A Hybrid Task Scheduling Approach Based on Genetic Algorithm and Particle Swarm Optimization Technique in Cloud Environment

Bappaditya Jana and Jayanta Poray

Abstract As per current trends, cloud services are becoming more popular day by day. Because these services satisfy several demands for heterogeneous resources without using dedicated IT infrastructure. In cloud platform, user gets shared pool of resources through Internet irrespective of any geographical location. But cloud services need to handle a gigantic amount of request, as the number of users increases exponentially. In order to manage a pool of requests till now no effective scheduling mechanism is available in practice. So to minimize the time delay and optimize the overall complexity, suitable scheduling methodology is very much required. In our study, we have presented a novel scheme for scheduling algorithm after the enhancement of genetic algorithm and particle swarm optimization technique. We have proposed a methodology that can provide a better response time from cloud provider and minimize the waiting time for particular clients in cloud environment.

Keywords Cloud computing · Genetic algorithm · Max–min scheduling
Min execution time · Scheduling algorithm · Particle swarm optimization
Hybrid task scheduling algorithm · Time hybrid task scheduling

1 Introduction

In recent advancement of information and communication technology, research on cloud computing becomes one of the most latest trends. Cloud computing appears as a fastest evolutionary virtual platform which promises for a good cost-effective environment including business model, education, personal intelligence, and many

B. Jana (✉) · J. Poray
Department of Computer Science & Engineering, Techno India University,
Kolkata, West Bengal, India
e-mail: bappaditya.j.in@ieee.org

J. Poray
e-mail: jayanta.poray@gmail.com

V. Bhateja et al. (eds.), *Intelligent Engineering Informatics*, Advances in Intelligent
Systems and Computing 695, https://doi.org/10.1007/978-981-10-7566-7_61

other diverse application for global users according to their need without investing software, hardware, or any other computing or networks infrastructure [1, 2]. The main purpose of this service to the user is to provide effective service with minimal cost [3]. As lots of advantages are coming from cloud services in our daily life, it is manually very difficult to assign or schedule a particular resource for a specific task [4]. So scheduling of task is one major challenging issue in cloud computing. Various methodologies have been proposed on this area. At present, some existing task scheduling approach using genetic algorithm and some cases modified form of genetic algorithm have been used in [5, 6, 7]. Some other existing algorithm such as FIFO scheduling, fair scheduling, and computing capacity scheduling is still facing some restrictions [8]. In cloud platform, redundant array of independence disks techniques is used to balance cost of the service and to maintain assurance of cloud data [9]. We have tried to propose a technique for task scheduling algorithm based on genetic algorithm (GA) and particle swarm optimization (PSO) technique and to optimize the existing issues in cloud platform. In our hybrid task scheduling algorithm, we have compared execution time with max–min scheduling execution and minimum scheduling time for a particular number of task and we have considered two times where execution time is less than other two cases. In our technique, we have shown that for a particular task, response time is comparatively better and waiting time is minimum than other two-mentioned scheduling approaches.

2 Overview on Cloud Services

Cloud system has two major components, one is cloud data distributor another is cloud service providers. In first cases, cloud collect data files from users or clients then splits into small clusters with respect to identical features and after splitting these are distributed according to the on-demand or requests of users. Memory distribution in cloud server is discussed in [10]. The distributed approach is maintained by redundant array of independent disk (RAID) technique. On the other hand, major goal of service providers is to store cluster of data in response according to the various requests showing availability of effective data and remove cluster of data as per requirement. Cloud service is a on-demand self-service. This service is generally divided into three categories, [11] such as Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), and Infrastructure-as-a-Service (IaaS). SaaS is on-demand software service and business model predicted by pay–per-use basis. In Saas, user can use data bases, hardware and software maintenance without any personal expenses. Again software is updated automatically without installing new software because in this service applications are hosted centrally. PaaS is one type of cloud computing in which cloud providers use an environment for developing applications and to provide various services through Internet. In Iaas, cloud provides an ideal computing platform to the client. Clients get virtual operating system and program execution environment, data bases, and Web server. Due to the

