Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparison of Particle Swarm Optimization and Genetic Algorithm for Load Balancing in Cloud Computing Environment

Er. Komal, Er. Arvinder Kaur
Computer Science & Engineering,
Chandigarh University,
Chandigarh, India

Abstract— Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IAAS), Platform-as-a-Service (PAAS) and Software-as-a-Service (SAAS). Particle Swarm Optimization and Genetic Algorithm is used in this work to generate the Make Span Value. A Particle Swarm Optimization algorithm has been presented to find the better solution for the problem of scheduling and load balancing in Cloud Computing environment. The goal of the technique is to resolve the problem of high consumption of system time while scheduling the incoming jobs according to available virtual machines in Cloud environment resulting into a balanced load distribution.

Keywords—Cloud Computing, Particle Swarm Optimization, Job Scheduling, Load Balancing, Genetic Algorithm.

I. INTRODUCTION

Cloud computing is not another thought in specialized world but rather it is a forthcoming innovation. Grid Computing, Utility Computing and dispersed frameworks have direct association with the Cloud Computing. It can be expressed that framework processing goes about as the spine to Cloud Computing. Cloud computing gives virtual assets and administrations with the goal of diminishing expense. Cloud computing is actualized and famous generally because of its properties of giving virtualization and reflection [1] [2].

In operating system, various strategies have been projected for the purpose of job scheduling. The algorithms or strategies proposed for job scheduling are SJF (Shortest Job First), FIFO (First in First Out), LIFO (Last in First Out), Priority Based, Greedy Algorithm. The basic aim of these algorithms is minimization of total execution time of all jobs. These algorithms are easily understandable and can be implemented [3]. In Cloud environment, there is no restriction on number of jobs at a time requesting for scheduling which becomes the issue of efficiency of existing operating system scheduling algorithms. These algorithms may produce unwanted results in Cloud Computing environment and thus are not feasible to be implemented.

To make use of existing algorithms, there is a need of optimization of these algorithms to generate better results. One more issue in job scheduling and load balancing is that the response time for various task is very high and load on the processor become a threaten of failure of processor. This leads to a need of the algorithm which can optimize the load balancing process [4]. In this paper, both the Genetic Algorithm [5] and Particle Swarm Optimization approach [6] has been implemented for scheduling and load balancing and a comparison is drawn on the basis of defined parameters to find the better approach for scheduling in Cloud computing environment.

The paper's organization is as follows:

- Section 2 describes the overview of the literature.
- Section 3 describes the proposed algorithm for optimization.
- Section 4 describes the result and analysis.
- Section 5 covers the conclusion and future work.

II. LITERATURE SURVEY

Radojevic and Zagar [4] proposed another calculation for burden adjusting called as CLBDM (Central Load Balancing Decision Module). The plan was proposed with the purpose of correspondence with all parts of PC structure, including workload balancers and application servers. CLBDM has effect sending choices on the heap balancers taking into account the collected data and inner estimations. It has determined that the execution of the composed structure can depend basically on isolating up work successfully over the taking an interest hubs in a circulated system of registering frameworks.

Ebehart and Kennedy [6] introduced particle swarm optimizer in an innovative structure. It has mentioned that genetic algorithm was much similar to particle swarm optimization. Likely genetic algorithm, particle swarm optimization also begun with the generation of population but unlike GA, it can assume particles and can initiate them with some random position and velocity and allows them to move freely in search space. It has implemented the technique over various applications and concluded that PSO has given better performance than other techniques. For example, due to crossover operator in GA, the immigration among subspecies of robots can be a serious issue; this issue should not be present with particle swarms.

AV. Karthick et al. [7] proposed a multi-queue scheduling scheme which can increase the client satisfaction and utilize energy of the system. Scheduling was the most important complex part in Cloud Computing and thus the major goal of global scheduler was to share the resources at most the maximum level. Researchers have given more importance to build a job scheduling algorithms that were well-suited and appropriate in Cloud Computing situation. Also in Cloud Computing, the user has to pay for services based on usage time that's why Job scheduling was one of the critical event in Cloud Computing.

