PalArch's Journal of Archaeology of Egypt / Egyptology

A TASK SCHEDULING APPROACH IN CLOUD COMPUTING TO MINIMIZE THE POWER COST IN DATACENTERS USING CROW SEARCH

Sudheer Mangalampalli
Research Scholar, Dept.of Computer Science and Engineering, Centurion University of Technology and Management, Odisha, India

Vamsi Krishna Mangalampalli
Professor, Department of CSE, Chaitanya Institute of Science and Technology, AP, India

Sangram Keshari Swain
Associate Professor, Department of CSE, Centurion University of Technology and Management, India

Sudheer Mangalampalli, Vamsi Krishna Mangalampalli, Sangram Keshari Swain : A Task Scheduling Approach in Cloud Computing to Minimize the Power Cost in Data centers Using Crow Search - Palarch's Journal of Archaeology of Egypt / Egyptology 17(6), ISSN 1567-214x

Keywords : Task Scheduling, Cloud Computing, Crow Search algorithm-CSA, make span , Power Cost Task Priority, VM priority, Ant Colony Optimization- ACO, Particle Swarm Optimization-PSO, Cuckoo Search algorithm- CS

Abstract
Cloud Computing is evolved as an enormous paradigm in the IT industry. This model enables the users to use the resources in the cloud in the form of services. Task Scheduling is an extensive problem in Cloud computing. To effectively map the tasks onto the suitable virtual machines in the cloud computing a Task scheduler is necessary. Task Scheduler in the cloud computing aimed at minimizing the makespan of the tasks and appropriately map the tasks to the suitable virtual machines. In the literature, many of the authors used evolutionay algorithms to solve the scheduling problem in Cloud Computing. The existing algorithm focuses on the metrics makespan, utilization of the resources. Power for the Datacenters is an important metric in the view of Cloud provider. In the existing algorithms, Power Cost metric at the datacenters was not that much highlighted. In this paper, our focus is to develop a multi objective task scheduling algorithm which schedules the tasks to the appropriate virtual machines by considering the electricity unit cost in datacenters by minimizing the makespan of the tasks and the Power Cost at the Datacenters. We have used Crow Search algorithm to solve the scheduling problem. It is simulated on the CloudSim simulator and it is compared with the existing algorithms ACO, PSO and CS and our approach surpass the existing algorithms in terms of makespan and the Power Cost.

1. Introduction

Cloud Computing Paradigm shifts the entire IT paradigm to a different level where everybody can access the virtual resources in the cloud as a service on demand. This model enables users to develop and host their applications on the cloud without any underlying hardware considerations. The computations, hosting and development of the applications on the cloud will be done seamlessly due to the on demand provisioning of the resources with the virtual infrastructure. The provisioning of the resources to the user will be done automatically based on the service level agreement made between the Cloud user and provider and based on the SLA the task scheduler will assign the virtual resources to the user automatically. This assignment of the virtual resources to the user generated tasks to be efficiently done by an effective task scheduling mechanism in Cloud Computing paradigm. Task scheduler plays an active role in assigning the tasks onto the appropriate virtual machines in the cloud. The quality of the services of the cloud depends on the effectiveness of the task scheduler. The primary goal of the task scheduling is to minimize the task execution time and effective utilization of the resources in the cloud. Power Cost at the datacenters is also one of the important parameter in the view of the cloud provider. If we have a scheduler which suitably maps the tasks to the virtual machines thereby reducing the makespan which can reduces the power cost at the datacenters. Power cost at datacenters can be calculated on hourly basis for the energy consumption. It depends on the electricity grid, diesel generators and the renewable resources. In the existing algorithms, many of the authors used Particle Swarm Optimization, Ant Colony Optimization and Cuckoo search algorithms to address the task scheduling problem in Cloud Computing but due to the diversified nature of the cloud computing, it is difficult to get the accurate convergence towards the solution. In this paper, we have used a nature inspired algorithm i.e., Crow Search Algorithm to solve the task scheduling in cloud computing by minimizing the makespan and the power cost at the datacenters. The key points of this paper are highlighted below.

