A HETEROGENEOUS DYNAMIC SCHEDULING MINIMIZED MAKE-SPAN FOR ENERGY AND PERFORMANCE BALANCING on *Sipht*

Saba Fatima[1], V. M. Vishwanath[2]
Department of Electronics and Communication Engineering[1,2]
Vivesvaraya Technological University, Belagavi, India[1],
S. L. N. College of Engineering, Raichur, India[2],
fatima.saba635@gmail.com[1]
vmviswanatha@gmail.com[2]

Abstract

GPU-enabled heterogeneous cloud computing system has gained immense popularity in real time due to its faster execution and adaptive processing. It can achieve the primary objectives such as high-energy conservation and performance balancing of heterogeneous cloud computing systems. In order to exploit the heterogeneous systems, it is very essential to achieve a high balancing between host CPUs and GPU accelerators to save energy and faster resource allocation in cloud computing environment. Therefore, here, we have introduced a Heterogeneous Dynamic Scheduling Minimized Make-span ($HDSMM$) using CPU-GPU core architecture for heterogeneous cloud computing systems to attain a trade-off between energy consumption and performance of information processing centers and increase execution time. To reduce high-energy consumption in various information processing centers in this research an effective modeling is designed. Experimental results verify superiority of our proposed $HDSMM$ model in terms of power consumption, average power and power sum.

Keywords*—GPU; Cloud Computing; Energy consumption.*

I. INTRODUCTION

In recent time, heterogeneous cloud computing applications inevitably becomes the main focus in various software industries such as Google, Amazon, Microsoft and Facebook etc. due to its faster computation, high storage capacity, high demand and 'pay-per-go' model. Moreover, heterogeneous cloud computing becomes an integral part of telecommunications, healthcare departments, pharmaceutical industries, Internet search, financial and business informatics. The heterogeneous cloud computing information processing centers gives user-required vital information, numerous resources and instantaneous scalability. Thus, this heterogeneous cloud computing applications becomes very essential in today's market and also can be very valuable application for future.

However, high demand of heterogeneous cloud computing applications can led to various problems such as improper resource utilization, high power consumption and performance degradation. Thus, a high-performance computing (HPC) application has taken immense progressin countering these issues using various deep learning frameworks [1], [12], [13]. This technique helps to provide proper resource utilization and adaptive computing to handle various tasks at a time. However, in last decade and so, various problems occurred in this HPC technique, such as large power consumption and hazardous to green computing. Therefore, in recent years, various researchers have provided different techniques to control these drawbacks like Constrained Earliest Finish Time (CEFT) technique [2],Hierarchical

Reliability-Driven Scheduling (HRDS) algorithm [3],Voltage and Frequency Island (VFI) algorithm [4], Contention-Aware Energy-efficient duplication (FastCEED) algorithm [5], Dynamic Voltage and Frequency Scaling (DVFS) [6], [14], [15].

Dynamic Voltage and Frequency scheduling (DVFS) Technique is one of the most famous methodologies in above mentioned all the techniques that handle numerous resources, reduces high-energy consumption and remains environmental friendly. In this DVFS technique, chip voltage is scaled down to decrease the energy consumption in cloud computing information centers. The two essential and integral parts of these techniques are CPUs and GPUs. Thus,DVFS can be termed as CPU-DVFS and GPU-DVFS where for the computation and increment in QOS both CPU-GPU cores are, utilized in cloud computing devices. However, CPU-DVFS provides unsatisfactory results in some cases due to its improper resource allocation and sluggish speed can degrade the performance of cloud information centers.

Thus, in recent years, GPU-enabled DVFS techniques have performed exceptionally well due to its parallel and high speed computation. In terms of resource utilization and performance of the system, GPU-enabled DVFS techniques perform better than the CPU enabled techniques. For example, various software companies Amazon, Microsoft Azure, Google, IBM has its own cloud services that works on the GPU core architecture. GPU computing is a self-governing task entity is, used to perform tasks and it works as a run-time platform for materialistic applications.