Swachil J. Patel et al. [8] has raised an issue of priority in job scheduling because some job requests have need of being scheduled first then all other remaining jobs which can adjust with longer waiting time. With this aim, author has presented an improvement in job scheduling scheme based on the priority in Cloud Computing. Also this algorithm need to be further improved for the minimization of make span.

Daji Ergu et al. [9] proposed task-oriented resource allocation model in Cloud Computing environment. Due to different computers with varying capacities in Cloud environment, allocation of resources becomes complex and difficult. The proposed scheme has formed a pair matrix of tasks and comparison was made on the basis of network bandwidth, cost of task, its reliability and its completion time. The main motive was to improve the consistency ratio when allocation was based on weights of tasks.

Tingting Wang et al. [10] has discussed that the load balancing issue was critical in Cloud scenario due to huge number of users and large data volume. Thus the requests of resource sharing and reuse were becoming more and more imperative. With the purpose of an efficient task scheduling strategy author has implemented load balancing using genetic algorithms so as to fulfill the user necessities and get better resource utilization. This strategy not only resulted into task scheduling sequence with shorter job and average job makespan, but has also satisfied the inter-nodes load balancing. But this strategy has assumed that there was no priority among jobs. However, in real Cloud Computing environment, it is not avoidable.

Ayed Salman et al. [11] has exhibited another undertaking task calculation that was in light of the standards of PSO in the circulated or parallel registering frameworks. PSO has taken after a populace based inquiry, which performs as indicated by the easygoing conduct of winged animals and fishes. PSO framework has joined neighborhood seek techniques with worldwide hunt strategies. Through self experience and neighboring knowledge, it endeavors to adjust investigation and abuse. Every individual component of the populace was called as a molecule that flies around in a multidimensional pursuit space in the hunt of the best arrangement. Particles may overhaul their position as per their own particular and their neighboring particles position, sending toward their best position or their neighbor's best position.

III. PROPOSED ALGORITHM

The research work begins with the design of the shortest job first algorithm for Cloud Computing. The methodology followed, starts its journey from exploring the insights of Cloud environment and its various advantages and research challenges. The design has been implemented in a simulated environment. The proposed work is divided into two parts:

- Implementation of Genetic Algorithm technique
- Implementation of Particle Swarm Optimization technique.

(a) Genetic Algorithm

Genetic algorithm is a part of evolutionary computing methods and is taking into account the characteristic choice procedure. This methodology can be used to solve various problems in cloud environment as scheduling algorithms such as shortest job first (SJF) and round robin (RR) cannot be applied to cloud environment as they may create undesirable results and limitless time. It is an algorithm based on mechanism of natural selection and inheritance theory. It has various following steps of implementation [10]:

Step1: Generation of fixed number of random chromosomes as population.
Step2: Calculation of fitness value of all chromosomes.
Step3: Selection of best chromosomes as parents using following rules:
Rule SL1: From P(g), calculate fitness function value of every solution and select two best solutions, denote as {p1, p2} for cross over operation.
Rule SL2: For mutation operation, select a random solution P3 from P(g).
Step4: Performance of crossover operator to generate new offspring from the parents.
Step5: Performance of mutation operator at each position.
Step6: Addition of offspring to the original population by replacing worst
Rule RU1: If a possible solution with low fitness value as compared to new solution is found, then replaces that solution by the new solution. It is performed to guarantee that next generation contains better solutions than current generation. If this is not applicable then apply the rule in RU2.
Rule RU2: A possible solution having equal fitness as new solution must be replaced by new one.
Step7: If stopping criteria is met, then end the process and take best chromosome as optimized solution.

Algorithm1: Genetic Algorithm_Scheduling Process
Input: Job Schedules Output: Best Job Schedule Step 1: Take population size = 30, generation g=1

Step 2: Initial population Generation Initialize → P(g) Step 3: For each solution, calculate the Fitness value, x€P(g). Step 4: Selection of parent solutions P1, P2 from the P(g), By applying Rule SL1 & SL2 Step 5: Generate new solution OS1, OS2 by applying operation of crossover: CROSSOVER (P1, P2) with Pc as crossover probability. Step 6: Generate new solution OS3 by applying operation of mutation: MUTATION (P3) with Pm as mutation probability. Step 7: Remove some worst solutions from current population P(g) By using rule RU1, RU2 to add newly generated solution Population: P(g+1). Step 8: Check for criteria by applying Rule TC1 and TC2, if stopping situation has arrived, display result else GOTO Step 3.

