Optimize Task Scheduling and Resource Allocation Using Nature Inspired Algorithms in Cloud based BDA

A.S. Manekar
Research Scholar, Department of Computer Science and Engineering, Klef, Green Fields, Vaddeswaram, Andhra Pradesh, India.
E-mail: asmanekar24@gmail.com

Dr. Pradeepini Gera
Professor, Department of Computer Science and Engineering, Klef, Green Fields, Vaddeswaram, Andhra Pradesh, India.
E-mail: pradeepini_cse@kluniversity.in

Received October 20, 2020; Accepted November 25, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V18SI01/WEB18049

Abstract

Task Scheduling and Resource allocation is a prominent research topic in cloud computing. There are several objectives associated with Optimize Task Scheduling and Resource allocation as cloud computing systems are more complex than the traditional distributed system. There are several challenges like resolving the task mapped to the node on which task to be executed. A simplified but near optimal proposed nature inspired algorithms are focus in this paper. In this paper basic idea about optimization, reliability and complexity is considered while design a solution for modern BDA (Big Data Application). Detailed analysis of experimental results, it is shown that the proposed algorithm has better optimization effect on the fair share policies which are presently available in most of the BDA. In this paper we focused on Dragonfly algorithm and Sea lion algorithms which are nature inspired algorithms. These algorithms are efficient for optimization purpose for solving task scheduling and resource allocation problem. Finally performance of the hybrid DA algorithm and Sea lion is compared with traditional techniques used for modern BDA using Hadoop MapReduce. Simulation results prove the efficacy of the suggested algorithms.

Keywords

Resource Allocation, Cloud, Big data, Deadline, Utilization Cost, Migration, Dragonfly Algorithm, Sea Lion Algorithm.

Introduction

In modern era of computing like bioinformatics, astronomy, physics, smart computing, weather data analysis and modelling for any data-driven scientific applications cloud computing with Big data application is used (Lee et al., 2019). Cloud computing needs computing resources on demand and virtualization is a prominent solution to this (Lee et al., 2019). It is necessary to schedule workflows on cloud in a way that reduces the cost of leasing resources. Scheduling tasks on resources is a NP hard problem, and use of meta-heuristic algorithms is an obvious choice for the same (Abd Elaziz et al., 2019). In this paper work is stimulated to Nature-inspired algorithms; particle swarm optimization like Dragonfly algorithm DA and Sea lion algorithm (Zhang et al., 2018). In cloud computing specially workflows in IaaS clouds using nature inspired algorithms are considered in this work. Metaheuristic optimization algorithms are simpler to implement and best suited for cloud computing (Buyya et al., 2009). Above all they can minimize local optima and they can be applied in a wide range of issues covering various disciplines (Zhang et al., 2019). In ultra large scale and high scalability like cloud computing large resource pools and idle resources are available. Dynamic allocation of application and mapping resources to these applications are treated with fair resource sharing policy in all BDA (Simic et al., 2019). In Modern cloud computing utilization of cloud infrastructure makes a large-scale optimization available to a significantly broad range of users offering a cloud-based optimization service. The aims of this research work are proposing a nature inspired metaheuristic EA for solving optimization problems. These solutions were mainly based on the classical laminate theory and on the fundamental works of Lekhnitskii (1968) (Zhang et al., 2017). These natures inspired metaheuristic EA are designed on multi-objective functions. In this work we have considered Deadline, Utilization Cost, and Migration Cost as objective for solving large-scale real-world optimization problems over heterogeneous computing resources. The main objectives of research are:

To make a clear review on different research works related to cost optimization- resource allocation and task scheduling in cloud and also to define the clear problem statement on this aspect.

To design a new model for cost optimization particularly resource allocation in cloud under big data analytics with Hadoop or Spark etc.

To introduce a new hybrid algorithm for solving the cost optimization problem that obviously enhances with respect to better convergence rate and speed, respectively.

To make a clear comparison of proposed model with other state-of-the-art models with respect to different analysis under cost saving, deadline and task completion.

