A Region-Driven Analysis of Processor Power Consumption for Various HPC Workloads

Sharda Dixit[#1], Amarjeet Sharma[#2], Anil Gupta[#3]
[#]Center for Development of Advanced Computing(C-DAC)
Pune, India

Abstract

Power consumption is one of the significant challenges for present and future HPC systems. The processor is one of the most power-hungry components at the node level of an HPC cluster. Various features are already implemented in processor hardware and OS/kernel, which optimize power consumption at the node level. We performed several experiments on a variety of workloads and observed power consumption patterns for CPU, memory and IO intensive regions of workloads. The investigations focused on correlating the CPU utilization, memory access, idle state residency, clock gated residency, and power consumption. Our results show that the scope of reducing energy consumed by workload, using employing frequency scaling, for compute-intensive workloads, is slim at the system software level.

Keywords - *DVFS, Clock-Gating, C-State, P-State.*

I. INTRODUCTION

Power consumption is one of the major challenges of Exascale computing [1] [2] [3] [4] [5] [6]. While the US DOE has set an upper limit of 20MW for Exascale systems [7], modern day super computers are already consuming about 15 MW power for delivering a peak performance of 93 PF (Sunway Taihulight) [8]. Meeting the target of 20MW requires coordinated efforts of reducing power consumption at various levels of the HPC facilities. In the absence of circuits, architecture, power conversion, power delivery and cooling technologies which are far more energy-efficient, the cost of ownership for exascale systems could be as much as ten times higher than today [7].

Our initial focus was to explore and utilize techniques for reducing processor power consumption at the node level. Literature survey suggests that memory-bound and I/O-bound phases of workloads present the opportunity of lowering processor frequency to gain significant power savings without any significant impact on performance. We started analyzing the power consumption behavior of a variety of HPC benchmarks and applications and noticed that a bulk of optimization mechanisms for non-compute intensive regions of workload are already implemented in the processor hardware. In this paper, we present the details of our findings followed by some recommendations.

The outline of the rest of the paper is as follows. Section II provides a brief overview of related work in this domain. Section III mentions the details of our experiments including test-bed, workloads, and utilities used. Section IV provides the details of our findings and analysis. Conclusion and recommendations are discussed in section V followed by future work in section VI.

.

II. RELATED WORK

A significant amount of work has been done in the direction of measurement, control, and optimization of CPU power consumption at the node level and cluster level for a variety of HPC workloads in the form of applications and benchmarks. One of the prominent approaches at the node level is to identify regions of the workload which are not computed intensive and reduce the processor operating frequency during these regions to save power.

The sweet spots in most of the HPC applications have been claimed [?] Where significant energy reduction can be achieved with an acceptable impact on runtime, even while using the static frequency scaling. A large amount of experimental data has been provided to support this claim. In [?], exercise has been repeated for several workloads at P-states 1-4. Up to 39.5% of energy reduction has been shown with 0.402% increase in runtime.

An optimal processor-DRAM frequency pair is modeled in to minimize the energy consumption within a given timeline. The proposed system, demonstrated on several benchmarks, shows total energy savings up to 22% for memory and compute-intensive applications with a performance penalty of about 4.8%. The workloads were executed at the highest and lowest processor frequencies with the same memory frequency to validate the variation in processor frequency. Same workloads were executed with highest and lowest memory frequencies with fixed processor frequency to validate the variation in memory frequencies. The proposed system selects the appropriate processor memory frequency pairs that show significant energy savings with a minimal performance-loss.

Some frameworks [11] [12] allow applications to communicate region hints, such as compute-bound, memory-bound, and communication-bound regions.

These region hints, along with the efficient resource allocation, also enable the processor frequency to be scaled down during non-compute intensive regions with little impact on performance.

An open source simulator, an extension to the Work-FlowSim, for energy optimization's in cloud computing has been presented in [13] to incorporate a power model. The performance for NASA's Montage, Sipht, and Inspiral projects has been evaluated on five DVFS governors. The energy savings with OnDemand governor and significant performance improvement has been shown with DVFS based scheduling strategy. The intra-host DVFS combined with inter-host DVFS scheduling is claimed to be more energy efficient.

A network utilization-based model and algorithm of a decision process reducing energy consumption has been presented in [14]. With [14], NAS benchmarks, the energy savings up to 25% with performance degradation less than 5% has been shown.

An online or runtime methodology for energy performance improvements has been introduced in [15]. It considers three HPC subsystems: processor, disk and interconnect. Energy savings are achieved with a frequency scaling-based system adaption policy using partial phase recognition of real-life workloads.

A frequency-scaling strategy, called DVFS-PhIT has been proposed in [16]. It detects the communication phases in parallel applications and calculates an optimal DVFS level within the phases and inter-phase time gaps. It also selects a proper CPU throttling level for the detected communication phases. A maximum of 14% energy saving with a performance loss of about 2% has been shown.