However, one must be attentive of the drawbacks such as high power consumption, imbalance between energy conservation and performance and improper resource allocationoccurs in utilizing existing CPU and GPU enabled DVFS techniquescan reduce the performance of the system. Therefore, to counter these issues, numerous researchers have shown their interest in this field and provided some essential work for faster GPU computing [16]. In [7], [8] and [17], a self-governing task scheduling technique adopted based on moldable task model with GPU core architectures. It provides high computational efficiency and proper task scheduling. An efficient task scheduling approach is adopted in [9], [10] and [11],for proper resource utilization and reduce energy consumption. However, the optimization problem and high computational complexity take place in these techniques.

Therefore, to reduce these existing optimization, computational complexity, power consumption and resource allocation problem, here, we have presented a Heterogeneous Dynamic Scheduling Minimized Make-span ($HDSMM$) using CPU-GPU core architecture for heterogeneous cloud computing systems to attain a trade-off between energy consumption and performance of information processing centers and increase execution time. The balancing between CPU and GPU-enabled DVFS techniques is very much essential to maintain a balance between energy consumption and high speed, which further helps to increase the performance of the system. Here, we have utilized $CloudSim$ simulator in our proposed $HDSMM$ model.

The contribution of work is as discussed below:

Here, we have presented a Dynamic Scheduling Minimized Make-span ($HDSMM$) using CPU-GPU core architecture for heterogeneous cloud computing information centers. A balancing between CPU DVFS and GPU DVFS are, attained to provide faster run time with parallel execution as well as minimization in energy consumption. There are two scheduling schemes such as static and dynamic scheduling which are utilized in GPU-accelerated information processing centers. The total task load of the cloud information processing center can be sub-divided into CPU and GPU processed tasks. The type of processor used decides the characteristics of a task load. Precisely, GPU enabled DVFS can handle tasks in a much

better way than a CPU enabled DVFS. The most essential objective of the proposed $HDSMM$ model is to decrease the net energy consumption in cloud information processing centers which is termed as the sum of $static\ and\ dynamic$ energy and performance enhancement of our model. $CloudSim$ Simulator is a type of toolkit to model and simulate cloud computing devices. It also provides the permission to describe the characteristics of information processing centers and their hosts, offered memory, network topology used and patterns for utilization of information processing centers. Our proposed $HDSMM$ model provides better outcomes in terms of execution time and power consumption than any other state-of-art-techniques.

This research work is categorized in following sections: Section IIdescribes our proposed methodology. Section III discusses the experimental results and evaluations and section IV concludes the paper.

II. HETEROGENEOUS DYNAMIC SCHEDULING MINIMIZED MAKE-SPAN ($HDSMM$)ARCHITECTURE

This section briefly defines the proposed $HDSMM$ architecture and its various building blocks. This section shows the proposed architectural diagram that defines proposed $HDSMM$ model. Here, a precise modeling is presented for reduce energy consumption, task load reduction and for proper resource distribution. In proposed $HDSMM$ model, the DVFS technique is, partitioned into CPU-DVFS and GPU-DVFS. CPUs and GPUs connections and partition of task loads between these CPUs and GPUs are, demonstrated in figure 1. Here, we have presented a Heterogeneous Dynamic Scheduling Minimized Make-span ($HDSMM$) Architectural for cloud computing heterogeneous system. A central resource scheduler used to handle resource allocation between CPUs and GPUs. A GPUs enabled parallel computing is used in our proposed $HDSMM$ architecture with much faster computational rate. Whenever a fresh task load is assigned, then all the task load is distributed into CPUs and GPUs tasks. A message passing scheme is required for communication between clusters. In our model, high computational efficiency, less energy consumption and efficient task distribution between CPU DVFS as well as GPU DVFS are key topics whose detailed modeling is presented below for heterogeneous cloud computing systems.

