Survey on Dynamic Resource Allocation Scheduler in Cloud Computing

Ms. Pooja Rathod
Computer Engineering, GTU

ABSTRACT

Cloud Computing is one of the area in the various fields related to computer science can be exercised for on demand dynamic resource allocation for providing reliable and guaranteed services to the consumer. Dynamic resource allocation should be done in such manner that it should not waste a resources and also less energy consumption. The scheduling dynamic resources for maximizing resource utilization, genetic algorithm concept with multiobjective is much faster than traditional other algorithms for resource predication and allocation. This paper provides detailed description of the dynamic resource allocation scheduling methods for cloud computing and comparative study provides the clear detail about the different techniques

Keyword: *Dynamic Resource Allocation, MOO- multiobjective optimization, GA – genetic algorithm. VM- virtual machine, PM- physical machine.*

1. INTRODUCTION

1.1 cloud computing

Cloud computing provides computing facilities such as CPU, memory, RAM, over a network without being physically present at consumer's location. Cloud computing offers computing capabilities as a service. It contains three types of services like IaaS (infrastructure as a service), PaaS (platform as service), and SaaS (software as a service).And all these services are being provided on the four types of cloud like public cloud, private cloud, hybrid cloud and community cloud.

1.2 Dynamic Resource Allocation scheduling

In cloud computing different cloud users request assortment of administrations according to their progressively evolving needs. So it is the job of cloud computing to benefit all the requested services to the cloud consumers. From the cloud suppliers' point of view cloud resources must be assigned in a reasonable way. It is additionally attractive to avoid wasting resources as a result of under-utilization and to avoid lengthy response time because of over-use. There are certain issue with respect to the dynamic resource assignment, for example, Resource Provisioning, Job Scheduling, Resource overbooking, Scalability, Load adjusting, Pricing, Availability, Overheads in Network I/O Workloads, Quality of Service (QoS). Dynamic resource allocation scheduler deals with the resource in such way to get most extreme usage of all resources with overseeing QoS and limiting the energy utilization.

2. NEED FOR DYNAMIC RESOURCE ALLOCATION SCHEDULING

Maximum utilization of available resources

Dynamic resource allocate scheduling is mainly required for the maximum utilization of all the available virtual machines and physical machine for particular requested service. So it also decreases the resource wastage.

Faster response for requesting service.

Cloud computing has lots of users nowadays so there will be much traffic in service request, as all the resource will execute as per proper scheduling it will give faster response to that service.

Energy efficiency.

Improving average utilization of resources makes the less energy consumption in virtual machine placement. Reduce cost of energy also reduces the overall cost of cloud computing environment

3. EXISTING METHODS FOR DYNAMIC RESOURCE ALLOCATION SCHEDULER

In this paper we have analyze some of the dynamic resource allocation scheduling techniques. And they are describe in following sections.

3.1 Dynamic Resource Prediction and Allocation for Cloud Data Centre Using the Multiobjective Genetic Algorithm.[1]

In this system it works with MOO formula, proposed GA and VM placement algorithm.
Where,

MOO: Formulate maximizing both CPU and memory of each active PM and minimizing the energy consumption of data canter.

The MOO problem of resource allocation in data center is defined as follows:

$$G(x) = \begin{cases} g1(x) = g_{cpu}(x) = \max C^{avg} \\ g2(x) = g_{memory}(x) = \max M^{avg} \\ g3(x) = g_{energy}(x) = \min E \end{cases}$$

Proposed GA

The proposed GA accurately predicts the CPU and memory utilization in next time slot, No matter the utilization tendency is stable raise and fall tendency or unstable fluctuation tendency.

The fitness function of a chromosome is designed to fit in with the *survival of the fittest*. It represents the deviation between prediction and reality, which is designed as follows:

$$F_c(t) = \alpha * f_{cpu}(t) + \beta * f_{mem}(t) + \gamma * f_{eng}(t)$$

where

$$f_{cpu}(t) = |C^{avg}(t') - C^{avg}(t)|$$
$$f_{mem}(t) = |M^{avg}(t') - M^{avg}(t)|$$
$$f_{eng}(t) = \frac{|E(t') - E(t)|}{E^{max}} * 100\%$$
$$\alpha + \beta + \gamma = 1.$$

The flowchart of the proposed GA for resource prediction is shown below:

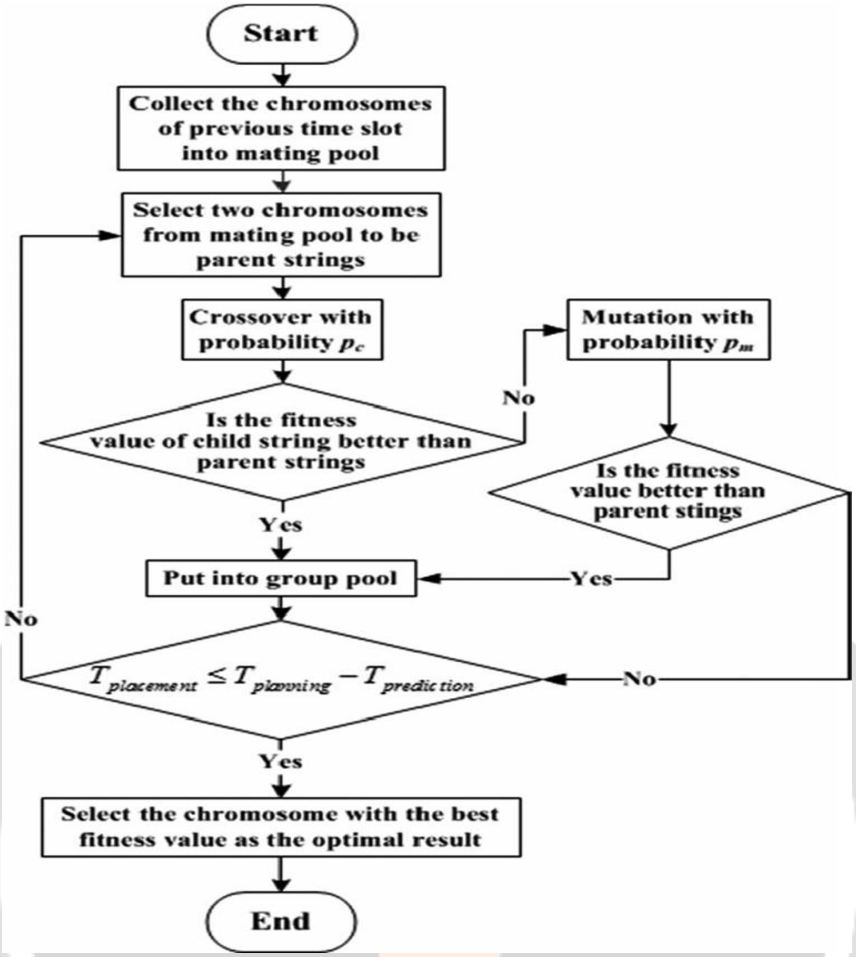

Fig 1: flowchart for proposed GA

VM placement algorithm:

The proposed VM placement algorithm reallocates VMs for the next time slot based on the forecast result of the proposed GA.

By using the VM placement algorithm, the resource utilization of each active PM is maximized, and the number of active PMs is minimized.

Algorithm 1: VM Placement Algorithm.

Input: V, P, C^{max}, M^{max}, E^{max}, S^{max}

01 $\Omega 1 = v_m$, $\Omega 2 = p_n$, $\Delta = \phi$
02 while $\Omega 2 \neq \phi$
03 repeat
04 $y = f(j) = \max_{j \in \{1...n\}} [(C^{max} - p_j^{cpu}) + (M^{max} - p_j^{mem})]$
05 $y^* = \arg\max f(j)$
06 $S = f_d(\Omega 2, p_j, y^*)$
07 $\Delta = \Delta \cup S$ [1]
08 if $(p_{S[n]}^{cpu} < C^{max})$ and $(p_{S[n]}^{mem} < M^{max})$
09 $v_{i,j} = v_{i,n}$
10 $p_{S[1]} = \phi$
11 end if

12 until $v_{i,j} = \Delta$
13 end while

The procedure of the proposed VM placement algorithm is shown in Algorithm 1. In lines 4–7, the difference in CPU and memory utilization between the maximum value of data center and PM pj is sorted in a descending order list S. For example, the CPU and memory utilization of PM p1 are 20% and 30%, and PM p2 are 50% and 60%. The PM p1 is ahead of PM p2 in the descending order list S. The PM with minimum CPU and memory utilization is put in the first order. Besides, let notation S[i] be the ith order in list. If two PMs have the same degree, the orders of them in the ordered list are arbitrary. In lines 8–11, VMs in the PM with minimum utilization are migrated to the PM in the last order of list S. Herein, the resource utilization of PM $pS_{[n]}$ should not exceed the maximum CPU utilization C^{max} and memory utilization $M^{max,}$ which is guaranteed in line 8. If the resource utilization of PM exceeds resource limitation, the remaining VMs will be migrated to a PM with the second-last order of list S. Once VMs in a PM are fully migrated to other PMs, the PM is shut down for decreasing the energy consumption of data center. According to the prediction result of GA, the VM placement algorithm aims to maximize the average of CPU and memory utilization and minimize the total energy consumption of data center with the least active PMs. The placement algorithm is executed at the end of each time slot after prediction. It should be finished and accomplished before next time slot.