The effect of voltage and frequency scaling on performance and energy has been studied in [17] on Intel Haswell and ARMv7 architectures. The results show significant energy saving on ARM architectures with optimal frequency compared to the maximum one. Intel processors are taking benefits from the frequency scaling already but the energy saving is not so significant due to DRAM modules and lower granularity on the frequency range.

.

III. EXPERIMENTAL ENVIRONMENT

This paragraph introduces the platform and systems used for experimentation purposes.

A. *Platform Used*

Experiments were conducted on an Intel S2600GZ server with 2x Intel Xeon CPU E5-2670 in a dual socket configuration along with 56GB of DDR3 RAM. Each CPU has eight physical cores with two threads in each core leading to a total of 16 logical CPUs per processor and 32 logical CPUs in total.

B. *Workloads*

1) LINPACK: The standard LINPACK (HPL) [18] for Intel 64-bit architecture is set up, and all other environment settings are kept at default to the underlying architecture. The LINPACK used, is compiled using Intel MKL Library [19]. Total 27 trial runs of 15 tests are performed with problem sizes ranging from 1000 to 45000.

2) STREAM: The STREAM [20] benchmark is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels. The STREAM ARRAY SIZE and NTIMES parameters are set to 60000000 and 1000 respectively.

3) Stress-ng: Stress-ng [21] is a suite designed to stress various physical subsystems of a computer as well as the different operating system kernel interfaces through the given stressors. Three stressors namely CPU, Cache, and AIO are used sequentially in the experiments to stress the processor, memory and i/o subsystems respectively.

4) LAMMPS: The LAMMPS 2018 stable version [22] is compiled for performing a parallel simulation of 300 aluminum crystal atom. A customized input script optimized for the platform mentioned above is provided as input for simulations. The simulation environment is kept to default wherever possible.

5) WRF: WRF 3.9.0.1 [23] is compiled on 64-bit intel architecture in parallel mode (dmpar) to perform a simulation of weather data of 12 hours for a single domain. WRF environment settings are kept at default.

6) GROMACS: The GROMACS [24] 2018 stable version is set up for performing a simulation of Lysozyme protein. The input data for simulation is taken from standard protein data bank in PDB format. The execution environment is maintained at default settings concerning the platform used.

C. *Utilities used*

1) Running Average Power Limits: Intel RAPL [25] provides a set of counters providing energy and power consumption information. We have used RAPL for power measurements of package, core, and DRAM.

2) Performance Counter Monitor: Intel Performance Counter Monitor (PCM) [26] is a tool to monitor performance and hardware counters on Intel processors. Intel PCM-Memory utility (a subset of PCM) is used in our experiments for measurement of system memory throughput (sum of read and write throughput) in MB/s.

3) PowerTop: PowerTop [27] is a Linux utility to measure and modify settings related to power consumption and management issues. It is used for measurement of C-state (C3, C6, and C7) residencies in the experiments.

4) ProcStat: Linux /proc/stat file provides various pieces of information about kernel and system activity. We have used the Linux /proc/stat interface to obtain CPU percentage utilization and idle percentage residency.

Note: *The required code segments from the above utilities have been combined into a single custom utility to achieve time synchronization across all the values measured in the experiments.*

IV. RESULTS AND ANALYSIS

This work aimed to investigate the relationship between the following parameters:

1) Processor power consumption (Package domain in RAPL)
2) CPU utilization
3) Idle state residency
4) Clock gated residency (C3, C6 and C7 states for Intel Sandy Bridge E5-2670 [28])
5) DRAM access bandwidth
6) DRAM power consumption (DRAM domain in RAPL)

All the measurements combine values from package-0 and package-1 (because of our dual socket setup). Correlation coefficient mentioned anywhere refers to Pearson correlation coefficient. Energy is calculated as the area under the power vs. time graph. ACPI-cpufreq driver has been used for all the experiments. Power and energy are also consumed when the processor is not executing any workload. This idle component of power and energy is reasonably constant and has not been removed from the power or energy readings of the different workloads mentioned below. Hyper-Threading was enabled for all the experiments other than where explicitly mentioned.

A. *Workload profile with default settings*

For this section, the frequency scaling governor was "conservative" for all the workloads. It scales the processor frequency dynamically according to workload requirements.

