International Journal of Engineering & Technology, 7 (2) (2018) 837-841

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i2.9531
Research paper

Range smart cluster monitor based guesstimate approach for resource scheduling in small size clusters

S. Gokuldev [1] *, Jathin R [1]

[1] Department of Computer Science, Amrita Vishwa Vidyapeetham, Mysuru, India
**Corresponding author E-mail: gokuldevs@gmail.com*

Abstract

Performing scheduling of tasks with low energy consumption with high performance is one of the major concerns in distributed computing. Most of the existing systems have achieved improved energy efficiency but compromised with QoS metrics such as makespan and resource utilization. A resource scheduling strategy for wireless clusters is proposed by making careful considerations on decisions that would im-prove the battery life of nodes. The proposed strategy also incorporates monitoring system with in the clusters for optimizing the system performance as well as energy consumption. The system ensures “Any case zero loss" performance wherein each cluster will be monitored by at least one cluster monitor. This is implemented by using predictive calculation at each cluster monitor to communicate only if absolutely essential, during assigning jobs to resources, selecting optimal resources by assigning the jobs to the most power efficient resource among the available idle resources within the cluster. The experimental result ensures improved system performance with low power consumption in homogeneous computing environment. The resource sharing strategy is experimentally analyzed, considering the important performance metrics such as starvation deadline, turnaround time, miss hit count through simulations. Significant results were observed with improved efficiency.

Keywords: *Cluster Monitor; Energy Efficiency; Monitoring Based Resource Scheduling; Node States; Small Size Clusters.*

1. Introduction

The increase in demand for high performance computing has directly resulted with increase in power consumption [1]. Power consumption in distributed computing is the biggest concern nowadays. Tianhe-2, the world fastest super computer, on an aggregate, consumes 24 megawatts including external cooling [2]. The use of high performance distributed system is better than using single high processor system. The attainable of using heterogeneous device as a collaborative computing resource is a step towards ubiquitous computing. Several local nodes are connected together to form a global cluster environment of resource sharing systems. If power consumption is minimized completely in each node [3] then maximum power efficiency can be achieved in cluster environments [4]. The major challenge is that the processors may take different time for processing different jobs and the system processing capacity may change over time. For example, if a resource ‘A’ takes 10 seconds for processing a job, then the same resource may take 15 seconds for processing the same job after some time.
Many scheduling algorithms [5] have been developed with maximum efficiency but declines in terms of power efficiency and involve high cost [6] [7]. Results from several scheduling algorithms and experiments carried out till date are highly optimistic and should be extended and carried on further until the vision of green computing is achieve.
The work proposes energy efficient resource strategy by introducing monitoring system which reduces node to node communication [8] by estimating different state of a node in the homogeneous computing environment. The cluster model with the monitoring systems is depicted in Fig. 1.

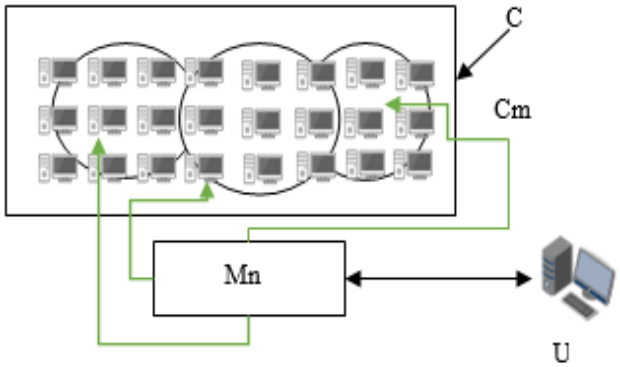


Fig. 1: A Logical Cluster Scheduling Architecture. C=Cluster, Cm=Cluster Monitor, Mn=Master Node, U=User.

The cluster monitor region view shown in Fig. 2 holds four states namely supra-positive, positive sleep, negative awake and neutral. The scheduling is performed based on resource states as mentioned above. The states of neighboring nodes are maintained in the local cache of each cluster monitor and hence pave way for an efficient scheduling of jobs with improved performance which is introduced in the introductory section.
The rest of paper is organized as follows: an in-depth literature survey on various scheduling strategies and energy efficiency algorithms are carried out in section 2. The proposed methodology in section 3 elucidates the monitor based resource scheduling algorithm. Section 4 deals with various experimental analyses considering various performance metrics that are used to evaluate the algorithm. Section 5 deals with the conclusion and the future work.

