

DEADLINE AWARE OPTIMIZATION IN RESOURCE ALLOCATION FOR REDUCING MIGRATION COST

AMITKUMAR S. MANEKAR¹ AND P. GERA

ABSTRACT. Nowadays, the scheduling of various tasks using the available resources is attaining a larger attention in cloud computing owing to the improved performance in cost optimization. Unluckily, scheduling approaches in traditional distributed computing are not able to work in an effective manner since, it is large-scale and dynamic. With virtualization techniques big data applications surges up to many extend. Compromises of cost and performance are someway achieved by these virtualized cloud computing platforms. One of the enormous challenges above virtualized platforms is managing resources. Reliance on a single cloud provider is a challenging task with respect to services like latency, QoS and non-affordable monetary cost to application providers. Managing distributed data centres and maximizing profit is a current problem. High cost and maintenance charges are leverage by data centres. We propose a multi-objective hybrid model with tight deadline (makespan) condition; for optimizes the execution time objective to meet the deadline constraint when the feasible solution hasn't been obtained. Experimental results and conclusions drew with extensive experiments based on different scales and different cloud resources. Detailed analysis of experimental results, it is shows that the proposed algorithm has a better optimization effect against the fair share policies which are presently available in most of the BDA with deadline constrained.

¹*corresponding author*

2010 *Mathematics Subject Classification.* 94A16, 49N30.

Key words and phrases. Deadline, Cost Optimization, Big Data, Cloud Computing, Data Migration Cost, PSO.

1. INTRODUCTION

Massive parallel processing by storage, networking infrastructure within a stipulated time frame for generating results are enormous tasks and tedious jobs in modern BDA [1]. A most important challenge faced by meta-heuristic optimization schemes such as GA is the rate of repetitions in obtaining optimized solutions. QoS in a large number of BDA is not satisfactory due to large enormous intensive data and scaling of resources on time due to the dynamic nature of the request [2]. In this paper, we are focusing on Deadline Aware Optimization in Resource Allocation for reducing migration cost. Metaheuristic algorithms are used in proposed hybrid algorithm. Arrangement of the article is in two sections first, the scheduling algorithm minimizes the makespan and second, it reduces the migration cost, both economic and ecological perspectives. Finally, in the experiment section, experimental results are discussed with enhanced performance. A recent study shows that every day 2.5 QB (Quintillion Bytes) of Data generated daily. These bulks of data are complex to capture, store, search, produce and analyse using a traditional database system [3,4]. 90 percentage of today's data is just created within the last couple of years of span. The data storage industry is now 4(X) times faster than previously. This data is variety and velocity in nature which leads to high demand of modern Big Data Processing Applications (BDA). Private Cloud frameworks are pay - as -go pre-paid with no real sharing of resources adopting mechanism of work-preserving fair -queuing for better utilization is designed [5,6]. Most earlier scheduler unable to manage non-MapReduce task due to less capability of optimization cluster and utilization of cluster which ultimately bottleneck for flexibility, efficiency, scaling, and performance. Hadoop 2.0 with YARN (Yet another Resource Negotiator to take care of resource utilization [1,7,8]. Cloud computing with major characteristics like completion time, the total cost for executing all users' tasks, utilization of the resource, power consumption, and fault tolerance, task scheduling is a complex problem in cloud computing.

2. RELATED WORK

Task Scheduling and Resource Allocation in cloud for BDA is a NP hard problem in cloud computing. With dynamic nature of the request and on demand

resource utilization makes this issue more complex. Price of engaging cloud for solving optimization problems is a crucial point in terms of revenue for the cloud service provider and cloud user. Task Scheduling and Resource allocation Optimization of total time and total cumulative uptime, leads to minimizing the cost of cloud resources utilization. BDA applications are applications that require a large amount of resources to solve complex problems. Ultimately cloud service provider and cloud user will be benefited. In this work a proposed algorithm basically works on EA which are metaheuristics. Proposed algorithms are designed and evaluated on the underlying infrastructure in terms of optimization duration and the cost of resource consumption for lowering infrastructure costs. EA and nature inspired metaheuristics algorithms have proven them self for solving a complex real-world optimization problem in modern BDA. Evolutionary Algorithms (EAs) are a population-based metaheuristic inspired by the survival of the fittest principle. EA are majorly intensive for real-world large-scale optimization and classification jobs. Task Scheduling and Resource allocation in cloud for BDA is a real-world large-scale optimization and classification tasks if uses with GA (Genetic Algorithms) involving thousands of objective functions for evaluations [1]. BDA use multiple sources to analyze this volume of data through parallel processing technologies like Hadoop, Spark, etc. In this information age, many sources generated voluminous data around us. On the basis of research gap and findings below objectives are consider for this work.

