

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/319669149>

Multi-objective Optimization of Grid Computing for Performance, Energy and Cost

Conference Paper in *Kurdistan Journal of Applied Research* · August 2017

DOI: 10.24017/science.2017.3.31

CITATIONS

0

READS

96

3 authors:



Ahmed Fanfakh

University of Babylon

16 PUBLICATIONS 49 CITATIONS

[SEE PROFILE](#)



Ali Yakoob Yousif

University of Babylon

9 PUBLICATIONS 13 CITATIONS

[SEE PROFILE](#)



Esraa Alwan

University of Babylon /College of science for women

3 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Compiler optimization [View project](#)



Machine learning for energy optimization [View project](#)

Multi-objective Optimization of Grid Computing for Performance, Energy and Cost

Ahmed Badri Muslim Fanfakh

Computer Dept.
College of Science for Women
University of Babylon, Iraq
wsci.ahmed.badri@uobabylon.edu.iq

Ali Yakoob Yousif

Computer Dept.
College of Science for Women
University of Babylon, Iraq
wsci.israa.hadi@uobabylon.edu.iq

Esraa Alwan

Computer Dept.
College of Science for Women
University of Babylon, Iraq
wsci.ali.yakoob@uobabylon.edu.iq

Abstract: *In this paper, new multi-objective optimization algorithm is proposed. It optimizes the execution time, the energy consumption and the cost of booked nodes in the grid architecture at the same time. The proposed algorithm selects the best frequencies depends on a new optimization function that optimized these three objectives, while giving equivalent trade-off for each one. Dynamic voltage and frequency scaling (DVFS) is used to reduce the energy consumption of the message passing parallel iterative method executed over grid. DVFS is also reduced the computing power of each processor executing the parallel applications. Therefore, the performance of these applications is decreased and so on the paid cost for the booking nodes is increased. However, the proposed multi-objective algorithm gives the minimum energy consumption and minimum cost with maximum performance at the same time. The proposed algorithm is evaluated on the SimGrid/SMPI simulator while running the parallel iterative Jacobi method. The experiments show that it reduces on average the energy consumption by up to 19.7 %, while limiting the performance and cost degradations to 3.2 % and 5.2 % respectively.*

Keywords: Multi-objective optimization, Grid computing, Parallel message passing iterative applications and DVFS.

1. INTRODUCTION

In the field of parallel computing over heterogeneous architectures such as grid or cloud, authors defined many optimization problems. Traditionally, they were introduced problem of execution time optimization techniques that maximized the speedup of the parallel applications over the heterogeneous platform. Moreover, providers of grid and cloud computing platform specified prices to use their own infrastructures. Therefore, minimizing the cost of booking nodes in the distributed heterogeneous architectures must be taken into consideration. For example about commercial grid and cloud see GridPP [1] and Amazon EC2 services [2]. Recently, energy consumption minimization problem become an important topic in the literature due to the increased in consumed power by the processors of higher frequency gears or processors with multi-core [3]. Many techniques used to reduce the total power consumption of these heterogeneous architectures. The most widely used tool to reduce the energy consumption of these platforms

is the dynamic voltage and frequency scaling (DVFS). This technique scales down the frequency of the processor to save its energy consumption [4]. While, the main drawbacks of this technique is reduced the performance of the processor. Therefore, the execution time of parallel applications is increased when the frequency of the processor executing them is decreased.

In this paper, the execution time, the energy consumption and the paid price for booking nodes in the grid platform is optimized at the same time. New optimization function and new online algorithm is presented to select the best vector of frequencies that gives the best trade-off between these three objective functions.

2. RELATED WORKS

In this section, three types of works to optimize energy-time trade-off, time-cost trade-off and time, energy and cost of reserving nodes trade-off of the grid parallel architecture is presented.

