

## **Scheduling Divisible Jobs to Optimize the Computation and Energy Costs**

Monir Abdullah<sup>1</sup>, Mohamed Othman<sup>2</sup>

<sup>1</sup>Information Technology Department, Tamar University, Yemen

<sup>2</sup>Department of Communication Technology and Network, Universiti Putra Malaysia, Malaysia

---

**ABSTRACT :** *The important challenge in cloud computing environment is to design a scheduling strategy to handle jobs, and to process them in a heterogeneous environment with shared data centers. In this paper, we attempt to investigate a new analytical framework model that enables an existing private cloud data-center for scheduling jobs and minimizing the overall computation and energy cost together. Our model is based on Divisible Load Theory (DLT) model to derive closed-form solution for the load fractions to be assigned to each machines considering computation and energy cost. Our analysis also attempts to schedule the jobs such a way that cloud provider can gain maximum benefit for his service and Quality of Service (QoS) requirement user's job. Finally, we quantify the performance of the strategies via rigorous simulation studies.*

**KEYWORDS:** *-Cloud computing, Divisible load theory, Energy saving, Scheduling, QoS.*

---

### **I. INTRODUCTION**

The clouds are typically large scale virtualized data-centers hosting a plenty of physical servers. Servers in some private clouds are often under-utilized depending on the application; it is always not equal demand. Moreover, the utilization of a data-center is different between weekends and working days and it varies during the day as well. The low utilization means that neither the capital investment in the servers nor the energy consumption is being used as effectively as it could be. Users can submit their jobs into cloud for computational processing or leave their data in cloud for storage. Different users has different QoS requirement. Cloud scheduler must be able to schedule the jobs such a way that cloud provider can gain maximum benefit for his service and QoS requirement user's job is also satisfied. Moreover, In order to improve the investment of the cloud infrastructure and the related energy consumption, enterprise data-centers consolidate various workloads on the minimum physical hardware.

Many scientific and engineering problems are data-driven and are computationally intensive like (satellite image processing, radar and sensor data processing and spectrum computation). Thus, some of above-mentioned applications have to handle huge amount of data. The objective of these data-driven computations is to minimize the processing time of computing loads. Since compute cloud [1] provides virtual computational service, these loads can be processed using compute cloud environment to reduce total processing time. The compute cloud environment provides heterogeneous computation platforms and multiple data storage units for the user applications to run. The data driven computing application in compute cloud system can be partitioned into smaller frames and can be processed simultaneously in the worker roles without any procedure. DLT model helps to schedule computationally intensive processing loads with communication delays [2]. The important challenge in compute cloud environment is to design a scheduling strategy to handle tasks, and to process them in a heterogeneous environment with shared data centers. To the best of our knowledge, the heterogeneous workload scheduling in cloud data centers is rarely evaluated in real life environments and there are no existing approaches for scheduling divisible batch load and transactional loads in compute cloud environments. Additionally, heterogeneous workloads are a fact of life in large-scale data centers, and current resource provisioning solutions do not act upon this heterogeneity [3]. Furthermore, the DLT model for scheduling the batch application sharing the transactional workload in the same cloud resources has not proposed. In this project, we attempt to investigate a new analytical framework model that enables an existing private cloud data-center to manage heterogeneous workloads: interactive and divisible batch loads. Additionally, the proposed model is expected to be successfully implemented in the private cloud to maximize the utilization of the computing infrastructure while providing service differentiation based on high-level performance goals and generating economic to the cloud provider. The rest of this paper is organized as follows: Section 2 deals with the literature review.

In Section 3, a scheduling model of indivisible jobs elaborated. In Section 4, a detailed description of the proposed DLT model is given. Section 5 presents the results and discussions to validate the proposed model. Finally, we summarize the findings and conclude the paper.

