Figure 1:

# Effective allocation of Resources and Task Scheduling in cloud Environment using social Group Optimization

S. Phani Praveen .K.Thirupati Rao ¿B.JanaKiramaiah

June 2023

[

Abstract Effective resource distribution to regulate load uniformly in heterogeneous cloud environments is crucial. Resource allotment which is taken after capable task scheduling is a critical worry in cloud environment. The incoming job requests are assigned to resources equally by load balancer in such a way that resources are utilized effectively. Number of cloud clients is very great in number, degree of approaching job requests is uninformed and information is tremendous in cloud application. Resources in cloud environment are constrained. Hence, it is not easy to deploy different applications with unpredictable limits and functionalities in heterogeneous cloud environment. The proposed method has two phases such as allocation of resources and scheduling of tasks. Effective resource allocation is proposed using social group optimization algorithm and scheduling of tasks using shortest-job-first scheduling algorithm for minimizing the makespan time and maximizing throughput. Experimentations are performed for accurate simulations on artificial data for heterogeneous cloud environment. Experimental results are compared with first-in, first-out and genetic algorithmbased shortest-job-first scheduling. Validity of the proposed method noticeably gives improved performance of the system in provisions of makespan time and throughput. Keywords Load balancer · Cloud platform · Heterogeneous cloud · SGO algorithm