In [5-7], proposed time-energy trade-off optimization algorithms for homogeneous, heterogeneous cluster and grid. The proposed algorithms were applied to parallel application with iteration to select the best frequencies that gives the best compromising between the performance and the energy consumption. In [8], mathematical optimization model was proposed to find the optimal frequency for each node depending on its computational power. They set the slowest node to maximum frequency in the system to keep application performance degradation to minimum. Authors in [9], proposed five algorithms. They address an algorithm called Multiple-Work flows-Slack-Time-Reclaiming to reclaim slack time using DVFS. The slack times is occurred when the fastest tasks waiting the slowest ones. The second type of works is optimized both execution time and the paid cost for computing resources in the distributed heterogeneous platforms. In [10], authors proposed online scheduling algorithm that takes into consideration the characteristics of cloud computing to compromise both execution time and cost with user input constrains. In [11], an ant colony multi-objective scheduling algorithm for the performance of parallel tasks and cost of used resources in the cloud is proposed. The algorithm depends on proposed cost model that consider the cost of the processing and storage resources. In [12], the authors proposed a heuristic swarm optimization algorithm to schedule both processing time and cost under the deadline and budget constraints. The proposed

algorithm works off-line to converge to the best possible solution.

In the third type of works, the time, energy and cost optimized at the same time to find the best possible trade-off between them. The complexity of multi-objective optimization problem with several parameters (more than two) is higher. To solve this problem, researchers used more complex techniques to find the best trade-off. They used offline methods that take more execution time such as heuristics methods, evolutionary algorithms, linear programming and etc. According to our knowledge, very small numbers of works have been introduced for this problem. For example, authors in [13], proposed a method for multi-objective workflow scheduling in clouds. The method depends on the DVFS technique to minimize energy consumption. They develop offline swarm algorithm to optimize the energy, performance and the cost of parallel tasks execution over data center. The algorithm finds many solutions and it allows the user to select the best frequencies.

The main contribution of this paper is to develop a method to optimize three objective functions, time, energy and cost at the same time. The proposed method works online without the need for training or learning with a very small overhead.

3. MULTI-OBJECTIVE MODELING

In this section, three objective functions are described. The three objectives are time, energy consumption and the cost of executing synchronous parallel applications over grid infrastructure. The time is the overall execution time of the parallel applications running over the grid. The energy consumption is the total amount of energy consumed of all nodes in the grid that executing the parallel applications. The cost is the total cost of the price that paid when booking nodes of the grid for computing. The next subsections, present the models of these three objectives.

3.1. THE EXECUTION TIME MODEL

The execution time of a parallel program is composed from computation and communication times. DVFS (Dynamic voltage and frequency scaling) is a modern techniques allowed in recent processors to minimize their frequencies to reduce their energy consumption. Using DVFS to change the frequency of the processor of the node in the grid will increase only the computation times linearly without changing the communication times, see [5] for more details. The process of changing the frequency of the processor can be expressed by the frequency scaling factor S , which is the ratio between the maximum frequency and the new frequency ($S = F_{max}/F_{new}$). Therefore, the frequency scaling factor S increase linearly the computation time only without affecting the communication time because the processor remains idle during these times.

In synchronous parallel applications the execution time of the program is execution time of the slowest task. In the grid structure where there is a heterogeneous nodes from different site which are different in term of computing

power (frequencies), a vector of frequency scaling factors are available. However, the execution time model of a parallel program can be defined by computing the maximum products of the computation time and the frequency scaling factors added to the minimum communication time as in the equation (1). The minimum communication time refer to that these times are without idle times, see [7].

$$T_{total} = \underset{j=1, \dots, M}{\text{Max}}_{i=1, \dots, N} (T_{cp_{ij}} \cdot S_{ij}) + \underset{j=1, \dots, M}{\text{Min}}_{i=1, \dots, N} (T_{cm_{ij}}) \quad (1)$$

Where N is the number of clusters, M is the number of nodes per cluster, T_{cp} is the computation time, T_{cm} is the communication time and S is the frequency scaling factor.

3.2. THE ENERGY CONSUMPTION MODEL

There are two power consumption meters for a processor. The first one is called dynamic power, which is consumed during the computation times. The latter is called static power, which is consumed during all the times that the processor is turn on [14].

In the considered heterogeneous grid platform, each node j in cluster i may have different dynamic and static powers from the nodes of the other clusters, noted as P_d and P_s respectively. Additionally, each node has different frequencies according to its computing power. However, it has different scaling factor value from the nodes of the other clusters. According to the grid platform, different vectors of frequency scaling factors are available. The frequency scaling operation is used to reduce the dynamic energy consumption by factor of S^{-2} . Whereas, the frequency scaling factor not affecting the static power, refer to [6,8]. Moreover, the frequency scaling factor increases the execution time according to the equation (1). However, the total energy consumption of a parallel application executed over grid is the sum of the total dynamic energies and the total of static energies from all the node of all clusters in the grid as in the equation (2).

$$E_{total} = \sum_{i=1}^N \sum_{j=1}^M P_{d_i} \cdot S_{ij}^{-2} \cdot T_{cp_{ij}} + \sum_{i=1}^N \sum_{j=1}^M P_{s_{ij}} \cdot T_{total} \quad (2)$$

This energy model was proposed in our previous work to predict and measure the energy consumption of parallel application with iterations, refer to [7].