3.2 Dynamic Resource Scheduling using Delay Time Algorithm in Cloud Environment. [3]

In this algorithm they implement Delay time algorithm which is having its efficient significance in scheduling and allocating the resources effectively

Here they are going to derive proposed idea in implementing Delay time algorithm which would make the users with efficient utilization of resources. Basic concept of the existing randomized scheduling algorithm is that, to calculate the inter arrival times of various jobs request. When scheduling the job request, jobs with the highest scheduled workload would be scheduled first. If multiple job requests have same density of the workload, random pick of the same density job request is proceeded the processing. This leads to a major issue that the job request with the minimal workload would face delay in processing. So they introduce Delay time algorithm by which it analyze, prioritize and schedule the job requests based on priority of time constraint deadline. This would help in efficiently allocating resources for the users without much delay in processing the request.

Thus this algorithm helps in increasing efficiency in utilization of resource without much delay in processing request.

3.3 Preemptible Priority Based Dynamic Resource Allocation in Cloud Computing with Fault Tolerance. [5]

In this paper they propose an algorithm which perform resource pre-emption from low priority task to high priority task and advanced reservation for resources considering multiple SLA parameters for deploying service. This algorithm is also effective for fault tolerance mechanism**.**

In this paper they focus on dynamic resource provisioning, they present a scheduling heuristic considering multiple SLA objectives, such as amount required CPU, network bandwidth, and cost for deploying applications in clouds. The scheduling present a flexible on demand resource allocation strategy included advanced reservation and pre-emptily mechanism for resources. This proposed algorithm dynamically responds to requested resource for the task. First it's locally checks for the availability of resource; if resource is free then it deployed new VMs to current task, If resource is not available then it's create new VM from globally available resource; if global resources are not available then it will check for resource if it's preemptable then it's migrate processes otherwise put the task into waiting list and apply advanced reservation scheme.

3.3.1 SCHEDULING STRATEGY

This proposed algorithm is generally based on SLA based resource provisioning and online adaptive scheduling for advanced preempt-able task execution. Two basic steps are required for effective utilization of cloud resources that meet the SLA objectives.

A Resource allocation and deployment of application

B. Scheduling heuristic description

When a scheduler receives a user's service request, it will first partition that task request into several subtasks and form a DAG. This process is called as static resource allocation. In [I] authors proposed greedy algorithms to generate static planning of allocation: CLS (cloud list scheduling algorithm).

Algorithm 1 shows a function forming a task list based on the priorities.

Algorithm 1 forming a task list based on priorities

Require (input): A DAG, Average execution time
AT of every task in the DAG
Ensure (output): A list of task P based on priorities
1. The EST is calculated for every task
2. The LST is calculated for every task
3. The T_p and B_p of every task are calculated
4. Empty list P and stack S, and pull all task in the list of task U
5. Push the CN task into stack S in decreasing order of their LST
6. While the stack S is not empty do
7. If top(S) has un-stacked immediate predecessors then
8. S ← the immediate predecessor with least LST
9. Else
10. P ← top(S)
11. Pop top(S)
12. End if
13. End while

3.3.2 SCHEDULING ALGORITHM

In this section propose an algorithm to handle the priority request from the customer, and provide the advanced reservation and preemption over the resources, it is a modified version of previous algorithm in Here the highest priority of task defines over AR task and task related to highest cost paying by the customers.

Algorithm 2: Priority based scheduling algorithm

In Priority based scheduling algorithm R is customer's service request, A is application data, S is SLA based terms and condition. These provided as input to the scheduler (step 1). When the request for a service send to cloud scheduler then scheduler divides it in many sub tasks as per their dependencies, for this purpose algorithm 1 is called. Algorithm 1 is also used to form a list of tasks based on their dependencies (step 2). In (step 3) scheduler get

global available VM list, and the entire resources list, that is available for deployment user's services from the each cloud. Used VM list is also provided to add deployed VM information. In (step 4-5) it uses the SLA terms to fmd a list of appropriate VMs that is capable to provisioning the request service R. In (step 6) scheduler get all the local information and lists, once it have all information then load balancer locally decides which VM is allocate to service request (step 7). When there are no VM and requested resource is available at that time then scheduler globally checks for free resources if any resource is free globally then it deploy service on the resource by creating a new instance of VM (step 8). Else if there are no extra resources available locally and globally then it checks for task priority , if task is high cost price task or it has advanced reservation then scheduler runs algorithm (3) (step 9). In (step 10) scheduler updates their list if any changes occurs, during the resource allocation. In any other cases provision request is add to queue of waiting list until the VM with desire resource is get free (step II). If after a certain period of time the service request is schedule and deployed then scheduler returns as successful deployment status otherwise it returns failure (step 14-19) to admin.