Copyright © 2018 S. Gokuldev, Jathin R. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

2. Literature survey

During the course of literature survey, it was found that several approaches for job scheduling algorithms has been implemented with the objective of optimizing cost, time and energy efficiency [9] [10] [11]. Several scheduling algorithms and experiments are still being carried out till date and they are highly optimistic and can be improved further by enhancing its performance. Hence, few of the advanced scheduling algorithm and energy efficiency algorithms have been discussed.

2.1. Review on scheduling

Ku Ruhana, Ku-Mahamud and Husna Jamal Abdul Nasir (2010) proposed an ant colony algorithm [12] where importance was given to reduce stagnation on resources. This system helps to record the status of resources in a matrix and provides quick updating of information in the table.

The system addresses scheduling with limited QoS. Dynamic scheduling can be included considering user QoS. The system uses optimal resources first for executing the jobs which may sometimes lead to high stagnation.

Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood, Alan Oxley (2011) implemented two Dynamic Multilevel Hybrid Scheduling algorithm namely MH and MDQ [13] based on fix time and dynamic time quantum in which the resources are maintained in queue in the decreasing order of CPU burst time. The resources are arranged only by considering CPU burst time.

The algorithm can be extended further using other factors that may reduce the total cost and improve the performance of the system. The system may not be cost effective while considering for more number of jobs.

Jian Xiao, Yu Zhang, Shuwei Chen and Huashan Yu (2012) introduced Application-Level Scheduling with Task Bundling(ALSTB) Approach [14] . The algorithm is divided into two levels, firstly application-resource pairing and job selection and secondly perform task selection from previously selected job and allocates the package to the selected resource.

The system does not select optimal resources. The performance of the system degrades with multi users and multiple task execution. Dynamic rescheduling can be included for better results. The system doesn't consider any issues about the IO contention.

Yu Zhang, Shuwei Chen, Ziqian Hu (2013) proposed a new scheduling algorithm Extended Application-Level Scheduling with Task Bundling Approach [15] which had overcome all the limitations of ALSTB algorithm, introduced some optimization for IO contention. This system has also reduced the overall communication. Shows better overall performance with increased resource utilization.

The system can be extended further by making network and workload fluctuation as dynamic.

Maria A. Rodriguez and Rajkumar Buyya (2014) proposed a Particle Swarm Optimization [16], a scheduling strategy for scientific workflows based on meta heuristics optimization technique. The algorithm reduces the overall workflow execution cost while meeting deadlines constraints.

The algorithm can be further extended by arranging the resources in the order of cost and other factors such that optimal resources can be used first and also with different optimization algorithm strategies for improving its efficiency.

B. Kalaiselvi and M. Thangamani (2016) introduced a model [17] for effective performance of job scheduling in grid computing where failure rate, resource's CPU speed and baud rate are considered and the jobs were successfully scheduled to the resources. This increases the performance of scheduling. This algorithm makes use of both static and dynamic load balancing policies to select effective sites and also implements simple computations. The model can be extended by considering more factors which may increase the overall performance.

2.2. Review on energy efficiency

Saurabh Kumar Garg and Rajkumar Buyya (2009) implemented Heterogeneity Aware Meta-scheduling Algorithm (HAMA) [18] that considers crucial information of global grid resource sites, such as cooling system efficiency (COP) and CPU power efficiency. Results show that HAMA can reduce up to 23% energy consumption. This algorithm with other scheduling strategy can give better results in terms of power consumption.

Lesandro Ponciano and Francisco Brasileiro (2010) introduced a strategy for opportunistic grids [19] which focused on two schemes, sleeping and wake-up strategies. This method has resulted in achieving better results on energy efficiency.

This model has not made improvisation on performance while considering multiple peers. The work has been an importance foundation for obtaining a better power efficiency in which task scheduling strategy can be improved.

Wei Liu, Hongfeng Li, Wei Du and Feiyan Shi (2011) proposed a novel scheduling algorithm Energy-Aware Task Clustering Scheduling (EETCS) algorithm [20], designed and implemented on clustering based and duplication based algorithms. The clustering scheduling is used to minimize the scheduling length and duplication based is used to improve energy efficiency.

A better heuristic approach can be introduced considering the user QoS and user deadline for increasing the overall performance so as to achieve energy efficiency.