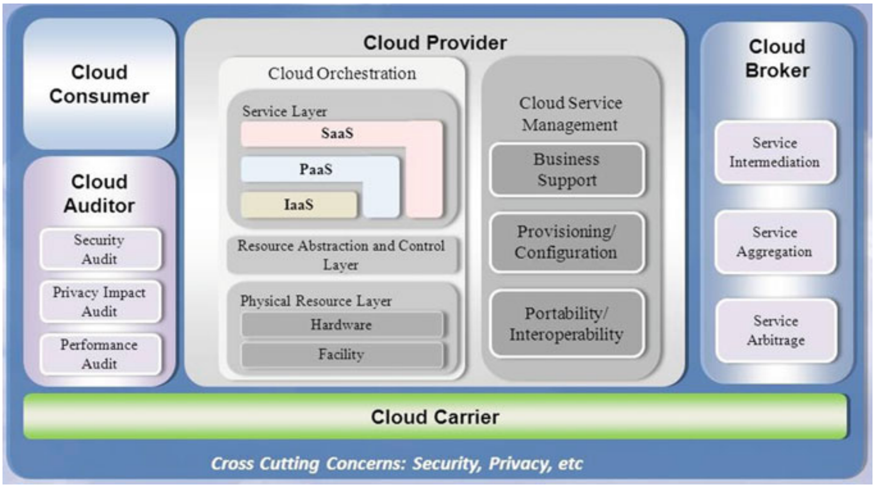

Fig. 1 Schematic diagram of cloud services

availability of central virtual architecture in cloud, many real-time operations are performed in cloud. The schematic diagram of overall cloud services is shown in Fig. 1, [12].

Clouds are classified into three types [13] such as public, private, and hybrid. Cloud behaves as virtual machine, and virtualizations of cloud are briefly described in [14].

3 Proposed Methodology

Genetic algorithm is based on biological evolution in population and concerned with Darwin's theory of evolution. Darwin proposed "Survival of the fittest" which implements method of scheduling in genetic algorithm [15, 16, 17]. In cloud computing, multiple resources have to be assigned according to the multiple requirements of jobs. So which resource will get preference for which task or job is a complex situation process. That's why absence of proper scheduling process, the time complexity will be high. Genetic algorithm is one type of evolutionary computing. Genetic algorithm was basically implemented on a set of chromosomes in ecological environment. At first, one group of chromosomes is considered as initial data set and then after generation, a new set known as offspring's is produced from parent set. This set of chromosomes is considered as a set of solutions in cloud computing and the set of solutions is known as population also. During generations, one population produces a newly updated population according to the expectation. A fitness function is used to find better solutions. Some genetic operator such as mutations, crossover, etc., is used repeatedly to get most optimized and simplest solution.

Algorithm 1: Genetic Algorithm

John Holland proposed genetic algorithm in 1962 which is a random search-based evolutionary method in biological environment [18]. Genetic algorithm includes six basics steps as **Initialization:** Generate population randomly from n number of solutions. **Use of fitness function:** f(x) is the fitness function of particular solution in existing population. **Selection:** According to the fitness value, two separate solutions are selected from the given population. These two individuals are considered as a parent for next coming generation. **Crossover:** Now two parents are coming in crossover based on probability and produce new offsprings. **Mutation:** Mutation of new child takes place at any position whenever the existing population tends to be consistent. **Accept and Replace:** After based on probabilistic mutation, the new solutions are generated and again replace new population as the present generation.

Algorithm 2: Particle swarm optimization technique

Kennelly and Eberhat proposed the concept of particle swarm optimization technique in 1995. It is one type of evolutionary search technique for a population in ecological environment where a particular set of random solution of existing population is considered as a particle. Every particle flies randomly with a velocity throughout whole search spaces and using a fit function they randomly fit according to swarm's historical behavior. This step is going repeatedly until getting better solutions.

4 Hybrid Approach

4.1 Evolutionary Algorithms as Optimizers

```
begin
solution representation
c: : = 0 ; // generation counter
initialize f(c)
evaluate f(c);
while(not terminate)
do
$f_1$(c) :  selection(f(c));
$f_2$(c) :  variation($f_1$(c));
Evaluate $f_2$(c);
f(c+1) : = survivor (f(c). $f_2$(c);
c : = c+1;
od
end
```

4.2 Traditional Evolutionary Approach in Binary-Coded Genetic Algorithms

4.2.1 Genetic Algorithms

- create a can for minimum cost having at most volume V.
- Objective function cost (d,h) : f(d,h) = Πdh + 2Πd2/4
- Constraint : Volume(d,h) : Πd2 h /4 ≥ V
- Representation in binary strings
- Fitness : Objective + penalty for constraint