Algorithm 3 Advanced reservations and preemption based cloud min-min algorithm

A cloud scheduler records execution schedule of all resources using a slot. A mappable task set is assigned to the algorithm, A mappable task set is a set whose predecessors task are allocated to VM and corresponding resource .If there is a high priority task (high cost) then it should be given advanced reservation according to given algorithm. If a task did not found appropriate resource in free condition and scheduler has a high priority task then this algorithm works in three steps. I) First it will check for earliest resource available time from all the clouds then it's check current requested resource is preemptable or not , if yes then switching perform between the tasks on same resource using a time quantum. So the deployment status of current task is true and all the succeeding tasks can be successfully deployed on the VM. 2) If the requested resource is not preemptable and an AR task is assigned to a cloud, first resource availability in this cloud will be checked by cloud scheduler. Since best-effort task can be preempted by AR task, the only case when most of resources are reserved by some other AR task. Hence there are not enough resources left for this AR task in the required time slot. If the AR task is not rejected, which means there are enough resources available for the task, a set of required VMs are selected arbitrarily. 3) If all the tasks running on cloud are AR task then our algorithm gives advanced reservation on the resource of earliest finish task. In preemptable priority if resource is preemptable then it just checks for nearest resource and assigns it to task then perform switching between tasks. When a task completes then it remove them from circular queue and return the result. In advanced reservation, if a resource is non-preemptable then scheduler just sends task checks request to all other cloud provider and receive the earliest available time of corresponding resource and then, the manager server of this cloud will first check the resource availability in this cloud. Since AR tasks can pre-empt best effort tasks, if the resources are reserved by some other AR tasks at the required time, then, AR task will capture the resource by advanced reservation when resource get free.

3.4 Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue.[10]

In this technique they propose an algorithm which considered Preemptable task execution and multiple SLA parameters such as memory, network bandwidth, and required CPU time.

In this, they present dynamic resource allocation mechanism for Preemptable jobs in cloud. They propose priority based algorithm, in which considering multiple SLA objectives of job, for dynamic resource allocation to AR job by pre-empting best-effort job.

They are describing SLA based resource provisioning and online adaptive scheduling for Preemptable task execution, these two methodologies which are combined in proposed algorithm for effective utilization of cloud resources to meet the SLA objective.

- Cloud Resource provisioning and scheduling heuristic.
  Provisioning can be done at the single layers alone. However, approach which we considered in aims to provide an integrated resource provisioning strategy. Thus, scheduling heuristics in considers the three layers. An aim of scheduling heuristic in is to schedule job on VMs based on the agreed SLA objective and creating new VMs on physical resources based on availabilities resources. This strategy helps to optimized application performance and at the same time reduces the possibilities of SLA violations. And, the integrated load-balancer in the heuristic ensures high and efficient resource utilization in the Cloud environment.

- Preemptable task execution
  When a scheduler receives customer's service request, it will first partition that service request into tasks in the form of a DAG. Then initially static resource allocation is done. In authors proposed two greedy algorithms, to generate the static allocation: the cloud list scheduling (CLS) and the cloud min-min scheduling
  (CMMS).
- Scheduling Algorithm
  In proposed priority based scheduling algorithm we have modified the scheduling heuristic in for executing highest priority task with advance reservation by preempting best-effort task as done in CMMS algorithm.

3.5 Improving Grouping Genetic Algorithm for Virtual Machine Placement in Cloud Data Centers. [8]

In the current work, they improve this algorithm by introducing a unique and efficient method for encoding and generating new solutions. Using vector packing problem, they model the problem of virtual machine placement and try to reduce power consumption by minimizing the number of used servers and also maximizing resource usage efficiency. The algorithm is tested over varying VM placement scenarios which show encouraging results.