- (1) To include the deadline parameter as the major constraint along with others, thereby planned to define a new objective function that aids in the execution of fine-grained resource allocation and migration as well.
- (2) To introduce a new hybrid algorithm for solving the cost optimization problem that obviously enhances with respect to better convergence rate and speed, respectively.
- (3) To comparison of proposed model with other state-of-the-art models deadline and task completion.

3. METHODOLOGY

This research work is totally dedicated to economic and ecological perspectives. Convergence, stability, and solution diversity is experimentally enhanced with these hybrid algorithms. Multiple independent tasks with shared resources

architecture is considered. Pay-as-go infrastructure of cloud is simulated and discussed analyses with conventional MCC's. Offloading decisions of all users were jointly optimized over a time with multi-objective function using simulation.

3.1. Problem Identification. In private cloud architecture where all resources are privately owned this problem is not as serious. Public cloud architecture with pay-as-go heavily charge for this problem and hence it's a bottleneck problem for both cloud service provider and cloud user in terms of the instances' cumulative up-time. In traditional typical way user has choice of either go for

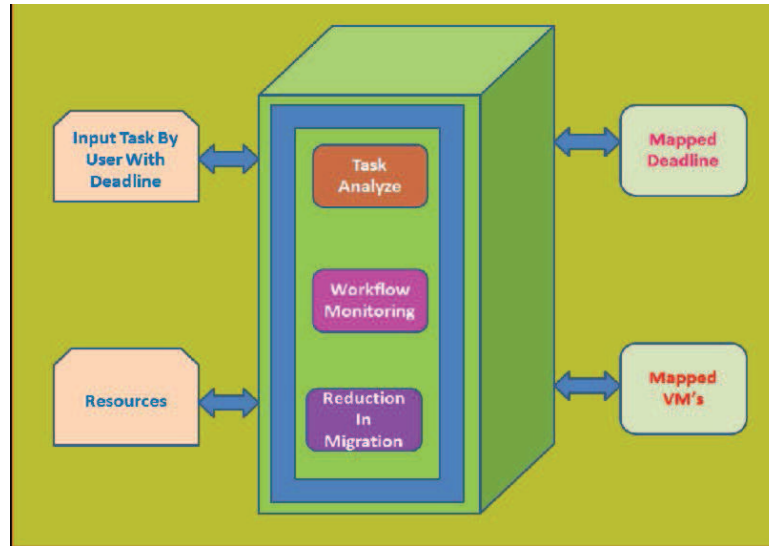


FIGURE 1. Typical System Architecture of Proposed Algorithm.

decide upon faster delivery of results or lower infrastructure costs. This research doesn't address the problem completely. Figure 1, Typically, system architecture of proposed algorithms is given. Figure 1, Shows the input task submitted by the user has to mapped with resources available in different VM's. VM's migration is depends on analysis made by task workflow and deadline parameter associated with task at the time of user submission.

3.2. VM Migration Cost. "Virtual machine migration plays important role in migrating Operating System instances across multiple physical machines". We consider VM's are heterogeneous in terms of resource capacity. The set $V_M = \{V_{M1}, V_{M2}, \dots, V_{Mm}\}$ is the set of finite number of VM's. Each VM is described

as follows. The set is the set of finite number of VM's. Each VM is described as follows

$$V_{Mj} = \{V_{MIdj}, V_{Mpj}, T_{Idj}, T_{quej}, H_{Idj}\}, \quad (1)$$

where V_{MIdj} is number of J^{th} virtual machine. Whereas V_{Mpj} is processing power of the VM. T_{Idj} is number of currently executing task. T_{quej} queue of waiting tasks assigned to this VM. And finally, H_{Idj} identification number of host where this VM is running.