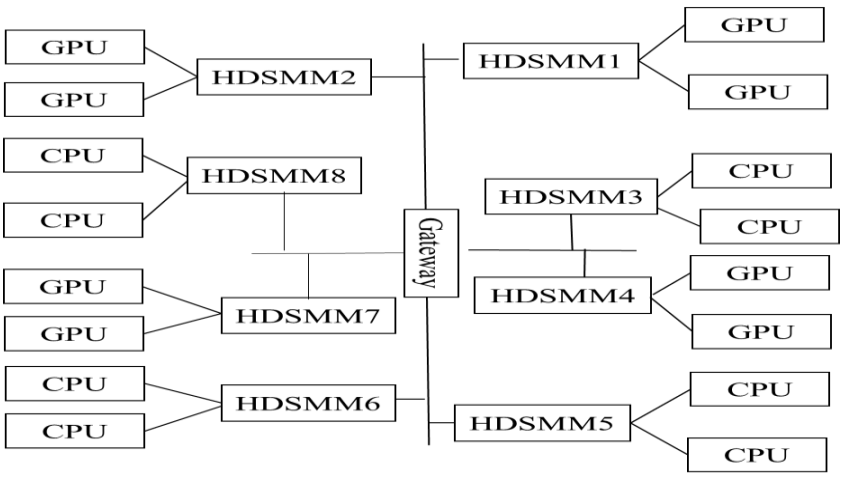

Figure 1: Architecture of our proposed $HDSMM$ model

A. Modeling for Adequate Task-load Reduction:

In this segment, an efficient task load reduction modeling is presented. In our proposed $HDSMM$ model, every job is distributed into sub-tasks. Our $HDSMM$ architecture consists of various stages where these stages either of distinct category or a consecutive group of distinct category. Let the overall number of tasks to be executed is expressed as R. Entire task load is partitioned into two different groups M and N. In proposed $HDSMM$ model, group M shows sub-tasksγ which is assigned to host CPU and group$N$ shows sub-tasksγ which is assigned to GPU. Then, the task computation is represented as,

$$R = R_M + R_N \tag{1}$$

$$R_M = \gamma . R \tag{2}$$

$$R_N = (1 - \gamma). R \tag{3}$$

Here, GPU processedsubtasks set can be expressed as γ and it ranges from 0 to 1. The allocated tasks to CPUs and GPUs can be denoted as R_M andR_N. The entire processing capabilities of CPUs and GPUs can be explored using the processing rate which is denoted as C for a specific task load R. The processing rate C, can be termed as the rate at which subtasks for a specified task load R can be finished in one sec. Subtasks for a specified task-load R can be defined as either same type or different and they can have their own processing units. In proposed $HDSMM$ model, the processing rate is dependent on the specified task load R and it can be either different or remain constant as well for similar type of tasks if the available processing units are capable of saturating processing.

Let the processing rates for heterogeneous devices, CPUs and GPUs can be denoted as C, C_M, C_N. For heterogeneous devices, CPUs and GPUs, the run-time can be expressed as D, D_M, D_N respectively. Then, execution time for heterogeneous devices, CPUs and GPUs can be expressed as following,

$$D_M = R_M . (C_M)^{-1} \tag{4}$$

$$D_N = R_N . (C_N)^{-1} \tag{5}$$

$$D = \uparrow (D_M, D_N + D_G) \tag{6}$$

Where, G represent a time-delay which is occurred wheneversub-tasks which are processed using GPU accelerators are offloaded and defined using equation (6). This time-delay Gcan occurs due to the time needed for sending computational information and activate kernel. Whenever, the execution time D_N of GPUs is much higher than time-delay G, then it is, discarded if it is not harming the model accuracy. Therefore, the entire execution time is, expressed as the summation of CPUs and GPUs time as well as the offloading delay time. Then the overall processing rate can be of heterogeneous devices can be computed as,

$$C = R.(D)^{-1} = \left[\uparrow \left((-\gamma).(C_M)^{-1}, (\gamma . (C_N)^{-1} + D_G.(R)^{-1})\right)\right]^{-1} \tag{7}$$

Where, assume that the execution time of GPUs are much higher than time-delay D_G and can be discarded and then, the equation (7) can be again defined as,

$$C = R\,.(D)^{-1} = \left[\uparrow\left((1-\gamma).(C_M)^{-1}, (\gamma\,.(C_N)^{-1})\right)\right]^{-1} \quad (8)$$

B. Modeling for Energy Consumption Control:

This section provides an efficient modeling for the controlling of high-energy consumption in information processing centers. The energy consumption in heterogeneous devices is distributed into three segments which is shown in following equation (9),

$$\mathbb{E} = \mathbb{E}_M + \mathbb{E}_N + E_H \quad (9)$$

Where, equation (9) represents the energy consumption of three different segments like host CPUs as M, GPU accelerators as N and host memory and its other essential portions as H. The overall energy can be of two types such as static energy $\mathbb{E}_A$ and dynamic energy$\mathbb{E}_B$. Here, Static energy $\mathbb{E}_A$ is independent of task load due to it carries task load free components whereas dynamic energy $\mathbb{E}_B$depends on the task load. The Static energy $\mathbb{E}_A$remain same throughout if all processing units works at a certain speed. A single static energy $\mathbb{E}_A$ model can be made if all the distinct type of static energies are combined together.