3.2. THE COST MODEL

According to the services provided by grid, administrator determines prices for booking nodes of the clusters for computing. Moreover, they also determine prices for transferring data between nodes. For example, the unit of transferring data is measured in MB per seconds. While the price of computing services is measured by hours. Therefore, the total cost model is the sum of the computation cost C_{comp} and the communication cost of transferring data C_{comm} as in the equation (3).

$$C_{total} = C_{comp} + C_{comm} \quad (3)$$

The cost of computation services is measured during the computation times of the message passing application. When DVFS is used the computation times are increased

linearly with the new frequency scaling factors as in equation (1). Then, the computation cost for the node j in the cluster i that has the price $Pcomp_{ij}$ is computed in the equation (4).

$$C_{comp} = \sum_{i=1}^N \sum_{j=1}^M (Tc_{p_{ij}} \cdot S_{ij} \cdot Pcomp_{ij}) \quad (4)$$

The cost of transferring data is measured by computing the price of communication $Pcomm_{ij}$ during the communication times for all nodes as in equation (5).

$$C_{comm} = \sum_{i=1}^N \sum_{j=1}^M (Tc_{m_{ij}} \cdot Pcomm_{ij}) \quad (5)$$

However, the total cost of executing a parallel application over grid is measured in as in the equation (6).

$$C_{total} = \sum_{i=1}^N \sum_{j=1}^M (Tc_{p_{ij}} \cdot S_{ij} \cdot Pcomp_{ij}) + \sum_{i=1}^N \sum_{j=1}^M (Tc_{m_{ij}} \cdot Pcomm_{ij}) \quad (6)$$

Indeed, in the grid each cluster has different price from other clusters according to its nodes' computing power and its communication network characteristics (e.g link bandwidth and network latency).

4. OPTIMIZATION FUNCTION OF TIME, ENERGY AND COST

The main goal is to optimize all time, energy and cost objectives at the same time when DVFS is used. Therefore, all the objective functions must minimize as in the equation (7).

$$F = \text{Min}(\text{time}, \text{energy}, \text{cost}) \quad (7)$$

As shown in the models (1), (2), (3), the relation between the frequency scaling factor and each one of these models is different. The scaling factor S increases both the time and the cost functions when it is increased. Whereas, this factor is decreased the energy consumption function when it is increased.

Therefore, the relation between these three objective functions are complex and nonlinear. The process of optimizing all of them at the same time is hard problem. In addition the nonlinear relation between them, they are measured using different metrics, for example, time in seconds, energy consumption in joules and the cost in dollars.

To solve this problem, the normalized value for each of these three functions is computed to solve the differences in metrics. The normalized execution time computes the ratio between the new execution time (after scaling down the frequencies of some processors) and the old execution with maximum frequency for all nodes as follows:

$$Tnorm = \frac{T_{new}}{T_{old}} \quad (8)$$

In the same way, the normalization is computed for both the energy and cost as follows:

$$Enorm = \frac{E_{new}}{E_{old}} \quad (9)$$

$$Cnorm = \frac{C_{new}}{C_{old}} \quad (10)$$

Where $Told$, $Eold$ and $Cold$ are computed as follows:

$$T_{old} = \text{Max}_{i=1, \dots, N} Tc_{p_{ij}} + \text{Min}_{i=1, \dots, N} Tc_{m_{ij}} \quad (11)$$

$$E_{old} = \sum_{i=1}^N \sum_{j=1}^M Pd_i \cdot Tc_{p_{ij}} + \sum_{i=1}^N \sum_{j=1}^M Ps_{ij} \cdot T_{old} \quad (12)$$

$$C_{old} = \sum_{i=1}^N \sum_{j=1}^M (Tc_{p_{ij}} \cdot Pcomp_{ij}) + \sum_{i=1}^N \sum_{j=1}^M (Tc_{m_{ij}} \cdot Pcomm_{ij}) \quad (13)$$