4.2.2 PSO Approach

- Let us consider all particles fly around the whole search space and velocity of each particle are not static, they are updated dynamically.
- $x_i = x_i + v_i$ where x_i is the location and vi is the velocity
- $V_i = v_i + c_1$ rnd() (P_i, best − x_i) + c2 rnd() ($P_g - x_i$) Where Pi is the location with best condition for i-th particles and P_g is the actual location of the particle ,selected as best and so far and c1,c2 in (0.2)
- According to the history, the momentum part is 1st term.
- The 2nd term is represented as thinking with private condition which is also known as cognitive part.
- The another remaining term is related with social components which means a collaboration.
- Inertia weight is Vi where $V_i = w V_i + \ldots$ (w in [0,4,0,9])
- Velocity restriction , V_i = min(Vmax,vi) and it prevents unbounded growth in velocity.
- Constricted PSO: $V_i = X(V_i + \ldots\ldots)$ (X = 0.7289) and it give better convergence.
- Bare bone PSO : Dropping V_i term
- Offspring are normally distributed around P_g and P_0
- Communication topologies :
- P_g is now replaced by I_g

Repeat the steps until a stopping criterion is meet to get a sufficient good fitness value.

Table 1 Average schedule length

Iteration number	Number of task	Max–Min scheduling time	Minimum execution time	Hybrid task scheduling algorithm
1	200	660	654	631
2	400	1350	1324	1286
3	600	2020	1996	1926
4	800	2772	2764	2692
5	1000	3446	3436	3376
6	1200	4164	4156	4116

5 Experimental Results

In our study, we have used a software for simulation known as cloudsim [18]. In [19] max–min algorithm maintains the task status that includes the updated time, execution time, and completion time to improve the performance. Our proposed algorithm is designed in cloudsim layer. Each service request from cloud user is called as task and which needs to be recorded by VMs. Each task has to be processed in VMs according to the processing capability of VMs. VMs have some memory and storage and controlled by millions of instructions. Associated band of number index entries found. Simulation number of task varied from 200 to 1200 with iterative increments by 200.

The data components give length of average scheduling and successful execution ratio which is shown in Table 1. According to Table 1, it is cleared that our presented methodology gives comparatively lower execution time than max–min scheduling time and minimum execution time. The hybrid-scheduling algorithm has decreased the scheduling length by 4.65% than max–min scheduling and by 4.15% than minimum execution time for 600 numbers of tasks which is shown in Fig. 2.

Similarly in Table 2, it is observed that our proposed methodology gives comparatively lower execution time than max–min scheduling time and minimum execution time. The hybrid-scheduling algorithm has decreased the scheduling length by 4.39% of max–min scheduling. On other hand by 5.61% than Minimum execution time for 400 numbers of task which is shown in Fig. 3.

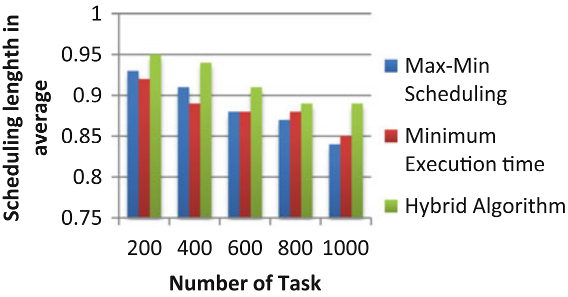

Fig. 2 Average schedule length

Table 2 Ratio of successful execution

Iteration number	Number of task	Max–Min scheduling time	Minimum execution time	Hybrid task scheduling algorithm
1	200	0.93	0.92	0.95
2	400	0.91	0.89	0.94
3	600	0.88	0.88	0.91
4	800	0.87	0.88	0.89
5	1000	0.84	0.85	0.89
6	1200	0.80	0.82	0.87