B S. Phani Praveen
phanisurapaneni.pvp@gmail.com

K. Thirupathi Rao
profktrao@gmail.com

B. Janakiramaiah
bjanakiramaiah@gmail.com

1 Department of Computer Science, Bharathiar University, Coimbatore, India

2 Department of Computer Science and Engineering, KL University, Guntur, India

3 Department of Computer Science

and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

## 1 Introduction

Cloud computing implementation has expanded essentially among organizations of all limits and individual clients. It has surfaced as a standout among the most vital processing method of the time. It is a utility administration where various servers are associated with Web. Clients need to rent or lease resources according to the use on hourly premise. Resources of cloud incorporate processors, software, storage, hardware, and so forth. Nowadays, every online client is making use of the cloud either directly or the other way round, in part or altogether for their own or business purposes. Resources are plenty, like the Web, for the client to get access to cloud. Business undertakings of different scale can utilize cloud computing without stressing on owning, refreshing, support, copyright, enrolling, and so forth. It has been viewed as the same stride as the utility computing after grid computing. Estimation of pricing technique is "pay as you go" on hourly premise. In cloud computing, extremely measurable resources are delivered as a utility using Web to various external clients [1]. A promising IT expansion, operation, and delivery model permit authentic delivery of resources and services over the web [2]. A resource provision method is proposed which integrates a common organizing rule for delivery of resource components, and a structural process as well as a cost-effective method—mainly a union of utility and grid computing. Cloud features are virtualization, elasticity, and scalability. Any device with IP address will combine to form a network and connects with Web, which makes users more virtual and remotely available. With a rapid improvement in technology researchers across world have granted environmental impact as a major standard for new resources. Hence, resource optimization holds high significance. Optimization of resources is essential measure for the performance of cloud. Due to the quick expansion in cloud adoption, various capacities of data center are rapidly increasing globally. The basic cause for higher server interruption is due to inappropriate mapping of load to resources [3]. Huge amount of heat is produced due to overfilling of physical machines. Energy utilization in cooling overfilling data center contains enormous amount of power consumption in cloud. But idle server consumes as huge as 50a completely used server. Nowadays, server expansion is an adverse condition of data center. Either the servers are overfilled or they are idle. Easiest ways to reduce energy misuse in cloud are optimal load balancing and effective task scheduling. This approach involves two autonomous but interconnected phases. (i) Optimal allocation of resources to tasks. (ii) Effective scheduling of allocated tasks. These two steps are autonomous to each other, but collectively impact efficiency of the cloud. Energy utilization in cooling overfilling data center contains enormous amount of power consumption in cloud [4]. To create an equilibrium between incoming job requests and server processing competence, load-balancing techniques are required. Load balancing includes two phases: (i) allocation of resources and (ii) scheduling of tasks. SGO resource allocation is able to balance the incoming requests in heterogeneous clouds. It is observed that various tasks are allocated to a single VM. So scheduling of tasks is the next vital step in order to get the improved throughput and minimum makespan time. Firstly, shortest job is used for scheduling to get high throughput and minimum makespan time. The following issues have been discussed in the present work: (i) allocation of resources using SGO and scheduling of tasks using SJF; (ii) influence of load balancer and scheduler on cloud nodes, with an objective of reducing makespan time and improving the throughput. To attain improved performance over all the nodes in cloud platform, optimal resource allocation and effective task scheduling for all incoming job requests are necessity. Load balancing mainly focuses on available resources that are evenly busy and refrain from overfilling of resources [5]. Load balancing increases computability and decreases carbon emission [6]. The key objectives of load balancing are best utilization of resources, increasing throughput, and minimum response time, and avoiding overfilled computing units. Performance of system is decreased due to misuse of resources and server breakdown in cloud. Overfilled servers lead to server break down and power consumption in data centers. Once workload is registered with cloud provider, job requests are divided into various tasks. Load balancing is allocating tasks to resources in a fair way such that server is neither overfilled nor idle. The main aim of the paper is to achieve reduction of makespan time and improvisation of throughput. The case studies used in this work address multicloud physical machines. Physical systems are divided into many virtual machines. Every application is classified into several jobs and these jobs are allocated to resource. Scheduling of tasks has high significance that enhances the efficiency of the complete cloud platform. Task scheduling decides the flow of execution of tasks by VMs. Hence, the two phases are scheduling and load balancing. These two phases are on two separate layers of abstraction. Basically, allocation of resources is nonrepresentational to load balancer and scheduler. It involves allotment of existing requests to VMs in a best possible way to improve throughput. Several jobs are allotted to a particular virtual machine. Hence, allotment of resources is followed by scheduling to improve efficiency of the system. In most of the cases, the prioritized task execution is important for the system performance. Therefore, effective load balancing among incoming resources and task scheduling becomes a challenging issue to be addressed, e.g., Google follows MapReduce scheduling [7], Facebook utilizes Fair Share Scheduler [8], Yahoo uses computability-based scheduler, and Amazon uses FIFO. All schedulers share advantages and drawbacks in overall delivery of technology. Scheduling in cloud with severe limitations is a NP complete problem. SGO is a population-based algorithm having the concept of social behavior of human as to approach solving a complex problems [9]. Effective allocation of resources, followed by scheduling of tasks is one of the primary worries in heterogeneous cloud platforms. In the present work, SGO-based load balancer is used for mapping the tasks to optimal resources and SJF scheduling is used to schedule execution of assigned tasks. In cloud platform, many varied servers are connected through Web. In virtualized system, various physical devices rifts many virtual VMs. VMs are the smallest unit of CPU, which deploy deliverables. VM management in its optimal way improves the efficiency of the system. Virtualization is used to hold numerous applications or users at the same period of time. Normally, a job request is divided into many smaller but autonomous tasks. Balanced allocation of resources reduces job makespan time. Fitness scheduling approach is addressed in the work, which deploys balancing of load among number of nodes, minimizing makespan and increasing throughput. The remaining paper is structured as follows: Sect. 2 shows the literature review and applications in a heterogeneous cloud system. Section 3 includes the resource allocation task mapping model. Section 4 describes proposed algorithm for minimum makespan mapping, and Sect. 5 discusses about performance metrics and experimental evaluation. The conclusion is given in Sect. 6.