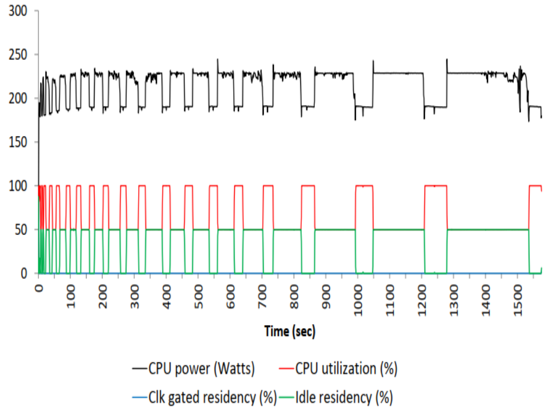

Fig. 1. LINPACK CPU Profile

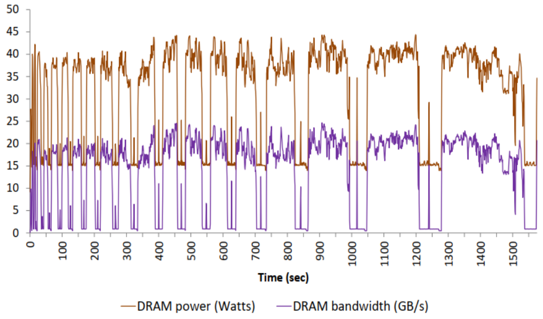

Fig. 2. LINPACK DRAM Profile

Linpack, a highly CPU intensive benchmark, the CPU utilization peaks (Fig. 1) at 100% but does not stay at 100% throughout the execution. Correspondingly, Idle residency graph is the opposite of the CPU utilization profile, and clock gated residency was almost invariably near 0. The observation that CPU utilization does not stay at 100% is rather interesting, and on further investigation, we found that the Intel MKL computational routines used by the HPL benchmark obtain the best performance using one thread per physical core.

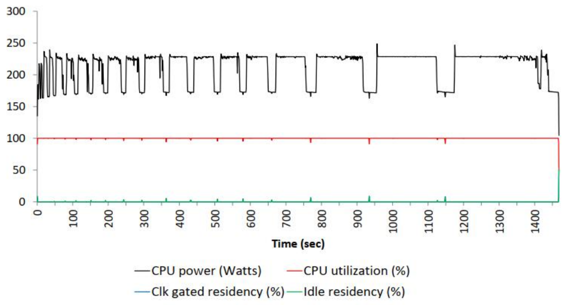

Fig. 3. LINPACK CPU Profile with Hyper-Threading disabled

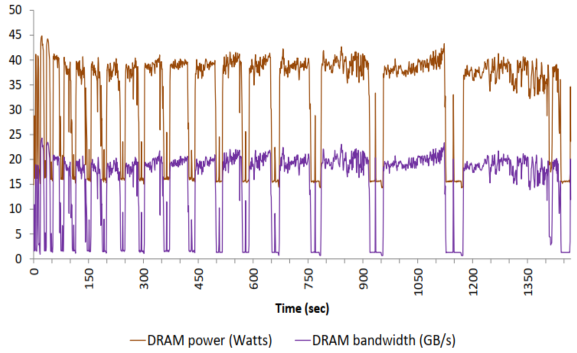

Fig. 4. LINPACK DRAM Profile with Hyper-Threading disabled

We repeated the same experiment with hyper-threading dis-abled and obtained the profile as shown in Fig. Three which shows the CPU utilization at an almost constant 100% with other values mostly unchanged. In our case, 15 different problem sizes were computed by Linpack. This corresponds to 15 peak regions in the CPU power consumption, DRAM access bandwidth and DRAM power consumption profiles (Fig. 2 and Fig. 4). Peak power consumption (229 W) almost maxes out to the TDP (230W) with occasional peaks of 245W (due to Intel TurboBoost feature) which exceed the TDP. Fig. (1 and 3) clearly shows that the processor hardware itself scales down its power consumption during periods of comparatively low activity.

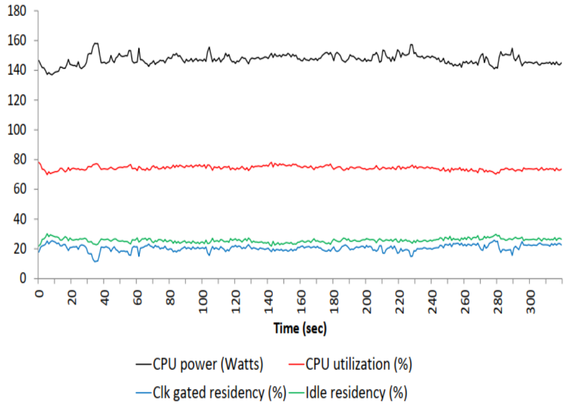

Fig. 5. STREAM CPU Profile

For the STREAM benchmark (Fig. 5), the CPU utilization was below 80% throughout execution as it is highly memory intensive. Correspondingly, Idle residency was above 20%, and it was closely followed by the clock gated residency (best case 26% and worst case 11%) as seen in Fig. 5. It shows that a majority of idle state residency was clock gated (due to features like Enhanced Intel SpeedStep Technology) leading to maximized power savings.