- A task scheduling mechanism is developed using Crow Search algorithm which is based on the food habits of the crows.
- For this approach, we will calculate the priorities of Tasks and Virtual machines to appropriately assign the tasks to the suitable virtual machines.
- This mechanism focuses on minimizing the makespan of the tsaks and the Power Cost at the datacenters.
- It is implemented using the Cloudsim [19] and the tasks are generated using the workload in the planetlab.

2. Related Works

In [1] task scheduling algorithm is formulated to minimize the makespan of the tasks in the cloud computing. This algorithm is modeled using the PSO algorithm. In this approach, logarithmic decreasing approach was used to get the faster convergence. It is compared with the existing artificial beec colony, dragonfly algorithms and the proposed algorithm gets fair convergence rather than the existing algorithms and makespan is greatly reduced over the existing algorithms. In[2] authors formulated a task scheduling algorithm is proposed to increase the resource utilization and minimize the total completion time of the tasks. This algorithm is modeled using the Genetic algorithm for mapping of the tasks onto the resources in the cloud. It is implemented on the cloudsim and it is compared against the basic round robin, Genetic algorithms and the proposed approach is outperformed in terms of the specified metrics. In[3], a hybrid task scheduling algorithm is developed by combining cuckoo search and oppositional based learning algorithms. This approach focuses on the minimization of makespan and the cost of the resources. It is implemented on the cloudsim and this approach is greatly minimizes the mentioned metrics over the existing algorithms PSO, IDEA and GA algorithms. In[8] a new paired task scheduling algorithm is formulated which aimed at minimizing the layover time of the tasks. This algorithm is modeled based on the Hungarian algorithm and it is implemented using MATLAB. It is compared against the Hungarian with lease time and converse lease time and FCFS algorithm and the proposed algorithm greatly reduces the lay over time rather than existing algorithms. In [5] a task scheduling algorithm is formulated which focuses on the scheduling length and ratio of successful execution of the task. It is modeled using the particle swarm optimization and it is implemented on the cloudsim. It is evaluated against the basic PSO, MIN-MIN, MAX-MIN algorithms and it surpasses the comparing algorithms interms of the mentioned metrics. In[6], a new task scheduling mechanism is proposed to minimize the makespan. It is modeled using the combination of chaotic local search with SOS to get the finer convergence and simulated annealing is also added to SOS to avoid local trapping. It is implemented in MATLAB and compared against the existing SOS variants and outperforms in terms of makespan and convergence. In[7], authors proposed a scheduling approach which can be used to efficiently allocate the tasks to the virtual machines in the cloud. It is modelled based on the ant colony algorithm by modifying the diversification and reinforcement due to which the ants won't be diverted from the path for the food. It is compared against the existing variants of ACO and the results outperforms in terms of maximization of the resource utilization. In[8] a task scheduling algorithm is formulated to minimize the energy consumption in cloud computing in heterogeneous environments. It is implemented in the MATLAB and it is evaluated against the synthetic datasets and it outperforms in terms of makespan and energy consumption. In[9], a task scheduling is formulated which can be used to effectively allocate the tasks to the virtual machines by using fission strategy by modifying the PSO in order to avoid the premature convergence. It is implemented on the customized platform and it is compared against the existing algorithms PSO, IPSO and FIFO and the results proved that it surpasses the existing algorithms in allocating the resources effectively and avoids the premature convergence. In [10], an algorithm is formulated which can be used to minimize the makespan of the tasks and increases the convergence rate. It is modeled by using genetic algorithm for the scheduling and for clustering the workloads k-means algorithm is used. It is implemented in MATLAB and it is showing the great impact over the existing algorithm GA in terms of makespan and the convergence rate. In [15], a hybrid task scheduling algorithm is formulated using GA and PSO algorithms. Task Priority is calculated and assigned the tasks to the resources in the cloud. It is implemented in the cloudsim and evaluated against existing GA, PSO algorithms and it is outperformed in terms of availability and scalability. In[12], a bi objective task scheduling strategy is developed to minimize the task execution time and the processing cost. It is modeled using genetic algorithm and it is implemented in Cloudsim and the experiments revealed that it surpasses the existing algorithms in the specified metrics. In[13], a scheduling mechanism is proposed to schedule the tasks optimally by minimizing the makespan and the total cost for the scheduling process. It is modeled using whale optimization algorithm and it is implemented on the cloudsim and it is evaluated against the existing schedulers PBACO and SLPSO and it shows a great impact over the existing algorithms in the view of specified metrics. In[14] ,focuses on the optimization of the processing cost for task scheduling in cloud computing. It is modelled using by adaptive IPSO algorithm. It is implemented in the cloudsim and it is compared with the existing PSO algorithm and it is outperformed in the view of minimization of total processing cost for the task scheduling. In[15], a task scheduling algorithm is proposed to minimize the makespan of the tasks. This algorithm is modeled by using the crow search algorithm. It is implemented in the cloudsim and it is compared against the MIN-MIN and ant colony approaches and it surpasses the existing approaches in the view of the specified metrics. From the existing literature, many of the authors used the metaheuristic algorithms mentioned as