The process of selecting frequencies that minimize these three objectives when considering the heterogeneity of the grid in terms of computing power, dynamic and static powers, computation to communication ratio (granularity), number of available frequencies and paid prices for each node is not easy. This problem can be solved by making the optimization process for energy, cost and execution time follow the same evolution according to the vector of scaling factors ($S_{11}, S_{12}, \dots, S_{NM}$). Therefore, the equation of the normalized execution time is inverted which gives the normalized performance equation, as follows:

$$Pnorm = \frac{T_{old}}{T_{new}} \quad (14)$$

However, the objective function that optimizes performance, energy and cost at the same time is computed the maximum perimeter of the triangle as in the equation (18). The perimeter of triangle computes the sum of round sides, denoted as $Rs1$, $Rs2$ and $Rs3$. Each side is computed as the distance between two objectives as follows:

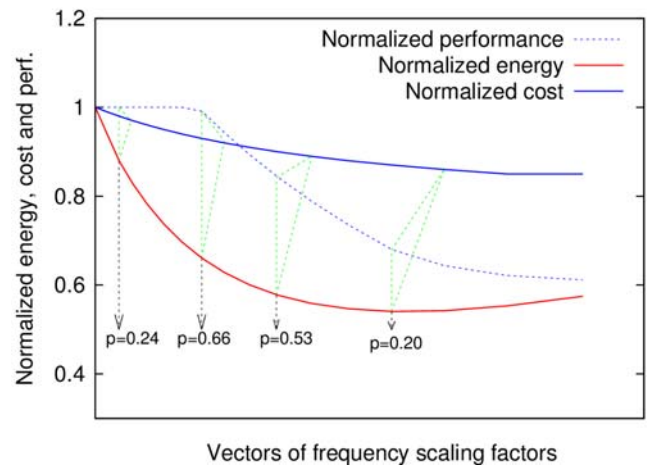
$$Rs1 = Pnorm - Enorm \quad (15)$$

$$Rs2 = Pnorm - Cnorm \quad (16)$$

$$Rs3 = Cnorm - Enorm \quad (17)$$

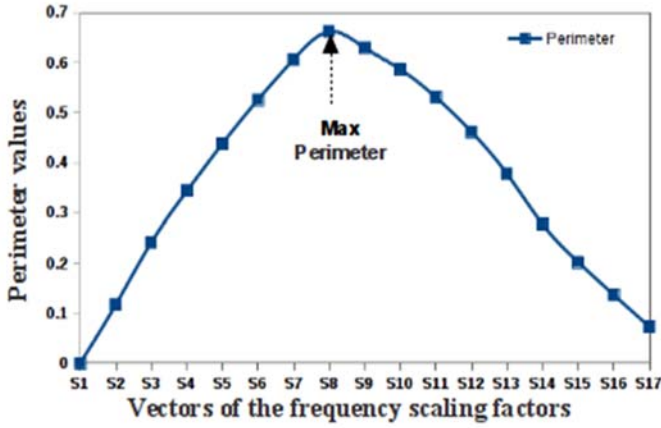
According to equations (15), (16) and (17), there are different distances values can be computed for each vector of frequency scaling factors. If the performance curve is higher than others then the distance value is positive. Otherwise, the distance value is negative. Therefore, the objective function is computed as follows:

$$Perimeter = \text{Max}_{\substack{i=1, \dots, N \\ j=1, \dots, M \\ k=1, \dots, F}} (Rs1 + Rs2 + Rs3) \quad (18)$$



Figure(1): Multi-objective optimization

Figure(1) shows there are different triangle perimeters for each vector of frequency scaling factors. The best vector is the one that gives the maximum perimeters according to the equation (18)



Figure(2): Perimeters of the three objectives

The maximum one gives the best trade-off between the energy consumption, performance and cost at the same time. Figure (2) demonstrates the computed perimeters for all available vectors of the frequency scaling factors.