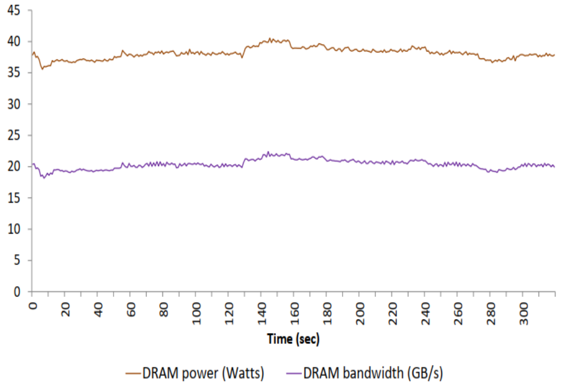

Fig. 6. STREAM DRAM Profile

DRAM access bandwidth stayed at 20 Gbps throughout execution (Fig. 6). This workload represents a highly memory intensive scenario which allows the processor sufficient scope to reduce its power consumption by taking advantage of clock gating.

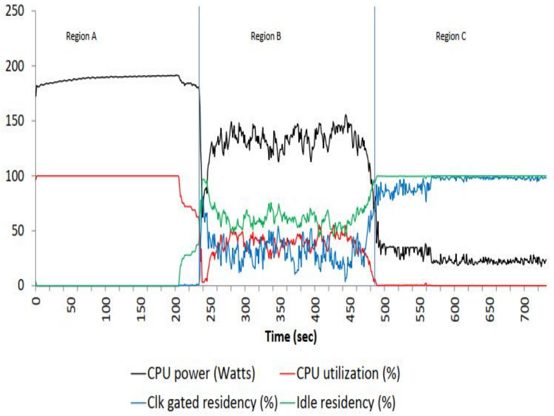

Fig. 7. Stress-ng CPU Profile

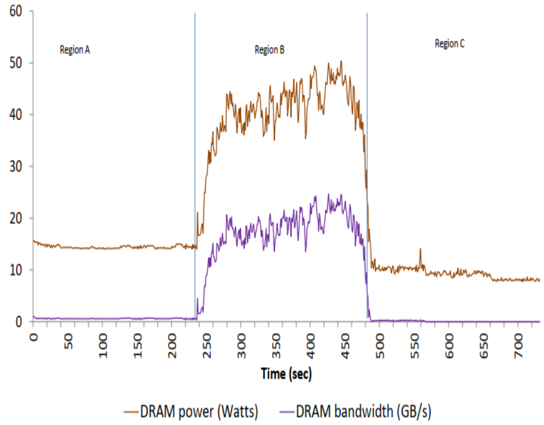

Fig. 8. Stress-ng DRAM Profile

Next, we used the stress-ng utility with three different types of stressors in the sequence of cpu-intensive (–cpu stressor), memory-intensive (–cache stressor) and I/O intensive (–aio stressor) modules (Fig. 7 & Fig. 8). During the CPU-intensive region (region A), the CPU utilization was almost stable at 100%. Correspondingly, Idle% and clock gated% were almost constantly near 0. CPU power consumption varies according to the particular CPU stress methods used by the utility. DRAM access shows minor variations due to changes in the CPU stress methods and the DRAM power consumption was stable at 14 -15W (Fig. 8).

For the memory intensive part (region B), stress-ng performs random and widespread memory reads and writes to thrash the CPU cache. CPU utilization varies from 3% to over 50%, and processor power consumption follows the same trend closely as seen in the graph (Fig. 7). The profile for idle state residency is almost a mirror image of CPU utilization, and a significant part of the idle state residency is clock gated (best case 82% and worst case 4%). If frequency scaling were performed, it would have a next-to-no effect on the clock gated cores. The DRAM access bandwidth shows a good amount of variation with a peak value of ˜24GB/s and the DRAM power consumption follows the same profile as DRAM access bandwidth (Fig. 8).

For the I/O intensive region (region C), CPU utilization is very low and idle state residency is very high. As a result, processor power consumption is also low. As is visible from the graph (Fig. 7), clock gated residency is quite high (best case 99% and worst case 70%). DRAM access and power consumption remain relatively stable (***Fig. 8***). Regions B and C of ***Fig. 7*** highlight the fact that the power consumption of the processor closely follows the CPU utilization. For durations of low CPU utilization, the processor automatically lowers its operating frequency (due to features like Enhanced Intel SpeedStep Technology) leading to reduced power consumption. Manual frequency scaling using ACPI-cpufreq interface is likely to produce no benefit in this case or might produce some benefit at the cost of performance.