PSO, CS, GA and ACO to solve the scheduling problem in cloud computing. Authors have tried to address the metrics like makespan, Resource utilization, Total Processing cost, Execution time. In this paper, we are proposing a task scheduling approach using a crow search algorithm which minimizes the makespan and the power cost at the datacenters.

3. Problem Formulation

Let us consider that k number of tasks namely $\{t_1, t_2, t_3, t_4, t_5, \text{----}\ t_k\}$,n virtual machines which can be represented as $\{V_1, V_2, V_3, V_4 \text{-----}\ V_n\}$. The k number of tasks are to be mapped onto the n number of virtual machines which were running in j datacenters. The datacenters are represented as $\{d_1, d_2, d_3, d_4 \text{----}\ d_j\}$ Initially, these k tasks were submitted to the task manager, after submission of the tasks the input tasks are to be analyzed for the calculation of the priorities of the tasks and the VMs based on the price per unit cost of the electricity at the corresponding datacenters.

We can define the problem in such a way that the suitable tasks are to be mpped optimally onto the appropriate virtual machines by calculating the priorties of both tasks and VMs while minimizing the makespan of the tasks and the power cost at the datacenters.

The below Table 1 represents the notations which can be used in the proposed algorithm

Table 1: Notations used in Architecture

Notation	**Meaning**
t_k	Number of Tasks
v_n	Number of Virtual Machines
d_j	Number of datacenters
vp	Priority of Virtual Machines
Tp	Priority of tasks
Lo_v	Load on the Virtual Machines
$capa_V$	Capacity of the Virtual machines
T_l	Length of a task
Lo_h	Load on the Physical machine
$T\ HR_k$	Threshold value
pr_n	Number of processing elements in a VM
pr_m	processing capacity of a VM
$Total_V$	Total capacity of all the VMS
ms_T	makespan of task
energyc	Energy Consumption
T_m	Processing capacity of a task
Tr	Processing elements of a task
EC_h	Highest Electricity cost among all the datacenters
EC_d	Electricty cost at a corresponding datacenter
PC	Power cost in datacenters

