Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406
International Journal of Advance Engineering and Research Development
A National Conference On Spectrum Of Opportunities In Science & Engineering Technology
Volume 5, Special Issue 06, April-2018 (UGC Approved)

STUDY OF MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR TASK SCHEDULING IN CLOUD COMPUTING

Miss. Pooja Kathalkar

Abstract:- . *Even though there is much advancement in the era of computers and Internet of Things (IoT) with respect to responsiveness, reliability and flexibility, still there is a room for improvement in scheduling, optimal resource allocation and management algorithms since these algorithms come under NP-hard and NP-complete complexity classes. Hence, there is a need to address these set of challenging problems using different techniques. In this paper, we mainly focus on scheduling the tasks using Modified Particle Swarm optimization algorithm. This algorithm reduced execution time and average response time with efficient utilization of cloud resource.*

Keywords:- *Task Scheduling, MPSO, Genetic Algorithm.*

INTRODUCTION

Nowadays, many clients demand for several resources (CPU, Memory etc.) on pay-as-you-go model and satisfying these types of tasks is very tedious and needs proficiency in managing these tasks from IaaS platform. Since, the efficiency of the cloud should not be hindered, hence throughput of the cloud should be directly proportional to the efficient utilization of cloud resources. Special focus is needed when these tasks change their demand for resources which are uncertain and handling the resources demand would be a great challenge in the cloud environment. Swarm Intelligence (Kennedy and Eberhart, 2001[7]) is a recent and emerging paradigm in bio inspired computing for implementing adaptive systems. In this sense, it is an extension of EC. While EAs are based on genetic adaptation of organisms SI is based on collective social behavior of organisms. As per definitions in literature, Swarm Intelligent encompasses the implementation of collective intelligence of groups of simple agents that are based on the behavior of real world insect swarms, as a problem solving tool. The word ―swarm comes from the irregular movements of the particles in the problem space.SI has been developed alongside with EAs. Some most well-known strategies in this area are discussed below. These trajectory tracking algorithms being inspired by the collective behavior of animals, exhibit decentralized, self-organized patterns in the foraging process. Swarm Intelligence Principles: SI can be described by considering five fundamental principles.

PSO has been found to be robust and is successfully applied in solving nonlinear, non differentiable multi-modal problems quickly. It is still in its infancy. Many research works have mentioned application of PSO in task scheduling. PSO is most successful meta-heuristic for generations of optimal scheduling solutions. PSO scans over solution space during each iteration and accumulates global best and local best solutions.[10]

1) Proximity Principle: the population should be able to carry out simple space and time computations. 2) Quality Principle: the population should be able to respond to quality factors in the environment.

3) Diverse Response Principle: the population should not commit its activity along excessively narrow channels.

4) Stability Principle: the population should not change its mode of behavior every time the environment changes.

5) Adaptability Principle: the population should be able to change its behavior mode when it is worth the computational price Particle Swarm Optimization Particle swarm optimization (PSO) is a computational intelligence oriented, stochastic, population-based global optimization technique proposed by Kennedy and Eberhart in 1995[8].

It is inspired by the social behavior of bird flocking searching for food. PSO has been extensively applied to many engineering optimization areas due to its unique searching mechanism, simple concept, computational efficiency, and easy implementation. In PSO, the term ―particles refers to population members which are mass-less and volume-less (or with an arbitrarily small mass or volume) and are subject to velocities and accelerations towards a better mode of behavior. Each particle in the swarm represents a solution in a high-dimensional space with four vectors, its current position, best position found so far, the best position found by its neighborhood so far and its velocity and adjusts its position in the search space based on the best position reached by itself (pbest) and on the best position reached by its neighborhood (gbest) during the search process.

Steps in PSO algorithm can be briefed as below:

1) Initialize the swarm by assigning a random position in the problem space to each particle.

2) Evaluate the fitness function for each particle.

3) For each individual particle, compare the particle‘s fitness value with its pbest . If the current value is better than the pbest value, then set this value as the pbest and the current particle‘s position, xi, as pi.

