Adaptive Power Management for HPC Applications

Sumit Kumar Saurav, Ganga Prasad G.L, Manisha Chauhan
Real Time Systems and Smart Grid Group
Centre for Development of Advanced Computing, 'C-DAC Knowledge Park',
Bengaluru, INDIA
sumitk@cdac.in, gpr@cdac.in, manishac@cdac.in

***Abstract*— Reduction of power and energy consumption is one of the major concerns and challenges for High Performance Computing (HPC). However, as we move towards Exascale, it will be power limited in future. The advent of Running Average Power Limit (RAPL) abstraction after sandy bridge processors has paved the way to manage power adaptively. This gives the fine-grained power measurement and control mechanism with integrated voltage regulator at core level based on power budget. The purpose of Adaptive Power Management System(APM) for HPC systems is to decide when to place power manageable components into various power saving states based on the power consumption of an application within the power budget. In HPC, jobs are distributed across various computing nodes and power management is more complex with respect to the placement of the components in required operating states. The real time power monitoring and controlling through RAPL interface gives an opportunity for adaptive management.**

In this paper, we describe fine-grained profiling of HPC applications and control mechanism at component level using RAPL and APM system. The idea is to profile the HPC applications at fine granular level by measuring the power consumed by various power manageable components of a node such as processors and DRAM so that more accurate power related information can be obtained and then adaptively learn and devise the Optimal Power Budget (OPB) of an application. The OPB information is stored in Knowledge Base (KB) .The devised OPB for HPC application with optimal number of processors is incorporated in the job scheduler to take power-aware scheduling decision.

Keywords— Adaptive Power Management; High Performance Computing; Knowledge Base; Optimal Power Budget; Running Average Power Limit

I. INTRODUCTION

The power consumption of High Performance Computing (HPC) system is one of the limiting factors to achieve Exascale performance. The estimated power consumption for Exascale systems which consists of millions of nodes is around 20MW [1], [2]. The performance is the prime concern for the HPC applications. Moreover, there is an inherent tradeoff between power and performance. The power consumption and performance of CMOS based devices such as processor and memory depend on clock frequency and operating voltage [4], [5]. The Dynamic Voltage and Frequency Scaling (DVFS) is a widely used power reduction technique which reduces power by tuning the operating voltage and frequency of the processor and memory devices and significantly contributes to save overall energy consumption of the workloads. Traditionally, DVFS has been applied only at a coarse timescale and it introduces the high overheads [10]. Typically, DVFS provides coarse-grained level voltage and frequency control and imposes voltage transition delay and slow response. However, the fine-grained DVFS at small time scales may lead to substantial energy savings for different workloads. The fine-grained onboard regulators have been incorporated with modern processors which enable the core level DVFS [6], [7]. The Intel's Sandy Bridge processor provides Running Average Power Limit (RAPL) interface which has mechanism to specify power limits, power measurement and control capabilities for power manageable subsystems. The RAPL abstraction has been used for power management. The modern Intel processors (code name Haswell) like Intel Xeon E5-1600v3 and E5-2600 v3 are also powered by Fully Integrated Voltage Regulators (FIVRs) which regulate the core level voltage and frequency [6], [7]. The FIVR enables Per-Core P-States (PCPS). The PCPS and fine granular switching of their states make the runtime system energy-aware. The Haswell architecture is also empowered by Uncore Frequency Scaling (UFS) which enables the processors to control frequency of uncore components. The Energy Performance Bias (EPB) and Energy-Efficient Turbo (EET) features are available in Haswell processor which also makes run time system power-manageable.

This paradigm has given us an opportunity to realize Adaptive Power Management (APM) system by exploiting the power management capabilities of the underlying hardware, as well as incorporating the context of the application and Quality of Service (QoS) parameters i.e. performance. The effective power management is a key to reduce overall power consumption of the system without compromising the performance, which increases the energy efficiency of an application. The prime challenge to build APM system is to decide when to place power manageable components in different power saving states at run time.

This paper describes an APM system which optimizes the overall power consumption and makes scheduler power-aware by providing Optimal power Budget (OPB) prior to scheduling decision.