4. System Architecture

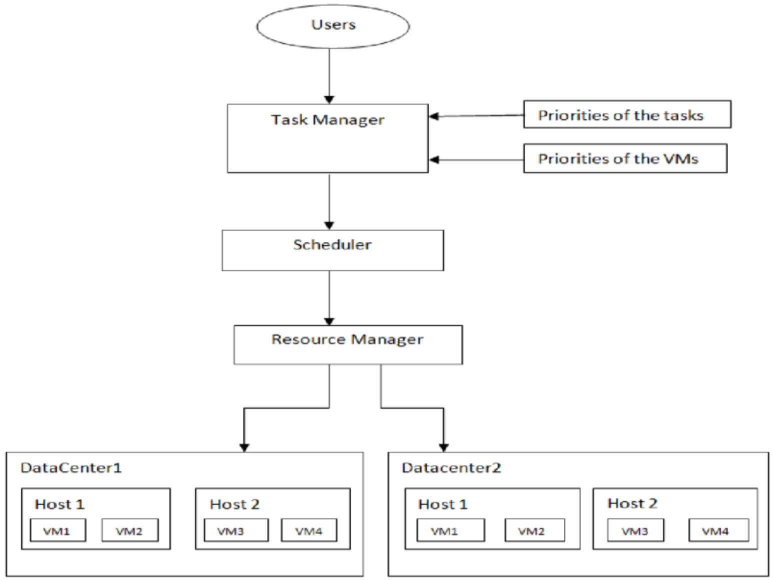

Fig. 1: Proposed system architecture

The proposed system architecture is shown in the above figure1. Initially the users give the input the tasks on to the cloud interface, upon receiving the inputs from the user task manager will submit these tasks to the scheduler which needs to map the tasks effectively to the virtual resources running in the datacenters. Here in this architecture whenever these tasks are submitted to the task manager, it needs to calculate the priorities of the tasks based on the size and processing capacity of the task and the VM priority based on the electricity cost at the corresponding datacenter. Upon identification of these priorities the suitable tasks are mapped optimally onto the Virtual resources are running in the physical hosts which were running in the datacenters.

For this to happen, priority of the tasks have to be calculated. The calculation of the priority of the task depends on the load on the virtual machines and load on the physical hosts. The complete load on all the virtual machines can be represented here as

$$LO_v = \sum_{n=1}^{p} \mathrm{LO_n} \qquad (1)$$

where n indicates number of virtual machines LO_v indicates the load on all the virtual machines.

To calculate the priority of the task we also need to calculate the entire load on the physical host. The complete load on the all the physical hosts be calculated as the ratio of load on the virtual machines to the summation of the physical hosts.

$$\mathrm{LO_h} = \frac{LO_v}{\sum_{i=1}^{n} \mathrm{p_h}} \quad (2)$$

where ph is the physical host After calculating the load on the virtual machines and the physical hosts we need to identify the threshold value to identify whether a virtual machine is ovelroaded, underloaded or balanced. The threshold value can be identified as

$$\mathrm{THR_k} = \mathrm{LO_h} * \mathrm{p_h} \qquad s(3)$$

We have identified the threshold value then a VM can be overloaded if

$$\text{If} \quad \mathrm{V} > \mathrm{THR_k} - \sum_{i=1}^{n} LO_v \qquad (4)$$

a VM can be underloaded if

$$\text{If} \quad V < THR_k - \sum_{i=1}^{n} LO_v \quad (5)$$

a VM can be balanced if

$$\text{If} \quad V = THR_k - \sum_{i=1}^{n} LO_v \quad (6)$$

To calculate the priority of the tasks, we also need to calculate the capacity of a VM. It can be calculated as

$$Capa_v = pr_n * pr_m \quad (7)$$

Where pr_n is number of processing elements in VM and pr_m is processing capacity of a VM.

Total capacity of all the VMs in a physcial host is calculated as

$$Total_v = \sum_{i=1}^{n} Capa_v \quad (8)$$

Priority of the task depends on the length and the processing capacity of the task. Length of the task can be calculated as

$$T_l = T_m * T_r \quad (9)$$

Now we can calculate the priority of the task can be calculated as given below

$$T_p = \frac{T_l}{Capa_v} \quad (10)$$

where Tl is the length of the task and capav is the capacity of the VM. The priority of a VM can be calculated based on the electricity price per unit cost at different data centers. It can be calculated as the ratio of highest electricty price among all the datacenters to the price at the corresponding datacenter. It can be given as