Task Model. Let

$$T_s = t_{s1}, t_{s2}, \dots, t_{sn} \quad (2)$$

be set of tasks that generated in CS. Every task has parameters like Task identification number, Size of task, Status of task indicating whether task is executing, waiting, completed, allocated, etc. t_{VMk} is number of VM to which this task is allocated.

3.3. Cost-Minimization and Deadline-Constrained Model. ANN framework was used in [3] that reduce the infrastructure cost and it also offers speedy delivery of results. However, there was no consideration on QoS requirements. In addition, PSO was implemented in [4] that attain minimized implementation time along with high efficiency; nevertheless, fault tolerance constraints are not taken into account. Swarm intelligence techniques (SI) are promising solution for handling large scale problem and relatively faster.

3.3.1. Description on Objective Functions. This work concerns on attaining three objective functions namely:

1. Deadline (Makespan): "Deadline otherwise termed as makespan or completion time is the total time taken to process a set of jobs for its complete execution". It can be derived based on the of the total tasks. The computation of based on Fig. 2 is given by equation (3):

$$ECT(task_1) = (VM_1 + VM_3 + VM_2 + VM_1)/MIPS_{data} \quad (3)$$

$$ECT(task_2) = \frac{VM_4 + VM_5 + VM_2}{MIPS_{data}} m. \quad (4)$$

Thus, the makespan can be computed as per equation (5).

$$Makespan = Max (ECT (task_1) \dots ECT (task_{N_T})). \quad (5)$$

2. Solution Encoding: In the presented research model for optimal resource allocation, it is planned to optimize the virtual machines, such that the objectives (migration cost, utilization cost and make span) could be reduced. Fig 1 Portrait the natural architecture of task processed in cloud infrastructure whereas fig 2. Illustrate the applied solution encoding. In Fig. 2, 1, 3, 2 and 1 refers to the virtual machines that are mapped to task 1, similarly, 4, 5 and 2 refers to the virtual machines that are mapped to task 2.



FIGURE 2. Solution encoding for Task Allocation

4. PROPOSED ALGORITHM

Even though, the existing DA model [6] results with precise estimations; it also involves few drawbacks like "minimal internal memory, and slow convergence". Therefore, to prevail over the drawbacks of existing DA, the concept of SeanLion is merged with it to introduce a new model. Hybrid optimization algorithms have been reported to be promising for certain search problems [8]. The procedure of proposed method model is as follows: DA includes two significant stages: (i) Exploration and (ii) Exploitation. The procedure of the proposed logic is as follows: the modelling for separation and cohesion are computed based on S^1 and S^2 . As per the new logic, the separation and cohesion formula are computed as per equation (6) and equation (7), where B denotes a random vector obtained from SeaLion algorithm [7], S^1 indicates the l^{th} nearer individual position, signifies current individual position S and U reveals the count of the nearby individuals.

$$C_i = -l = 1U2BS - Sl \quad (6)$$

$$C_i = -l = 1U2BS - Sl. \quad (7)$$

Similarly, the formula for alignment A_i is modeled as given in equation (8), where, Q_1 symbolizes the velocity of l -th nearby individual. Attraction to food is evaluated as per equation (9), where S^+ points out food source position and S refers to the present position of individual

$$A_i = \frac{\sum_{l=1}^U Q_l}{U} \quad (8)$$

$$F_i = S^+ - S. \quad (9)$$

Distraction to enemy is represented by equation (10), in which the enemy position is indicated by S^- and S refers to the present position of individuals. So as to update the dragonfly's position, two vectors namely step (ΔS) and position S are computed as specified below

