SAAS FOR ENERGY EFFICIENT UTILIZATION OF HPC RESOURCES OF LINEAR ALGEBRA CALCULATIONS

HRACHYA ASTSATRYAN WAHI NARSISIAN[*] AND GEORGES DA COSTA[†]

Abstract. The most important factor of High performance computing (HPC) systems nowadays is to limit or decrease the power consumption while preserving a high utilization. And with the availability of alternative energy, which powers such systems, there is a need to maximize the usage of alternative energy over brown power. For now, the usage of alternative energy is varying in time due to different factors such as sunny days, the wind, etc. and it is crucial to have an energy-aware algorithm to maximize the usage of this energy. In this paper a SaaS service is presented to optimize a usage of alternative energy, to reduce the power consumption and to preserve a best possible percentage of resource utilization.

Key words. Linear algebra, HPC, Energy efficient utilization, DVFS, Cloud service

AMS subject classifications. 15A06, 15A24, 15A30

1. Introduction.

Over the past few years, resilience has become a major issue for HPC systems, especially for large petascale and future exascale systems [1, 2], taking into account different challenges in this area varying from the deployment, maintenance and the shifting of more heterogeneous resources to a public and private cloud computing infrastructures. HPC infrastructures, such as HPC clusters, consume a lot of energy, not only by their computing elements, but also for cooling, networking and other facilities. The power consumption of data centers is increasing due to several aspects, such as increasing the data volume to deal with, the need for more calculation facilities etc [3], which in its term leads to serious environmental issues (including e-waste and CO2 emission). Thus, reducing the energy consumption of such HPC resources will play an important role to decrease the total energy consumption of these centers.

Taking into account the vast number of applications running on these systems, it is crucial to optimize or find ways to improve the energy efficiency for HPC applications and to build an automatic scheduling environment for optimization of the usage of such infrastructures in the context of keeping the performance regardless of energy consumption [4].

There has been a big amount of investigations and research to resolve these environmental challenges, especially by adopting renewable energy supply techniques (such as solar panels), data centers partially powered with green energy and it is very important to maximize the utilization of this energy when it is available [5]. In this work, a SaaS service is introduced that consist of the following two steps to reduce the power consumption of HPC clusters:

- To add more computing resources to the submitted jobs, if they are available.
- To suggest the usage of alternative energy period, which, in turn, will decrease the power consumption cost, because the price of brown energy is much higher than the price of alternative energy.

The studies have been carried out in the domain of linear algebra simulations lying at the heart of most calculations in scientific computing.

[*] Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia, P. Sevak 1, Yerevan 0014, Armenia (hrach@sci.am, wahi@sci.am).

[†] Institute of Computer Science Research, University of Toulouse, France (dacosta@irit.fr).

The remainder of this paper is divided into the following sections: section 2 is a related work, section 3 shows the methodology and the service, section 4 illustrates implemented experiments and their results, section 5 represents the case studies and, finally, section 6 is the conclusion.

2. Related work. The HPC clusters have crucial roles in the context of energy consumption because the energy consumption of the networking, data and computational infrastructures increases exponentially over the time. Therefore, it is important to study and analyze the power consumption of such infrastructures, as they incur tremendous energy costs and CO2 emissions. For instance, the power consumption of U.S. data centers is expected to grow to 140 billion kWh by 2020 [6].

Several methods and techniques have been suggested and implemented to save energy at data centers and HPC resources. At the hardware level, the Dynamic voltage and frequency scaling (DVFS) method mainly is used, which is a well-known and efficient technique for reducing energy consumption in modern processors. For instance, the paper [7] analyzes the impact on power consumption of two DVFS-control strategies when applied to the execution of dense linear algebra operations on multi-core processors. Another example is to adopt DVFS method to decrease a power consumption of an application without affecting the performance of HPC resources [8, 9]. Reduces clock frequencies for MPI ranks that lack computational work, this technique showed an average measured energy savings of 10.6% and a most of 21.0% over regular application runs [10].

There are several studies related to measuring and control of the power and energy consumption of HPC systems by various components in the software stack [16, 12, 13]. In the cloud environments, the efficient computing resource utilization, and power consumption reduction are being addressed through the intelligent workload, server consolidation, and other mechanisms [14].

In recent years, significant amount of studies have been devoted to optimizing HPC applications for green energy sources, which expose rapid changes in power's availability due to the use of local renewable energy. There are studies, for instance, that discuss the issue of how deploying a scheduler, which can predict available renewable energy and can reschedule jobs preserving their deadlines [15], based on information gathered from the HPC load offer power-aware scheduling [16]. Our approach is to focus on combining the energy consumption of a real scientific application running on HPC production system by not only scheduling the jobs during alternative energy period, but also increasing more computational resources to the jobs in term of CPU cores. These approaches have been taken into account in the SaaS service for a maximum utilization of resources.