The rest of the paper is organized as follows. In Section II, we describe the related work, and in Section III, we review the Power Management Capabilities of Modern Intel Architecture. In Section IV, the proposed methodology for APM is described. Section V presents the experimental setup used to test the proposed methodology. Section VI has a case study

978-1-4673-6615-1/16/$31.00

that demonstrates the Experimental Results. Section VII presents some concluding remarks.

II. Related Work

The power optimization for HPC applications is one of the major research area and recent studies have addressed several facets of processor power limiting and budgeting aspect. The most important aspect to reduce power consumption in HPC system is the accurate and fine-grained measurement along with detailed analysis of power consumption [20]. Typically, power measurement is done using hardware power meters such as Watts up? .NET meter which gives overall power consumption of the system [14]. Moreover, the detailed analysis requires fine-grained power consumption pattern at subsystem level. The Intel's RAPL interface provides such capability to measure, monitor and control power usage for different domains [6]. In [7], Hackenberg et al. provide an energy efficiency feature survey of the Intel Haswell Processor which describes energy efficiency optimization strategies such as dynamic voltage and frequency scaling (DVFS) along with enhanced RAPL implementation and its improved accuracy in measurement mechanism. In [11], Langer et al. have analyzed the energy vs. time trade-off for power in over-provisioned HPC data centers. This work demonstrates curve fitting technique and interpolation mechanism for performance prediction for different configuration in over-provisioned system. In [15], Energy Efficient Rescheduling Algorithm for HPC application is described by selecting the appropriate combination of voltage and frequency for optimal energy consumption. Power Aware Algorithm for Scheduling in High Performance Computing is proposed based on optimal voltage and frequency in [16]. Berakoo et al. [19] have analyzed software based power measurement technique and its effectiveness. The different state-of-the-art power measurement and profiling aspects are reviewed in [20]. In [21], Narayan discusses about software based power measurement model and its limitations. Although hardware approach provides accurate power related information, it only gives coarse-grained power consumption which is inadequate to take control decision at subsystem level. In [22], Sistla et al. have described the user level control of power management policy and EPB range selection with power and performance tradeoffs.

III. Power Management Capablities of Modern Architecture

The modern system architectures are incorporated with power efficient and power-manageable subsystems which help to achieve high energy efficiency for the running applications. The processor architecture like Intel Haswell, AMD Bulldozer, IBM power 8 and upcoming Heterogeneous System Architecture (HSA) are designed to achieve the high energy efficiency along with fine-grained power measurement and monitoring capabilities.

A. Running Average Power Limit (RAPL)

The RAPL provides a standard interface for monitoring and controlling the power consumption of sub-components like processor and memory. RAPL is not an analog power meter; rather it uses a software based power model which estimates energy usage by using hardware performance counters and I/O models. RAPL interfaces consist of non-architectural Model Specific Register (MSRs). The RAPL interface reads MSRs which contain power consumption and control information. Each RAPL domain supports set of capabilities like power limit, energy status, performance status, power information and policy. The RAPL support was introduced with Sandy Bridge processors and more features have been added to extend its usage since then. In RAPL, platforms are divided into domains to attain fine-grained control. These domains include package, power planes i.e. CPU core (PP0), graphics uncore (PP1) and DRAM controller. Each level of the RAPL hierarchy consists of RAPL interface MSRs [6], [7]. The MSR_RAPL_POWER_UNIT register describes the unit of power, energy and time. Fig.1 illustrates MSR_RAPL_POWER_UNIT register which contains specific unit information for different measuring parameters like power, time and energy status for all the domains. The MSR_PKG_POWER_LIMIT register describes the power limit which can be set to restrict the power usage of specific domain. The MSR_PKG_ENERGY_STATUS reports the actual energy usage for the package domain. The MSR_PKG_POWER_INFO MSR reports package power range information [7].