## II. LITERATURE REVIEW

Several research works have been conducted on the scheduling of workloads on cloud environments. In this section, we compare our proposed work to most relevant ones. DLT has proven to be a valuable tool in handling large-scale computational loads on networked systems for various aerospace data and image processing applications [4]. Although DLT uses linear modeling, recent studies also show the use of the DLT paradigm in handling computation that demands a nonlinear style of processing [5]. DLT was successfully applied for Scheduling divisible loads on large scale data Grids and produced competitive results [6, 7, and 8]. Recently, DLT paradigm was investigated to design efficient strategies to minimize the overall processing time for performing large-scale polynomial product computations in compute cloud environments. A compute cloud system with the resource allocator distributing the entire load was considered to a set of Virtual CPU Instances (VCI) and the VCIs propagating back the processed results to resource allocator for post processing [4]. Furthermore, a programming pattern for programmers was proposed to easily develop high performance applications on dynamic and heterogeneous cloud environments using DLT paradigm [9]. This pattern uses a performance-based approach to distribute workloads within a program to working nodes to reduce scheduling overhead. Moreover, the scheduling strategy should be developed for multiple workflows with different QoS requirements. In [10, 11], a multiple QoS constrained scheduling strategy of multi-workflows (MQMW) was considered to address this problem. The strategy can schedule multiple workflows which are started at any time and the QoS requirements are taken into account. Here, the indivisible jobs only considered.

Several aerospace applications have been shown to benefit through the direct use of the DLT paradigm in processing radar and satellite image processing [12, 13], large-scale database search problems [15], satellite remote sensing and monitoring, [14], etc., to quote a few. Study in [16] considers using a different objective of optimizing the monetary cost of processing a divisible load especially when utilizing public domain networks. Our work could be treated as an extension of our previous work in [17] by considering one more factor which is energy costs in order to reach better results in a minimal computation and energy cost.

## III. SCHEDULING INDIVISIBLE JOBS

In cloud computing, end users do not own any part of the infrastructure. The end-users simply use the services available through the cloud computing paradigm and pay for the used services. The cloud computing paradigm can offer any conceivable form of services, such as computational resources for high performance computing applications, web services, social networking, and telecommunications services [18].

A. **Notations and definitions:** The notations and definitions that are used throughout this paper are shown in Table 1.

**TABLE 1**  
*Notations and Definitions*

Notation	Definition
$M$	Total number of the machines
$N$	Total number of jobs
$\omega_i$	Processing cost per instruction for machine $i$
$\sigma_i$	The cost per instruction for processing unit $i$
$\beta_j$	The delay cost of job $j$
$I_j$	Expected instruction count per job $j$
$C_j$	Instruction count per job $j$
$E_i$	Energy cost for machine $i$
$P_i$	Millions instruction per second can be executed by machine $i$

**B. Cost Model:** Let consider the following cost factor:  $\sigma_i$  be the cost per instruction for processing unit  $i$  and  $\beta_j$  indicates the delay cost of job  $j$ . Suppose,  $M$  machines with  $N$  jobs and assign these  $N$  jobs into  $M$  machines, in such an order that following condition can be satisfied: From user side, finish time ( $T_f$ ) must be less than the worst case completion time ( $T_{wcc}$ ), scheduling must be done such way to preserve QoS and to avoid possible starvation.

$$T_f \leq T_{wcc} \quad (1)$$

This condition must be satisfied anyhow, otherwise the job is considered as a failure job and the corresponding scheduling is illegal. From cloud provider side, to minimize the cost spend on the job: Suppose  $i^{th}$  machine is assigned to  $j^{th}$  job. Then the cost for execution job  $j$  is  $C_j * \sigma_i$  (while  $C$  is the instruction count). Let  $\Psi_j$ , estimated delay cost for job  $j$ , can be defined as:

$$\Psi_j = \begin{cases} 0 & T_d \geq T_f \\ \beta * (T_f - T_d) & T_d < T_f \end{cases} \quad (2)$$

Where,  $T_d$  is the deadline for job  $j$  and  $T_f$  is the estimated finish time, when job  $j$  is assigned to machine unit  $i$ . Thus overall cost to execute all  $N$  jobs can be given by:

$$\zeta = \sum_{i=1}^N ((C_i * \sigma_i) + \Psi_j) \quad (3)$$

Thus, cloud provider's aim is to schedule jobs (i.e find a permutation:  $N \rightarrow N$ ) such a way which minimize the value of  $\zeta$ .

$$\text{Min}(\zeta) = |\text{Min} \sum_{i=1}^N ((C_i * \sigma_i) + \Psi_j)| \quad (4)$$

As there are  $N$  number of jobs and  $M$  number machines and assuming that all machines are capable to perform any job, then there are total  $N * M$  numbers of way to assignment. And if  $N=M$ , then it need  $N!$  assignments, which has an exponential complexity  $O(N!)$ . Thus this problem is a kind of NP-Complete problem. A probabilistic search algorithm can solve this assignment problem in finite time.

