PAAS: Power Aware Algorithm for Scheduling in High Performance Computing

H. V. Raghu[1], Sumit Kumar Saurav[2], Bindhumadhava Bapu S[3]
Real Time Systems and Smart Grid Group
Centre for Development of Advanced Computing, 'C-DAC Knowledge Park',
Bangalore, INDIA
raghuhv@cdac.in[1], sumitk@cdac.in[2], bindhu@cdac.in[3]

***Abstract*— Energy consumption has become the major concern in High Performance Computing (HPC) systems as far as operational cost, reliability of systems and environmental impacts are concerned. The optimization of energy consumption in HPC systems is challenging task. The frequency/voltage used for execution of a job has a key impact in overall energy consumption. In this paper, we have implemented Power Aware Algorithm for Scheduling (PAAS), which focuses on energy reduction and load balancing aspects of HPC systems. The PAAS guides the scheduler to take intelligent decisions based on the information available in knowledge base. The Scheduler provides an optimal energy aware schedule for each node to minimize the makespan across nodes and to reduce energy consumption. The knowledge base gives the optimal frequency and voltage where energy consumption is minimal for a particular job. The Dynamic Voltage and Frequency Scaling (DVFS) knobs are tuned across the nodes based on predicted optimal frequency and voltage. We used hardware based power measurement device Watts up? .NET meter for individual computing nodes as well as Multi-Agent framework for monitoring and analyzing energy consumption. We evaluated the algorithm on the experimental test-bed with the set of scientific applications. Our technique showed an average measured energy savings of 12.64% and a maximum of 13.5%.**

Keywords*— *Power Aware Scheduling, Power Aware Algorithms; High Performance Computing; Dynamic Voltage and Frequency Scaling; Energy Optimization.

I. INTRODUCTION

Power Management and Energy Optimization in HPC system has became one of the major research challenges in order to reduce overall energy consumption. The research in HPC was focused on performance, throughput, low turnaround time, load balancing and less attention had been given to power optimization. HPC system needs more power with the increasing computing requirements. The power requirement of the HPC system leads not only to high operational cost but also reliability and environment protection concerns. The power consumption of US-based Titan supercomputer consumes 8.2 megawatt to deliver the performance of 17.6 Pflop/s and it costs around 9 million dollars per year [1].

There are various participating component in HPC infrastructure where we can look for the energy optimization from underlying infrastructure to development and execution of an application. There are various approaches that can be used to reduce energy consumption of HPC systems such as power efficient hardware, cutting down the power of underutilized hardware components when the system load is not high, implementing power aware compilers and schedulers, using DVFS controls etc [2]. CPU consumes the maximum amount of total system power for compute intensive jobs. DVFS is an efficient way to manage and reduce dynamic power consumption of the CPU during computation by decreasing the supply voltage/frequency of systems [3, 4]. The process of selecting the optimal frequency/voltage for each application to minimize energy consumption while meeting the QoS requires exploration of various combinations, which is more time consuming to compute [5]. Hence, we use predictive decision tree based model using knowledge base to get optimal operating frequency/voltage while scheduling jobs. The power aware scheduling is required for energy efficient resource utilization. The scheduler has to take intelligent decision while scheduling jobs based on energy utilizations. The voltage and frequency parameters have to be considered while scheduling the job.

In this paper, we have implemented the Power Aware Algorithm for Scheduling based on optimal frequency/voltage. The knowledge base contains energy profiles of various HPC applications of different workloads. The identification of near optimal frequency/voltage is based on minimal energy consumption. The rest of this paper is organized as follows. In Section II, related work in power aware schedulers for High Performance Computing is discussed. In Section III, experimental set-up with power measurement and analysis using CDAC Multi-Agent Framework [CMAF] [14] for HPC applications is discussed. In Section IV, Power Aware Algorithm and Scheduling (PAAS) as an optimization problem for minimizing energy consumption of the system is described. In Section V, results are presented to show the performance of PAAS.Conclusions and future works are presented in Section VI.

II. RELATED WORK

The energy optimization and efficient power management in HPC system has became one of the major research areas in order to reduce the power budget.