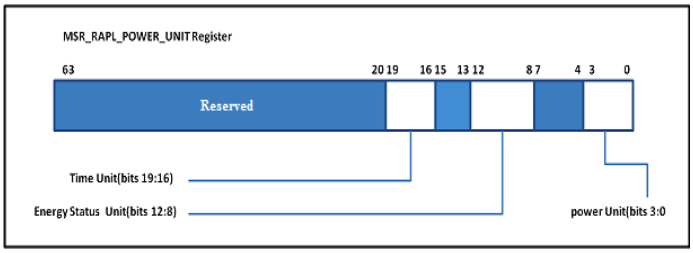

Fig.1. Model Specific Register (MSR_RAPL_POWER_UNIT) details

The MSR_PKG_PERF_STATUS MSR contains total time for which the package was throttled for P-states and T-states transition due to RAPL limits. A P-state is an operational state for a processor whereas a T-state refers to throttling the processor clock to lower frequencies in order to reduce thermal effects. The domain specific MSRs are available for each domain like PP0/PP1 and DRAM. These MSRs report the usage in terms of power, time and energy consumed, as well as it has specific bit range to set power limit. This fine granular measurement at different domain level can be used for effective power management for the scheduled application in real time.

B. Fully Integrated Voltage Regulator (FIVR)

The Intel's 4th generation microprocessors are powered by a 140-MHz, multiphase FIVR. This is the industry's first large-scale deployment of high current switching regulators integrated into a VLSI die and package. The FIVR provides individual voltage for each core which enables core level state transition. FIVR increases the number of voltage rails,

allowing each domain to be set at the minimum possible voltage which reduces both leakage and dynamic power. FIVR enables Per-Core Power-State. The FIVR technology has impacted the energy efficiency using transition of core states with very minimal transition latency. Similarly, UFS has enabled the control for uncore components i.e. Last Level Cache [7], [8].

C. Energy Performance Bias (EPB)

The EPB influences the selection of operating frequency of the processors. The EPB enables user level control by setting IA32_ENERGY_PERF_BIAS MSR with value from 0-15. The values represent a sliding scale, where 0 (the default reset value) corresponds to highest performance level and 15 corresponds to the maximum energy savings [7]. The power management system can use EPB to reduce power by making appropriate settings at run time.

IV. PROPOSED METHODOLOGY

A. System Model

The Proposed Methodology is experimented on our test bed describe in Section V. The Thermal Design Power Limit (TDP) of each processor of our test bed is 130W. The TDP is the maximum amount of thermal power that a processor dissipates but it can consume more than the TDP limit. The available power domains on compute nodes are PKG, PP0, PP1 and DRAM [6]. These power domains are monitored and regulated in power capping environment.

B. Energy Consumption Model

The power consumption of CMOS logic based integrated circuits i.e. processors and DRAM, is composed of dynamic and static power. The dynamic power consumption is caused by the charging and discharging of capacitive load which originates from the switching of logic gates state. The static power consumption is due to leakage current regardless of switching activity. The power consumption of silicon CMOS logic circuit is expressed as

$$P_{Total} = P_{dynamic} + P_{static} \quad (1)$$

where P_{Total} is the total power consumed by CMOS circuit and $P_{dynamic}$ is calculated as

$$P_{dynamic} = AC\ v_{dd}^2\ f_{clk} \quad (2)$$

In (2), A is the percentage of active gates, C is the capacitance, v_{dd} is the operating voltage and f_{clk} is operating clock frequency. The correlation between clock frequency and operating voltage can be expressed as

$$f_{clk} \propto (v_{dd}-v_t)/v_{dd} \quad (3)$$

where v_t is the threshold voltage. Power is the rate of energy consumption and the total energy consumption can be written as

$$E_{Total} = (P_{dynamic} + P_{static}).\ \Delta t \quad (4)$$

The total energy consumption of a CMOS based circuit can be expressed as [3], [4]

$$E_{Total} = (AC\ v_{dd}^2 f_{clk} + v_{Ileak}).\Delta t \quad (5)$$

where E_{Total} is the energy consumed by CMOS circuit, I_{leak} denotes leakage current and Δt is the execution time.