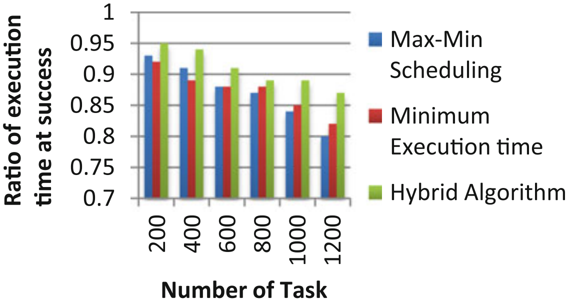

Fig. 3 Ratio of successful execution

6 Conclusions

We have presented a cost-effective hybrid task scheduling algorithm based on principle of genetic algorithm and coupled with particle swarm optimization technique which can provide a better response time from cloud provider and minimize the waiting time for a particular client in cloud computing environment. In future, we will focus on model implementation of dynamic task allocation in account of data center load factor, minimization of electric cost, and internal communications.

References

1. Zhu, K., Song, H., Liu, L., Gao, J., Cheng, G.: Hybrid genetic algorithm for cloud computing applications. In: 2011 IEEE Asia-Pacific Services Computing Conference (APSCC), pp. 182–187. IEEE (2011)
2. Paul, M., Sanyal, G.: Survey and analysis of optimal scheduling strategies in cloud environment. In: Proceedings of the IEEE International Conference on Information and Communication Technologies, Georgia, USA, pp. 789–792 (2012)

3. Lv, Z., Halawani, A., Feng, S., Li, H., Rehman, S.U.: Multimodal hand and foot gesture interaction for handheld devices. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **11**(1s), Article 10, 19 pages (2014)
4. Tangang, Zhan, R., Shibo, Xindi: Comparatively analysis and simulation of load balancing scheduling algorithm based on cloud resource. J. Springer (2014)
5. Yin, H., Wu, H., Zhou, J.: An Improved Genetic Algorithm with Limited Iteration for Grid Scheduling. In: IEEE Sixth International Conference on Grid and Cooperative Comput ing, 2007. GCC 2007, Los Alamitos, pp. 221–227 (2007)
6. Guo, G., Ting-Iei, H., Shuai, G.: Genetic simulated annealing algorithm for task scheduling based on cloud computing environment. In: IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 60–63 (2010)
7. Wang, J., Duan, Q., Jiang, Y., Zhu, X.: A new algorithm for grid independent task schedule genetic simulated annealing. In: IEEE World Automat ion Congress (WAC), 2010, Kobe, pp. 165–171 (2010)
8. Randles, M., et al.: Biased random walks on resource network graphs for load balancing. J. Springer (2009)
9. Qiyi, H., Tinglei, H.: An optimistic job scheduling strategy based on QoS for cloud computing. In: Proceedings of the IEEE International Conference on Intelligent Computing and Integrated Systems, Guilin, China, pp. 673–675 (2010)
10. Wei, Z., Pierre, G., Chi, C.: CloudTPS: scalable transactions for web applications in the cloud. IEEE Trans. Serv. Comput. **5**(4), 525–539 (2012)
11. Dean, J., Ghemawats, S.: MapReduce simplified data processing on large clusters[C]. In: Proceedings of the 6th Symposium on Operating System Design and Implementation, pp. 137–150. ACM, New York (2004)
12. Cryptographic Key Management Issues & Challenges in Cloud Services Ramaswamy Chandramouli Michaela Iorga Santosh Chokhani, NISTIR 7956, http://dx.doi.org/10.6028/NIST.IR.7956
13. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
14. Beloglazov, A., Buyya, R.: Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Trans. Parallel Distrib. Syst. **24**, 1366–1379 (2013)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1992)
16. Goldberg, E.: The existential pleasures of genetic algorithms. In: Winter, G. (ed.) Genetic Algorithms in Engineering and Computer Science, pp. 23–31. Wiley, New York (1995)
17. Yusoh, M., Izzah, Z., Maolin, T.: Clustering composite SaaS components in cloud computing using a grouping genetic algorithm. In: IEEE Congress on Evolutionary Computation, pp. 1–8 (2012)
18. Buyya, R., Ranjan, R., Calheiros, N.: Modeling and Simulation of scalable cloud computing environments and the cloudsim toolkit: challenges and opportunities. In: Proceedings of the 7th High Performance Computing and Simulation Conference, Leipzig, Germany, pp. 1–11 (2009)
19. Mao, Y., Chen, X., Li, X.: Max-Min task scheduling algorithm for load balancing in cloud computing. J. Springer (2014)