$$V_p = \frac{EC_h}{EC_d} \quad (11)$$

Where EC_h is the highest electricty price per unit cost among all the datacenters and EC_d is the Electricity price per unit cost at the corresponding datacenter. Finally we have calculated the priorities of task and VMs through which tasks will mapped optimally onto the suitable virtual machines. Tasks will be mapped onto the VMs as if a task is having highest priority it should be mapped onto a high prioritized VM i.e, VM which is running in a datacenter where the electricity cost is low. If a task comes onto the scheduler, then it will probe the resource manager to get the available resources in the physical hosts which were running at the datacenters. It will get the information and it will notify the information about the resources to the scheduler. Then the scheduler need to check both the priorties of Task and VM, then it will map the tasks optimally to the VMs in the datacenter where the price per unit cost is less. For example, if a high priority task arrives at the scheduler, then scheduler checks for the priority of both the tasks and VM and it have to check a suitable VM which can process this task with the less electricity cost. If all the VMs are running in the physcial host are busy in processing the high prioritized tasks rather thn the current task then the scheduler have to identify another VM which is suitable to the corresponding tasks based on the constraints i.e, Priorities of Task and VM.If the VM is busy in processing the tasks considerably having less priority rather than the current task then the scheduler will migrate the low prioritized task to the next VM in the same host or in the different host to run the current task into the VM. In this task scheduling approach, our aim is to minimize the makespan of the tasks and to minimize the power cost at the datacenters. The primary objective of any task scheduling algorithm is to minimize the makespan which can be defined as the total time taken for execution of the tasks. It can be calculated as

$$m_t = a_n + Et_k \quad (12)$$

Where an represents the availability of a virtual machine and Et_k represents execution time of a task on a virtual machine In this paper, we also focussed to minimize the power cost at the datacenters. Scheduling of the tasks in the cloud also impacts the power cost in the datacenters. If we can schedule the tasks based on their priorities to the suitable virtual machines based on the electricity cost at the datacenters then power cost at the datacenters can be minimized. Here in this approach we are not only scheduling the tasks according to the priority but also scheduling the VMs according to the electricity cost. Generally, Power cost in the datacenters is calculated based on the consumption of the energy per hourly basis. Power cost in the datacenters mainly depends on the [16] energy retrieving from the grid, diesel generators and renewable energy resources. Power cost in the Datacenter can be calculated as given below.

$$PC = cost^{g} + \left(cost^{d} * \alpha\right) + cost^{r} \quad (13)$$

where $cost^{g}$ represents the hourly energy cost incurred due to the electrical grid, $cost^{d}$ represents the energy consumed cost due to the usage of the power from the diesel generators, α indicates whether the diesel generator switched on or off. Here the diesel generators used in the datacenters as a power backup for the electrical grids and the green resources. $cost^{r}$ represents the energy consumed cost due to the renewable resources i.e, green resources. Note that the power cost is calculated from all these resources are in hourly basis. Where $cost^{g}$ can be calculated as follows

$$cost^{g} = \sum Energy_{c}(d,h) * p(d,h) \quad (14)$$

where $Energy_{c}(d,h)$ represents consumption of energy at the datacenter per hour and $p(d,h)$ represents the energy price per hourly basis at the datacenter.

5. Crow Search Algorithm

Crows are [17] intellectual birds in nature. They are having their own style of collecting food and store the food at particular places. They can remember the places where their food was stored easily. They can remember other faces of the crows and they can warn other crows due to the disagreeable nature with other crows. They have the cleverness to communicate with other crows and they will observe other crows where the others were storing their food. They can also save the food from other crows when the others were trying to steal their food. Crows generally lives in the form of groups. They follow other crows to identify the food storage places of others. In this approach, each crow will remember the storage location of their food for every iteration. Consider k iterations for this algorithm and there are n number of crows are present in the group. For an instance, Two crows named as i and j are selected among the n number of crows in the group. Now we can identify the current position of the crow i at kth iteration is $x_{i,k}$ and its best food storage location is memorized by the crow i is represented as $m_{i,k}$. Now if the crow wants to change its food storage location, it needs to find and update its storage location. For that assume if crow i decided to change its food location and it flies up and at the same instance of time another crow j is also flies upto identify the location for the storage of food and in this case the crow j does not know that the crow i is following its pathway then the new food location can be calculated as follows

$$x_{i,k+1} = x_{i,k} + r_{i} * fl_{i,k} * (m_{j,k} - x_{i,k}) \quad (15)$$

Where r_{i} is the random value chosen between 0 and 1, $fl_{i,k}$is the flight length of the crow i. The searching process for the new food storage location can be done locally or globally based on the flight length chosen by the crow. if the flight length is less than 1 then the searching will happens locally and if the flight length is greater than 1 the searching process can be done globally. In the another case if the crow J knows that crow i is following its path then it will take a random path by diverting the crow i to protect its food at the location. if $r_{j} \geq awp_{i,k}$ then the food location can be identified as in the above mentioned eqaution16. Otherwise it will take a random position to divert the other crow. In this algorithm, Awareness probability will mitigates the unbalance between intensification and diversification.