## 2 Literature Review

Several techniques on scheduling of tasks have been projected for a variety of computing domains like distributed and parallel processing [10–17]. Authors used PTC (Predicted Time to Compute) is used to represent total time of executed tasks on VM. Estimated value depends on the features like size of information, server size, cpu speed, and bandwidth [18] and [19]. Scientific researchers used GA balancing load and cloud analyst as simulating instrument. They have chosen single time period for their experiments, which is impractical. Authors had compared AGA with job spanning and JLGA [20]. Scientific researchers have proposed LBACO, simulated using CloudSim, and compared the result with FCFS. Some have integrated GA with knapsack problem with a variety of fitness to find best way of mapping tasks to resources [21]. Authors used MapReduce service composition to address large scale composition problems [5]. Authors have used ball and bin method to balance the load [10]. Researchers have proposed a LBS algorithm having two classes in cloud environment: In the first class, OLB is practiced, and in second LBMM is used to assign task to proper service node. Several meta-heuristic methods are applied to task scheduling to achieve optimal solution in allocation of resources [22,23]. Scientific researchers compared the load balancing techniques of HFB, BRS, and AC and used ACCNT in cloud. Maguluri et al. [24] have similarly compared two balancing of loads and scheduling of tasks in cloud clusters, but arrival rate of jobs defined in simulation range from 0.1 to 1, and job requests allocated are non-preemptive. Stochastic hill climbing is used for allocation of resources to VMs and is analyzed using cloud analyst. Authors have implemented GA load balancing and compared with FCFS [10,18,25]. Many Authors have proposed online scheduling algorithms called CLS and CMMS in IaaS. These scheduling divides tasks into two types, namely reserved and unreserved modes used in applications. Drawback of CLS is that it chooses the minimum execution time of cloud, irrespective of cloud ready time, which results in load imbalance. Tejaswi et al. [26] proposed a solution for job scheduling using GA-based technique. Performance is evaluated in terms of makespan time [27]. Researchers have analyzed uneven scaling in cloud environment. First, they have identified the VMs which are overfilled and needed to be migrated. Identified hot spot VMs which are emitting heat energy are being overfilled. By minimizing the hot spots and migrating VMs green cloud computing is achieved [19]. Three task scheduling algorithms for multi-cloud were suggested, which reduces makespan time and increases average cloud usage. But the major issue is that cloud can execute only one task assigned to it at a given point of time, when unreserved tasks are preempted on arrival of reserved tasks, and then, unreserved task will be executed on the same cloud which could lead to imbalanced workload.

## 3 Resource Allocation Task Mapping Model

Figure 1 represents the architecture of the cloud model supporting SGO-based mapping of tasks to balance the load evenly across all VMs followed by SJF scheduling. The major components of this architecture are as follows:

1.**Clients :**Clients can record their application and implement that application throughout the world by using a cloud agent in the heterogeneous cloud.

2.**Cloud Agent :**Agent acts as a medium between group of service providers and clients. To implement application, agent needs resources from several service providers. Service-level agreement needed to be signed, and task type should be defined like reserved tasks or unreserved tasks at the registration phase.

3.**SGO Load Balancer :**Each and every application is categorized into many tasks. Every task is classified into two categories: (i) reserved tasks (ii) unreserved tasks, based on the existing resources in cloud system and predicted makespan time, and tasks are assigned to resources by using SGO operators.

4.**Shortest-Job-First Scheduler:**Since jobs are many with altering capacities, implementation of the tasks is scheduled on the basis of SJF as an alternative of conventional FIFO when the tasks are allotted to resources.

Allocation of resources in cloud environment is a three layer model. First layer consists of interaction of client and cloud agent. Second layer consists of optimal mapping of job requests and accessible resources. Allotment of tasks into suitable VMs in balanced way is implemented in second layer. Third layer involves optimal execution of tasks keeping throughput at very high and minimum makespan. Cloud agent registers all incoming requests with distinct ID. Depending on the utilization of resource types and time period of resource usage of service is calculated. Significance of each resource is measured iteratively in little quantum of interval, which is vital for better efficiency of the system. Task mapping to resources is done using SGO. SGO is one of the fast and best heuristic approach when search space is huge [9]. Batch processing is used to keep higher utilization of server and waiting time is reduced in queue. Latency and waiting time are key criteria to choose cloud provider. SGO mapping scheme is compared with FIFO and GA-based SJF, whereas efficiency is compared with throughput and average makespan time

Let j is total number of job requests. Job request j(i) splits into multiple tasks t(i). Total number of tasks are represented using Eq. (1):

Total tasks(i) = J i=1 task(i).

Number of tasks combined in a group are mapped to resources in well balanced manner. Poison distribution is used for incoming job requests

MakespanTime(t) =k j=1 TaskTime(i, j)

where r represents the ith task in job assigned position, TaskTime (i, j) denotes time taken for task execution i on jth node of cloud p. The average time of execution of a job is given by the formula:
AvgjobTime = j t=1 MakespanTime(t) J .
Length of job request is autonomous of temporal and spatial job request. Throughput and AvgTimeofjob are major metrics for efficiency of the system. Although, in some cases, it is not possible to exempt node having remote data transmission. Task mapping to cloud node is represented into m*2 matrix, where m denotes task and second row represents the cloud to which task is alloted. Workload can be calculated by adding workloads of all VMs. Where physical device divides into k VMs. The load(L) of a node can be measured by adding the loads of all VMs running on it.