A. Cabrera et al. [6] presented an analytical model for predicting the energy consumption for the High Performance Linpack (HPL) benchmark. The derived model can be used to know the energy consumed by the HPL in advance over target architecture. This can be integrated to the schedulers for energy aware scheduling decision. A. S. Araujo et al. [7] evaluated an energy-aware application level list scheduler that uses heuristics based on critical path to determine processor frequency of each core to save energy. X. Ruan et al. [8] proposed an energy efficient scheduling algorithm using dynamic voltage scaling technique to provide energy saving to clusters. This algorithm takes the advantage of processor idle times to lower the processor voltages without significant increase in the execution time, which in turn reduces energy consumption of parallel applications running on clusters. J. M. Pierson and H. Casanova [9] investigated the capabilities provided by DVFS and formalized the problem to compute the bound on the optimal solution based

CFP13UCC-USB/13 $26.00 © 2013 IEEE
DOI 10.1109/UCC.2013.71

on several power consumption modes of the server. This approach quantifies the potential improvements that job placement algorithms can achieve by exploiting DVFS capabilities. L. Wang et al. [10, 11] developed scheduling heuristics to present application experience for reducing power consumption of parallel tasks in a cluster with the DVFS technique. They use slack time for non-critical jobs, extend their execution time and reduce the energy consumption. M. Etinski et al. [12] explained two power aware parallel job scheduling policies for HPC systems. The first policy assigns job frequency based on the predicted job performance and the second policy uses system utilization to decide when to run jobs at reduced frequency. M. Etiniski et al. [13] illustrated DVFS enabled processors and power aware scheduling leads to minimize energy with high performance. A. K. Coskun et al. [24] reliability-aware job scheduling and power management approaches for chip multiprocessors. C. Augonnet et al. [25] designed StarPU, which provides convenient way to provide parallel tasks over heterogeneous hardware and easily develop and tune powerful scheduling algorithms.

This paper illustrates PAAS which considers energy consumption and voltage/frequency as a scheduling parameter across the nodes. PAAS gives the optimal makespan and ensures the load balancing across the nodes such that the total energy consumption of the system is minimized. It predicts the optimal frequency/voltage from the knowledge base for each incoming jobs based on decision tree approach. CMAF [14] provides the power measurement and management aspect.

III. Experimental Setup

The experimental setup shown in Fig. 1, consists of five HP DL380 G7 servers which are housed in a 32U 19" Rack and connected to the in-house network with a 16 Port Ethernet switch. The Watts up? .NET meter is used to measure the power consumption of individual nodes and the combined power consumption of the cluster. The power meter has been connected between power supply AC (Alternating Current) input of the each cluster node and the socket connected to the external supply.

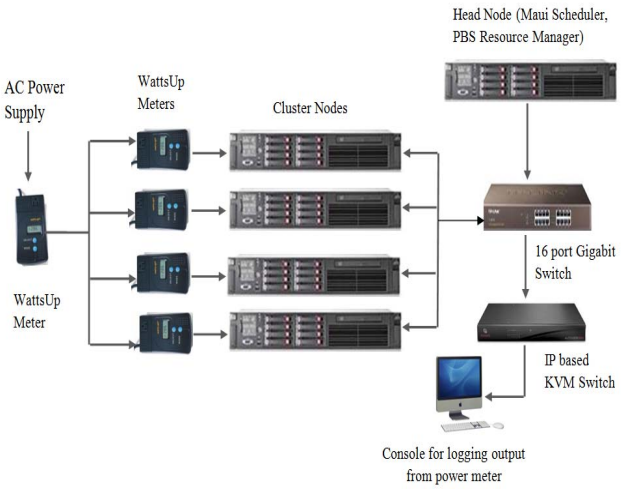

Fig. 1. Power profiling of HPC applications

The data from the power meter has been logged in the remote system and it can be downloaded to a Windows or Linux machine. The five servers are connected to IP based KVM switch (for remote GUI access) with a console attached. These five systems are clustered using PBS resource manager and Maui scheduler in Red Hat Enterprise Linux operating system. Power measurement, monitoring and analysis is performed using CMAF [14] for standard HPC benchmarks with respect to resource utilization (CPU, memory, disk etc), operating frequency/voltage and energy profile [15]. The energy and system usage profile of HPC applications are stored in knowledge base.