3. Service and Methodology. Nowadays, it is a challenge to balance the use and performance of HPC systems. In the meantime, periodical increase of energy costs is forcing infrastructure providers to operate the resources within an energy budget or to decrease energy usage. Therefore, resource management in the context of HPC refers to the process of assigning and scheduling workloads to resources. In the case of alternative energy sources, it is a necessity to maximize the utilization of alternative energy over brown energy.

Brown energy is energy that comes from conventional fossil fuels, such as oil or coal. The combustion of these fuels releases harmful emissions into the environment. Renewable or “green” energy comes from clean sources such as the wind, the solar cell that are more sustainable and are better for the environment.

The provided service or the scheduler is considered as a SaaS or "software as a Service" because it provides an easy way to handle job submission into Linux cluster, SaaS can be considered as a thin client model for software provision, where it is providing the user a point of access to software running on servers.

The total power consumption of HPC cluster is given by the following equation, where T_e is the total energy consumption of the system

$$T_e = \sum_{i=1}^{n} E_i \tag{3.1}$$

Where E_i is the energy consumption per each job; n is the number of jobs completed per unit time. We consider two prices of the electricity: one is for the brown power and the second one is for the alternative energy. The energy consumption of each job is given by the following equation:

$$E_i = Time_i \cdot Price_i \cdot Nodes_i \tag{3.2}$$

The $Time_i$ is given in seconds; the $Price_i$ is given for one watt and the $Nodes_i$ is a crucial factor, because, depending on the number of nodes there are different numbers for power consumption.

There is a single restriction that the jobs must be completed during a single day, in other words, there is a number of tasks, which need to be completed during a 24 hours time period. The ultimate goal of the work is to reduce the total energy consumption T_e by the suggested methodology (see fig. 3.1).

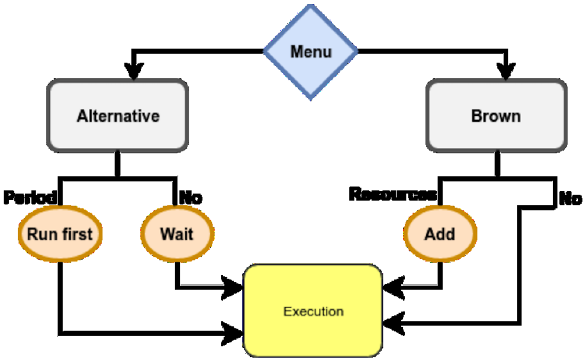

FIG. 3.1. *Schematic Illustration of Methodology.*

The job script checks the number of idle nodes to decide the free resources, then, after the submission, the script, based on the size of the job, allocates more resources, if they are available. The crucial factor is the job priority and the resources available; in case the job has a higher priority, the script will automatically alerts SLURM scheduler to pause the jobs on the required resources to run the higher priority job first. The "Period" word on the schematic illustration refers to the jobs, which are coming in the alternative energy period, so they have priority to go first, the "Resources" on the other hand shows that during the brown energy period there is a possibility to add more resources if they are available. The main drawback here is that when there are many high priority jobs, the other jobs will get delayed for much longer than expected, but, as mentioned before, there is a restriction of a quantity of completed jobs for each day, so, on average, the time of delay will be acceptable.

The command line interface is used by users to submit jobs, the small matrix indicate 4096 size matrix, the medium 8192 size, and finally the large is 12288 size.

The following two scenarios are available after the submission:

- If the user chooses to submit a job during brown energy, the script automatically checks, if there are any available computational resources and adds them to the job. If there are no resources, the job is being executed as requested. This procedure is hidden from the user, so the user doesn't know about it.
- If the submission is selected to be done during the alternative energy period, the script checks the time (there are two periods of alternative energy), if it is in those periods, the job will be executed at once, and all other jobs on the selected computing element will be put on hold.

The above-mentioned results are accomplished by increasing the priority of the alternative energy queue to be higher than the others. If the execution is not in the alternative energy period, the jobs stay on hold till that time. The approach gives some benefits, such as to hide the complexity of creating scripts to submit jobs, to increase the time execution, and to maximize the usage of alternative energy by finishing the maximum number of jobs.

4. Experiments Results. Scientific computing [17] aims at constructing mathematical models and numerical solution techniques for solving problems arising in science and engineering. The solution of linear system of equations lies at the heart of most calculations in scientific computing. For the past twenty years, there has been a great deal of activity in algorithms and software for solving linear algebra problems. For this reason, a specific case of the linear algebra problems has been chosen, namely, the matrix multiplications.