(b)Particle Swarm Optimization

PSO adopts extremely basic methodology that appears to be useful for optimizing an extensive variety of functions [11]. It has association with the hypothesis of ECT. It appears to lie between genetic algorithm and evolutionary programming. It is a dependent of estimation processes and is proposed by Kennedy and Eberhart [6]. It was firstly implemented for involvement of collective behavior of bird flock or fish school as representation of the movement of components or particles in simulation. Particle Swarm Optimization (PSO) has recently come into sight as a well-known heuristic approach, which can be applied to various large and complex problems, like knowledge extraction in data mining, task scheduling problem, electric power systems, etc.

The step by step description of particle swarm optimization algorithm is given below to have clear understanding of its working.

- Initialize the swarm or the solution space
- Calculate the fitness of each particle
- Update individual and global best values
- Update velocity and position of each particle
- Go to step2, and repeat until termination condition

Particle Updating: The following equation gives a brief description of PSO update process:

Vpi+1 = ωvpi + c1rand1 × (pbestp – xpi) + c2rand2 × (gbest – xpi) (3.1)

Xpi+1 = xpi + vpi+1 (3.2)

where:

vpi at iteration i, is the velocity of particle p

vpi+1 at iteration i + 1, is the velocity of particle p

ω is inertia weight

ca is coefficient of acceleration; a = 1, 2

randr is random number from 0 to 1; r = 1, 2

xpi at iteration i, is p particle's current position

pbestp is the best position of particle p

gbest in a population, is position of best particle

xpi+1 at iteration i + 1, is p particle's position

The steps to be followed while implementing the PSO algorithm have been listed in Algorithm 3. The first step is to perform the value assignment process of particle's position and velocity by method of random generation. Assuming that jobs are independent and preemption is not allowed.

Fitness Function: Fitness function in evolutionary techniques plays a vital role. It is an objective function based on which the best survival will be selected. Shortest Job First algorithm has a basic principle to find the job with shortest completion time. It searches the whole system and found the one with least job completion time and executes it firstly. In the same manner, all jobs are processed.

Algorithm2: Proposed_PSO_Scheduling Process
Input: jobs, resources, schedules Output: best execution schedule according to fitness function Begin Generate a swarm with p particles having m*n dimensions 1. Initialize all parameters 2. Initialize particles positions randomly and velocity of each particle vi randomly.

3. repeat
For i =1 to iterations
Compute fitness using fitness evaluation function
If fitness of a schedule is greater than previous, update it to new best solution
If f(Xi) > f(pbesti) , then
Pbesti = Xi
end
Now compute gbest that is globally best from all the pbest schedules
If f (pbesti) > f (gbesti), then
gbesti = pbesti
end
Velocity updation using equation (3.1)
Position updation using equation (3.2)
end for loop until stopping criteria is met

According to the algorithm, each solution is now updating the velocity parameter based on the evaluations of the personal best and the global best. Each solution or schedule keeps observing its own best position and the group keeps saving the global best position.

IV. RESULTS AND ANALYSIS

The calculated results from the load balancing and job scheduling in proposed techniques of genetic algorithm and particle swarm optimization are taken under four scenarios. In all the four scenarios both the number of the virtual machines and jobs are fixed. All the scenarios use shortest job first mechanism for job scheduling. System Response Time for any technique can be the total time consumed by the system for executing the algorithm. The SRT that is system response time for both the techniques is also calculated for all the job sets.