5. MULTI-OBJECTIVE ALGORITHM

This section presents the proposed multi-objective algorithm (1), that depends on the proposed multi-objective function (18) to select the best vector of frequency scaling factors that gives the best trade-off between energy, performance and cost for parallel message passing iterative applications executed over grid. The algorithm works online during executing the iterative parallel application. It gathers the initial information after the first iteration of iterative application. The gathered information are computation time, communication time, maximum frequency, minimum frequency, dynamic power, static power and the prices of services for each node in the grid. Depending on these information, the algorithm predicts the performance, energy and the cost for each available vector of frequency scaling factors. It selects the frequencies vector that maximizes function (18). At the beginning, the algorithm computes the initial frequencies proportionally to the gathered computation times to start its search space as follows:

$$F_{ij} = \frac{Max(Tcp_{ij}).Fmax_{ij}}{Tcp_{ij}}, i = 1, \dots, N, j = 1, \dots, M \quad (19)$$

Where F_{ij} is the initial frequency, $Fmax_{ij}$ is the maximum frequency and Tcp_{ij} is computation time from the first iteration for node j in the cluster i .

Algorithm (1)

Gathering the initial information;

Computing the initial frequencies as in equation (19);
Computing the initial frequencies scales
 $S_{ij} = Fmax_{ij} / F_{ij};$
Computes the initial time, energy and cost using equations (1),(2) and (6) respectively;
Computes the initial perimeter as in equation (18);
While (perimeter > 0 or reaching to min frequencies) **do**
Scale down all frequencies .
Computing the new frequencies scales
 $S_{ij} = Fmax_{ij} / F_{ij};$
Computes the new time, energy and cost using equations (1),(2) and (6) respectively;
Computes the new perimeter as in equation (18);
if (new perimeter > initial perimeter) **then**
{
store all new frequency scaling factors in Sbest vector;
initial perimeter = new perimeter;
}
End while
return the best frequencies vector from Sbest vecto;
End algorithm

6. EXPERIMENTAL RESULTS

The Multi-objective algorithm (1) was applied to parallel message passing iterative Jacobi method, algorithm that executed over grid. The experiments were conducted over SimGrid/SMPI simulator [15]. This simulator introduces good and easy tools to simulate the grid platform.

6.1. EXPERIMENTS CONFIGURATION

The parallel iterative Jacobi method computes the global maximum residual (GMRES) to converge, see [16] for more details. The used errors accuracy of the convergence of the iterative algorithm is equal to $1 \cdot 10^{-4}$. The grid platform was simulated using SimGrid/SMPI simulator with four different types of heterogeneous computing clusters. Each cluster in the grid platform has nodes with different characteristics from the other clusters such as computing power (frequency), the dynamic and static powers, available frequencies rang, network latency, link bandwidth, computation price for each node (dollar/hours) and communication price for data transfer between nodes (Dollars/hours). As an example about the grid platform see figure (3). Table (1) shows the data used in the experiments configuration.

Table (1): Data for experiments configuration

cluster	Max freq. GHz	Min freq. GHz	Dynamic power	Static power	Comp. Price/h	Comm. Price/h
Cls1	2.50	1.20	20 W	4 W	1.5\$	0.8\$
Cls2	2.66	1.60	25 W	5 W	2.0\$	1.2\$
Cls3	2.90	1.20	30 W	6 W	3.0\$	2.5\$
Cls4	3.40	1.60	35 W	7 W	4.0\$	3.0\$

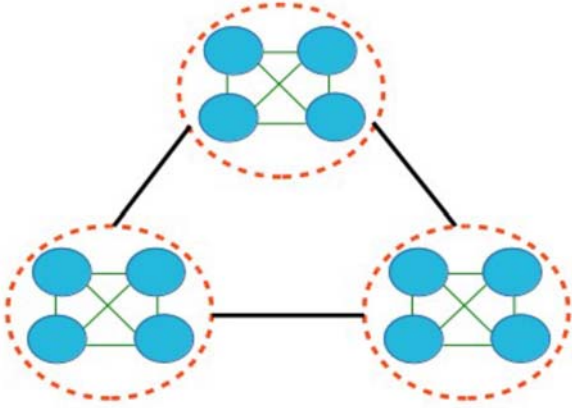
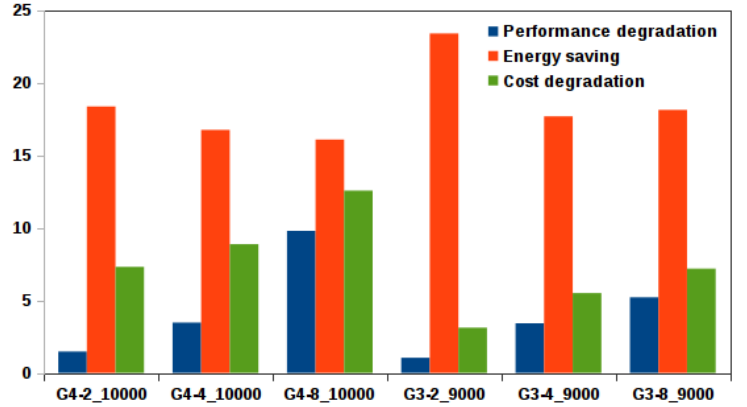