6. Proposed Algorithm

To schedule the tasks effectively onto the appropriate virtual machines in the cloud computing environment, here we are using crow search algorithm to optimally map the tasks onto the VMs. Initially for the scheduling process, we have identified the number of tasks (k), the VMs (n) and the number of iterations for CSA algorithm. After identifying the VMs, Tasks and the number of iterations we need to initialize them. Initially for scheduling process, here in this approach, we will calculate the priorities of the tasks (Tp) and the priority of VMs (Vp) based on the electricity price is identified . After calculation of the priorities for scheduling process initially a random task is identified and based on the awareness probability it needs to map the task to the corresponding virtual machine and also need to consider the priorities as constraints to map the tasks to the VMs. The mapping process and the assignment of task to the VM is as calculated as below

$$V_x = MT_{i,k} + r_i fl_{i,k} * (MT_{j,k} - MT_{i,k}) \quad (16)$$

Now we need to identify the new fitness values are feasible or not i.e. makespan and power cost. If the new fitness values are less than the previous values then update these values and move them from the temporary mapping table (TMT)to the mapping table(MT). If the new fitness value are not less than the current values then the the mapped VM will be copied from (MT) to (TMT). These steps were repeated for k until all the tasks were completed their execution based on the mapping of tasks done using the equation 16. Now the makespan and power cost are identified and if they are less than the previous values then the values are moved from (MT) to (TMT). otherwise the current values of makespan and powercost are maintained which are located at the mapping table(MT).

Input:

Tasks $\{t_1, t_2, t_3, t_4, t_5,---- t_k\}$,

Output: Optimal mapping of the tasks to the suitable Virtual machines while minimizing the makespan and the Power Cost at the Datacenters.

Step1: Start

Step2: Initialize the number of tasks(T_k), number of virtual machines(V_n),number of datacenters(d_j), number of iterations(n), flight length(f l), Awareness Probability(awp)

Step3: calculate the priority of tasks Tp

Step4: calculate the priority of VMs Vp

Step5: repeat

Step6: for k= 1:n

Step7: select a task T_i randomly

Step8: r_i = random value between 0 and 1

Step9: r_j = random value between 0 and 1

Step10: if $r_j \geq awp_{i,k}$

Step11: $V_x = MT_{i,k} + r_i fl_{i,k} * (MT_{j,k} - MT_{i,k})$

Step12: else

Step13: V_x = randomly map the task to the VM

Step14: endif

Step15: evaluate nm_t and nPC values for the task T_i in V_x

Step16: if (nm_t <m_t) ^(nPC_t<PC_t)

Step17: TMT[i] = V_x
Step18: else
Step19: TMT[i] =MT[i]
Step20: endif
Step21: evaluate nm_t and nPC values from TMT
Step22: if (nm_t <m_t) ^(nPC_t<PC_t)
Step23: update mapping values from TMT to MT
Step24: $m_t = nm_t$
Step25: $PC_t = nPC_t$
Step26: endif
Step27: Increment the count
Step28: until $count \leq n$
Step29: end

7. Simulation and Results

This algorithm is implemented in the cloudsim simulator[19]. For this simulation, tasks were generated randomly by using the workload in the planetlab. The simulation ran in the system with intel i5 processor,8 GB RAM and 500GB hard disk. We have considered 1000 cloudlets for this simulation.The simulation settings were taken from[18] Simulation settings were represented in the below table2 .