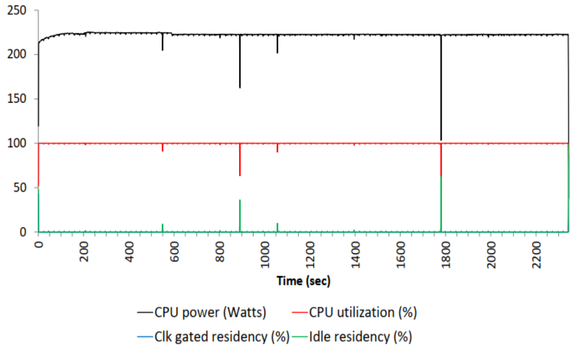

Fig. 9. GROMACS CPU Profile

For GROMACS (Fig. 9), the CPU utilization is very high throughout the entire duration of execution. Idle state residency is very low. Processor power consumption profile is almost identical to the cpu utilization profile and shows sharp drops during periods of low cpu utilization. The correlation coefficient of processor power consumption with CPU utilization is 0.97. GROMACS represents a highly cpu intensive profile with minimal scope for idle residency or clock gating.

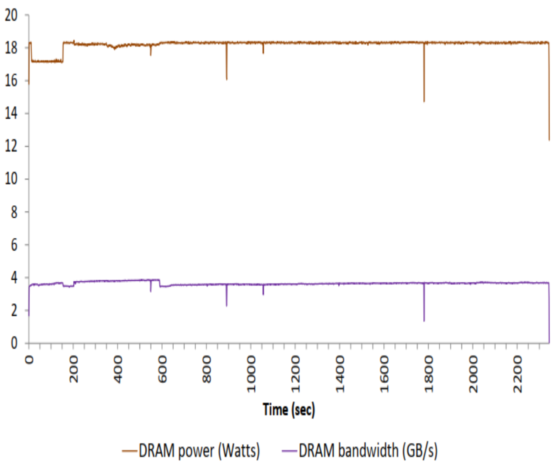

Fig. 10. GROMACS DRAM Profile

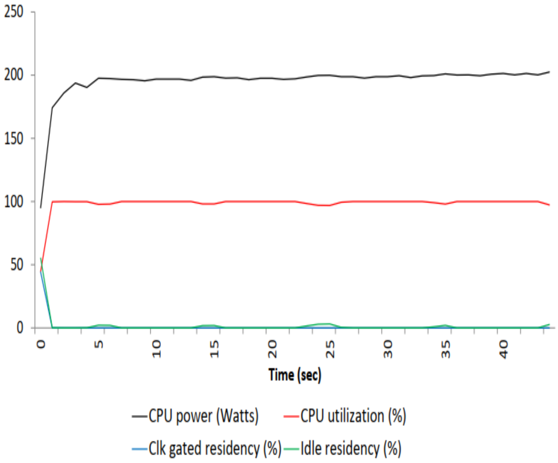

Fig. 11. WRF CPU Profile

Our observation with WRF (Fig. 11) is somewhat similar. The CPU utilization is very high throughout the major duration of execution. Idle state residency is very low. Like with GROMACS, the processor power consumption profile is very similar to the CPU utilization profile (correlation coefficient of 0.95).