Figure (3): An example about grid of three clusters

6.2. RESULTS OF MULTI-OBJECTIVE ALGORITHM

The multi-objective algorithm (1) was executed many times over parallel Jacobi iterative program. Two grid platform scenarios were used, the first one used grid with four types of clusters as in table (1) and the later used grid with three cluster (the first three clusters in table(1)). The first and second grid scenarios were applied to parallel Jacobi method that solve a two dimensional spare matrix of size 10000*10000 and 9000*9000 respectively. Moreover, each grid scenario utilized different number of nodes for each cluster. Three different configurations were used to each scenario. For example, G4-2_10000 scenario means the grid is composed from 4 clusters each with 2 nodes, total number of nodes equal to 8, that applied to parallel Jacobi method solving the spare matrix of size 10000*10000.

Figure (4) presents the results of applying the multi-objective algorithm. The results are the percentages of energy saving, performance degradation and cost degradation. Both performance and cost degradations percentages are increased when the number of clusters increased. The increase in the number of clusters is increased the communications times due to the high latency of the external communication network between clusters. While the inverse is happened for the energy saving that decreases when the number of cluster and the total number of nodes is increased. The increased in the communication times will decrease the granularity, which is the computation to communication ratio. The highest



Figure(4): The results of performance degradation, energy saving and cost degradation

granularity gives highest energy saving results because the dynamic energy reduces by double during the computation times with the reduced frequencies as in the equation (2) and verse versa. However, the average of the performance degradation, energy saving and cost degradation of the first scenario (grid with four cluster) are equal to 4.9%, 17% and 9.5% respectively. Whereas, the second scenario (grid with three clusters) the average results are equal to 3.2%, 19.7% and 5.2% for performance degradation, energy saving and cost degradation respectively.

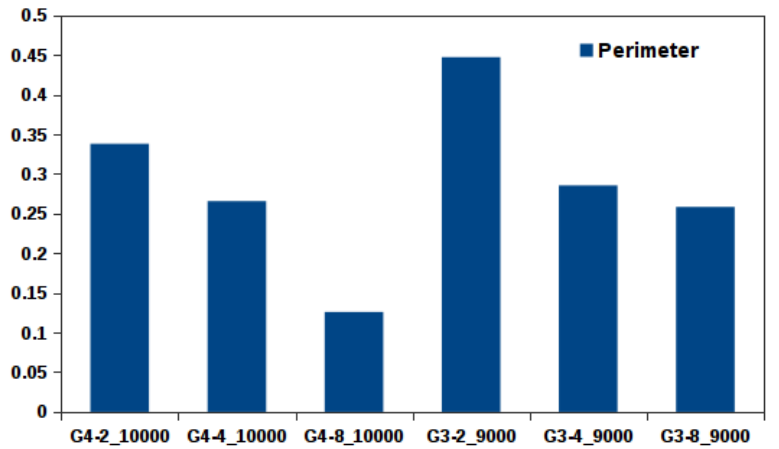


Figure (5): The perimeter percentages

Correspondingly, the perimeter of the performance degradation, energy saving and cost degradation can be computed as in the figure(5). It shows that the best scenario (maximum perimeter) to optimize all these three objectives at the same time is G3-2_9000. This scenario gives the highest energy saving and the lowest degradation for both performance and cost compared with other scenarios, see figure (4).

4. CONCLUSION

In this paper, new multi-objective optimization algorithm that selects the best frequency gears for distributed grid platform is presented. It gives the minimum energy consumption and minimum cost with maximum performance for message passing iterative applications.

The proposed algorithm defines a new optimization function that computes the maximum perimeter between the normalization values of the execution time, energy consumption and cost. The simulated results over SimGrid simulator show that the algorithm reduces up on average the energy consumption by 19.7 % and limiting the degradation of both the performance and the cost by 3.2 % and 5.2 % respectively.