IV. Power Aware Scheduling

A. System Model

The experimental setup consists of five homogenous nodes. The head node is acting as a central resource controller. All nodes are connected in the form of clustered architecture. The head node schedules the job across the nodes based on optimal frequency and voltage.

B. Energy Model

DVFS enabled processors can execute jobs by using discrete set of voltage and frequency pairs (v_i,f_i) ordered in this manner $\{v_1<v_2<.....<v_n\}$ and $\{f_1<f_2<\ldots<f_n\}$[21]. The servers used in our experimentations support the voltage range {12, 13,…,19 volts} and frequency range {1.6,1.73,….2.4GHz}. The power consumption of a processor is composed of static power and dynamic power. Static power is the power consumed while there is no circuit activity. The power consumed while inputs are active is dynamic power. When inputs have AC supply, capacitances start to charge and discharge. As a result it increases the power dissipation.

$$P_{\text{total}} = P_{\text{static}} + P_{\text{dynamic}}$$

Energy consumption in terms of static and dynamic power is

$$E = (P_{\text{static}} + P_{\text{dynamic}}).\ \Delta t$$

According to [16], the total energy consumption can be given as:

$$E = (ACv^2f + vI_{\text{leak}}).\Delta t$$

Where,

A is the percentage of active gates, C is the capacitance load of all gates, v is the operating voltage, f is the processor frequency, I_{leak} is the leakage current and Δt is the time duration.

To reduce the total energy consumption, the processor frequency (f) and voltage (v) value can be reduced dynamically based on the instruction throughput and system utilization to yield a cubic reduction in the dynamic (Cv^2f) power with or without affecting the execution time [17]. The process of dynamically altering the supply voltage and operating frequency is commonly referred to as DVFS.

C. Problem Definition

The increase in performance of HPC systems has been followed by increase in power consumption [18]. The high power dissipation leads to problems like reduced reliability, increased operation cost, less stability etc. Energy consumption of the job

can be the main parameter to be considered while scheduling the jobs, in order to save energy [8, 9, 26]. Executing the jobs at high frequency/voltage leads to more energy consumption. Hence, there is a need to predict optimal frequency/voltage for each job to make scheduler power aware.

D. Power Aware Algorithm for Scheduling (PAAS)

Predicting optimal frequency/voltage for each job is done using the information available in the Knowledge Base (KB). The KB will have the information about optimal frequency/voltage based on various runs of HPC applications. When the scheduler receives a new job from the resource manager, it checks for the availability of the same job information in KB with same parameters. If the job is not present in the KB, then it selects the frequency/voltage to execute the job based on the CPU utilization and corresponding optimal frequency/voltage given in Table-I. CMAF [14, 15] monitors the real time CPU utilizations of the computing nodes. When there is change in the phase of the CPU utilization, the corresponding optimal frequency/voltage is chosen from Table-I. The job execution behavior and frequency/voltage of execution is analyzed with the energy consumption. By analyzing the job behavior and performing multiple runs of the application with varied frequency/voltage combinations, the optimal operating points are determined and stored in the knowledge base.

If the job is present in KB with the same input parameters, then a corresponding optimal voltage/frequency is selected. If the job is present with different input parameters, then the near optimal frequency is predicted based on decision tree approach [19, 20]. The set of rules are framed and near optimal frequency/voltage is predicted based on dynamic threshold value. The dynamic threshold value is based on similarity of job size. The DVFS knob is tuned to the predicted voltage and frequency.