The overall power consumption of a node is an aggregation of dynamic and static power consumed by power manageable and unmanageable components of a computer system [3].

$$P_{Node} = \Sigma P_{NodeDyn} + \Sigma P_{NodeStatic} \quad (6)$$

$$P_{NodeDyn} = P_{cpu} + P_{DRAM} + P_{disk} + P_{nic} + \Delta \quad (7)$$

In (6), the $P_{Node,}$ $P_{NodeDyn}$ and $P_{NodeStatic}$ represent the total, dynamic and static power consumption of a node respectively. In (7), the $P_{cpu,}$ P_{DRAM}, P_{disk} and P_{nic} denote CPU, DRAM, disk and Network Interface Card (NIC) power consumption respectively.

C. Problem Statement and Research Work

The power and energy are key challenge for future HPC systems to achieve Exascale performance. The coarse-grained power profiling of an application is inadequate to determine power consumption at subsystem level. The power efficient hardware, fine-grained power measurement and adaptive control mechanism are imperative to realize adaptive runtime system for energy efficient HPC systems. Moreover, the adaptive power budgeting and controlling must not sacrifice the performance of the application.

The prime goal of our investigation is to realize an APM system which reduces overall power consumption for the scheduled applications by exploiting modern hardware power management capabilities with the consideration of the application's QoS in autonomic and effective manner for HPC facilities.

The proposed work includes the following tasks:

I. Fine-grained power profiling of HPC application to devise OPB and Optimal Configuration (OC) for initial schedule.
II. Real-time QoS parameters monitoring and controlling for power manageable components.

D. Optimal Configuration and Optimal Power Budget

The Optimal Configuration (OC) of an application is the number of Processing Element (PEs) required for running an application with minimal energy consumption. This information is used by job scheduler to schedule the application. The over-provisioned system requires an additional base power and has an associated tradeoff between the performance gained and the additional power required for over-provisioning. In proposed work, OC is devised from power profile of an application by running at maximum clock frequency with varying operating voltage levels for different number of PEs. In Fig. 5, we have illustrated the SFM (Seasonal Forecast Model) Optimal Configuration. The OPB is the power budget of an application for an optimal configuration which maximally saves the energy with negligible or no impact

on performance. The APM system uses RAPL interface to set the OPB in specific domain's MSR for the target configuration. The head node schedules the job across the nodes based on devised OC and OPB.

E. The Proposed Adaptive Power Management System

The notion of Adaptive Power Management System is to design an autonomic and self managed system which exploits the power management abstractions of the underlying hardware and the QoS characteristics of workloads to reduce the overall energy consumption by placing the power manageable subsystems into the required power saving state while operating in power constrained environment within OPB limit. The QoS parameters are the runtime requirements of the systems i.e. performance, power consumption and execution time for a given workload. The real time monitoring of these parameters enables the APM system to take appropriate control decision in order to save power.

The power profiling and characterizing the scheduled workload at subsystem level are the building blocks of the APM system. The workload characterization helps to build correlations among the different parameters like power consumption of subsystems with performance, CPU and memory usage etc. This information is the initial decision making construct for Control Decision Engine (CDE) which learns progressively from such correlations and takes decision in autonomic manner.

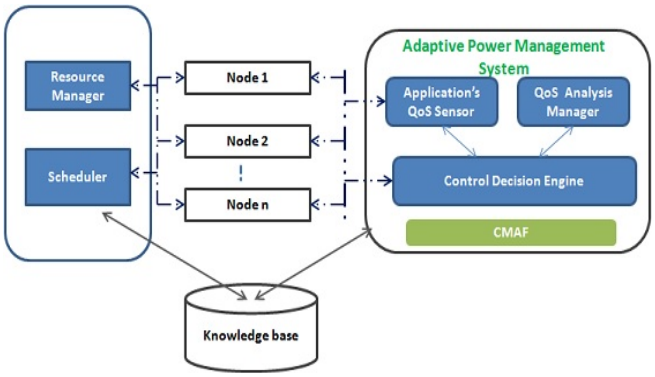

Fig. 2. Adaptive Power Management System Block Diagram

In Fig. 2, we describe the architecture of the proposed APM system. The APM system has four different components namely, Application QoS Sensor (AQS), QoS Analysis Manager (QAM), CDE and C-DAC Mobile Agent Framework (CMAF). The AQS is a real time monitoring system which monitors various parameters like power consumption, CPU, DRAM and Uncore usage etc. by reading MSRs through RAPL. The CDE takes decision to place power manageable components into learnt power saving states. The QAM analyzes and maintains the correlations among the QoS parameters and devises thresholds which can be updated dynamically. This information is maintained in KB with execution context and used for decision making. The CMAF helps to gather and propagate decisions of the CDE across the target nodes on HPC platform. The AQS keeps monitoring power and performance and CDE tunes the power and performance requirements of an application in real-time with the help of the QAM such as EPB's IA32_ENERGY_PERF_BIAS MSR value ranging 0-15 based on power and performance requirements.