Related Works

In 2019, Lee et al., introduce the task scheduling algorithm a hybrid approach using two of the most widely used biologically-inspired heuristic algorithms. They used genetic algorithms (GAs) and the bacterial foraging (BF) algorithms by considering some characteristics of both algorithms. First time researches consider makespan further it evolved as deadline of task. With makespan they also focused on energy consumption. In 2018, Chen et al., contributed a general multi-user mobile cloud computing (MCC) considering mobile users and their task submission. In 2019, Simic et al., introduced large-scale optimization in the elastic cloud environment. WoBinGO framework running in the IaaS environment simulation carried out for various comprehensive analyses. Scope of makespan is still not address. In 2019, Najme et al., experiment a hybrid task scheduling algorithm named FMPSO that is based on Fuzzy system and Modified Particle Swarm Optimization technique to enhance load balancing and cloud throughput. In 2018, Fredy et al., works on a polynomial-time algorithm that combines a set of heuristic rules and a resource allocation technique in order to get good solutions on an affordable time scale. In 2019, Zhang et al., heuristic algorithms in cloud task scheduling problems are address and studied in depth. Simulation based multiobjective proposed algorithms with characteristics of a cloud task scheduling strategy to minimize the task completion time and execution cost (MCTE) for the smart grid cloud. The experimental results show MCTE is well for the smart grid cloud. In 2018, Sobhanayak et al., proposed task scheduling algorithm using a hybrid approach. Second time used biologically-inspired heuristic algorithms, the genetic algorithms (GAs) and the bacterial foraging (BF) algorithms in the computing cloud. In 2017, Yingfeng et al., have implemented a dynamic game theory based two-layer scheduling method to reduce makespan, the total workload of machines and energy consumption to achieve real-time multi-objective flexible job scheduling.

Problem Definition

Heterogeneity in infrastructure with parameters like resolvability, extensibility and maintainability of underlying hardware support integrity in processing massively large data generated by different sources like public web, social media, university, business, research, sensors etc (Zhang et al., 2017); (Zhang et al., 2017). Mainstream cloud computing solutions are completely different and insufficient to process and compute massive intensive data hence need of modern BDA on available cloud is in tremendous demand (Zhang et al., 2019).

Task Scheduling and Allocation Problem

In cloud numerous physical Machines (PM) are combined parallel processing HPC (High performance computing) architecture having enough bandwidth. Each physical machine is mapped with various virtual machines (VM's). VM's are responsible to finish up task practically within minimum execution time (Simic et al., 2019); (Zhang et al., 2017).

For formulation of problem we have consider typically IaaS architecture with NPM Physical machines having NVM virtual machines.

$$CSys = \lfloor PM_1, PM_2, ...PM_{N_{PM}} \rfloor \quad (1)$$

In Eq. (1), the PM's present in cloud is denoted by $PM_i = [i = 1,...N_{PM}]$ and it is represented as shown in Eq. (2).

$$PM_i = \lfloor VM_1, VM_2, ...VM_k, ...VM_{N_{VM}} \rfloor \quad (2)$$

In Eq. (2), $VM_k,\ k = 1,2...N_{VM}$ denotes k^{th} virtual machine in cloud and to identify total number of virtual machines a count is indicated by N_{VM}. Eq. (3) explain a simple attributes of VM_k.

$$VM_k = [S_{IDV\,k},\ M_{IPS\,k}] \quad (3)$$

Cloud User has to submit the task time to time for execution. The major objective is to minimize the makespan by allocating the optimal set of tasks to be executed on VMs.

$$E_{CT_{l,k}} = \frac{task - length}{M_{IPS_k}}, k = 1,2\ldots N_{VM}, l = 1,2\ldots N_T \quad (4)$$

Where $ECT_{l,k}$ refers to the required execution time of l^{th} task on k^{th} VM.

Description on Objective Functions

Proposed algorithm considers Multi-objective function consisting of Deadline (Makespan), Utilization cost, Migration cos. First understand the basic definitions of all three.