#### IV. PROPOSED DLT SCHEDULING MODELS

In the proposed DLT model, both computation and energy cost are considered when the load is divided. Firstly, the computation time fraction is derived as in [18]. After that, the energy cost fraction is proposed. Finally we combine both fractions. The proposed DLT model works stepwise as follows:

- [1]. The load fraction is calculated using processing cost,  $\omega_i$
- [2]. The load fraction is calculated using energy cost,  $E_i$ .
- [3]. The two fractions are combined to produce the closed form solution.
- [4]. The total cost is calculated using cost model.

##### A. Computation Cost Fraction

The computation cost fraction has been proposed successfully in [17]. The optimal schedule is that all nodes finish the processing with the same cost. Based on that, we will have:

$$\alpha_i = \frac{1}{\omega_i * \sum_{j=1}^M \frac{1}{\omega_j}} \quad (5)$$

##### B. Energy Cost Fraction

By considering energy cost instead of processing cost in equation (5), we have:

$$\alpha_i = \frac{1}{E_i * \sum_{j=1}^M \frac{1}{E_j}} \quad (6)$$

**C. Combination DLT Closed Form :** After getting the computation cost and the energy cost fractions in Equations (5) and (6), respectively, the combination fraction (closed form) ( $CE_i$ ) is calculated as:

$$CE_i = \frac{1}{\omega_i \cdot \sum_{j=1}^M \frac{1}{\omega_j}} + \frac{1}{E_i \cdot \sum_{j=1}^M \frac{1}{E_j}} \quad (7)$$

$$\alpha_i = \frac{CE_i}{\sum_{j=1}^M CE_j} \quad (8)$$

When we apply this model to divide the load, it gives better results than previous model and the load is balanced efficiently. The cost diagram of proposed DLT model is shown in Fig. 1 and 2.

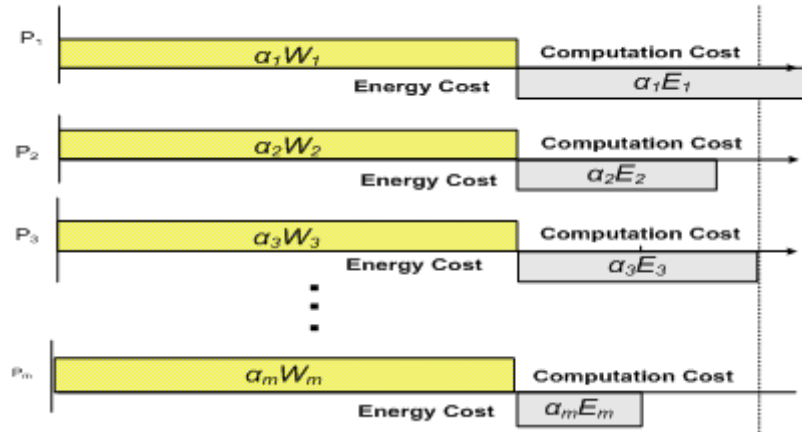


Fig. 1: DLT Processing Cost Diagram with Energy Cost of Load Distribution with  $M$  Machines