4) Identify the particle that has the best fitness value. The value of its fitness function is identified as guest and its position as pg.
5) Update the velocities and positions of all the particles using (1) and (2).
6) Repeat steps 2–5 until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value)

Advantages over Genetic Algorithm:

(a) PSO is easier to implement and there are fewer parameters to adjust.
(b) PSO has a more effective memory capability than GA.
(c) PSO is more efficient in maintaining the diversity of the swarm, since all the particles use the information related to the most successful particle in order to improve themselves, whereas in Genetic algorithm, the worse solutions are discarded and only the new ones are saved; i.e. in GA the population evolve around a subset of the best individuals. There are many similarities between the PSO and EAs. Both of them initialize solutions and update generations, while the PSO has no evolution operators as does the latter. In a PSO, particles try to reach the optimum by following the current global optimum instead of using evolutionary operators, such as mutation and crossover. It is claimed that the PSO, in addition to continuous functions, has been showing stability and convergence in a multidimensional complex space also.

LITERATURE REVIEW ON PSO

Originally PSO was proposed in [1] where PSO was proposed as an optimization tool. Two types of PSO namely, Discrete PSO and Continuous PSO versions were proposed. With several passes over the search space and updating local best and global best solutions during each pass, PSO performed much faster than ACO or GA.

In [2] authors introduced the concept of inertia weight into the original PSO. With introduction of inertia weight PSO could converge even faster. Initially inertia weight was proposed to lie in the range [0.9, 1.2], which can improve performance of PSO. Different values of inertia allowed better control over solution search space. Higher values of inertia weight will result in overshooting the and lower values will trap search in definite area in search space.

In [4] authors exploit PSO for optimizing overall tasks completion cost in a workflow and respecting the given deadline constraints. The proposed metaheuristic approach based on PSO succeeds whereas IC-PCP fails to meet application's deadline. In comparison IC-PCP failedto meet deadline constraints as IC-PCP ignored VM boot time. Results prove that PSO performs better than current state-of-the-art algorithms. Proposal considered deadline constraint. Proposal generates constraint makespan and performs cost evaluation for various workflows like Montage, Ligo etc. When compared to SCS, proposed algorithm is capable of generating better schedules and achieved cost optimization.

In [5] authors proposed mathematical model using a Load Balancing Mutation (balancing) Particle Swarm Optimization (LBMPSO) and considered reliability and availability as the objective parameters of proposals. LBMPSO used an algorithm to generate schedule and allocation for cloud computing environment. Algorithm considered available resources for generation of schedule and allocation patterns. Basic PSO suffers from free VMs, allocation of more than one task to same VM, allocation of same tasks to multiple VMs and premature convergence. LBMPSO takes into account execution time, transmission time, make span, round trip time, transmission cost and load balancing between tasks and achieved reliability in task scheduling. Idea of LBMPSO is to reschedule failure tasks to available VM. LBMPSO performance was compared with standard PSO, random algorithm and Longest Cloudlet to Fastest Processor (LCFP) algorithm to show that LBMPSO can save in make span, execution time, round trip time, transmission cost.

In [6], authors proposed a model for resource-task mapping which could reduce execution cost and also designed a PSO based heuristic to allocate tasks to resources. Both computation cost and data transmission cost are calculated by using the workflow application. Authors compared results of heuristic against "Best Resource Selection" (BRS) heuristic and found that PSO based task scheduling could result into three times cost savings.

In [7], authors compared three popular heuristic approaches namely PSO, GAs and MPSO for efficient task scheduling in cloud environment. MPSO algorithm improved makespan characteristics when compared with PSO and GA.

Modified Particle Swarm Optimization Algorithm

Particle Swarm Optimization was developed by Eberhart and Kennedy in 1995 and it has been widely used stochastic optimization technique based on the behavior of animals and birds. In MPSO, the particle is represented by its position and velocity; these particles keep track of local best (LB) and global best (GB) values; fitness function determines the LB and GB values. In the scheduling approach, particles refer to VMs; LB refers to under-loaded VM from all LB values. The algorithm iterates continuously to get the new LB and GB values. In MPSO, GB does not remain the same in each iteration when compared to PSO. The position and velocity of a particle are updated based on the following Equations (1) and (2) respectively.