Deadline (Makespan)**:** Completion time is the total time taken to process a set of jobs for its complete execution is term as deadline of task. This time also called as Makespan its is calculated as,

$$M_{kspan} = Max\left(ECT\left(task_1\right).....ECT\left(task_{N_T}\right)\right) \quad (5)$$

Utilization cost**:** The cost incurred to utilize the virtual machine is term as Utilization cost. It is calculated on in VM's basis,

$$Cost\left(task_1\right) = Cost\left(VM_1\right) + Cost\left(VM_2\right) + . \\ .....Cost\left(task_{N_T}\right) \quad (6)$$

Migration cost**:** It can be evaluated based on the switching, which takes place between the virtual machines that are assigned to each physical machine. In general, there exists two types of switching modes in physical machines, they are: inter PM switching and intra PM switching. If the switching takes place between the same PM, it is said to be intra PM switching, whereas, if the switching takes place between two diverse PM, it is said to be inter PM switching.

$$M_{GC} = Max\left(M_{GC}\left(task_1\right) + ...M_{GC}\left(task_{N_T}\right)\right) \quad (7)$$

Methodology

Map Reduce Framework for Objective Calculation

Now-a-days, huge amount of data is generated from social media, mobile devices, IoT and many other emerging applications. Therefore, data processing and analytics have become really important in all the major domains, such as research, business and industry (Abd Elaziz et al., 2019). Resource sharing among applications will be prevented. Second, in static resource allocation, the user has to manually set the amount of resources each application is going to use. Even with dynamic resource allocation, the user still has to set the initial amount of resources. This proposal aims to introduce a new resource allocation model under MapReduce. Fig.2. Shows Map Reduce model is operated under two phases:

"the map phase and the reduce phase" (Buyya et al., 2009). Under the mapping phase, the master node divides the input into small splits of shared data that are then assigned to worker nodes. Therefore, the final Multi objective function MOF is defined as shown in Eq. (8) (Zhang et al., 2017).

$$MOF = Avg(Ranks) \tag{8}$$

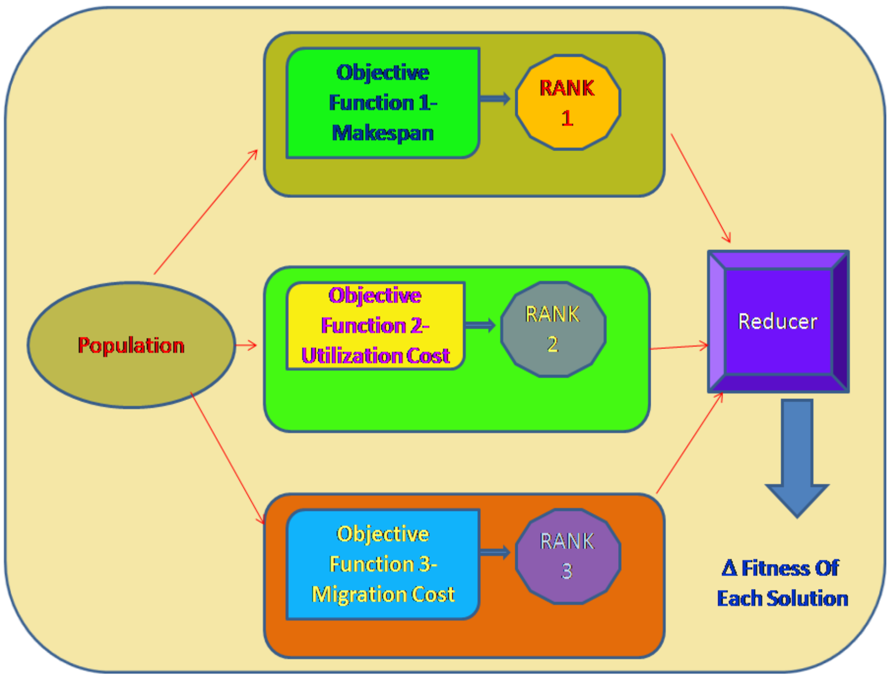


Figure 1 MapReducing framework With Ranks

Multi-objective Optimization to VM Deployment

This model insists the utilization of optimization concept for resource allocation and migration with respect to certain constraints or parameters like Deadline, cost saving and utilization cost (resource saving). Moreover, this proposal aims to define a new objective function that combines the execution of fine-grained resource allocation and migration process. For this, a new hybrid algorithm that hybrids the concept of Dragonfly (DA) and Sea Lion Algorithm (SLnO) is introduced. DA (Zhang et al., 2019) is a new meta-heuristic optimization approach that solves the single-objective, discrete and multi-objective problems. The SLnO algorithm (Abd Elaziz et al., 2019) imitates the hunting behaviour of sea lions in nature. Moreover, it is inspired by sea lions' whiskers that are used in order to detect the prey.