The PBLAS library of ScaLAPACK (Scalable Linear Algebra PACKage) [18] package has been used in the experiments, which is a library of high-performance linear algebra routines for distributed memory message-passing computers and networks of workstations supporting Parallel Virtual Machine and/or Message Passing Interface. PBLAS can be seen as a parallel version of the BLAS (Basic Linear Algebra Subprograms) [19]. Using the PBLAS, three separate levels of operations (vector-vector, matrix-vector and matrix-matrix) can be performed in parallel, which is widely used by various scientific applications in the domain of High-performance linear algebra. The PBLAS Level 3 routines, which are the most computing-intensive, perform distributed matrix-matrix operations.

The PDGEMM double precision routine of the PBLAS is used for experiments, as it is one of the most widely used layer three routine. The Linux cluster consists of a single controller node and four computational nodes (each node has 2 CPU cores Intel Xeon E5420, 1 GB RAM, Ubuntu 14.04 X86 64 OS)is used as a testbed infrastructure for conducting the experiments. Taking into account the testbed infrastructure's parameters (mostly RAM), three different sizes of matrices are studied: small (4096×4096), medium (8192×8192) and large (12288×12288). The Simple Linux Utility for Resource Management (Slurm) [21] used for jobs scheduling which is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system both for large scale and small Linux clusters. To ensure that the results were statistically sound and the value of the execution time and power consumption of each matrix is reliable, for each random execution the experiment is run ten times and the arithmetic mean is taken.

The results of the experiments are shown in the figure 4.1:

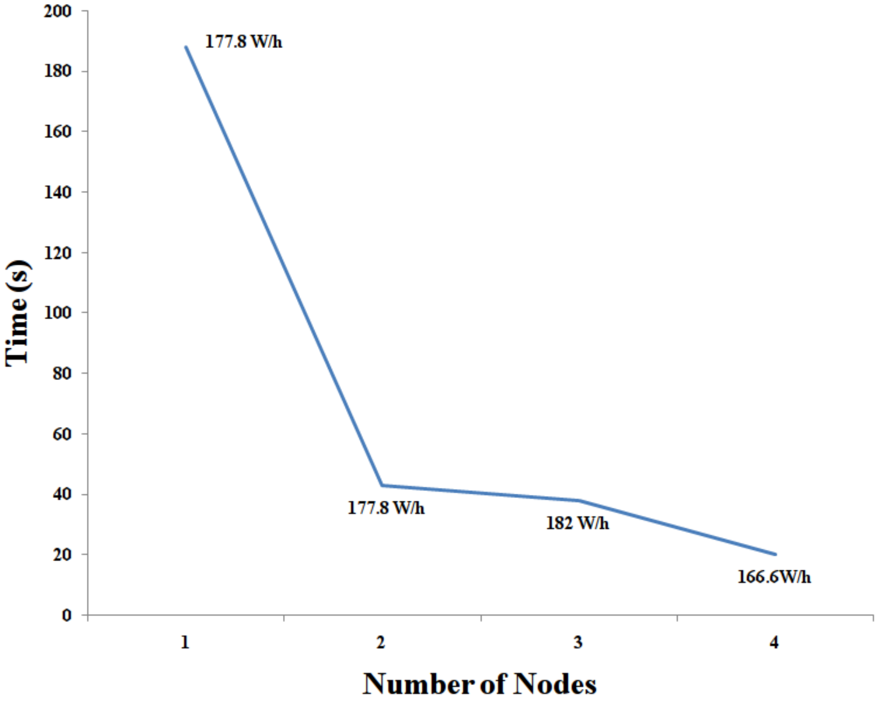

Fig. 4.1. *Experiments results for small matrices*

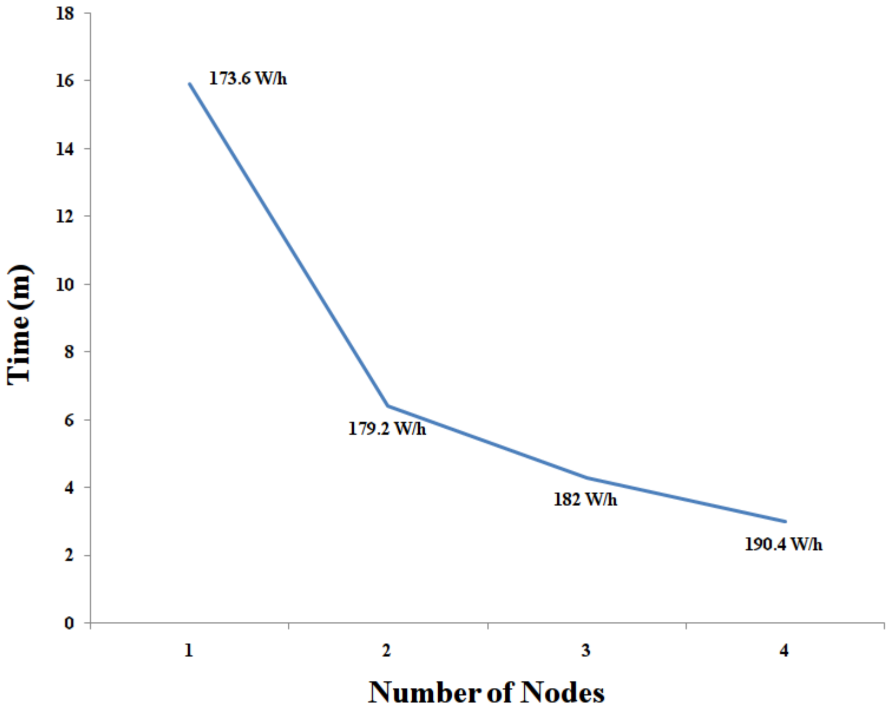

Fig. 4.2. *Experiments results for medium matrices*

The 12288 matrix is only fitted on the whole cluster, and the execution time is: 11 minutes, the power consumption is: 191.8 W/h. Based on these results:

- For small size matrices, the power consumption is becoming almost the same if the number of nodes is increased.
- For medium matrices, there is an execution time improvements in case of using four nodes, and the power consumption remains almost the same.
- For large matrices, the best result is taken when the job is executed on four nodes in term of execution time and power consumption.

5. Use Case. As a use case randomly generated jobs have been submitted received that were from two different applications [22]. The power consumption of the server on the idle state has been measured in order to check the power consumption during jobs execution, because the power meter is showing the overall power consumption of the server. The total energy consumption for each hour on the idle state was 149 watt. The total energy consumption obtained for a single day was about $T_e = 1385$ kilowatt with the following breakdown results:

- 168 small matrix jobs (mix on 2, 3 and 4 nodes).
- 167 medium matrix jobs (mix on 2, 3 and 4 nodes).
- 72 large matrix jobs.

The energy consumption costs per day were about 126.7€(the mentioned price is for the corresponding Kilowatt). The same amount of jobs have been executed with the same node numbers and with the following scenarios in order to study the effectiveness of the suggested model. In the case of only big matrix jobs are executed during the alternative energy period, the total energy consumption was about 1039 kilowatt The price for this amount is: Price = 87.7€(the mentioned price is for the corresponding KW), and the same number of jobs is executed only in eighteen hours.

The same mix of matrices are executed during the alternative energy period and in this case, the total energy consumption is the same: T_e =1120€

As stated from the suggested model there is a reasonable amount of reduced energy consumption and in the meanwhile, the utilization of the resources is increased by cutting out the time for the same number of jobs to be executed.

6. Conclusion. By using the suggested methodology there is an effective gain in two separate domains:

- The same amount of jobs can be run in a shorter time, which gives a possibility to increase the number of completed jobs per day.
- The price of consumed electricity is less due to effective usage of alternative energy period.

For future work, it is planned to enhance the methodology by taking into account the weight of each submitted job depending on several factors such as power consumption, time etc, which will give an opportunity to deploy the mentioned methods for various scientific applications.

Acknowledgments. The European Commission through the VI-SEEM (Virtual Research Environment in Southeast Europe and the Eastern Mediterranean), EaPConnect Projects, the COST Action IC1305, 'Network for Sustainable Ultrascale Computing (NESUS)', and The Armenian State Thematic Project entitled "Towards the Green e-Infrastructures" supported the work presented in this paper.

REFERENCES

[1] J. Dongarra, P. Beckman, and et.al , *The International Exascale Software Roadmap*, International Journal of High Performance Computing, 25:1 (2011), pp. 3–60.

[2] T. Hey and S. Tansley, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, First ed., p. 284, 2009.

[3] E. H. D'Hollander, J. J. Dongarra, I. Foster, L. Grandinetti, and G. R. Joubert, *Transition of Hpc Towards Exascale Computing*, IOS Press, p. 232, 2013.

[4] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, *United States Data Center Energy Usage Report*, LBNL-1005775 p. 257, June 2016.