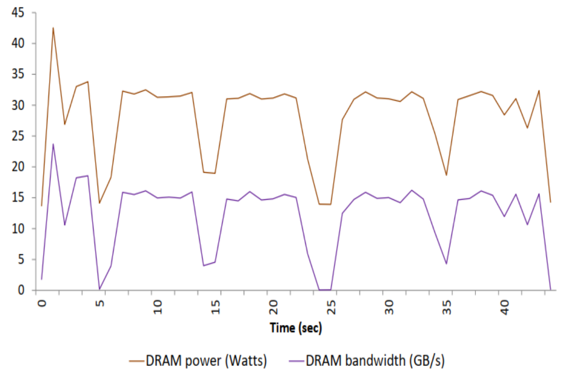

Fig. 12. WRF DRAM Profile

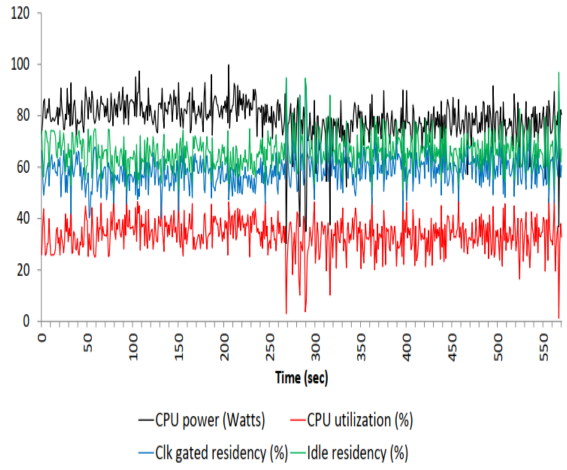

Fig. 13. LAMMPS CPU Profile

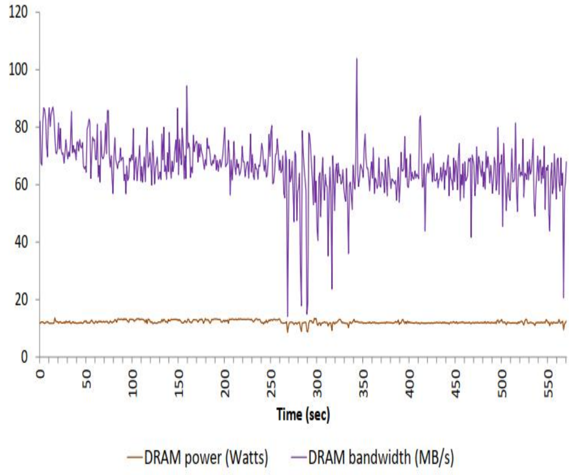

Fig. 14. LAMMPS DRAM Profile

For LAMMPS (Fig. 13), the CPU utilization remains below 40% and shows some variation throughout the application execution with occasional peaks reaching up to 49%. Idle state residency varies between 51% and 97% with clock gated residency between 40% and 95%. Processor power consumption stays within 100W and has a correlation factor of 0.8 with CPU utilization percentage. It shows that the processor power consumption responds dynamically to the CPU utilization and power consumption reduces automatically during regions of low compute activity. Clock gated residency is noticeably high during the entire execution period

B. Workload profile with frequency scaling

For the next section, frequency scaling governor was changed to "userspace," and the workloads were repeatedly executed for four frequency values - 2.6GHz, 2.2GHz, 1.7GHz and 1.2GHz. Time taken for the execution of a workload is treated as the relative metric for measuring performance. Change in energy or time taken is measured relative to the readings for conservative scaling governor.

TABLE - I
ENERGY VS RUNTIME AT DIFFERENT FREQUENCIES FOR LINPACK

S#	Freq (GHz)	Energy(J)	Time(S)	%Energy Reduction	%Time Reduction
1	Conservative	343329.06	1575		
2	2.6	300257.89	1694	-12.54	7.55
3	2.2	258192.66	1832	-24.80	16.32
4	1.7	226461.76	2182	-34.04	38.54
5	1.2	221732.58	3004	-35.42	90.73

For LINPACK, as seen in table I, lowering of frequency does reduce the total energy consumed by the workload at the cost of performance. For a 12% reduction in energy, the time taken is increased by 7.5%. It shows LINPACK is not a good candidate for power optimization using frequency scaling.

TABLE - II
ENERGY VS RUNTIME AT DIFFERENT FREQUENCIES FOR STREAM

S#	Freq (GHz)	Energy(J)	Time(S)	%Energy Reduction	%Time Reduction
1	Conservative	47732.93	319		
2	2.6	39780.89	331	-16.66	3.76
3	2.2	30706.35	299	-35.67	-6.27
4	1.7	25534.82	331	-46.50	3.76
5	1.2	23380.17	391	-51.02	22.57

STREAM (table II) shows mixed results. At 2.6 GHz, energy consumed is reduced by 16% but the time taken increases by 3.7%. But, at 2.2GHz, the energy consumption reduces by 35.6% along with a 6% reduction in time taken. It shows that 2.2GHz might act like a sweet-spot for this particular workload on the test setup and holds potential for significant power reduction and performance improvement.

TABLE - III
ENERGY VS RUNTIME AT DIFFERENT FREQUENCIES FOR STRESS-NG

S#	Freq (GHz)	Energy (J)	Time(S)	%Energy Reduction	%Time Reduction
1	Conservative	83496.49	731		
2	2.6	75829.83	769	-9.18	5.20
3	2.2	71314.90	886	-14.59	21.20
4	1.7	64224.21	988	-23.08	35.16
5	1.2	60517.55	1182	-27.52	61.70

TABLE - IV
ENERGY VS RUNTIME AT DIFFERENT FREQUENCIES FOR GROMACS

S#	Freq (GHz)	Energy(J)	Time(S)	%Energy Reduction	%Time Reduction
1	Conservative	527851.01	2344		
2	2.6	468702.88	2696	-11.20	15.02
3	2.2	423004.84	3119	-19.86	33.06
4	1.7	392046.85	3976	-25.73	69.62
5	1.2	384787.21	5466	-27.10	133.19

TABLE V
ENERGY VS RUNTIME AT DIFFERENT FREQUENCIES FOR WRF

S#	Freq (GHz)	Energy(J)	Time(S)	%Energy Reduction	%Time Reduction
1	Conservative	10820.73	44		
2	2.6	7865.82	44	-27.31	0
3	2.2	6762.51	47	-37.50	6.82
4	1.7	5474.12	53	-49.41	20.45
5	1.2	5701.82	70	-47.31	59.10