Algorithm to select optimal frequency/voltage

/* Job $J_k=\{j_1(s_1,d_1),j_2(s_2,d_2),....,j_n(s_n,d_n)\}$
CPU_frequency = f_i, Voltage = v_i, Wall time = d_i
Knowledge Base *KB* = {
JobName = J_k, No_of_nodes = N_i,
Processing_Elements = PE_i,
Execution_Time = T_i, Size = s_i,
Optimal_Operating_Points(f_i,v_i) ← $O_i(f_i,v_i)$ }
MAX_FREQ ← 2.4 GHz
MAX_VOLT ← 19 Volts
*/
Build decision_ tree() using *KB*;
for job j_i ,*i* **from** *1* **to** *n* **do**
for JobName J_k,*k* **from** *1* **to** *m* **do**
compare j_i **and** J_k **in** *KB*;
if($j_i \neq J_k$ **) then**
set
{
frequency/voltage ← Based on *Table-I*
if(*change in CPU utilization phase***)**
Update the frequency/voltage accordingly;
}
store the results in *KB*;
else
$J_k(j_i, PE_i, s_i)$ → *decision_tree*(j_i, PE_i,s_i);
Optimal_Operating_Points(f_i,v_i) ← *decision_tree*(j_i, PE_i,s_i);
end else
end if
end for J_k
end for j_i

KB contains previously executed HPC jobs with their corresponding information. If an entirely new job arrives, then execute the job with the frequency/voltage based on Table-I to minimize energy consumption. Table-I contains the best measured frequency/voltage to reduce power consumption for various phases. If the submitted job matches with the job present in the KB, then the decision tree gives optimal operating points for execution.

TABLE I. CPU UTILIZATION PHASES AND CORRESPONDING OPTIMAL FREQUENCY/VOLTAGE

Phase	CPU Utilization (%)	$O_i(f_i,v_i)$
1	0-10	1.6, 12
2	10-20	1.73, 13
3	20-30	1.73,13
4	30-40	1.86,14
5	40-50	2.0,15
6	50-60	2.13,16
7	60-70	2.13,16
8	70-80	2.26,17
9	80-90	2.39,18
10	90-100	2.40,19

The optimal frequency/voltage for CPU utilization phases is determined by performing experimentation in the test bed cluster with various combinations using HPC benchmarks and real time applications such as LINPACK, NAS (IS, CG, BT, SP, LU), STREAM, Effective bandwidth benchmark (b_eff), Seasonal Forecasting Model (SFM).

Finding Optimal frequency/voltage from decision tree

/* PE_{REQ}= Required PE's
*/
decision_tree(j_i, PE_i,s_i)
{
if *(*($j_i = J_k$) **AND** ($s_i \leq s_k$) **AND** ($T_k \leq d_i$) **AND**
($PE_{REQ} \leq PE_i$)*)*
then
execute j_i **in** $O_i(f_i,v_i)$
else
if
Similarity($j_i(s_i)$) ≥ *dynamic_threshold_value; [21]*
return $O_i(f_i,v_i)$;
else
frequency/voltage ← Based on *Table-I*
if(*change in CPU utilization phase***)**
Update the frequency/voltage accordingly;
end else

end if
end else
end if
return $O_i(f_i,v_i)$;
}

Fig. 2 shows the Power aware scheduling framework for HPC cluster system. When a HPC cluster receives a job from a user, the resource manager decides to accept the job or not. If the job is accepted then the resource manager will place them in a queue. The acceptance or rejection of the newly submitted jobs will be based on meeting deadlines and QoS parameters. Resource manager tracks the availability of resources such as free nodes, CPU, memory etc. and communicates with an agent running on the compute nodes to obtain resource availability status. We developed a power aware algorithm for scheduling that smartly selects the job from the queue based on their energy profiles and the execution time. Scheduler is responsible for assigning jobs to the resources according to predetermined policies and resource availability. The details of the jobs such as CPU frequency, voltage, number of PE's used, energy consumption, execution time and input size are stored in the knowledge base by conducting several experimentations.