The algorithm 1 describes selection of OC and OPB while scheduling. The HPC jobs are characterized by input parameters like problem size, wall time etc. The wall time is the maximum execution time for an application. The APM system collects an additional input parameter which describes the extent of power saving i.e. performance mode, balanced mode, power saving mode etc. while scheduling the jobs. The KB suggests OC and OPB to job scheduler based on decision tree approach [16], [23] and [24]. The real time monitoring of each power domain like PKG, PP0, and DRAM enable adaptive tuning of power budget dynamically based on fine-grained power profile, usage patterns and thresholds.

Algorithm 1 : To select OPB and OC from KB

/* Job J_k={$j_1(s_1,d_1,op,),j_2(s_2,d_2,$ op),….,$j_n(s_n,d_n,$ op)}
Wall time = d_i size = s_i
Knowledge Base KB = {J_{Name} = J_k, Nodes = N_i, Cores=C_i,
Execution Time= T_i , Size = s_i,
OC (Nodes, Core) ← OC_i (N_i, C_i)}
OPB_{domain} ← PB_i
Domain = {PKG, PP0, DRAM} }
*/

1. Build decision_ tree() using KB;
2. for job j_i , i from 1 to n do
3. for J_{Name} J_k ,k from 1 to m do
4. compare j_i and J_k in KB;
5. if($j_i \neq J_k$) then
6. set {
7. If(op) then
8. EPB [IA32_ENERGY_PERF_BIAS MSR]← Set mode using RAPL
9. Monitor (Qos i.e. performance, usage of each domain and set limits)
10. else
11. EPB [IA32_ENERGY_PERF_BIAS MSR]← Set Performance mode using RAPL
12. Monitor (Qos i.e. performance, usage of each domain and set limits) }
13. Store the results in KB else
14. Jk(j_i, PE_i,s_i,mode) → decision_tree(j_i, PE_i,s_i,mode);
15. for each domain
16. OPBdomain←decision_tree(j_i, PE_i,s_i, mode);
17. end for
18. Monitor QoS parameters
19. end else end if
20. end for J_k
21. end for j_i

The algorithm sets *IA32_ENERGY_PERF_BIAS MSR* limits to operate on performance mode if QoS parameter, OPB limit and OC are not available. The EPB limit is based on power
The algorithm sets IA32_ENERGY_PERF_BIAS MSR limits to operate on performance mode if QoS parameter, OPB limit and OC are not available. The EPB limit is based on power and performance tradeoffs [22]. The EPB limits based on CPU utilization is described in TABLE I. The APM control engine sets EPB limits based on CPU usage dynamically.

TABLE I. EPB AND CPU USAGE CO-RELATION

EPB Limit	CPU Utilization in %	Remarks
0-3	80-100	Performance
4-7	50-80	Balanced
8-11	20-50	Balanced
12-15	0-20	Power Saving

The incorporation of FIVR and PCPS as discussed in section III has also given the opportunity for aggressive power management at core level. The APM System also exploits P-STATE and C-STATE transition in adaptive manner. The proposed APM system exploits all possible power management capabilities to enforce control decision predicted by the decision engine to reduce power consumption.