Joanna Kołodziej, Samee Ullah Khan, Lizhe Wang and Albert Y. Zomaya (2012) addressed on independent batch scheduling in computational grid with an objective of global minimization of makespan and energy efficiency. Three GA [21] based algorithms for energy aware grid schedulers were developed. A better optimization of the makespan had resulted for the improvement of the load balancing of resources.

The scope is to improve the flow time considering the user QoS factors into account. Multilevel Hybrid scheduling can also be adapted for better performance.

Sergio Nesmachnow, Bernabe Dorronsoro, Johnatan E. Pecero and Pascal Bouvry (2013) implemented a mathematical model [22] for energy consumption in the multicore HC grid systems. The model had resulted in fast scheduling. A two level model has been introduced in which a very short execution time has been achieved even for largest problem instances.

The system can be extended by including powerful optimization for improving further assignment. The algorithm can also be improved in terms of search capabilities, move and swapping of tasks.

Nidhi Jain, Inderveer Chana (2014) implemented an Energy Efficient and High Performance (EEHP) scheduling algorithm [23] reduced the power consumption without degrading the system performance. Only the resources which are proved as energy and performance efficient by joule sort benchmark have been used and tasks are prioritized by their complexity. A special focus was given to power efficiency without affecting the performance.

The system can also be improved by including other power efficiency factors to improve the overall system performance.

From the extensive review of literature, it has been quite clearly indicated that high performance computing is potentially game changing with respect to the endless possibilities they present. The main problem identified from scheduling algorithm is that effective scheduling has resulted in more usage of power and energy efficient scheduling algorithm lacks in effective utilization of resources. The energy efficient algorithms can be further extended with optimal scheduling.

The proposed system is designed to optimize both the power consumption of resources and better utilization of resources through monitoring based scheduling strategy which ensures 'Any case zero loss' performance by introducing cluster monitors which is explained in the following section.

3. Monitoring based resource scheduling strategy

Cluster monitor are selected randomly by master node and assigns the further responsibilities of scheduling. For every ten nodes in cluster there should be at least one cluster monitor assigned. The objective is that each and every node is monitored by at least one cluster monitor. The cluster monitor will record state of all neighboring system. The cluster monitor will further assign the jobs to nodes depending on state which is listed in Table 2.

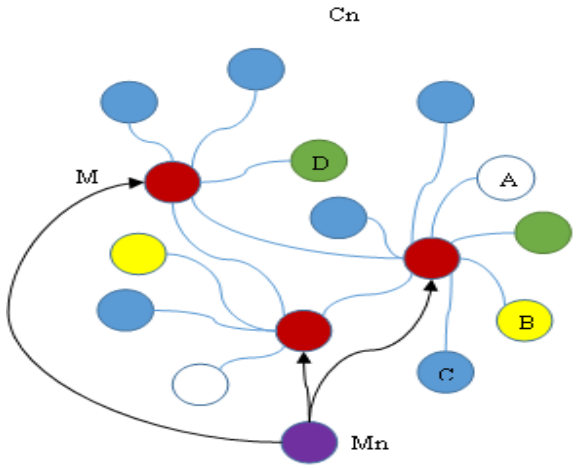


Fig. 2: Cluster Monitor Region View. Mn=Master Node, Cn=Compute Nodes, M=Cluster Monitor, A=Negative Awake, B=Neutral, C=Positive Sleep, D=Supra Positive Sleep.

The further scheduling of jobs by cluster monitor and how the cluster monitor interacts with its neighboring nodes is clearly depicted in the Fig. 2.
The number of cluster monitors in the cluster M is given by:

$$M=N/10 \quad (1)$$

where 'N' represents total number of nodes in cluster environment excluding master node.

Table 1: Cluster Monitor Table

Id	State	Resource Capacity	Idle time
2	Supra positive Sleep	10^4mips, 300 mb	Highest
4	Positive sleep	2Gflops, 700 mb	Medium
7	Negative awake	10^7 mips, 1600 mb	Very low

The selection of neighbor nodes by cluster monitor

$$\text{Rand } \{(\{s_1\}, \{s_2\}, \{s_3\}. \{s_N\}) \mid x = \text{rand } (1\ldots..10)\} \quad (2)$$

Where rand () is random selection function with theoretical equal probability selection and S represents the nodes available in a cluster.
The nodes have processors, hardware, memory, storage space that can be used to performed tasks. The percentage of idle resources is high and the objective is to use the resources which are idle from long time and having good resource capacity. The state of the node [24] is assigned as:

- Supra positive sleep: Resource capacity of a node is much greater than the positive value and resources are given priority based on longest interval idle time.
- Positive sleep: Resource capacity of a node is greater than the positive value and here the resources idle time lies in medium range
- Negative awake: Resource capacity of a node is lesser than the than the negative value
- Neutral awake: nodes outside the region of a cluster monitor.