$$C_i = -\sum_{l=1}^{U}\left(2\overline{B}S - S_l\right) \quad (9)$$

$$O_i = \frac{\sum_{l=1}^{U_b} S_l}{U_b} - 2\overline{B}S \quad (10)$$

The Levy flight is computed as given in Eq. (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 Eq. (12), where $\Gamma(x) = (x-1)$. Algorithm 1 depicts pseudo code of the proposed MF- DA model (Zhang et al., 2017).

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

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

Algorithm 1 reveals the pseudo code of presented Proposed Hybrid Algorithm scheme.

Algorithm 1 : Proposed Hybrid Algorithm

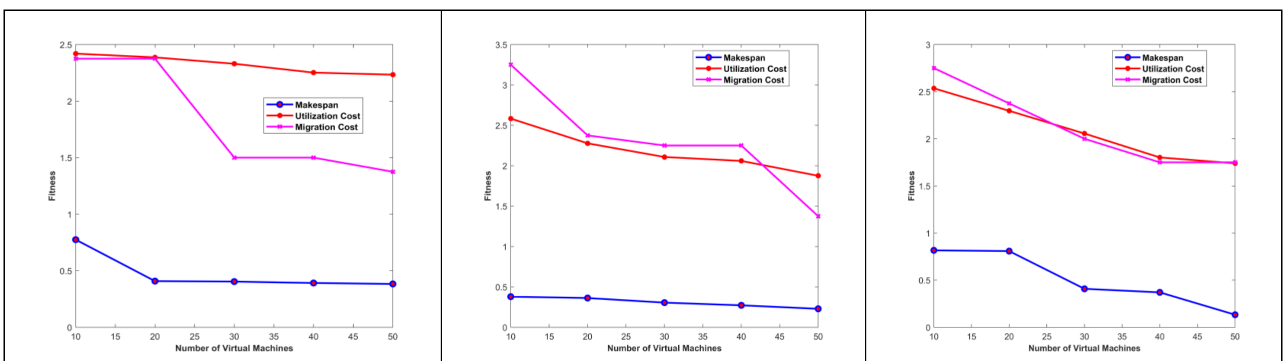
```
Initialization
While end condition cannot be obtained
   Compute objective value
   Update h, q, a, c, f and b
   Evaluate C A, O, En and F as per Eq. (9,10)
   Update the nearby radius
   If dragonfly include one nearby dragonfly
      Update velocity and position based on Eq. (10) and Eq. (11)
   else
      Update levy based on Eq. (12)
   end If
   New positions are verified on the basis of variable boundaries
end While
```

Results and Discussion

Proposed Hybrid Algorithm approach was implemented in Python with Matlab Libraries importing and the analysis was held on synthetic data set. Conventional methods like MTA-S (Lee et al., 2019), PSO+GA (Zhang et al., 2019) and MSDE (Abd Elaziz et al., 2019) schemes are compared with Proposed Hybrid Algorithm algorithms. Analysis was carried out on proposed model by varying the No. of tasks and Analysis on Proposed model by varying the No. of VM finally we had Analysis by varying No. of VM: Proposed Vs. Conventional method as shown in Table.1.