V. EXPERIMENTAL SETUP

The experimental setup consists of five homogenous HP BL460c Gen8 Intel Xeon E5-2680 v3 servers (refer to TABLE II.) which are mounted in a 2U Rack and connected to the in-house network with a 16 Port Ethernet switch. We use Watts up? .NET meter for power measurement and calibration purpose. The Watts up power meter is connected to the individual nodes. The power meter has been connected between the power supply AC input of each cluster node and the socket connected to the external supply. All servers are connected to IP based KVM switch with a console attached. The compute nodes are clustered using Portable Batch System (PBS) resource manager and Maui scheduler with Red Hat Enterprise Linux operating system. The C-DAC Mobile Agent Framework (CMAF) is used for acquiring power and resource utilization (CPU, Memory, and Disk) data from each compute node and controlling power usage [17]. We use Intel® Power Governor (power_gov) software utility for monitoring and controlling power at very fine granularities. The power monitoring and controlling is available for each domain i.e. package, core, graphics, uncore and DRAM. The knowledge base (KB) contains power and system usage information of HPC applications.

TABLE II. EXPERIMENTAL SETUP DETAILS

Processor Features	Details
Processor	2x Intel Xeon E5-2680 v3
Frequency Range	1.2 -2.5 GHZ
Max Turbo Frequency	3.3 GHZ
Energy Performance Bias	Balanced
Uncore Frequency Scaling	Enabled
Per Core P State	Enabled
TDP(Thermal Design Power)	130 W

VI. EXPERIMENTAL RESULTS

In this section, we illustrate the power profile of LINPACK and Seasonal Forecast Model (SFM). These experiments support the goal of the proposed APM System. We conducted all experiments in power capping environment in order to devise an OC and OPB. The work load characterization at subsystem level is an important measure to devise the correlation among different parameters like CPU usage with power consumption.

A. Case Study 1: Linpack

The Linpack benchmark is a measure of computer's floating-point rate of execution. This is determined by running a computer program that solves a dense system of linear equations. The Linpack is a compute intensive benchmark which exhorts the processors. On our test bed setup, we experimented with different power budgets, viz. 115 Watt, 100 Watt, and 85 Watt and tried to devise optimal power budget with 8 cores and 16 cores. We found the OPB for LINPACK to be 85 Watt in both the scenarios as shown in Fig 3. The optimal power budget gives an approximate energy saving of 11% with degradation in performance by 3%.

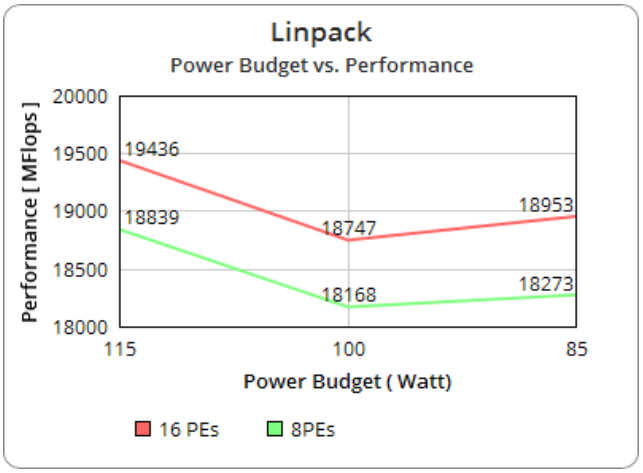

Fig. 3. LINPACK Power Budget vs. Performance

B. Case Study 2: SFM

Seasonal Forecast Model (SFM) is a climate model. This climate model is being used for predicting the Indian summer monsoon rainfall in advance of a season. The experiment has been conducted for different number of processors (4, 8, 12, and 16) in power capping environment. SFM is executed using different voltage and frequency combination. We observed that as the number of processor increases, the time to compute reduces significantly at every voltage level. These experimental results indicate that the optimal configuration for SFM with 720 hours forecast is 12, as illustrated in Fig.4. We also conducted experiment for SFM to devise an OPB. We executed SFM at different power budgets, viz. 115 Watt, 100 Watt and 85 Watt. The results are presented in Fig. 5, which indicates an approximate energy saving of 13% with degradation in performance by 2%.

