC on MIC: Early experiences with native and symmetric modes. In: *Proceedings of the 7th international conference on PGAS programming models* (2013). Ohio State University.
- Mathew B, Rai N, Gupta A, et al. (2015) Exploiting computing power of xeon and intel xeon phi for a molecular dynamics application. In: *Proceedings of the symposium on high performance computing (HPC '15) society for computer simulation international*, Alexandria, VA, USA, pp. 9–16. San Diego, CA, USA. <http://dl.acm.org/citation.cfm?id=2872599.2872601>.
- Misra G, Kurkure N, Das A, et al. (2013) Evaluation of rodinia codes on Intel Xeon Phi. In: *2013 4th international conference on intelligent systems, modelling and simulation*, Bangkok, Thailand, 7 August 2016, pp. 415–419. DOI: 10.1109/ISMS.2013.118. IEEE.
- Newburn CJ, Dmitriev S, Narayanaswamy R, et al. (2013) Offload compiler runtime for the Intel Xeon Phi Coprocessor. In: *Proceedings of the 2013 IEEE 27th international symposium on parallel and distributed processing workshops and PhD forum (IPDPSW '13)*, pp. 1213–1225. IEEE Computer Society, Washington, DC, USA. DOI: 10.1109/IPDPSW.2013.251.
- Open MPI Project (2016) Portable Hardware Locality (HWLOC). (2016). Available at: <https://www.open-mpi.org/projects/hwloc/>. (accessed 7 August 2016).
- Park J, Bikshandi G, Vaidyanathan K, et al. (2013) Tera-scale 1D FFT with Low-communication algorithm and intel xeon phi coprocessors. In: *Proceedings of the international conference on high performance computing, networking, storage and analysis (SC '13)*, Denver, CO, USA, ACM, New York, NY, USA, Article 34, p. 12. DOI: 10.1145/2503210.2503242.
- Saini S, Jin H, Jespersen D, et al. (2015) Early multi-node performance evaluation of a knights corner (KNC) based NASA supercomputer. In: *2015 IEEE international parallel and distributed processing symposium workshop (IPDPSW)*, Hyderabad, India, pp. 57–67. DOI: 10.1109/IPDPSW.2015.140.
- Sainz F, Belln J, Beltran V, et al. (2015) Collective offload for heterogeneous clusters. In: *2015 IEEE 22nd international conference on high performance computing (HiPC)*, Bangalore, India, pp. 376–385. DOI: 10.1109/HiPC.2015.20.
- Saule E, Kaya K and Çatalyürek Ü (2014) *Performance Evaluation of Sparse Matrix Multiplication Kernels on Intel*

- Xeon Phi*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 559–570. DOI: 10.1007/978-3-642-55224-3_52
- Schmidt MW, Baldrige KK, Boatz JA, et al. (1993) General atomic and molecular electronic structure system. *Journal of Computational Chemistry* 14(11): 1347–1363. DOI: 10.1002/jcc.540141112
- Sodani A, Gramunt R, Corbal J, et al. (2016) Knights landing: Second-generation Intel Xeon Phi Product. *IEEE Micro* 36(2): 34–46. DOI: 10.1109/MM.2016.25
- Teodoro G, Kurc T, Kong J, et al. (2014) Comparative performance analysis of Intel (R) Xeon Phi (TM), GPU, and CPU: a case study from microscopy image analysis. In: *2014 IEEE 28th international parallel and distributed processing symposium*, Phoenix, AZ, USA, pp. 1063–1072. DOI: 10.1109/IPDPS.2014.111.
- Weaver V (2011) Reading RAPL energy measurements from Linux. (2011). Available at: <http://web.eece.maine.edu/vweaver/projects/rapl/>. (accessed 7 August 2016).
- Wood J, Zong Z, Gu Q, et al. (2014) Energy and power characterization of parallel programs running on Intel Xeon Phi. In: *2014 43rd international conference on parallel processing workshops*, Minneapolis, MN, USA. pp. 265–272. DOI: 10.1109/ICPPW.2014.43.
- Wu Z and Huang N (2009) Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in Adaptive Data Analysis* 1(1): 1–41. DOI: 10.1142/S1793536909000047

Author biographies

Gary Lawson is a PhD candidate of the Modeling, Simulation and Visualization Engineering Department at Old

Dominion University. His research interests include high-performance computing, energy and performance analysis, visualization, and computer graphics. His most recent work focuses on power trace modeling and analysis for large-scale simulations.

Masha Sosonkina is a professor of Modeling, Simulation and Visualization Engineering Department at Old Dominion University. Her research interests include high-performance computing, large-scale simulations, energy and performance analysis, and adaptive algorithms.

Tal Ezer is a professor of ocean earth and atmospheric sciences at Old Dominion University. His research interests include numerical modeling of ocean circulation, physical processes in the ocean, and data analysis; in recent years, his research focused on analysis of climate change and sea level rise and their impact.

Yuzhong Shen is an associate professor and graduate program director of the Department of Modeling, Simulation, and Visualization Engineering at Old Dominion University. His research interests include visualization and computer graphics, modeling and simulation, and signal and image processing.