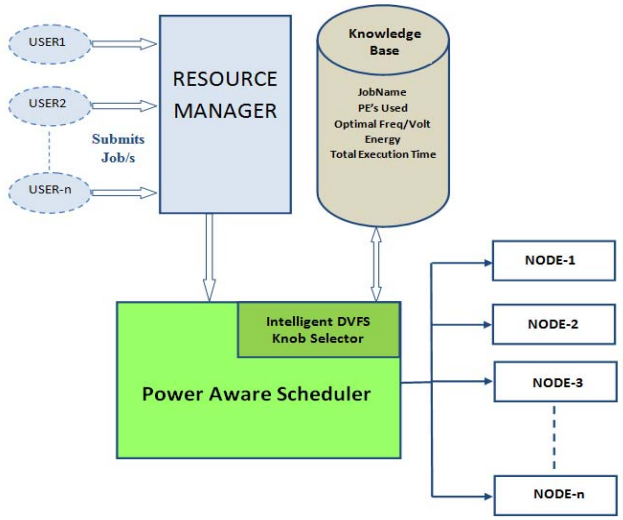

Fig. 2. Power Aware Scheduling Framework for HPC Cluster System

There are *'j'* jobs to be scheduled across N identical nodes. The shortest processing time has been taken in consideration for scheduling. We need to schedule all jobs across the nodes with minimal makespan (MK) which is very nearer to the optimal makespan (MK_{OPT}).

The optimal makespan is given by:

$$MK_{OPT} = \sum_{j=1}^{n}(T_i.j_i)\ /\mathrm{N}\ \ ,\mathrm{i} = 1,2,..,\mathrm{n}$$

The start time of the job is ST_j, completion time of the job is CT_j, and T_i is the execution time of the job. To schedule the jobs with proper load balancing, the following condition needs to be satisfied. Let MK_{OPT} be the optimal makespan of the node at any given instance,

$$(\ CT_j=ST_j+T_i) \le MK_{OPT}\ \forall\ j\in J\ \ \textbf{and}$$

$$MK_{OPT} \ge T_i\ \forall\ j\in J\ .$$

The Job is assigned to a node based on shortest processing time and executes it within the makespan. The makespan is adjusted dynamically based on incoming jobs for all iteration. Here x_i is a variable indicates whether job j_i is assigned to node N_i. The objective is to minimize the makespan and load balancing among all computing nodes dynamically. The first set of constraints ensures that each job is scheduled on one of the nodes and the second ensures that each node has makespan of at most *'t'*.

Minimize *'t'*, Subject to:

$$\sum_{i\in N} x_i = 1\ \ ,if\ job\ j\ is\ selected, else\ 0.$$

$$\sum T_i.x_i \le \mathrm{t}\ ,\quad \mathrm{i}\in \mathrm{N},$$

Such that:

$$\sum T_i.x_i \cong MK_{OPT}\ ,\quad \mathrm{i}\in \mathrm{N},$$

$$x_i \in \{0,1\}$$

Job Selection Algorithm

/* Job $j_i=\{j_1(s_1,d_1),j_2(s_2,d_2),....,j_n(s_n,d_n)\}$
Execution Time $T_i=\{T_1,T_2,....,T_n\}$
N = Total number of nodes.
N_A=Number of nodes available. */
Step 1: **Initialize** $j_i \leftarrow \{1,2,....n\}$
for j_i, i =1 **to** n **do**
for $\mathrm{N_k}$, k =1 **to** N **do**
Step 2: Calculate the optimal time of makespan
$MK_{OPT} = \sum_{i\in j_i} T_i\ /N_A$
Step 3: **Minimize** $MK_{OPT} - \sum_{i\in j_i} T_i\ .x_i$
$x_i \in \{0,1\}$
Step 4: **Set** $A_k = \{i | x_i = 1, i \in j_i\}$
where $\{x_i | i \in j_i\}$ is from step 3.
$j_i = j_i - A_k$
end for
end for
return A_k;

Each job will have the execution time associated with it, which is calculated by using the knowledge base [20]. After running this algorithm, we get $\mathrm{A_k}$ which is the job chosen for execution at node N_k.