TABLE VI
ENERGY VS RUNTIME AT DIFFERENT FREQUENCIES FOR LAMMPS

S#	Freq (GHz)	Energy(J)	Time(S)	%Energy Reduction	%Time Reduction
1	Conservative	47776.30	570		
2	2.6	35570.54	557	-25.55	-2.28
3	2.2	35913.67	652	-24.83	14.38
4	1.7	38957.63	836	-18.46	46.67
5	1.2	47335.32	1174	-0.92	105.96

The readings for stress-ng (table III) combine the results from the three separate stressors mentioned previously. For a 9% decrease in energy consumption, the time took increases by 5% showing that it is not a suitable candidate for frequency scaling.

For GROMACS (table IV), the observations are similar to LINPACK. An 11% decrease in energy consumption pushes up the time taken by 15%. Frequency scaling is not useful in this scenario.

WRF (table V) shows a 27% decrease in energy consumption with no change in runtime when the frequency is scaled to 2.6GHz. At 2.2GHz, energy consumption reduces by 37%, but runtime also increases by 7%. For the test setup, 2.6GHz can serve as a sweet spot for this workload.

LAMMPS (table VI) shows a 25% decrease in energy consumption along with a 2.2% decrease in time taken at 2.6GHz. Time taken goes up by 14% at 2.2GHz with a 24.8% reduction in energy consumed. Hence, for the test setup, 2.6 GHz can serve as a sweet spot for this workload to maximize performance with a reduction in energy consumed.

V. CONCLUSION

The results presented in this paper lead to two important conclusions.

1) As is visible in the power consumption profiles of the different workloads, the processor hardware autonomously scales down the power consumption during periods of low activity.
   a. For workloads like STREAM, stress-ng (–cache and –aio stressor) and LAMMPS, a significant amount of clock-gated residency is visible which helps in reducing the processor power consumption during periods of reduced load. For these three workloads, the clock gated residency closely follows the idle residency which shows that the processor does a good job of autonomously detecting the scope of reducing power consumption by utilizing the clock gating feature during idle periods.
2) Frequency scaling, as a technique for power optimization, is only suitable for certain kinds of workloads.
   a) The results show that for highly CPU intensive HPC workloads like LINPACK or GROMACS, frequency scaling is not a suitable option as reducing the frequency leads to a significant increase in run-time which is not acceptable for HPC applications. For synthetic workloads like stress-ng (3 stressors combined) as well, frequency scaling did not fare well as it significantly increased runtime.
   b) For highly memory-intensive workloads like STREAM or LAMMPS, frequency scaling might lead to a reduction of both energy and runtime as shown in tables II and VI. It might require multiple iterations to obtain the optimum value of frequency.

While the results reported in this paper are the node level measurements, the results are equally crucial for cluster level considerations where the outcomes reported effectively get magnified on a bigger scale. Our results show that, for non-compute intensive workloads, multiple iterations can be carried out to find a sweet spot that leads to both energy reduction and performance improvement. The number of iterations may increase with variations in input parameters. We also show that even in the absence of any manual frequency scaling at the system software level, the processor itself autonomously reduces the power consumption (using clock gating and other features) during periods of idleness. The power consumption profiles for non-compute intensive workloads clearly show that during the periods of inactivity, the clock gated residency closely follows the idle state residency. It is highlighted by the fact that the mean of the difference between the idle residency (%) and clock gated residency (%) is only 7.4% for LAMMPS and 5.1% for STREAM.

VI. FUTURE WORK

In the future, we intend to explore hardware platforms from other vendors like AMD, IBM and newer Intel architectures like Broadwell which provides independent power management in hardware [30]. We plan to extend our work with other benchmarks and HPC applications and present a comparison across the different architectures.

REFERENCES

[1] Exascale Challenges, Scientific Computing World, July 2014, https://www.scientificcomputing.com/feature/exascale-challenges

[2] Power Problems Threaten to Strangle Exascale Computing, IEEE Spectrum, Dec 2015, http://spectrum.ieee.org/computing/hardware/power-problems-threaten-to-strangle-exascale-computing

[3] Technical Challenges of Exascale Computing, JASON, The MITRE Corporation, April 2013, https://fas.org/irp/agency/dod/jason/exascale.pdf

[4] Amarasinghe S, Campbell D, Carlson W, et al. (2009) Exascale Software Study: Software Challenges in Extreme Scale Systems. DARPA Report http://mrichards.ece.gatech.edu/wp-content/uploads/sites/462/2016/08/ECSS-report-101909.pdf