Evaluation through System Response Time and Load:

Table 4.1 Overall system response time with 30machines

Number of Jobs	**Virtual Machines**	**System Response Time (micro-sec)**	
		GA	**PSO**
30	30	6.7548	3.4788
50	30	5.5224	3.6036
100	30	6.7704	4.3992
200	30	10.4833	6.2868
500	30	29.5934	13.0471

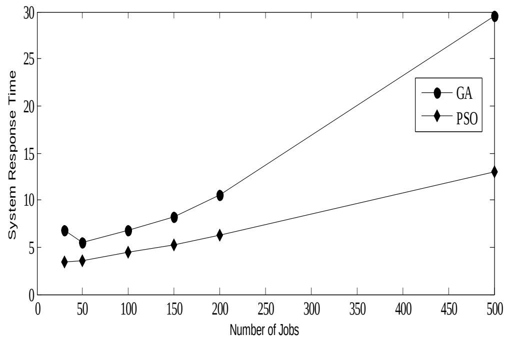

Figure 4.1: Comparison of System Response Time

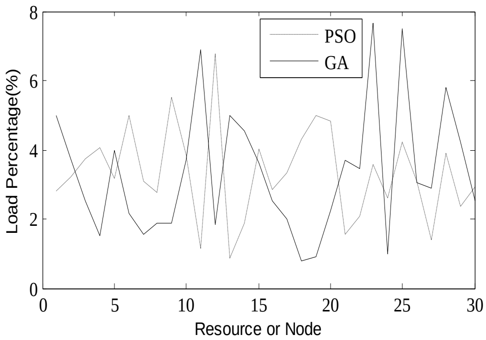

Figure 4.2: Load Calculated

Table 4.2 Overall system response time with 50machines

Number of Jobs	**Virtual Machines**	**System Response Time (micro-sec)**	
		GA	**PSO**
50	50	5.772	4.1184
100	50	6.8796	4.3992
150	50	9.0481	5.3352
200	50	11.4661	6.1932
500	50	33.509	13.2913

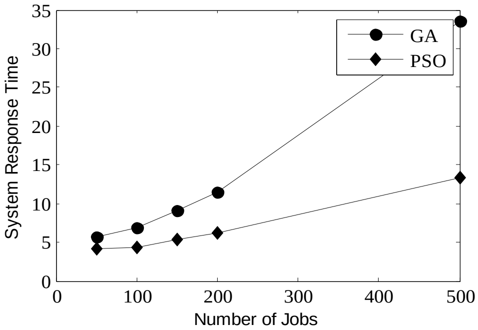

Figure 4.3: Comparison of System Response Time

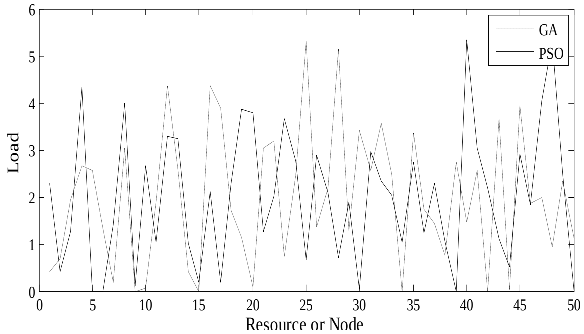

Figure 4.4: Load Calculated

After testing the algorithm techniques for all the scenarios it can be evaluated that system response time in genetic algorithm increases rapidly with increase in number of jobs whereas in case of particle swarm optimization technique, the increment in time is slow and less even for maximum.
Evaluation through Makespan Generated:

Table 4.3: Total Makespan for different Iterations

Number of Iterations	**Total Makespan(sec)**	
	GA	**PSO**
20	2163	1700
40	2658	1928
60	2451	1934
80	3244	1929
100	2726	1923