[5] C. Li, A. Qouneh and T. L, *iSwitch: Coordinating and optimizing renewable energy powered server clusters*, Proc. of ACM International Symposium on Computer Architecture, 2012, pp. 512–523.

[6] E. R. Masanet, R. E. Brown, A. Shehabi, J. G. Koomey, and B. Nordman, *Estimating the Energy Use and Efficiency Potential of U.S. Data Centers*, Proc. of IEEE 99 volume 8, August 2011, pp. 1440-1453.

[7] M. F. Dolz, F. D. Igual, R. Mayo, and E. S. Quintana-Ort, *DVFS-control techniques for dense linear algebra operations on multi-core processors*, Computer Science - Research and Development, 27:4 (2012), pp. 289-298.

[8] H. Astsatryan, W. Narsisian, A. Kocharyan, G. da Costa, A. Hankel, and A. Oleksiak, *Energy Optimization Methodology for e-Infrastructure Providers*, Concurrency and Computation: Practice and Experience, (2016), doi: 10.1002/cpe.4073.

[9] H. Astsatryan, W. Narsisian, G. da Costa, and T. Gurout, *Dynamic Voltage and Frequency Scaling for 3D Classical Spin Glass*, Proc. of IEEE Computer Science and Information Technologies (CSIT), September 2015, pp. 121-124.

[10] A. Tiwari, M. Laurenzano, J. Peraza, L. Carrington, and A. Snavely, *Green queue: Customized large-scale lock frequency scaling*, Proc. of 2012 Second International Conference on Cloud and Green Computing, November 2012, pp. 260-267.

[11] Y. Georgiou, T. Cadeau, D. Glesser, D. Auble, M. Jette, and M. Hautreux, *Energy Accounting and Control with SLURM Resource and Job Management System*, Proc. of 15th International Conference, ICDCN 2014, January 2014, Volume 8314 of the series Lecture Notes in Computer Science, pp. 96-118.

[12] A. Vishnu, S. Song, A. Marquez, K. Barker, D. Kerbyson, K. Cameron, and P. Balaji, *Designing energy efficient communication runtime systems: a view from PGAS models*, Journal of Supercomputing, 63:3 (2013), pp. 691-709.

[13] H. Shoukourian, T. Wilde, A. Auweter, and A. Bode, *Monitoring Power Data: A first step towards a unified energy efficiency evaluation toolset for HPC data centers*, Environmental Modelling & Software, 56 (2014), pp. 13-26.

[14] Md. H. Ferdaus and M. Murshed, *Energy-Aware Virtual Machine Consolidation in IaaS Cloud Computing*, Cloud Computing, Part of the series Computer Communications and Networks, October 2014, pp. 179-208.

[15] I. Goiri, Md. E. Haque, K. Le, R. Beauchea, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, *Matching renewable energy supply and demand in green datacenters*, Ad Hoc Networks, 25 (2015), pp. 520-534.

[16] C. Dupont, *Renewable Energy Aware Data Centres: The Problem of Controlling the Applications Workload*, Proc. of Second International Workshop, EDC 2013, Volume 8343 of the series Lecture Notes in Computer Science, 2014, pp. 16-24.

[17] A. Petitet , H. Casanova , J. Dongarra , Y. Robert , and R. C. Whaley, *Parallel and Distributed Scientific Computing: A Numerical Linear Algebra Problem Solving Environment Designer's Perspective*, Handbook on Parallel and Distributed Processing, 1999.

[18] L. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. Whaley, *ScaLAPACK: A Linear Algebra Library for Message-Passing Computers*, Proc. of SIAM Conference on Parallel Processing for Scientific Computing, March 1997, pp. 1-15.

[19] S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, and K. Remington, *An Updated Set of Basic Linear Algebra Subprograms (BLAS)*, ACM Transactions on Mathematical Software, 28:2 (2002), pp. 135-151.

[20] H. Astsatryan, V. Sahakyan, Yu. Shoukouryan, M. Dayd, A. Hurault, R. Guivarch, A. Terzyan, L. Hovhannisyan, *Services Enabling Large-Scale Linear Systems of Equations and Algorithms based on Integrated P-Grade Portlal*, Grid Computing, 11:2 (2013), pp. 239-248.

[21] A. B. Yoo, M. A. Jette, M. Grondona, *SLURM: Simple Linux utility for resource management*, Proc. of 9th International Workshop, JSSPP 2003, June 2003, Volume 2862 of the series Lecture Notes in Computer Science, pp. 44-60.

[22] H. Astsatryan, T. Gevorgyan, and A. Shahinyan, *Web Portal for Photonic Technologies Using Grid Infrastructures*, Software Engineering and Applications, 5:11 (2012), pp. 864-869.