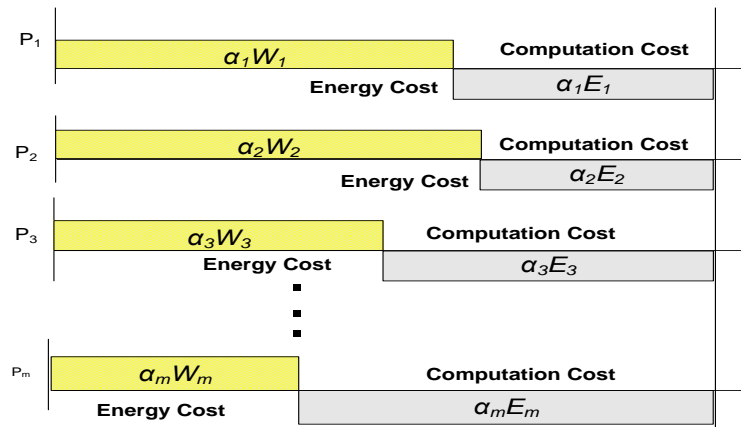


Fig. 2: DLT Energy and Computation Cost Diagram of Load Distribution with  $M$  Machines

**D. Optimality Criterion:** In all the literature related to the divisible load scheduling domain so far, an optimality criterion [19] is used to derive an optimal solution as follows. It states that in order to obtain an optimal processing time, it is necessary and sufficient that all the sites that participate in the computation must stop at the same time. Otherwise, load could be redistributed to improve the processing time.

**E. Scheduling Cost**

The total load that will be processed by machine  $i$  can be defined as:

$$L_i = \alpha_i * \sum_{j=1}^M C_j \quad (9)$$

The cost of the  $L_i$  load execution in machine  $i$  is calculated by:

$$Cost_i = L_i * (\omega_i + E_i) \quad (10)$$

Based on DLT optimality criterion in previous Section, all machines will finish the processing at the same based on the DLT model. Because of that, the finish time  $Tf_i$  of each job will be:

$$Tf_i = \frac{\sum_{j=1}^M C_j}{\sum_{j=1}^M P_j} \quad (11)$$

So that, the estimated delay time for job  $i$ , can be defined as:

$$DelayTime_i = Tf_i - Deadline_i \quad (12)$$

Here also we will implement same rule of indivisible jobs (see Equation (2)).

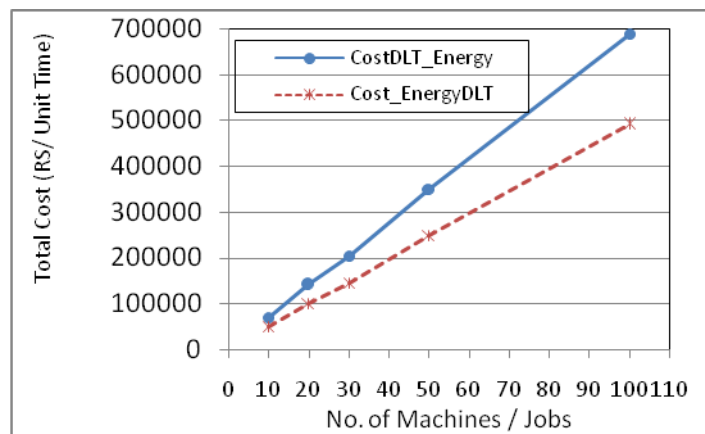
$$DelayCost_i = DelayTime_i * \beta_i \quad (13)$$

The total cost ( $T_c$ ) for scheduling  $N$  jobs on  $M$  machines is:

$$T_c = \sum_{i=1}^M L_i * (\omega_i + E_i) + \sum_{j=1}^N DelayCost_j \quad (14)$$