Table 2: Simulations settings

S.No	Entity Name	Quantity
1	Cloudlets	100-1000
2	Length of the Tasks	500000 to 200000000
3	Datacenters	10
4	Capacity of VM	4096MB
5	Hypervisor	Xen
6	No of VMs	100
7	No of Hosts	500
8	Processing elements Capacity	174/247/355 MIPS
9	Capacity of the Physical Host RAM	32GB
10	No of Processing Elements to taks	1

7.1 Calculation of Makespan

The primary objective of any task scheduler in cloud computing is to minimize the makespan. It is defined as the total time taken to execute a task. The effectiveness and quality of a task scheduler depends on the makespan. So here in this simulation we have considered 1000 cloudlets generated from the cloudsim workload and ran the simulation.From the below table 3 we can observe that the proposed algorithm using CSA is surpassed the ACO, PSO,CS algorithms interms of makespan.

Table 3: Caclulation of makespan

No of Tasks	ACO	PSO	CS	CSA
100	1587.5	1345.7	1326.5	1012.6

500	1996.9	1878.9	1756.8	1232.5
1000	3164.8	2563.5	2146.9	1567.8

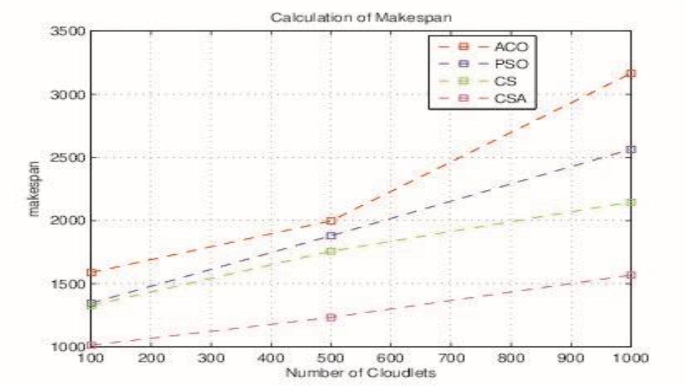

Fig. 2: Calculation of makespan

7.2 Calculation of Power Cost at Datacenters

Power Cost at datacenters is also one of the important parameter in the view of Cloud Provider. For running a datacenter, huge amount of energy is to be required there by consumption of huge power can be done at the datacenters. Power Cost is considered as operational costs for the cloud provider. If we can schedule the tasks appropriately based on the processing capacity and size of the tasks to an appropriate virtual machine then we can reduce the energy consumption thereby we can also reduce the power cost at the datacenters. Generally power cost is measured at the datacenters interms of the energy consumption per hour. In this simulation we have considered 120 hours for calculating the power cost at the datacenters. we have chosen the number of datacenters as 10 as mentioned in the table for simulation settings. We have compared our proposed approach with the existing algorithms ACO, PSO and CS. From the table 4 we can observe that the proposed approach which uses CSA algorithm is showing signficant improvement over the existing approaches.

Table 4: Caclulation of Power Cost at Datacenters

No of Tasks	ACO	PSO	CS	CSA
24	2502	3543	3345	1867
48	2998	2456	3765	2256
72	2765	3567	4245	2278
96	4567	4124	4376	3245
120	5657	4879	5208	4126