[5] ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, DAPRA IPTO, September 28, 2008, http://www.cse.nd.edu/Reports/2008/TR-2008-13.pdf

[6] John Shalf, Sudip Dosanjh, and John Morrison, Exascale Computing Technology Challenges, Springer-Verlag Berlin Heidelberg 2011, https://www.nersc.gov/assets/NERSC-Staff-Publications/2010/ShalfVecpar2010.pdf

[7] Top Ten Exascale Research Challenges, ASCAC Subcommittee Report, Department of Energy, USA, February 10, 2014, https://science.energy.gov/˜/media/ascr/ascac/pdf/meetings/20140210/ Top10reportFEB14.pdf

[8] Top500, https://www.top500.org/

[9] James H. Laros III, Measuring and Tuning Energy Efficiency on Large-Scale High-Performance Computing Platforms, SAND2011-5702, August 2011

[10] Sundriyal, V. & Sosonkina, M. J Supercomput (2016) 72: 1549. https://doi.org/10.1007/s11227-016-1680-4

[11] J. Eastep, S. Sylvester, C. Cantalupo, B. Geltz, F. Ardanaz, A. Al- Rawi, Livingston, F. Keceli, M. Maiterth, and S. Jana, Global extensible open power manager: A vehicle for HPC community collaboration on co-designed energy management solutions, in High Performance Computing - 32nd International Conference, ISC High Performance 2017, Frankfurt, Germany, June 18-22, 2017, Proceedings, 2017, pp. 394412.

[12] R. E. Grant, M. Levenhagen, S. L. Olivier, D. DeBonis, K. T. Pedretti, and J. H. L. III, Standardizing power monitoring and control at exascale, Computer, vol. 49, no. 10, pp. 3846, Oct 2016, ISSN: 0018-9162.

[13] Cotes-Ruiz IT, Prado RP, Garca-Galn S, Muoz-Expsito JE, Ruiz-Reyes N (2017) Dynamic Voltage Frequency Scaling Simulator for Real Workflows Energy-Aware Management in Green Cloud Computing. PLoS ONE 12(1): e0169803. https://doi.org/10.1371/journal.pone.0169803

[14] Georges Da Costa, Jean-Marc Pierson. DVFS governor for HPC: Higher, Faster, Greener. 23rd Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2015), Mar 2015, Turku, Finland. 2015 23rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp.533-540, 2015, ¡10.1109/PDP.2015.73¿. ¡hal-01387826¿

[15] Ghislain Landry Tsafack Chetsa, Laurent Lefevre, Jean-Marc Pierson, Patricia Stolf, Georges Da Costa. Beyond CPU Frequency Scaling for a Fine-grained Energy Control of HPC Systems. 24th International Symposium on Computer Architecture and High-Performance Computing, Oct 2012, New York City, United States. 2012.

[16] Sundriyal, V., Sosonkina, M. & Zhang, Z. J Supercomput (2014) 68: 777. https://doi.org/10.1007/s11227-013-1062-0

[17] Nikl V., Hradecky M., Keleceni J., Jaros J. (2017) The Investigation of the ARMv7 and Intel Haswell Architectures Suitability for Performance and Energy-Aware Computing. In: Kunkel J., Yokota R., Balaji P., Keyes (eds) High-Performance Computing. ISC 2017. Lecture Notes in Computer Science, vol 10266. Springer, Cham

[18] ” HPL Benchmark,” http://www.netlib.org/benchmark/hpl/

[19] ” Intel MKL Benchmarks,” https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite

[20] ” Stream Benchmark,” https://www.cs.virginia.edu/stream/

[21] Stress-ng benchmark, http://kernel.ubuntu.com/cking/stress-ng/

[22] ” Lammps Benchmarks,” http://lammps.sandia.gov/bench.html

[23] ” WRV V3 Parallel Benchmark”, http://www2.mmm.ucar.edu/wrf/WG2/ benchv3/

[24] ” Gromacs Benchmark,” http://www.gromacs.org/About Gromacs/ Benchmarks

[25] ” Running Average Power Limit,” https://01.org/blogs/2014/running-average-power-limit-%E2%80%93-rapl

[26] ” Performance Counter Monitor,” https://software.intel.com/en-us/articles/intel-performance-counter-monitor

[27] ” PowerTop,” https://01.org/powertop

[28] ” Intel Xeon Processor E5-1600/E5-2600/E5-4600 Product Families”, Datasheet - Volume One, May 2012, https://www.mouser.com/ds/2/612/xeon-e5-1600-2600-vol-1-datasheet-263646.pdf

[29] ” Intel Turbo Boost Technology 2.0”, https://www.intel.in/content/www/in/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html

[30] ” HWPM - Hardware Power Management,” https://software.intel.com/en-us/articles/intel-xeon-processor-e5-2600-v4-product-family-technical-overview