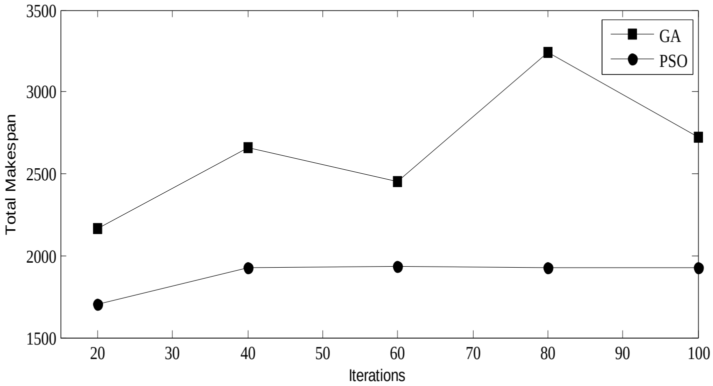

Figure 4.5: Comparison of Total makespan

These results and discussions over the difference in the system response time by Genetic algorithms and particle swarm optimization gives the clear picture that the particle swarm optimization takes much lesser time than genetic algorithms. There are reasons that support the more efficiency in particle swarm optimization is Easy to Execute, Fewer Mathematical operators, Less Complex than GA.

V. CONCLUSION

Cloud Computing is internet based computing in which resources are provided to users on demand. A Particle Swarm Optimization algorithm has been presented to find the better solution for the problem of scheduling and load balancing in Cloud Computing environment. The goal of the technique is to resolve the problem of high consumption of system time while scheduling the incoming jobs according to available virtual machines in Cloud environment. In this study evolutionary techniques that are genetic algorithm and particle swarm optimization, are implemented in Cloud Computing using shortest job first scheduling methodology. On the basis of results it has been evaluated that PSO is better than GA if system response time is considered as it takes lesser response time and and generate a low makespan than Genetic Algorithm approach.

REFERENCES

[1] Dinh, Hoang T., et al. "A survey of mobile cloud computing: architecture, applications, and approaches." Wireless communications and mobile computing 13.18 (2013): 1587-1611.
[2] Zhang, Shufen, Hongcan Yan, and Xuebin Chen. "Research on key technologies of cloud computing." Physics Procedia 33 (2012): 1791-1797.

[3] Silberschatz, Abraham, Peter Baer Galvin, and Greg Gagne “Operating System Principles” John Wiley & Sons, 2006.

[4] Radojevic, Branko, and Mario Zagar. "Analysis of issues with load balancing algorithms in hosted (cloud) environments." MIPRO, 2011 Proceedings of the 34th International Convention. IEEE, 2011.

[5] Gu, Jianhua, et al. "A new resource scheduling strategy based on genetic algorithm in cloud computing environment." Journal of Computers 7.1 (2012): 42-52

[6] Eberhart, Russ C., and James Kennedy. "A new optimizer using particle swarm theory" Proceedings of the sixth international symposium on micro machine and human science. Vol. 1. 1995.

[7] AV.Karthick “An Efficient Multi Queue Job Scheduling for Cloud Computing”, International conf. on Computing and Communication Technologies (WCCCT), March 2014, 164 – 166.

[8] Patel, Swachil J., and Upendra R. Bhoi. "Improved Priority Based Job Scheduling Algorithm in Cloud Computing Using Iterative Method" Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on IEEE, 2014.

[9] Ergu, Daji, et al. "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment" The Journal of Supercomputing 64.3 (2013): 835-848.

[10] Wang, Tingting, et al. "Load Balancing Task Scheduling Based on Genetic Algorithm in Cloud Computing" Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on. IEEE, 2014.

[11] Salman, Ayed, Imtiaz Ahmad, and Sabah Al-Madani. "Particle swarm optimization for task assignment problem." Microprocessors and Microsystems 26.8 (2002): 363-371.

[12] Pandey, Suraj, et al. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." Advanced Information Networking and Applications (AINA), 24th IEEE International Conference on. IEEE, 2010.

[13] Abedifar, Vahid, et al. "An optimized virtual network mapping using PSO in cloud computing." Electrical Engineering (ICEE), 21st Iranian Conference on. IEEE, 2013.

[14] Qing, Wang, and Zheng Han-chao. "Optimization of task allocation and knowledge workers scheduling based-on particle swarm optimization." Electric Information and Control Engineering (ICEICE), International Conference on. IEEE, 2011

[15] Zhang, Lei, et al. "A task scheduling algorithm based on PSO for grid computing." International Journal of Computational Intelligence Research 4.1 (2008): 37-43.