$$E_i = S^- + S. \quad (10)$$

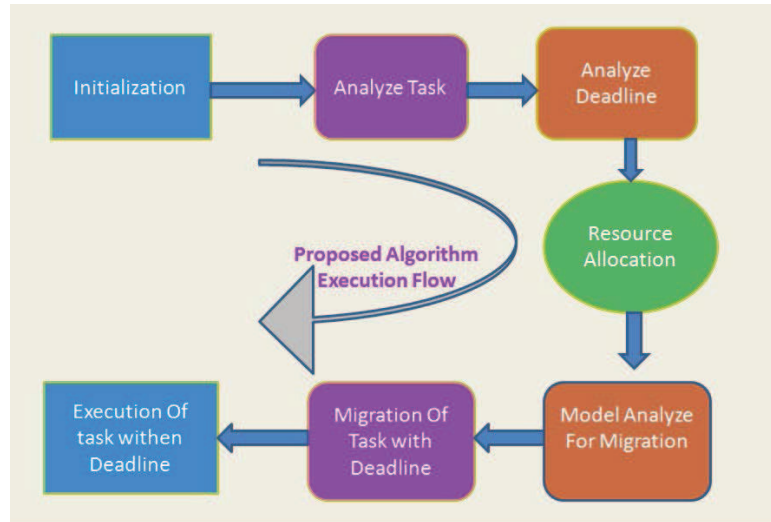


FIGURE 3. Flow of Execution of Proposed Algorithm.

The Levy flight is computed as given in equation (11), in which η is a constant value and r_1, r_2 indicates random integers that lies among $[0, 1]$. Further, δ is determined in equation (12), where $\Gamma(x) = (x - 1)$ Algorithm 1 depicts pseudo

code of the proposed hybrid version of multi objective PSO model and flow in portrait in Fig 3.

$$Levy(x) = 0.01 \times \frac{r_1 \times \delta}{|r_2|^{\frac{1}{\eta}}} \quad (11)$$

$$\delta = \left(\frac{\Gamma(1+\eta) \times \sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1+\eta}{2}\right) \times \eta \times 2^{\left(\frac{\eta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (12)$$

Algorithm 1 : Proposed Method

Initialization

While end condition cannot be obtained

 Compute objective value

 Update h,q,a,c,f and b

 Evaluate C , A , O , E_n and F as per equation (6-9)

 Update the nearby radius

 If dragonfly include one nearby dragonfly

 Update velocity and position based on equation (10) and equation (11)

 else

 Update levy based on equation (12)

 end If

 New positions are verified on the basis of variable boundaries

end While

5. RESULTS AND DISCUSSION

The presented task scheduling model using Proposed Method approach was implemented in Python along with libraries of MATLAB and the analysis was held. Here, synthetic dataset was used for analysis purpose. Accordingly, the performance of the presented technique was compared over the other conventional methods like MTA-S [1], PSO+GA [4] and MSDE [5] schemes. In addition, analysis was carried out by varying the number of tasks from 250, 500 and 1000. Moreover, examination was performed by varying the number of VM from 20 and 40.

5.1. Computational time Analysis by varying No. of VM. The computational time analysis of the suggested proposed model is summarized in this section by varying the counts of VM from 20 and 40. Table I reveal the time (represented

TABLE 1. ANALYSIS ON PROPOSED MODEL OVER EXISTING MODELS BY VARYING THE NUMBER OF VIRTUAL MACHINES

VM=20				
No.of tasks	MTA-S [1]	PSO+GA [4]	MSDE [5]	Proposed Method
200	7.300469	1.585228	2.834687	0.944865
600	21.8097	20.57337	21.47189	3.744312
1200	38.1482	1.585228	37.94493	12.98874
VM=40				
200	7.11672	1.558116	2.934807	1.033701
600	12.29334	7.368898	9.946115	3.32982
1200	38.53181	34.13123	37.9366	12.69217

in minutes) utilized by the presented proposed approach over the conventional models with respect to varied number of tasks from 200, 400, 600, 800, 1000 and 1200. For a better performance of the system, the computing time has to be minimal, which is attained by the presented model.

5.2. Computational Analysis by Varying No. of Tasks. Table II exposes the computational analysis of proposed method model over conventional models by varying the number of tasks from 250, 500 and 1000. Accordingly, the obtained outcomes for the offered proposed method model have shown minimal values for all counts of VM such as 10, 20, 30, 40 and 50. Primarily, the computing time of suggested method for task=250 is 85.1, 43.35 and 66.35 percent-Superior to traditional MTA-S, PSO+GA and MSDE models when the number of VM is 50. Furthermore, the computing time of implemented scheme for task =500 has obtained a minimal value of 2.524837, whereas, the evaluated MTA-S, PSO+GA and MSDE schemes has obtained the higher computing time values of 11.09581, 5.413038 and 7.63648 when the number of VM is 50. As a result, the development of the presented proposed method model is verified from the investigation outcomes.