The proposed algorithm works as follows. Dynamic priority queue is marinated at each cluster monitor. Nodes are prioritized as manner supra-positive sleep>positive sleep>negative awake. This design choice is majorly used to reduce the battery drain in nodes due to frequent network interaction. The execution time (ET) of simulation is given by,

$$ET= \text{stop} - \text{start} \quad (3)$$

where 'stop' represents the total time taken for simulation and 'start' is the time when the simulation starts its execution.
The hit count and miss count performance metric is given by:

$$\text{Hit count}=SE/N \quad (4)$$

where 'SE' represents total numbers of job successfully executed.

$$\text{Miss count}=UE/N \quad (5)$$

where 'UE' represents total number of jobs executed unsuccessfully.
Job scheduling is done in two levels the head cluster node assigns jobs to cluster monitor ,then, the cluster monitor are responsible for further delegation of job to its neighboring nodes depending on its priority list, which in turn is decided by state and ideal time of the node. The objective of cluster monitor is that to maintain state information of their neighboring node. The algorithm for monitoring based resource scheduling is given below:

Input: Neighbor node state
Output: Job assignment and status update
Priority_queue<status>q;
While (all neighbour)
Quash (status)
While (z:incoming jobs)
Status a = q.pop()
a.assign(z)
Update state (a)
q.push (a);
For all neighbours in q
Check (long idle interval time)
Increment node_status priority

The states are not always constant. It changes according to job assigned, time it takes for execution of jobs and processing capacity of job. Thus priority of nodes changes by above factor and also by idle interval time of node, by this all resources are utilized efficiently that make up the wireless cluster. The cluster monitor will keep the track of long ideal interval with respect to each system and assign jobs accordingly.

4. Simulation results

Initially to evaluate the performance of the algorithmic strategy, few performance metric were considered and been identified as been effectiveness for the proposed work.
The simulation was tested by providing an input of 200 jobs and 500 jobs with varying number of resources. The main objective of simulation is to compare the execution time, miss/hit ratio, energy consumption by considering "with cluster monitoring" and "without cluster monitoring" in the cluster computing environment.
The benefits of using cluster monitoring environment are:
Maximum starvation deadline: The maximum starvation deadline for varying number of resources in a cluster during simulation is shown in Fig. 3 and Fig. 4.



Fig. 3: Maximum Starvation Deadline-1.



Fig. 4: Maximum Starvation Deadline-2.

The resources "without cluster monitoring" in the computing environment is found to be idle for long time without any job assigned to it.

The simulation results with 200 jobs and 500 jobs are shown in Fig. 5 and Fig. 6 respectively.



Fig. 5: Execution Time for 200 Jobs.



Fig. 6: Execution Time for 500 Jobs

The overall simulation execution time is less for the system "with cluster monitor" when compared to "without cluster monitoring" environment.

Miss/hit ratio: The number of jobs successfully executed to the total number of jobs is defined has hit. The number of jobs unsuccessfully executed to the total number of jobs in cluster is defined has miss. The result of simulation is shown in Fig. 7 and Fig. 8.

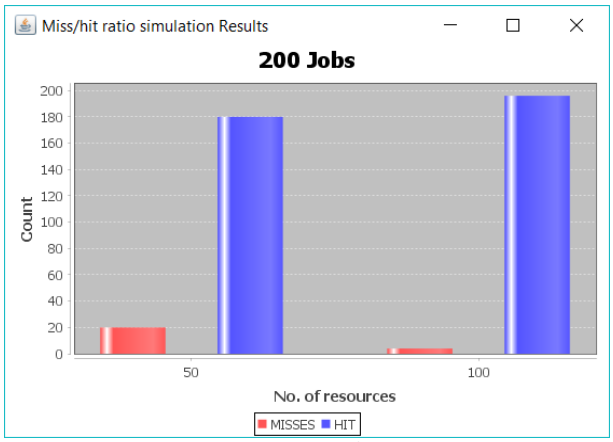


Fig. 7: Hit/Miss Count for 200 Jobs.