$x(t + 1) = x(t) + v(t)$ (1)

$x(t + 1) = v(t) + c1r1(LB−x(t)) + c2r2(GB−x(t))$ (2) where, $x(t)$ is current position of particle/Current load of a VM. LB is least under loaded VM from cluster. GB is least under loaded VM from all LB values. c1 and c2 acceleration coefficients , usually c1= c2=2. r1 and r2 are random numbers between (0,1).

Scheduling of VMs Using MPSO Algorithm

Here, we use a MPSO technique for scheduling the VMs against incoming tasks. Cloud receives tasks in a rapid rate from the outside world, assigning and executing these tasks is a challenging issue. Number of incoming requests/tasks i.e.{x1,x2,x3,...,xn} have to be scheduled on VMs i.e. {vm0,vm1,vm2,...,vmk}. Here, our proposed MPSO algorithm is deployed for scheduling the tasks in a balanced way. The MPSO algorithm plays an important role in assigning the incoming tasks to the VMs as efficiently as possible. We experimented this work for a private cloud which receives the tasks in a batch of ten (can be extended) and these tasks are assigned to the VMs. Clustering depends on the number of VMs taken for experimentation. Algorithm 1 gives the complete details of task scheduling by MPSO. In every iteration, each cluster will identify the least loaded VM referred to as local best (LBz) and the smallest among these VMs referred to as global best (GB). The next task is allocated to the VM that is associated with GB. If the GB remains the same in the subsequent iteration, then GB is updated with second least LBz from the cluster list Cz. The same process is continued until all the tasks are executed. The time complexity of this algorithm is 0(n.z). Since z is a constant hence the time complexity of MPSO algorithm will be 0(n) in polynomial time.

Algorithm

Task Scheduling Using MPSO 0: Initialisation:{vm0,vm1,vm2,...,vmk}count = 0 Local Best (LBz) = 0 Global Best (GB) = 0 say VMs ={vm0,vm1,vm2,...,vmk} say Clusters, Cz ={c1, c2, c3,..., cz} Cluster size = k/Cz

1: for all incoming requests{x1,x2,x3,...,xn}
2: each cluster Cz = least loaded VM
3: Assign each one of them as LBz from Cz
4: end for
5: Assign GB = least LBz
6: Next task allocated to VM which contains GB
7: if (Next allocation == last used GB) then skip
8: goto step 2 for next least LBz
9: else
10: goto step 6

In MPSO, the particle is represented by its position and velocity; these particles keep track of local best (LB) and global best (GB) values; fitness function determines the LB and GB values.

CONCLUSION

Bio inspired algorithms are going to be a new revolution in computer science. The scope of this area is really vast since as compared to nature, computer science problems are only a subset, opening a new era in next generation computing, modeling and algorithm engineering. . The MPSO algorithm is more efficient in scheduling the tasks when compared to other algorithms.

REFERENCES

[1] http://www.swarmintelligence.org/tutorials.php, accessed on May 14, 2015.
[2] Y. Shi and R. Eberhart, “A modified particle swarm optimizer”, Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on. IEEE, (1998).
[3] Zhao, G. (2014). ‘Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment’, International Journal of Grid and Distributed Computing. 7, 1, .33-42
[4] Rodriguez, M. A., &Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. Cloud Computing, IEEE Transactions on, 2(2), 222- 235.
[5] Awad, A. I., El-Hefnawy, N. A., &Abdel_kader, H. M. (2015). Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments. Procedia Computer Science, 65, 920-929.
[6] Pandey, S., Wu, L., Guru, S. M., &Buyya, R. (2010, April). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on (pp. 400-407). IEEE.
[7] Abdi, S., Motamedi, S. A., &Sharifian, S. (2014). Task scheduling using Modified PSO Algorithm in cloud computing environment. In International Conference on Machine Learning, Electrical and Mechanical Engineering (pp. 8-9).

[8] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948.
[9] Binitha S, S Siva Sathya , “A Survey of Bio inspired Optimization Algorithms”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012
[10] Jyoti Thaman, Manpreet Singh, “Current Perspective in Task Scheduling Techniques in cloud Computing: a review”, International Journal in Foundations of Computer Science & Technology (IJFCST) Vol.6, No.1, January 2016