6. CONCLUSION

This work addresses many features of cloud computing with a deadline scheme. As a result finally, we are able to discuss possible solutions and limitations. There

TABLE 2. ANALYSIS ON PROPOSED MODEL OVER EXISTING MODELS BY VARYING THE NUMBER OF TASKS

Tasks=250				
No.of VM	MTA-S [1]	PSO+GA [4]	MSDE [5]	Proposed Method
10	7.215419	1.982824	3.38725	1.104184
20	7.36295	1.990773	3.428076	1.135769
40	7.775287	2.052254	3.4506	1.161805
60	7.809198	2.053858	3.457907	1.163534
Tasks=1000				
10	26.32465	19.22036	22.81883	7.62611
20	26.34147	19.27335	23.01395	7.630493
40	26.63477	19.72564	23.11377	7.675852
60	27.51809	24.56872	26.74261	7.826187

are some grand challenges 1) Job Scheduling- jobs scheduling with laxity. Job scheduling is a parameter for interpretation between time (to run) and resources available i.e. VM's completely depends on parallelism of job. 2) Pricing - The proposed model presumes that deadlines are very much clear to the user before proceeding to job execution. 3) Learning - Our work significantly learned many things from observation. Most of the time framework observed time series sequence can be used to the prediction of the future. 4) Robustness- practically to run efficiently framework need robustness. Robustness should be accountable in both planned and unexpected failures in any dimension. Majorly resource failure is a big challenged to achieve robustness. With experiments we have been prove that result of proposed method for reducing cost is drastically optimized. We found that as a result, the development of the proposed method model is verified from the investigation outcomes on POS, MTA and MSDE. In future we will experiment performance analysis of deadline and migration cost with computing time values.

REFERENCES

- [1] M. ZAHARIA, T. DAS, ET.AL.: *Apache spark: A unified engine for big data processing*, Communications of the ACM, **59**(11) (2016), 56-65. <https://doi.org/10.1145/2934664>

- [2] V. SIMIC, B. STOJANOVIC, M. IVANOVIC: *Optimizing the performance of optimization in the cloud environment-An intelligent auto-scaling approach*, Future Generation Computer Systems, **101** (2019), 909-920.
- [3] G. PRADEEPINI ET. AL: *Opportunity and Challenges for Migrating Big Data Analytics in Cloud*, IOP Conference Series: Materials Science and Engineering, **225** (2017), art. id 012148. <https://doi.org/10.1088/1757-899X/225/1/012148>
- [4] J. DEAN, S. GHEMAWATAND: *MapReduce: simplified data processing on large clusters*, Communications of the ACM, **51**(1) (2008), 107–113. <https://doi.org/10.1145/1327452.1327492>
- [5] M.A. ELAZIZ, S. XIONG, K.P.N. JAYASENA, L. LI: *Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out*, European Journal of Mechanics a Solids, **66** (2017), 1-14.
- [6] M. JAFARI, M.H. BAYATI CHALESHTARI: *Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution*, Knowledge-Based Systems, **169** (2019), 39-52.
- [7] R. MASADEH, B. MAHAFAZH, A. SHARIEH: *Sea Lion Optimization Algorithm*, International Journal of Advanced Computer Science and Applications, **10** (2019), 388-395.
- [8] M. MARSALINE BENO, I.R. VALARMATHI, S.M. SWAMY, B. R. RAJAKUMAR: *Threshold prediction for segmenting tumour from brain MRI scans*, International Journal of Imaging Systems and Technology, **24**(2) (2014), 129-137. DOI: <https://doi.org/10.1002/ima.22087>
- [9] X. WANG, X. CHEN, C. YUEN, W. WU, M. ZHANG, C. ZHAN: *Delay-Cost Tradeoff for Virtual Machine Migration in Cloud Data Centers*, Journal of Network and Computer Applications, **78** (2017), 62-72. <https://doi.org/10.1016/j.jnca.2016.11.003>
- [10] G. PRADEEPINI ET AL.: *Experimenting cloud infrastructure for tomorrows big data analytics*, International Journal of Innovative Technology and Exploring Engineering, **8**(5) (2019), 885-890.

DEPARTMENT OF CSE, KLEF, VIJAYWAD, A.P, INDIA.
 E-mail address: asmanekar24@gmail.com

DEPARTMENT OF CSE, KLEF, VIJAYWAD, A.P, INDIA
 E-mail address: pradeepini.gera@gmail.com