Node Selection Algorithm

/* $A_k(\mathrm{PE_{REQ}}, \mathrm{T_i})$
$\mathrm{N_k}$ = Nodes available for execution
$\mathrm{PE_{REQ}}$= PE's required for job execution.
$\mathrm{PE_{AVL}}$= Available PE's in node $\mathrm{N_k}$
*/
Step 1: **Initialize** $\mathrm{A_i} \leftarrow \{\mathrm{A_1},\mathrm{A_2},....,\mathrm{A_n}\}$, $\mathrm{PE_{AVL}}$=x.
Step 2: Sort $\mathrm{A_i}$ based on least executing time.
Step 3: **for** $\mathrm{A_i}$, i=1 **to** n **do**

for N_k, k=1 **to** m **do**
if ($PE_{REQ} \leq PE_{AVL}$)
schedule A_i to N_k
$PE_{AVL} = PE_{AVL} - PE_{REQ}$
else
place A_i in the job queue
end if
end for
end for

V. Experimental Results

This section demonstrates the PAAS results on our test bed. We have performed several experiments on various HPC jobs for different voltage and frequencies, different number of PE's with different workloads and stored the results in knowledge base. We have used phc_intel module for scaling voltage levels and acpi-cpufreq is used to set frequencies. PHC (Processor Hardware Control) is an acpi-cpufreq patch built with the purpose of enabling undervolting on the processor [22]. The servers support the voltages {12,13,14,15,16,17,18,19 volts} and frequencies {1.6,1.73,1.86,2.0,2.12,2.26,2.39,2.4GHz}. The sample details of the information stored in the knowledge base is given in Table-II. The details stored are for the defined problem size. The execution time and energy consumption varies based on the problem size. Here, we have considered a defined problem size for illustration purpose.

TABLE II. Energy Profile in Knowledge Base

J_i	PE_i	T_i (sec)	E_i (Ws)	$O_i(f_i, v_i)$
Linpack	16	401	173500	2.3,18
Linpack	4	403	131750	2.39, 18
SFM	4	166	160650	2.39, 17
SFM	8	201	148146	2.39, 18
SFM	12	264	124721	2.39, 16
SFM	16	544	139187	2.39, 18
NAS-ft	4	24	4149	2.26, 17
NAS-cg	4	30	5122	2.0, 15
NAS-lu	4	72	12538	2.39, 18

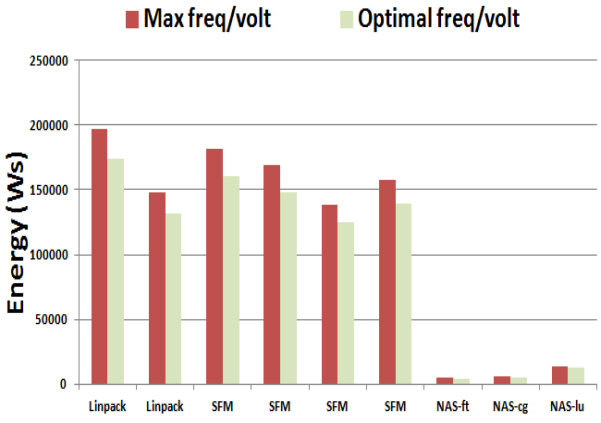

Fig. 3. Energy comparision for Maximum freq/volt and optimal freq/volt

Fig. 3 presents the differences in energy consumption of various HPC jobs executing at maximum freq/volt versus jobs running at optimal freq/volt. The jobs running at optimal freq/volt shows significant reduction in energy consumption ranging from 11-14%. Table-III shows the maximum freq/volt and optimal freq/volt for execution with its corresponding energy consumption.

TABLE III. Energy Reduction Using Optimal Freq/Volt

J_i	E_i (Ws) [Max(f_i,v_i)] 2.4GHz,19v	E_i (Ws) [$O_i(f_i,v_i)$]	$O_i(f_i, v_i)$	Energy reduction (%)
Linpack	196922	173500	2.3,18	13.5
Linpack	147560	131750	2.39, 18	12
SFM	181534	160650	2.39, 17	13
SFM	168886	148146	2.39, 18	14
SFM	138752	124721	2.39, 16	11.25
SFM	157281	139187	2.39, 18	13
NAS-ft	4729	4149	2.26, 17	14
NAS-cg	5736	5122	2.0, 15	12
NAS-lu	13917	12538	2.39, 18	11

The time sharing and space sharing model of job execution have been taken into consideration for load balancing among the nodes while scheduling the jobs. The job is allocated to particular node based on Node Selection Algorithm. Table-IV lists the jobs in the queue to be allocated for execution.