$$\mathbb{E}_A = \mathbb{E}_{A_M} + \mathbb{E}_{A_N} + E_{A_H} \quad (10)$$

The high power consumption can occur due to several reasons, in which some reasons are presented as follows,

i. For the overall run time D, the system can drain till static energy$\mathbb{E}_A$,
ii. For the CPU runtimeD_M, the system can drain till dynamic energy $\mathbb{E}_{B_M}$,
iii. For the GPU run timeD_N, the system can drain till dynamic energy $\mathbb{E}_{B_N}$,
iv. Whenever, GPUs required more time to finish its task i.e. $D_N > D_M$ then the host CPU needed additional power $\mathbb{E}_G$ to handle GPUs along with CPU static energy $\mathbb{E}_{A_M}$. Then, the entire energy model can be represented as the summation of four energy modules which are described in above section and can be expressed as,

$$\mathbb{P} = D.\mathbb{E}_A + D_M.\mathbb{E}_{B_M} + D_N.\mathbb{E}_{B_N} + (D_N - D_M).\mathbb{E}_G \quad (11)$$

$$\beta = R.(\mathbb{P})^{-1} = \mathbb{E}_A.\uparrow\left[\left((1-\gamma).(C_M)^{-1}, \gamma.(C_N)^{-1}\right) + \left\{(1-\gamma).\mathbb{E}_{B_M}.(C_M)^{-1}\right\} + \gamma.\mathbb{E}_{B_N}.(C_N)^{-1} + \mathbb{E}_G.\uparrow(\gamma.(C_N)^{-1} - \{(1-\gamma).(C_M)^{-1}\}, 0)\right] \quad (12)$$

Where, all the modules of equation (11) can be measured straightly and this equation (11) shows the total energy of the model. The efficiency of energy β can be expressed as ratio of task load R to the energy $\mathbb{P}$ in task operations/ joule. The average processing rate to the average power ratio can be described as the task operations/ watts or G-flops/ watts. The complete energy efficiencyβ can be described with the help of equation (4) and (11) which is demonstrated in equation (12).

Here, equation (12) defines the complete energy efficiency β . For simplification, assume that the offloading delayD_G is very less in comparison with D_N and can be discarded. The complete energy efficiency β rely upon various factors such as computational distribution γ, static energy $\mathbb{E}_A$of the model, performance of the model, components and energy distribution of GPUs and CPUs.

Here, β_N and β_M represents the GPUs and CPUs energy efficiency. Then, the individual GPUs and CPUs efficiencies can be expressed as,

$$GPUs = C_N.\left(\mathbb{E}_{B_N}\right)^{-1} \quad (13)$$

$$CPUs = C_M.\left(\mathbb{E}_{B_M}\right)^{-1} \quad (14)$$

In our proposed $HDSMM$ model, DVFS is portioned into host CPU and GPU accelerators i.e. CPU DVFS and GPU DVFS. This DVFS technique is familiar and eminent technique to reduce power consumption in cloud computing environment. A special voltage-frequency pair is used in power states and the processing units which are connected with DVFS gives a support to power states. This power states are mostly responsible for the high energy consumption. If these power states remains high then it needs larger processing rates as well as high frequencies which can cause larger power consumption. For CPU DVFS, the model parameters $C_M, \mathbb{E}_{A_M} and\ \mathbb{E}_{B_M}$ are the elements of CPU frequency. Likewise, For GPU DVFS, the model parameters $C_N, \mathbb{E}_{A_N} and\ \mathbb{E}_{B_N}$ are the elements of CPU frequency.The DVFS-enabled GPUs can simply switch between performance or power states. For example, the Tesla K20 which is a DVFS-enabled GPUprocessor, can maintain up to six power or performance states and can easily switch with each other, according to the requirement. In proposed $HDSMM$ model, both processing speed and power are elements of CPU and GPU frequencies. The total energy, energy efficiency and performance are the elements of $\gamma, \mathbb{f}_M\ and\ \mathbb{f}_N$. From this above equations and description, it is verified that the elements like $C_N, C_M, \mathbb{E}_{B_M}\ and\ \mathbb{E}_{B_N}$ changes whenever the certain assigned frequency of the model changes.