Fig. 8: Hit/Miss Count for 500 Jobs.

The simulation of "with cluster monitor" environment has maximize the hit count and reduces the job failure.

Energy consumption: The energy consumed in the overall simulation is shown in Fig. 9 and Fig. 10.

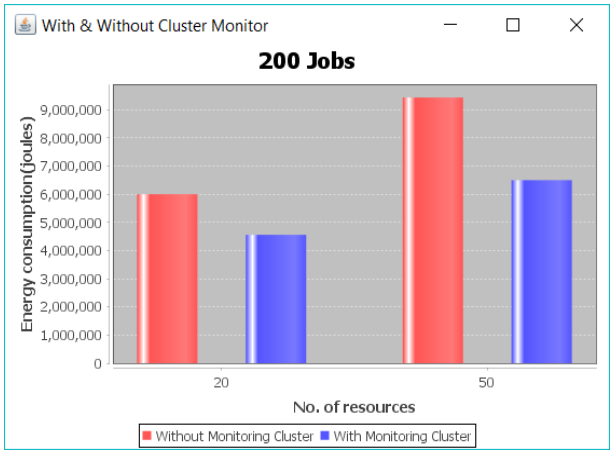


Fig. 9: Energy Consumption for 200 Jobs.



Fig. 10: Energy Consumption for 500 Jobs.

The energy consumption is minimised upto 37% when cluster monitors were incorporated in the computing environment.

5. Conclusion

An algorithmic approach to schedule resources in cluster environment is been proposed with a key focus to reduce the energy consumption of nodes, with an assumption of different node states. The overall reduction in the communication process incorporated with "Any case zero loss" performance of the entire cluster node makes the work more significant. The proposed monitoring based scheduling algorithm makes better scheduling decisions thus improving the efficiency and overall performance of the system in terms of better scheduling and energy efficiency while compared with recent other existing scheduling strategies.
The scope is to include few additional performance metrics such has average CPU utilization and other QoS factors to further enhance its performance.

References

[1] Swamy, S. R., & Mandapati, S. (2017). A Fuzzy Energy and Security Aware Scheduling In Cloud. *International Journal of Engineering & Technology*, *7*(1.2), 117-124. https://doi.org/10.14419/ijet.v7i1.2.9021.

[2] Yang, X. J., Liao, X. K., Lu, K., Hu, Q. F., Song, J. Q., & Su, J. S. (2011). The Tianhe-1a Supercomputer: Its Hardware and Software. *Journal of Computer Science and Technology*, *26*(3), 344-351. https://doi.org/10.1007/s02011-011-1137-8.

[3] Rakesh, N., Shakir, M., Kalamani, P., & Maheswari, B. U. (2017, January). An Energy Saving Algorithm Using Heterogeneity Aware Protocol In Wireless Sensor Networks To Sustain Lifetime Of Nodes. In *Inventive Systems and Control (ICISC), 2017 International Conference on* (Pp. 1-5). IEEE.

[4] Duy, T. V. T., Sato, Y., & Inoguchi, Y. (2010, April). Performance Evaluation Of A Green Scheduling Algorithm For Energy Savings In Cloud Computing. In *Parallel & Distributed Processing, Workshops and Phd Forum (Ipdpsw), 2010 IEEE International Symposium on* (Pp. 1-8). IEEE.

[5] Pillai, A. S., Singh, K., Saravanan, V., Anpalagan, A., Woungang, I., & Barolli, L. (2017). A Genetic Algorithm-Based Method for Optimizing the Energy Consumption and Performance of Multiprocessor Systems. *Soft Computing*, 1-15.

[6] Vasudevan, S. K., Anandaram, S., Menon, A. J., & Aravinth, A. (2016). A Novel Improved Honey Bee Based Load Balancing Technique In Cloud Computing Environment. *Asian Journal of Information Technology*, *15*(9), 1425-1430.

[7] Valentini, G. L., Lassonde, W., Khan, S. U., Min-Allah, N., Madani, S. A., Li, J., ... & Li, H. (2013). An Overview of Energy Efficiency Techniques in Cluster Computing Systems. *Cluster Computing*, *16*(1), 3-15. https://doi.org/10.1007/s10586-011-0171-x.