TABLE IV. Jobs in Queue

Jobs (J_i)	PE_{REQ}	T_i	$O_i(f_i, v_i)$
J_1	4	32	2.39, 17
J_2	4	20	2.39, 16
J_3	8	20	2.39, 16
J_4	8	20	2.26, 17
J_5	4	26	2.26, 17
J_6	12	24	2.39, 18
J_7	4	30	2.13, 16
J_8	4	45	2.0, 15

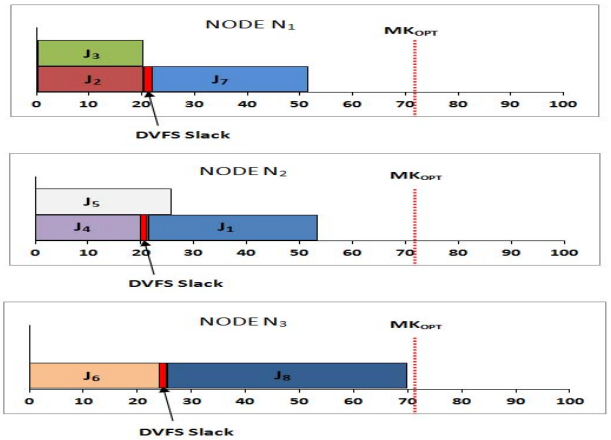

Fig. 4. Task graph of jobs.

Fig. 4 shows the task graph with optimal makespan for the jobs listed in Table-III. In this task graph, all jobs have completed its execution before MK_{OPT}, where MK_{OPT} represents the optimal makespan time. When the parallel jobs are executing, the frequency/voltage of the cores is controlled by using core based DVFS knob [23].

VI. Conclusions and Future Work

In this paper, we presented power aware scheduling based on optimal frequency/voltage. The optimal frequency/voltage is predicted based on the energy profiles of various HPC jobs available in the knowledge base using decision tree approach. PAAS also deals with load balancing across the nodes to minimize energy consumption. The makespan scheduling is implemented in both time and space sharing mode of execution based on availability and requirements. The makespan based schedule gives better utilization as well as energy saving. The observed reduction in energy consumption by using PAAS is 11% - 14% with an average measured energy savings of 12.64% and a maximum of 13.5%.

In future, we planned to validate the experimental results of PAAS on larger HPC cluster and in heterogeneous environment.

Acknowledgment

The authors would like to thank Dr. Sarat Chandra Babu, Prof. Soumyendu Raha, Prof. T. Matthew Jacob, R.K.Senthil Kumar, B.Jayanth, Bhavyasree Unni, Nazia Parveen, Manisha Chauhan, Ramesh Naidu and Mohit Ved for their support and valuable help while conducting this research.

References

[1] Top500 Supercomputers [Online]. Available : http://www.top500.org/list/2011/11/100, [Accessed: May-2013].

[2] J. Hikita, A. Hirano, and H. Nakashima, "Saving 200kW and $200K/year by Power aware Job/Machine Scheduling," IEEE International Symposium Parallel & Distributed Processing, 2008.

[3] K. H. Kim, R. Buyya, and J. Kim, "Power Aware Scheduling of Bag-of Tasks Applications with Deadline Constraints on DVS-enabled Clusters," Seventh IEEE International Symposium on Cluster Computing and the Grid, CCGRID, 2007.

[4] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power Optimization of Variable-Voltage Core based Systems," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 12, pp.1999.

[5] P. M. Alvarez, E. Levner, and D. Mosse, "Power Optimized Scheduling Server for Real-Time Tasks," Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02).

[6] A. Cabrera, F. Almeida, V. Blanco, and D. Gimenez, "Analytical modelling of the energy consumption for the High Performance Linpack," 21st Euromicro International Conference on Parallel, Distributed and Network based processing, 2013.