III. PERFORMANCE EVALUATION

The exploitation of heterogeneous cloud computing systems is enormously enhanced in recent years due to extensive use of their cloud computing resources for the internet itself. These cloud computing resources consist of various digital equipment's, software, gadgets and networking tools etc. However, the demand is very high in contrast to the available cloud resources and power consumption in information processing centers also become very high due to the extensive utilization of these cloud resources which can degrade the performance of the system. Therefore, the performance of heterogeneous cloud computing systems must be enhance in order to handle high demand of cloud resources by various subscribers all over the world. A well-known conventional CPU-GPU DVFS approach can be utilized to achieve a trade-off between performance and large energy consumption for heterogeneous cloud devices. Therefore, here, we have adopted a Heterogeneous Dynamic Scheduling Minimized Make-span ($HDSMM$) using CPU-GPU cores for heterogeneous cloud devices to accomplish a trade-off between energy consumption and performance of the model. In proposed $HDSMM$model, different jobs are usedlike 30, 50, 100, and 1000 to evaluate execution time. In the following sections, Power sum, average power and energy consumption outcomes demonstrated in graphical form. A $Sipht$ scientific datasetis utilized to verify proposed $HDSMM$ architecture. In proposed $HDSMM$ model, various job sizes as 30, 60,100 and 1000 are considered using $Sipht$ scientific dataset. The proposed $HDSMM$ model simulated on 64-bit windows 10 OS with 16 GB RAM which contains an INTEL (R) core (TM) i5-4460 processor.It contains 3.20 GHz CPU. This project is simulated using $EclipseWS$ Neon.3 editor and code is written in JAVA.

a) Comparative Study:

In recent years, heterogeneous systems has received high praise from all over the world in different fields like healthcare solutions, software industries, trading and medical applications etc. Therefore, cloud computing has started to add heterogeneity support to cope up with extensive demand. Furthermore, to enhance the efficiency of heterogeneous systems for future use, GPU instances are favored in contrast to traditional CPU-based cloud resources. However, high energy consumption and inappropriate resource allocation techniques causes high degradation in performance of cloud computing systems. Subsequently, to ensure proper resource allocation, less energy consumption and trade-off between high efficiency and power consumption, here, we have presented a Heterogeneous Dynamic Scheduling Minimized Make-span ($HDSMM$) using CPU-GPU cores for heterogeneous cloud computing devices. Our proposed $HDSMM$ model helps to boost throughput and performance of the heterogeneous systems and provide effective adaptive resource scheduling. Here, we have conducted various experiments using the proposed $HDSMM$ model to find energy consumption,power sum and average power results which are demonstrated in table 1 with the help of $Sipht$ scientific dataset for various jobs 30, 50,100 and 1000. Our proposed $HDSMM$ technique ensures very less energy consumption for $Sipht$ scientific dataset for $\text{Sipht}\,30$ is 2812.991014 Watts, $\text{Sipht}\,60$ is 3158.219947 Watts,$\text{Sipht}\,100$ is 3174.261302 Watts and $\text{Sipht}\,1000$ is 11211.22691 Watts demonstrated in table 1 which is highly reduced compare other state-of-art techniques using similar statistics. Table 1 also demonstrates Execution time to finish the task using the proposed $HDSMM$ technique for various jobs as 30, 50,100 and 1000 with the help of $Sipht$ benchmark. The average power outcomes for $\text{Sipht}\,30$ is 21.99945901 W, $\text{Sipht}\,60$ is 21.9994593 W,$\text{Sipht}\,100$ is 21.9994591 W and $\text{Sipht}\,1000$ is 21.99946127 W.

b) Graphical Representation:

This section provides graphical representation of our simulated experiments for various jobs using $Sipht$ scientific dataset and compared with traditional state-of-art-techniques in terms of average power, energy consumption and power sum. Here, figure 2 shows Power Sum results in contrast to DVFS technique using proposed $HDSMM$ model for scientific dataset Sipht for different jobs as 30, 50,100 and 1000.Here, figure 3 shows Average Power results in contrast to DVFS technique using proposed $HDSMM$ model for scientific dataset Sipht for different jobs as 30, 50,100 and 1000.Here, figure 4 shows Energy consumption results in contrast to DVFS technique using proposed $HDSMM$ model for scientific dataset Sipht for different jobs as 30, 50,100 and 1000.these outcomes concludes the superiority of proposed $HDSMM$ prototype in terms of average power, power consumption and power sum using $Sipht$ scientific dataset.