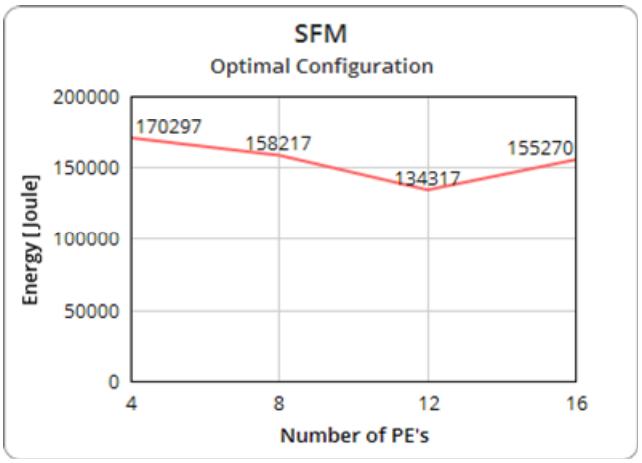

Fig. 4. SFM Optimal Configuration

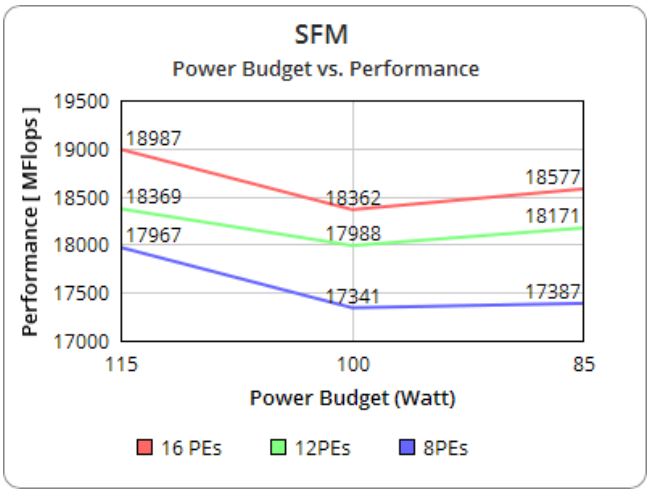

Fig. 5. SFM Power Budget vs. Performance

VII. CONCLUSIONS AND FUTURE REMARKS

In this paper, we explored the power management characteristic of modern system architecture and proposed APM system based on fine granular power profile and adaptive control mechanism. We devised the OPB and OC for HPC applications which can be used for scheduling purpose to build a power-aware job scheduler. On our test bed, we observed overall energy saving of 11-15% with degradation in performance by 2-4%. The long term goal of this research work is the progressive exploration of predictive decision making algorithms and its incorporation with APM system which takes an autonomic control decision to optimize the overall energy consumption.

In future, we want to validate the experimental results of different HPC applications at heterogeneous platform to devise effective decision algorithms for APM system. The future goal of APM system is to build sustainable and self-managed software which takes decision in autonomous manner.

ACKNOWLEDGMENT

The authors would like to thank Dr. P.V. Ananda Mohan for his valuable suggestions and support during writing this paper. The authors also would like to thank Dr. Sarat Chandra Babu, R.K. Senthil Kumar, A.C. Natraj, H.V. Raghu and B.Jayanth for their encouragement while conducting this work.

REFERENCES

[1] J.Dongarra,P.Beckman, and Terry Moore, "The International Exascale Software Project Roadmap", International Journal of High Performance Computing Applications, Sage Publications,Vol.25 Issue 1, pp. 3-60, Feb. 2011.

[2] K.Iskra, K.Yoshii, and R.Gupta , P.Beckman "Power Management for Exascale" Mathematics and Computer Science Division Argonne National Laboratory,USA.

[3] Rodero, S. Chandra, M. Parashar, R. Muralidhar, H. Seshadri, and S. Poole "Investigating the Potential of Application-Centric Aggressive Power Management for HPC Workloads" in International Conference on High Performance Computing (HiPC), pp. 1-10, Dec 2010.

[4] D Suleiman, Muhammed A Ibrahim,and I Hamarash" ,Dynamic Voltage Frequency Scaling (DVFS) For Microprocessors Power and Energy Reduction" in 4th International Conference on Electrical and Electronics Engineering, 2005.

[5] T.Kolp, A.Zhai, and S.S. Sapatnekar "Enabling Improved Power Management in Multicore Processors through Clustered DVFS" in Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, pp. 1-6, March 2011.

[6] Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2, 253669-039US, May 2011.