In the future, we plan to take into consideration another objectives such as the CPU temperature and the energy consumed by the communication devices. Moreover, it is also interested to apply the proposed algorithm to other iterative method that solves very big sparse matrix over a larger number of nodes.

5. REFERENCE

- [1] GridPP, Distributed Computing for Data-intensive Research, Online available: <https://www.gridpp.ac.uk>.
- [2] Amazon Elastic Compute Cloud ,Online available: <https://aws.amazon.com/ec2>
- [3] V. W. Freeh, F. Pan, N. Kappiah, D. K. Lowenthal, and R. Springer, "Exploring the energy-time tradeoff in MPI programs on a power-scalable cluster," in Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05) - Papers – Volume 01, ser. IPDPS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 4a–4a.
- [4] N. B. Rizvandi, J. Taheri, and A. Y. Zomaya, Some observations on optimal frequency selection in DVFS-based energy consumption minimization, *J. Parallel Distrib. Comput.*, vol. 71, no. 8, pp. 1154–1164, Aug. 2011.
- [5] Jean-Claude Charr, Raphaël Couturier, Ahmed Fanfakh, Arnaud Giersch. Dynamic Frequency Scaling for Energy Consumption Reduction in Synchronous Distributed Applications. ISPA 2014: The 12th IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 225-230. IEEE Computer Society, Milan, Italy, 2014.
- [6] Jean-Claude Charr, Raphaël Couturier, Ahmed Fanfakh, Arnaud Giersch. Energy Consumption Reduction with DVFS for Message Passing Iterative Applications on Heterogeneous Architectures. The 16th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing. pp. 922-931. IEEE Computer Society, INDIA, 2015.
- [7] Ahmed Fanfakh, Jean-Claude Charr, Raphael Couturier, Arnaud Giersch. Optimizing the energy consumption of message passing applications with iterations executed over grids. *Journal of Computational Science*, 2016.
- [8] T. Rauber and G. Runger, Analytical modeling and simulation of the energy consumption of independent tasks, in Proceedings of the Winter Simulation Conference, ser. WSC '12. Winter Simulation Conference, 2012, pp. 245:1–245:13.
- [9] Jiang, J., Lin, Y., Xie, G., Fu, L., Yang, J.: Time and energy optimization algorithms for the static scheduling of multiple workflows in heterogeneous computing system. *Journal of Grid Computing* 1–22, 2017.
- [10] Ke Liu, Hai Jin, Jinjun Chen, Xiao Liu, Dong Yuan, Yun Yang, A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform, *The International Journal of High Performance Computing Applications*, 445-456, 2010.
- [11] L. Zuo, L. Shu, S. Dong, C. Zhu and T. Hara, A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing, in *IEEE Access*, vol. 3, no. , pp. 2687-2699, 2015.
- [12] Amandeep Verma, Sakshi Kaushal, A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling, *Journal of Parallel Computing*, Volume 62, Pages 1-19, 2017.
- [13] Sonia Yassa, Rachid Chelouah, Hubert Kadima, and Bertrand Granado, Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments, *The Scientific World Journal*, vol. 2013, Article ID 350934, 13 pages, 2013.
- [14] E. Le Sueur and G. Heiser, Dynamic voltage and frequency scaling: The laws of diminishing returns, in proceedings of the 2010 Workshop on Power Aware Computing and Systems (HotPower'10), Oct. 2010.
- [15] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, Versatile, scalable, and accurate simulation of distributed applications and platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, Oct. 2014.
- [16] J. Burgerscentrum, Iterative solution methods, *Applied Numerical Mathematics*, 51(4): 437-450, 2011.

Biography

Dr. Ahmed Badri Muslim Fanfakh, Lecturer at the University of Babylon, College of Science for Women, Computer Department. Google Scholar: <https://scholar.google.com/citations?user=dhEIqUIAAA&hl=en>

Dr. Ali Yakoob Yousif, Lecturer at the University of Babylon, College of Science for Women, Computer Department. Google Scholar: <https://scholar.google.com/citations?user=9bkhTv0AAA&hl=en>

Dr. Esraa Alwan , Lecturer at the University of Babylon, College of Science for Women, Computer Department. Google Scholar <https://scholar.google.com/citations?user=aOxeL5QAA&hl=en>