- **Efficiency of resource usage**

This criterion reflects how well the resources of different types are utilized. The goal is to fully
Utilize utilize the resource in all dimensions. It is important to mention that an overly high utilized server might be in danger of poor performance. To be specific, response time of a server which is an important performance indicator, increase by utilization. Therefore, in accordance to maximum accepted response time which defined in SLA, the maximum allowed utilization for servers should be computed. To prevent CPU usage of a server from reaching 100%, the capacity of CPU resource is lessened by parameter G which is a controlling parameter defined by data center managers. The main idea behind this is that 100% utilization can cause severe performance degradation and VM live migration technology consumes some amount of CPU processing capability on the migrating node.

Power consumption

Recent studies show that server power consumption scales linearly with CPU utilization. We defined the power consumption of the i-th server as a function of the CPU utilization.

Scalarizing multi-objective optimization problem

Aiming to maximize resource usage efficiency and minimizing the server's power consumption simultaneously, one can consider VM placement as a multi objective problem. However, by scalarizing the optimization formulation, we reduced the complexity of the problem to that of a single objective one. The placement problem can therefore be formulated as:

Minimize

$$\sum_{j=1}^{m} P_j - K * \sum_{j=1}^{m} \text{UsageEFF}_j$$

In their work, they only consider two dimensions namely CPU and memory to characterize VMs and servers. If two VMs are assigned to the same server, the CPU/Memory utilization of the sever (host) is estimated as the sum of CPU/Memory utilization accounted for, by the two VMs.

IMPROVED GROUPING GENETIC ALGORITHM (IGGA) The grouping genetic algorithm (GGA) proposed is a version of genetic algorithm heavily modified to suit the structure of grouping problems. Those are the problems where the aim is to find a good partition of a set or group together the members of the set. As we stated earlier the VM placement problem usually model as bin packing problem which is categorized to grouping problem. Falkenauer in [7] clearly demonstrates that classical genetic algorithm performs poorly on such problems. So he enhanced the algorithm through new encoding and special genetic operators.
This work proposes an improved version of grouping genetic algorithm (IGGA) for solving the VM placement problem in cloud data centers. After modelling the problem as a vector packing problem, IGGA attempts to find a placement solution with minimum power consumption and maximum resource efficiency

4. CONCLUSIONS

Cloud computing is emerging as a new paradigm of large scale distributed computing. It has moved computing and data away from desktop and portable PCs, into large data centers. Resource allocation is one of the critical issues in cloud computing. Costing for the Cloud Computing is highly dependent over the Utilization of Resources. Dynamic Resource Utilization scheduler is a technique that is used to increase the utilization of the resources and decrease the resource wastage as well as it also manages the energy consumption

In this paper, several methods for dynamic resource scheduling surveyed on. Genetic algorithm for VM placement, Delay Time Algorithm and Priority Based Dynamic Resource Allocation etc. These method makes better utilization of available resources and minimize the energy consumption

5. ACKNOWLEDGEMENT

I would like to thank my parents for their unconditional support

6. REFERENCES

[1] Fan-Hsun Tseng, Xiaofei Wang, Li-Der Chou, Han-Chieh Chao and Victor C. M. Leung "Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic Algorithm" 2017 IEEE.

[2] H.GoudarziandM.Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in Proc. IEEE Int. Conf. Cloud Comput., Jun. 2012, pp. 750–757.

[3] Ajay Thomas S, Ms. santhiya C "Dynamic Resource Scheduling using Delay Time Algorithm in Cloud Environment" 2017 Second International Conference On Computing and Communications Technologies(ICCCT'17).

[4] Mashayekhy, an Online Mechanism for Resource Allocation and Pricing in Clouds, IEEE Transactions on Computers, 12 June 2015.

[5] Shubhakankshi Goutam, Arun Kumar Yadav, "Preemptable Priority Based Dynamic Resource Allocation in Cloud Computing with Fault Tolerance" 2015 IEEE International Conference on Communication Networks (ICCN).

[6] Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen, Zhong Ming, "Adaptive Resource Allocation for Preempt able Jobs in Cloud Systems," in 10th International Conference on Intelligent System Design and Application, Jan. 2011, pp. 31-36.

[7] E. Falkenauer, "A hybrid Grouping Genetic for Bin packing Algorithm", Journal of Heuristics, 2, 5-30, 2014

[8] Shahram Jamali, Sepideh Malektajiv, "Improving Grouping Genetic Algorithm for Virtual Machine Placement in Cloud Data Centers" 2014 IEEE.

[9] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Lio, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing", Journal of Computer and System Sciences, 2013.

[10] Chandrashekhar S. Pawar, Rajnikant B. Wagh, "Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue" 2013 IEEE