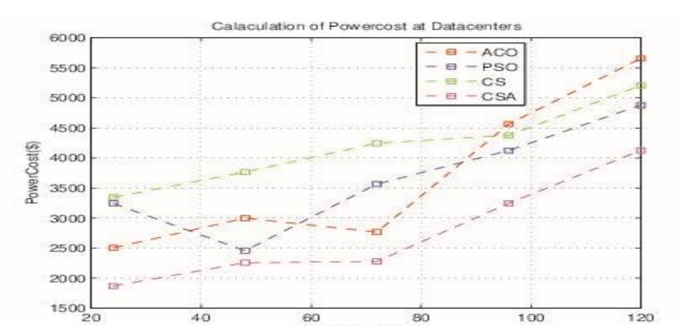

Fig. 3: Calculation of Power Cost at datacenters

8. Conclusion and Futurework

Task scheduling is a huge challenge in cloud computing. In the existing algorithms authors focused on the mapping of tasks onto the virtual machines by considering metrics like makespan and the resource utilization. In this paper we have proposed new task scheduling approach which analyzes the task for calculating the priorities of the tasks and the priorities of VMs based on the electrcity cost at the different datacenters. After calculating these priorities the scheduler need to map the tasks to the Virtual machines optimally by minimizing the makespan and the powercost at the Datcenters. This algorithm is modeled by using a nature inspired algorithm known as crow search and it is implemented in the Cloudsim simulator and we have compared our approach with the existing algorithms ACO, PSO and CS algorithms. From the results, it shows that our proposed approach is outpeformed interms of makespan and the powercost at the datacenters. In future, we want to check the effciency of our algorithm by deploying it in the openstack cloud environment.

References

1. Huang, Xingwang, et al. "Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies." *Cluster Computing* (2019): 1-11.
2. Singh, Krishnapal, et al. "Optimization of task scheduling using genetic algorithm in federated cloud environment." *International Journal of Computer Engineering and Sciences Research* 1 (2018): 24-30.
3. Krishnadoss, Pradeep, and Prem Jacob. "OCSA: task scheduling algorithm in cloud computing environment." *International Journal of Intelligent Engineering and Systems* 11.3 (2018): 271-279.
4. Panda, Sanjaya Kumar, Shradha Surachita Nanda, and Sourav Kumar Bhoi. "A pair-based task scheduling algorithm for cloud computing environment." *Journal of King Saud University-Computer and Information Sciences* (2018).
5. Jana, Bappaditya, Moumita Chakraborty, and Tamoghna Mandal. ”A task scheduling technique based on particle swarm optimization algorithm in cloud environment.” Soft Computing: Theories and Applications. Springer, Singapore, 2019. 525-536.
6. Choe, SongIl, et al. "Improved Hybrid Symbiotic Organism Search Task-Scheduling Algorithm for Cloud Computing." *KSII Transactions on Internet and Information Systems (TIIS)* 12.8 (2018): 3516-3541.
7. Moon, YoungJu, et al. "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments." *Human-centric Computing and Information Sciences* 7.1 (2017): 28.
8. Panda, Sanjaya K., and Prasanta K. Jana. "Efficient task scheduling algorithms for heterogeneous multi-cloud environment." *The Journal of Supercomputing* 71.4 (2015): 1505-1533.
9. Shao, Youwei. "Research on cloud task scheduling algorithm of particle swarm optimization algorithm based on fission mechanism." *International Journal of Multimedia and Ubiquitous Engineering* 11.8 (2016): 145-152.
10. Sun, Zheng. "Multi-Task Scheduling Algorithm Based on Genetic Algorithm in Cloud Computing Environment." (2016).
11. Kumar, AM Senthil, and M. Venkatesan. "Task scheduling in a cloud computing environment using HGPSO algorithm." *Cluster Computing* 22.1 (2019): 2179-2185.

12. Aziza, Hatem, and Saoussen Krichen. "Bi-objective decision support system for task-scheduling based on genetic algorithm in cloud computing." *Computing* 100.2 (2018): 65-91.

13. Sreenu, Karnam, and M. Sreelatha. "W-Scheduler: whale optimization for task scheduling in cloud computing." *Cluster Computing* (2019): 1-12.

14. Zhao, Shasha, et al. "Research on cloud computing task scheduling based on improved particle swarm optimization." *International Journal of Performability Engineering* 13.7 (2017): 1063-1069.

15. Kumar, KR Prasanna, and K. Kousalya. "Amelioration of task scheduling in cloud computing using crow search algorithm." *Neural Computing and Applications* 32.10 (2020): 5901-5907.

16. Aslam, Shahzad, et al. "Towards Energy Efficiency and Power Trading Exploiting Renewable Energy in Cloud Data Centers." *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*. IEEE, 2020.

17. Askarzadeh, Alireza. "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm." *Computers & Structures* 169 (2016): 1-12.

18. Devi, D. Chitra, and V. Rhymend Uthariaraj. "Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks." *The scientific world journal* 2016 (2016).

19. Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and experience* 41.1 (2011): 23-50.