[7] A. S. Araujo, C. A. S. Camargo, G. G. H. Cavalheiro, and M. L. Pilla, "Towards a power-aware application level scheduler for a multithreaded runtime environment," 22nd International Symposium on Computer Architecture and High Performance Computing, 2010.

[8] X. Ruan, X. Qin, Z. Zong, K. Bellam, and M. Nijim, "An Energy-Efficient Scheduling Algorithm using Dynamic Voltage Scaling for Parallel Applications on Clusters," Proceedings of the 16th IEEE International Conference on Computer Communications and Networks (ICCCN), Hawaii, 2007.

[9] J. M. Pierson and H. Casanova, "On the utility of DVFS for Power-Aware Job placement in Clusters," Proceeding of the 17th International Conference on Parallel Processing, Euro-Par 2011.

[10] L. Wang, G. Laszewski, J. Dayal, and F. Wang, "Towards Energy Aware Scheduling for precedence constrained Parallel Tasks in a cluster with DVFS," Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID 2010.

[11] L. Wang, S. U. Khan, D. Chen, J. Kolodziej, R. Ranjan, C. Xu, and A. Zomaya, "Energy aware parallel task scheduling in a cluster," ELSEVIER - Future Generation Computer Systems, 2013.

[12] M. Etinski, J. Corbalan, J. Labarta, and M. Valero, "Power-Aware Parallel Job Scheduling," First International Workshop on Energy Efficient High-Performance Computing (EEHiPC) Held in Conjunction With 17th IEEE/ACM International Conference on High Performance Computing (HiPC), 2010.

[13] M. Etinski, J. Corbalan, J. Labarta, and M. Valero, "Utilization driven Power-Aware Parallel Job Scheduling," Springer-Verlag Computer Science-Research and Development, Volume 25, Issue 3-4, pp 207-216, 2010.

[14] S.Venkatesh, B. S. Bindhumadhava and A. A. Bhandari, "Implementa-tion of automated Grid software management tool: A Mobile Agent based approach," Proceedings of International Conference on Information and Knowledge Engineering, pages 208-214, June 2006.

[15] H. V. Raghu, Ankit Kumar, and B. S. Bindhumadhava, "High Performance Systems: An Agent based Application Power Profiling," 18th annual International Conference on Advanced Computing and Communications ADCOM - 2012.

[16] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, et al. "Leakage current: Moore's law meets static power," Power Aware Computing, Published by IEEE Computer Society, vol. 36, no.12, December 2003.

[17] T. Kolpe, A. Zhai, and S. S. Sapatnekar, "Enabling Improved Power Management in Multicore Processors through Clustered DVFS," In proceeding of Design, Automation and Test in Europe, France, pp.293-298, Mar 2011.

[18] R. Ge, X. Feng, H. Pyla, K. Cameron, and K. W. Feng, "Power Measurement Tutorial for the Green500 List," June 27, 2007.

[19] 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.

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

[21] Jirong Li, "Dynamically Threshold Value Determination in the Optimal Fuzzy-Valued Feature Subset Selection," 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2012.

[22] Processor Hardware Control https://wiki.archlinux.org/index.php/PHC, [Accessed: June-2013].

[23] T. Kolpe, A. Zhai, and S. S. Sapatnekar, "Enabling Improved Power Management in Multicore Processors through Clustered DVFS," In proceeding of Design, Automation and Test in Europe, France, pp.293-298, Mar 2011.

[24] A. K. Coskun, R. Strong, D. Tullsen, and T. Rosing. "Evaluating the Impact of Job Scheduling and Power Management on Processor Lifetime for Chip Multiprocessors," Proceedings of SIGMETRICS 2009: The 11th ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.169-180, June 2009.

[25] C. Augonnet, S. Thibault, R. Namyst, and P .A. Wacrenier, "StarPU: a unified platform for task scheduling on heterogeneous multicore architectures," published in "Concurrency and Computation: Practice and Experience", pp.187-198, 2011.

[26] Y. Ma, B. Gong, and L. Zou, "Energy-Efficient Scheduling Algorithm of Task Dependent Graph on DVS-Unable Cluster System," 10th IEEE/ACM International Conference on Grid Computing,2009.