[7] D.Hackenberg, R.Schone, T.Ilsche, Daniel Molka, J.Schuchart,and R. Geyer "An Energy Efficiency Feature Survey of the Intel Haswell Processor" in International Parallel and Distributed Processing Symposium Workshop (IPDPSW) Hyderabad, IEEE, pp. 896-904, May 2015.

[8] E.A. Burton, G.Schrom, F.Paillet, W.J. Lambert, K.Radhakrishnan,M.J. Hill, and J. Douglas ,"FIVR – Fully Integrated Voltage Regulators on 4thGeneration Intel® Core™ SoCs" in Twenty-Ninth Annual IEEE Applied Power Electronics Conference and Exposition (APEC), pp. 432-439, March 2014 .

[9] B.Unni ,N. Parveen,Ankit Kumar , and B. S. Bindhumadhava, "An intelligent energy optimization approach for MPI based applications in HPC systems" Springer, *CSI Transactions on ICT*, Volume 1, Issue 2, pp. 175-181. June 2013.

[10] S.Eyerman and L.Eeckhout Ghent University, Belgium "Fine-Grained DVFS Using On-Chip Regulators," in *ACM Transactions on Architecture and Code Optimization (TACO)*, Volume 8 Issue 1, April 2011.

[11] A.Langer, H. Dokania and L.V. Kale, "Analyzing Energy-Time Tradeoff in Power Overprovisioned HPC Data Centers" in International Parallel and Distributed Processing Symposium Workshop (IPDPSW) Hyderabad, IEEE, pp. 849 – 854,May 2015.

[12] H.David, E.Gorbatov, Ulf R, R.Khanna, and C.Le , "RAPL: Memory Power Estimation and Capping" ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED) Austin, USA , pp. 189–194, Aug.2010.

[13] Balaji Subramaniam and Wu-chun Feng Virginia Tech, "Towards Energy-Proportional Computing for Enterprise-Class Server Workloads" in Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE) , ACM, pp. 15-26,April 2013.

[14] Yongpeng Liu and Hong Zhu"A Survey of the Research onPower Management Techniques for High Performance Systems" Software-Practice & Experience John Wiley & Sons, Inc. New York, NY, USA, vol.40 Issue11, pp. 943-964, October 2010.

[15] M. Chauhan, N.Parveen, S.K. Saurav, and G. L Ganga Prasad, "Energy Efficient Rescheduling Algorithm for High Performance Computing"

Parallel Computing Technologies (PARCOMPTECH), Banaglore, IEEE, pp. 1–7 , Feb 2015.

[16] H. V. Raghu , S.K Saurav , and Bindhumadhava Bapu S, "PAAS: Power Aware Algorithm for Scheduling in High Performance Computing" IEEE/ACM 6th International Conference on Utility and Cloud Computing Dresden, IEEE Computer Society , pp.327-332 ,Dec.2013.

[17] H. V. Raghu, Kumar, and B. S. Bindhumadhava, "High Performance Systems: An Agent based Application Power Profiling," 18th annual International Conference on Advanced Computing and Communications (ADCOM),IEEE, pp. 59 – 65, Dec. 2012.

[18] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, et al. "Leakage current: Moore's law meets static power," in *IEEE Computer*, vol. 36, no.12, pp. 68-75,Dec.2003.

[19] G.Bekaroo, C.Bokhoree, and C.Pattinson "Power Measurement of Computers: Analysis of the Effecetiveness of the Software Based Apporaoch" International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 5, May 2014.

[20] Hui Chen and Weisong Shi "Power Measuring and Profiling:State-of-the-Art" Department of Computer Science Wayne State Universty Detroit, MI,USA.

[21] D.Narayan "Software Power Measurement". Microsoft Research, Technical Report MSR-TR-51,2005.

[22] K.V.Sistla et al."User level control of power management policies" U.S Patent US9098261 B2,August 2015.

[23] MD. A. Iqbal, S. Rahman, S. I. Nabil, and I. V. A. Chowdhury, "Knowledge Based Decision Tree Construction with Feature Importance Domain Knowledge," 7th International Conference on Electrical and Computer Engineering, 20-22 December, 2012.

[24] O. Maimon and L. Rokach, "*Data Mining and Knowledge Discovery Handbook*," Springer Second Edition, 2010.