## V. RESULTS AND DISCUSSIONS

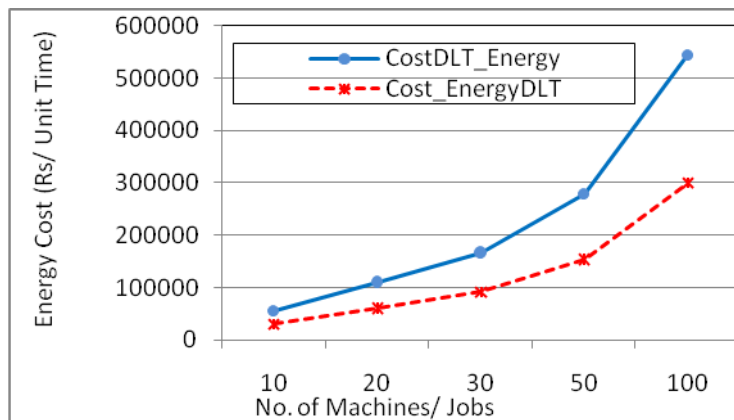
To measure the performance of the proposed DLT model, randomly generated experimental configurations will be used. These configurations are normally utilized in this area of research [18]. To evaluate the performance of the model, it has been simulated to find best schedule for different number of jobs and different number of machines. A number of jobs having different attributes are generated randomly and also a number of machine units having random attributes are generated randomly. We have examined the overall performance of the model by running it under 100 randomly generated cloud configurations. For instance,  $N$  different jobs (10, 20, 50 and 100) having different characteristics are given generated randomly. Similarly  $M$  different machine units (10, 20, 30, 40, 50, and 100) attributes with random characteristics are generated. When was applied the cost model, we have varied the job parameters uniformly: job deadline (1 to 10), delay cost (1 to 10), expected instruction count per job ( $I$ ) (100 to 1000). Also we uniformly distributed process units as: Millions Instruction Per Second (MIPS) can be executed by node ( $P$ ) (10 to 100), computation cost ( $\omega$ ) (1 to 10) and energy cost ( $E$ ) (10 to 100). The simulation results proved that the proposed model will give good results in terms of computation and energy cost. Thus, we will compare the performance of the models with different random configuration. The performance of the models was compared in Fig. 3.



**Fig. 3:** Total Cost vs. No of Machines/Jobs in the Cloud

In Fig. 3, the total cost is plotted against the number of processing nodes/jobs. It is clear that, the *Cost\_EnergyDLT* model is the best for any number of processing nodes/jobs. That is because this model takes into account the energy cost for each machine when it divides the loads.

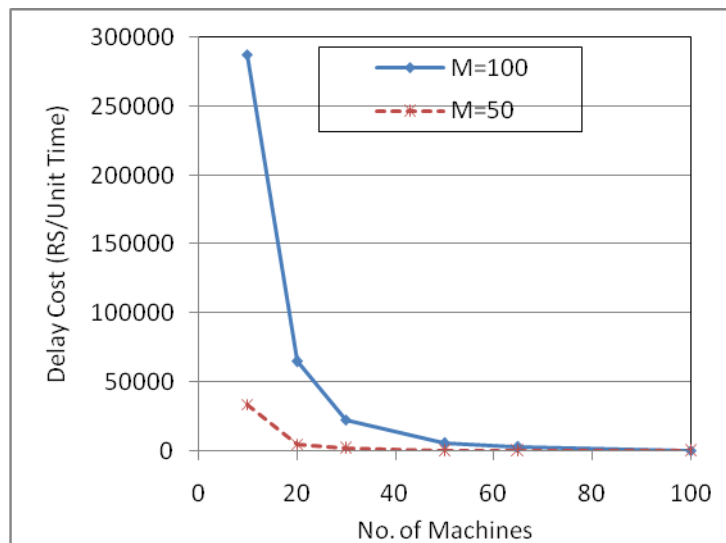
When we compare the two models for calculating the energy cost only, the *Cost\_EnergyDLT* model also produces better result. The result is clearly depicted in Fig. 4.



**Fig. 4:** Energy Cost vs. No of Machines/Jobs in the Cloud

The proposed models are also implemented with varied number of processing nodes. Fig. 5 clearly demonstrates the performance of the two models.

The delay cost is plotted against the number of nodes/jobs. It is observed that, when the number of nodes is increased, the delay cost decreases. When the number of processing node is more enough for processing the jobs, the delay cost will be zero.



**Fig. 5:** Delay Cost vs. No of Machines in the Cloud

## VI. CONCLUSION

In this paper, the problem of scheduling divisible jobs on cloud platforms is addressed. We use DLT model to derive closed formsolutions for the scheduling jobs considering energy cost along with computation cost. In the proposed models, the optimality criterion is utilized to ensure an optimal solution. By and large, the experiment results of the proposed models show better performance. That is because the proposed model considers both the computation time along with the node energy cost.