Table 1: Various parameters comparison for proposed $HDSMM$ technique vs DVFS using scientific model $Sipht$

Parameters	**DVFS**				***HDSMM***			
	***Sipht*30**	***Sipht*60**	***Sipht*100**	***Sipht* 1000**	***Sipht*30**	***Sipht* 60**	***Sipht*100**	***Sipht*1000**
	VM=30	VM=30	VM=30	VM=30	VM=30	VM=30	VM=30	VM=30
Power Sum (W)	10734800.61	22566020.59	33620561.59	335969269.8	9783513.615	10291173.27	9934905.132	16022788.43
Average Power (W)	28.65571784	28.6557203	28.65572104	28.65572239	21.99945901	21.9994593	21.9994591	21.99946127
Power Consumption (Wh)	4367.658563	11228.74085	20813.04776	1070996.931	2812.991014	3158.219947	3174.261302	11211.22691

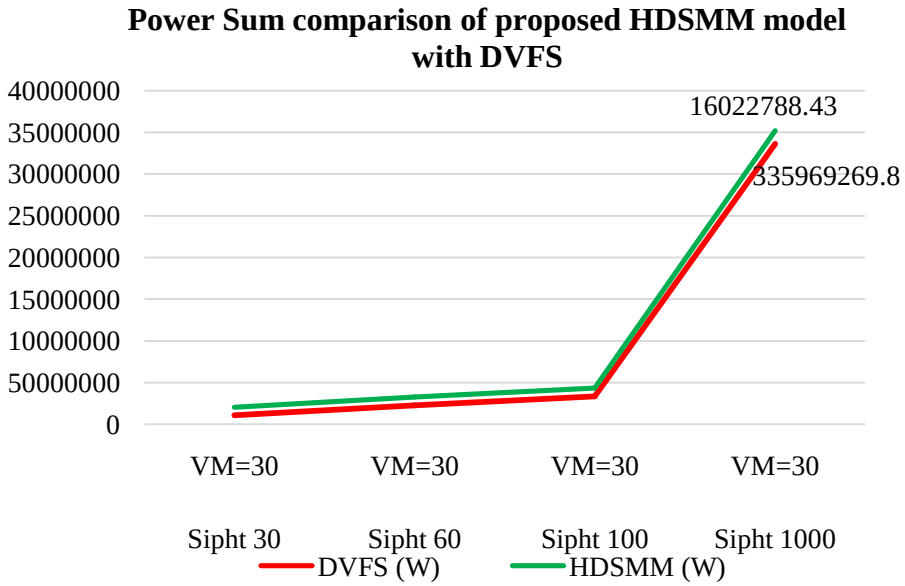

Figure 2: Power Sum comparison using our *HDSMM* technique with DVFS on *Sipht*

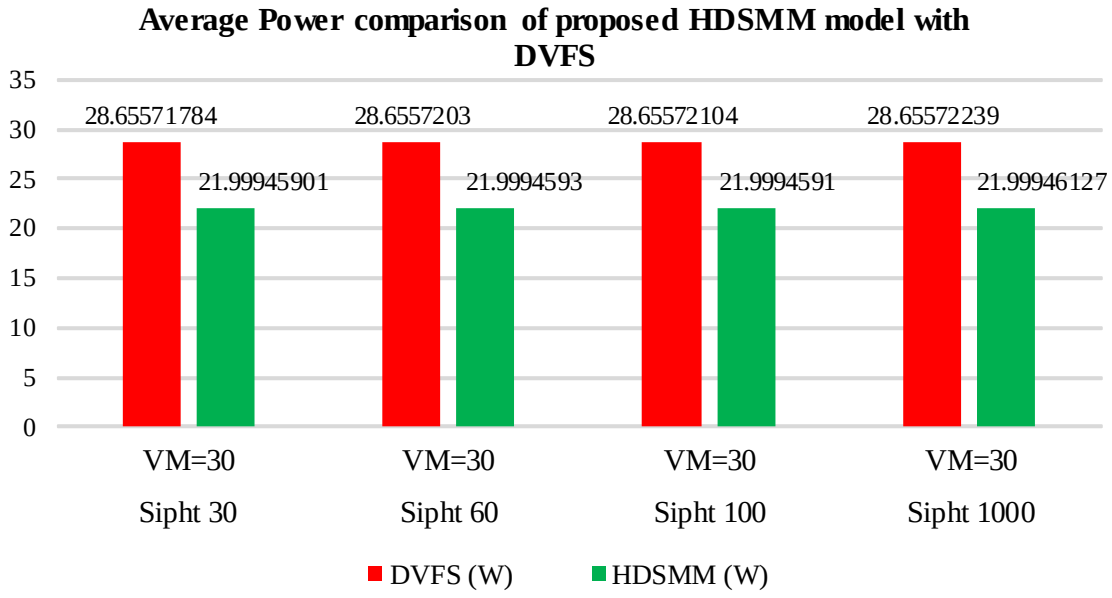

Figure 3:Average Power comparison using our HDSMM technique with DVFS on *Sipht*

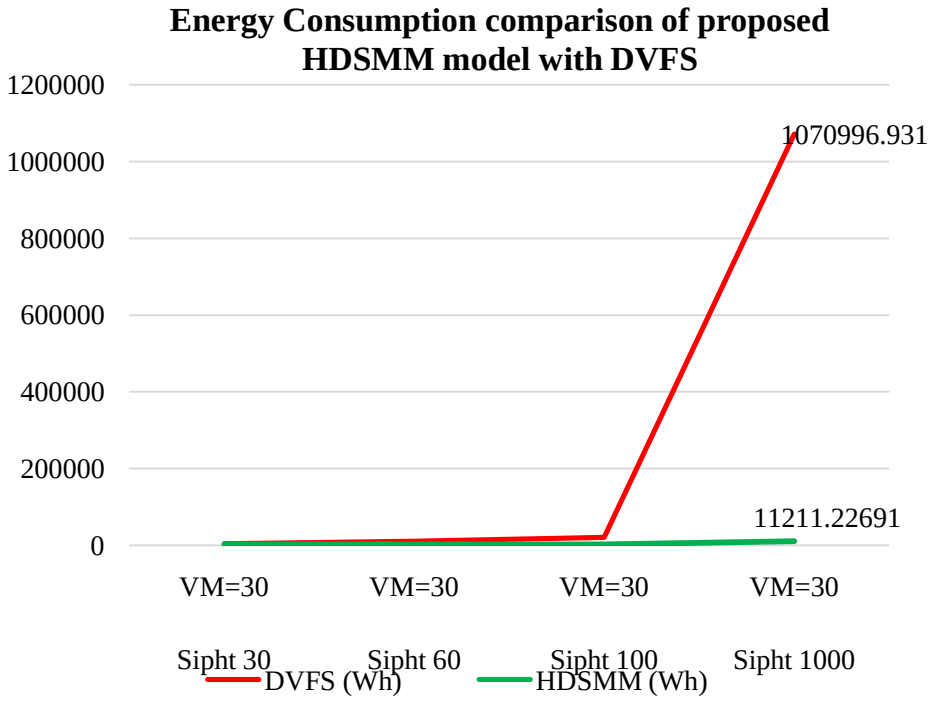

Figure 2: Power Consumption comparison using our *HDSMM* technique with DVFS on *Sipht*

IV. CONCLUSION

The importance of controlling high power consumption and achieving high tradeoff between efficiency and energy consumption and effective resource allocation is very essential for heterogeneous systems. Therefore, here, we have presented a Heterogeneous Dynamic Scheduling Minimized Make-span (*HDSMM*) using CPU-GPU core architecture for heterogeneous cloud computing systems to reduce energy consumption and hence, enhance the efficiency of the system. GPU-enabled DVFS approach helps to decrease task load and achieve better resource utilization due to high speed and faster implementation. An effective modeling is presented to decrease energy consumption and enhance performance. A balancing between CPU and GPU enabled DVFS technique is achieved as CPU-enabled DVFS help to reduce energy consumption whereas GPU-enabled DVFS technique performs much faster processing. Experimental outcomes are compared with traditional state-of-art-techniques in terms of average power, energy consumption and power sum. Our proposed *HDSMM* technique ensures very less energy consumption for *Sipht* scientific dataset for Sipht 30 is 2812.991014 Watts, Sipht 60 is 3158.219947 Watts,Sipht 100 is 3174.261302 Watts and Sipht 1000 is 11211.22691 Watts which is very less in contrast to conventional state-of-art-techniques and conclude high superiority of our proposed *HDSMM* model. In future work, an effective modelling to cost reduction for heterogeneous cloud computing systems will be presented.