Table 1 Analysis on Proposed Model Over Existing Models by Varying the Number of Tasks

Tasks=250				
No. of VM	**MTA-S [1]**	**PSO+GA [2]**	**MSDE [3]**	**CD-SLnO**
10	5.186605	4.637483	3.604205	1.856467
20	5.159657	3.841948	3.408779	1.723177
30	5.138312	3.586152	3.36563	1.411334
40	5.045676	3.468356	3.04663	1.381092
50	4.643918	3.111177	2.758316	1.330547
Tasks=500				
10	5.645396	4.867836	4.209766	2.070877
20	5.358196	4.499468	3.77954	1.671485
30	5.295545	4.021937	3.669577	1.554451
40	5.143199	3.768941	3.507199	1.527223
50	4.577035	3.315331	3.025743	1.159788
Tasks=1000				
10	5.243726	4.223274	4.098746	2.033993
20	5.23148	4.189252	4.074642	1.826618
30	5.008607	4.101139	3.597413	1.487983
40	4.861747	3.769428	3.520763	1.307546
50	4.579923	3.641037	2.868058	1.207604



(a) **(b)** **(c)**

Figure 2 Analysis on proposed CD-SLnO technique by varying the number of tasks from (a) 250 (b) 500 and (c) 100

With Graph representation in Fig. 2 we can analyse that the fitness of makespan, utilization cost and migration cost using Proposed Hybrid Algorithm model has reached the minimal value. Makespan with a lower value of 0.75 For virtual machines is 10 whereas migration cost minimal value of 0.4 and finally utilization cost reduced value of 1.9. With Fig. 2 We can say that whenever we add more number of virtual machines the cost is drastically reducing for 20, 30, 40 and 50 numbers of virtual machines. For Analysis on Proposed model by varying the No. of VM we varied number of tasks from 200, 400, 600, 800, 1000 and 1200. We observe that Here, the fitness (makespan, utilization cost and migration cost) is critically outperform. For varying task we achieved minimal value of 0.3 to 1.1. For conventional methods analysis is showing the fitness of presented scheme for VM=20 is 60.46%, 54.74% and 48.17% better than existing MTA-S, PSO+GA and MSDE models when the number of tasks is 200.

Conclusion

As stated objective in earlier section and result based analysis for Task Scheduling and Resource allocation model, we tried a Proposed Hybrid Algorithm for resource allocation along with migration process. Parameters like deadline, utilization cost and migration cost we critically evaluated and analyses for Task Scheduling and Resource allocation. Here proposed Hybrid Algorithm for modern BDA in cloud has proven and shown benchmark. In result and analysis section Primarily, the computing time of suggested method for task=250 was 85.1%, 43.35% and 66.35% superior to traditional MTA-S, PSO+GA and MSDE models when the number of VM was 50. With variation and enhancement we achieved minimal time for computation. Thus, the enhancement of the presented Proposed Hybrid Algorithm was proved in an effectual manner. Further in future we are planning to enhance this take with more multiobjective functions and parameter for minimizing overall cost for both the cloud service provider and cloud users.

References

Lee, J.W., Jang, G., Jung, H., Lee, J.G., & Lee, U. (2019). Maximizing MapReduce job speed and reliability in the mobile cloud by optimizing task allocation. *Pervasive and Mobile Computing*, *60*, 101082.

Abd Elaziz, M., Xiong, S., Jayasena, K.P.N., & Li, L. (2019). Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. *Knowledge-Based Systems*, *169*, 39-52.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, *25*(6), 599-616.

Simic, V., Stojanovic, B., & Ivanovic, M. (2019). Optimizing the performance of optimization in the cloud environment–An intelligent auto-scaling approach. *Future Generation Computer Systems*, *101*, 909-920.

Zhang, H., Shi, J., Deng, B., Jia, G., Han, G., & Shu, L. (2019). MCTE: minimizes task completion time and execution cost to optimize scheduling performance for smart grid cloud. *IEEE Access*, *7*, 134793-134803.

Zhang, Y., Wang, J., & Liu, Y. (2017). Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact. *Journal of Cleaner Production*, *167*, 665-679.

Jena, R.K. (2017). Energy efficient task scheduling in cloud environment. *Energy Procedia*, *141*, 222-227.

Zhang, Y.W., Wang, C., & Liu, J. (2018). Energy aware fixed priority scheduling for real time sporadic task with task synchronization. *Journal of Systems Architecture*, *83*, 12-22.