## REFERENCES

- [1] L. Robert, G. Yunhong, S. Michael, and Z. Wanzhi., Compute and storage clouds using wide area high performance networks, *Future Generations Computer Systems*, 25 (2), 2009, 179-183.
- [2] B. Veeravalli, D. Ghose, M. Venkataraman, and T.G. Robrtazzi, *Scheduling Divisible Loads in Parallel and Distributed Systems*, Wiley-IEEE Computer Society Press, 1996.

- [3] J. Zhan, L. Wang, X. Li, W. Shi, C. Weng, W. Zhang, and X. Zang, Cost-aware Cooperative Resource Provisioning for Heterogeneous Workloads in Data Centers, *IEEE Transaction on Computers*, 62(11), 2013, 2155 – 2168.
- [4] N. Iyer, V. Bharadwaj, S.G Krishnamoorthy. On Handling Large-Scale Polynomial Multiplications in Compute Cloud Environments using Divisible Load Paradigm, *IEEE Transactions on Aerospace and Electronic Systems*, 48 (1), 2012, 820 – 831.
- [5] J. T. Hung, and T. G. Robertazzi. Scheduling nonlinear computational loads. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3), 2008, 1169 – 1182.
- [6] M. Abdullah, M. Othman, H. Ibrahim, and S. Subramaniam, An integrated approach for scheduling divisible load on large scale data grids, in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 4705, 2007, 748–757.
- [7] M. Othman, M. Abdullah, H. Ibrahim, S. Subramaniam: A2DLT: Divisible Load Balancing Model for Scheduling Communication-Intensive Grid Applications. *Lecture Notes in Computer Science (LNCS)*, Springer, Heidelberg, Part I. LNCS, 5101, 2008, 246 – 253.
- [8] M. Abdullah, M. Othman, H. Ibrahim and S. Subramaniam, Optimal workload Allocation model for Scheduling Divisible Data Grid Applications, *Future Generation Computer Systems*, 7 (26), 2010, 971 – 978.
- [9] Wen-Chung Shih, Shian-Shyong Tseng, and Chao-Tung Yang. Performance Study of Parallel Programming on Cloud Computing Environments Using MapReduce, *International Conference on Information Science and Applications (ICISA)*, 2010, 1 – 8.
- [10] M. Abdullah, M. Othman, An Improved Genetic Algorithm for Job Scheduling in Cloud Computing Environment, *Global Journal on Technology*, 2, 2012, 291- 296.
- [11] M. Xu, I. Cui, H. Wang, Y. Bi. A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing, *IEEE International Symposium on Parallel and Distributed Processing with Applications*, 2009, 629-634.
- [12] Ko, K. and Robertazzi, T. G. Equal allocation scheduling for data intensive applications. *IEEE Transactions on Aerospace and Electronic Systems*, 40 (2),2004, 695 - 705.
- [13] Hung, J. T. and Robertazzi, T. G. Switching in sequential tree networks. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3),2004, 968 - 982.
- [14] Moges, M. and Robertazzi, T. G. Wireless sensor networks: Scheduling for measurement and data reporting. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1), 2006, 327-340.
- [15] Ko, K. and Robertazzi, T. G. Signature search time evaluation in flat file databases. *IEEE Transactions on Aerospace and Electronic Systems*, 44(2), 2008, 493 - 502.
- [16] C. F. and Robertazzi, T. G., Optimizing a divisible load nonlinear cost function. In *Proceedings of the 2005 International Conference on Information Sciences and Systems*, Johns Hopkins University, 2005, 16 - 18.
- [17] M. Abdullah, M. Othman, Cost-based multi-QoS job scheduling using divisible load theory in cloud computing, *Procedia Computer Science* 18, 2013, 928-935.
- [18] M. Mezmaza, N. Melabb, Y. Kessaci b, Y.C. Lee, E. G. Talbi, A.Y. Zomayac, D. Tuytens, A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems, *Journal of Parallel and Distributing Computing*, 71, 2011, 1497 – 1508.
- [19] V. Bharadwaj, D. Ghose, T. Robertazzi, Divisible load theory: a new paradigm for load scheduling in distributed systems, *Cluster Computing* 6 (1), 2003, 7 – 17.