REFERENCES

[1] DepeiQian. 2016. High performance computing: a brief review and prospects.National Science Review 3, 1 (2016), 16–16

[2] M.A. Khan, Scheduling for heterogeneous systems using constrained critical paths, Parallel Computing. 38 (4) (2012) 175–193.

[3] X. Tang, K. Li, M. Qiu, E.H.M. Sha, A hierarchical reliability-driven scheduling algorithm in grid systems, J. Parallel Distrib. Comput. 72 (4) (2012) 525–535
[4] J. Han, X. Wu, D. Zhu, H. Jin, L. T. Yang, and J. Gaudiot, “Synchronization-aware energy management for vfi-based multicore real-time systems,” IEEE Transactions on Computers, vol. 61, no. 12, pp. 1682–1696, 2012.
[5] J. Singh, S. Betha, B. Mangipudi, and N. Auluck, “Contention aware energy efficient scheduling on heterogeneous multiprocessors,” IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 5, pp. 1251–1264, May 2015.
[6] G. Xie, Y. Chen, X. Xiao, C. Xu, R. Li and K. Li, "Energy-efficient Fault-tolerant Scheduling of Reliable Parallel Applications on Heterogeneous Distributed Embedded Systems," in IEEE Transactions on Sustainable Computing, vol. PP, no. 99, pp. 1-1.
[7] R. Bleuse, S. Hunold, S. Kedad-Sidhoum, F. Monna, G. Mounié and D. Trystram, "Scheduling Independent Moldable Tasks on Multi-Cores with GPUs," in IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 9, pp. 2689-2702, Sept. 1 2017.
[8] A. Prakash, H. Amrouch, M. Shafique, T. Mitra and J. Henkel, "Improving mobile gaming performance through cooperative CPU-GPU thermal management," 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2016, pp. 1-6.
[9] K. Li, “Energy and time constrained task scheduling on multiprocessor computers with discrete speed levels,” Journal of Parallel and Distributed Computing, vol. 95, pp. 15 – 28, 2016.
[10] C. Chen, “Task scheduling for maximizing performance and reliability considering fault recovery in heterogeneous distributed systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 521–532, 2016.
[11] H. Arabnejad and J. G. Barbosa, "List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, pp. 682-694, March 2014.
[12] Y. Wang, L. Zhang, Y. Ren and W. Zhang, "Nexus: Bringing Efficient and Scalable Training to Deep Learning Frameworks," 2017 IEEE 25th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Banff, AB, Canada, 2017, pp. 12-21.
[13] R. D. Fonnegra, B. Blair and G. M. Díaz, "Performance comparison of deep learning frameworks in image classification problems using convolutional and recurrent networks," 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, Colombia, 2017, pp. 1-6.
[14] N. Farazmand and D. R. Kaeli, "Quality of Service-Aware Dynamic Voltage and Frequency Scaling for Mobile 3D Graphics Applications," 2017 IEEE International Conference on Computer Design (ICCD), Boston, MA, USA, 2017, pp. 513-516.
[15] S. Höppner et al., "Dynamic voltage and frequency scaling for neuromorphic many-core systems," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.
[16] C. E. Giles and M. A. Heinrich, "M2S-CGM: A Detailed Architectural Simulator for Coherent CPU-GPU Systems," 2017 IEEE International Conference on Computer Design (ICCD), Boston, MA, USA, 2017, pp. 477-484.
[17] Y. Cao and H. Wang, "A Task Scheduling Scheme for Preventing Temperature Hotspot on GPU Heterogeneous Cluster," 2017 International Conference on Green Informatics (ICGI), Fuzhou, China, 2017, pp. 117-121.