[8] Luo, L., Wu, W., Di, D., Zhang, F., Yan, Y., & Mao, Y. (2012, June). A Resource Scheduling Algorithm of Cloud Computing Based On Energy Efficient Optimization Methods. In *Green Computing Conference (IGCC), 2012 International* (Pp. 1-6). IEEE.

[9] Berl, A., Gelenbe, E., Di Girolamo, M., Giuliani, G., De Meer, H., Dang, M. Q., & Pentikousis, K. (2010). Energy-Efficient Cloud Computing. *The Computer Journal*, *53*(7), 1045-1051. https://doi.org/10.1093/comjnl/bxp080.

[10] Lim, D., Ong, Y. S., Jin, Y., Sendhoff, B., & Lee, B. S. (2007). Efficient Hierarchical Parallel Genetic Algorithms Using Grid Computing. *Future Generation Computer Systems*, *23*(4), 658-670. https://doi.org/10.1016/j.future.2006.10.008.

[11] Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010, April). A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. In *Advanced Information Networking and Applications (Aina), 2010 24th IEEE International Conference on* (Pp. 400-407). IEEE. https://doi.org/10.1109/AINA.2010.31.

[12] Chen, W. N., & Zhang, J. (2009). An ant colony optimization approach to a grid workflow-scheduling problem with various QoS requirements. *IEEE transactions on systems, man, and cybernetics, part c (applications and reviews)*, *39*(1), 29-43. https://doi.org/10.1109/TSMCC.2008.2001722.

[13] Ku-Mahamud, K. R., & Nasir, H. J. A. (2010, May). Ant colony algorithm for job scheduling in grid computing. In *Mathematical/Analytical Modelling and Computer Simulation (AMS), 2010 Fourth Asia International Conference on* (pp. 40-45). IEEE.

[14] Shah, S. N. M., Mahmood, A. K. B., & Oxley, A. (2011). Dynamic multilevel hybrid scheduling algorithms for grid computing. *Procedia Computer Science*, *4*, 402-411. https://doi.org/10.1016/j.procs.2011.04.042.

[15] Xiao, J., Zhang, Y., Chen, S., & Yu, H. (2012, September). An application-level scheduling with task bundling approach for many-task computing in heterogeneous environments. In *IFIP International Conference on Network and Parallel Computing* (pp. 1-13). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35606-3_1.

[16] Zhang, Y., Chen, S., & Hu, Z. (2013, June). A scheduling algorithm for many-task computing optimized for IO contention in heterogeneous grid environment. In *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on* (pp. 1541-1544). IEEE.

[17] Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE transactions on cloud computing*, *2*(2), 222-235. https://doi.org/10.1109/TCC.2014.2314655.

[18] Thangamani, M. (2016). Grid Computing for Effective performance of Job Scheduling. *Science and Management (ISJCRESM)*, *1*(1).

[19] Garg, S. K., & Buyya, R. (2009, December). Exploiting heterogeneity in grid computing for energy-efficient resource allocation. In *Proceedings of the 17th International Conference on Advanced Computing and Communications.*

[20] Ponciano, L., & Brasileiro, F. (2010, October). On the impact of energy-saving strategies in opportunistic grids. In *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on* (pp. 282-289). IEEE. https://doi.org/10.1109/GRID.2010.5698003.

[21] Liu, W., Li, H., Du, W., & Shi, F. (2011, August). Energy-aware task clustering scheduling algorithm for heterogeneous clusters. In *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on* (pp. 34-37). IEEE. https://doi.org/10.1109/GreenCom.2011.14.

[22] Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor Crummey, J., & Tallent, N. R. (2010). HPC Toolkit: Tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience*, *22*(6), 685-701.

[23] Nesmachnow, S., Dorronsoro, B., Pecero, J. E., & Bouvry, P. (2013). Energy-aware scheduling on multicore heterogeneous grid computing systems. *Journal of grid computing*, *11*(4), 653-680. https://doi.org/10.1007/s10723-013-9258-3.

[24] Lin, J., Cheng, A. M., & Song, W. (2014). A practical framework to study low-power scheduling algorithms on real-time and embedded systems. *Journal of Low Power Electronics and Applications*, *4*(2), 90-109. https://doi.org/10.3390/jlpea4020090.