Asian Journal of Information Technology 13 (9): 561-565, 2014
ISSN: 1682-3915

Comparison of Ant Colony Optimization and Particle Swarm Optimization in Grid Scheduling

B. Booba and T.V. Gopal
Department of Computer Science and Engineering, Anna University, Chennai, India

Abstract: Computational grids are a modern trend in distributed computing applications includes searching and sharing of resources for a particular job in geographically distributed heterogeneous computing systems. Grid computing allows finding efficient allocation of resources to jobs submitted by users by making appropriate scheduling decisions. In a grid environment an important issue associated with efficient utilization of resources can be done by job scheduling. As per the demand of scheduling, the job scheduling is implemented as an integrated part of parallel and distributed computing. It selects the correct match of resource for a particular job providing an increase in job throughput and performance. It is often difficult to find an exact resource for a defined job to make the scheduling of job efficiently an Ant Colony algorithm is proposed for allocating optimal resources to each job at a minimal execution time. In this project, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) is proposed to solve and find an exact resource allocation by choosing shortest and the optimal path for a required specific job, minimizing the schedule of length of jobs with minimum make span and execution time. This study distinguishes both optimization methods and concluded it with its best performance.

Key words: Grid computing, job scheduling, ACO, PSO, grid scheduling

INTRODUCTION

The term grid is the combination of several resources in computer science from multiple administrative domains to obtain a common goal dealing as a distributed system non-interactive workload involving large number of files. Grid computing (Foster and Kesselman, 2003) has been considered as loosely coupled, heterogeneous and geographically dispersed from a conventional high performance cluster computing system. Grid scheduling provides a network by applying the resources of several computers to solve a single problem technically and scientifically at a same time. It requires a large amount of data and number of computer processing cycles to solve a problem. This study represents the concept of ACO and PSO to generate an efficient optimal job scheduling in minimum time period. In this study, researchers propose a technique by comparing the two methods. Such as ACO and PSO to obtain an optimal job scheduling in grid computing.

LITERATURE REVIEW

Job scheduling in grid computing depends upon the Quality of Service (QoS). Job scheduling is an NP complete problem which does mapping of jobs to specific physical resources. Job scheduling in grid is done by using two schedulers: grid scheduler and local scheduler (Berman *et al*., 2003; Schopf, 2004). A grid scheduler or broker selects the resources in grid environment and has no control over local resources as the resource are distributed whereas a local scheduler only manages a single site or cluster and owns the resource. Researchers proposed an approach to solve job scheduling in grid with time uncertainties. Shakerian *et al.* (2011) PCO based scheduling heuristics is analysed for data intensive applications is used to compute cost and data transmission cost. As compared with ACO, it results three times cost saving and provides good distribution of the workload onto resources. According to Kousalya and Balasubramanie (2008a, b), the jobs using the ACO technique in grid system provide a real distributed real time system with no global control for schedulers. It senses the current system environment freely and proceeds for further jobs by keeping updating of optimal resources, information and allocates the resource in dynamic, scalable distributed environment.

ACO AND PSO IN GRID SCHEDULING

Grid scheduling deals with the design of real time operating system, multitasking and multiprocessing

Corresponding Author: B. Booba, Department of Computer Science and Engineering, Anna University, Chennai, India

operating system (Chang *et al*., 2009; Parsa and Entezari-Maleki, 2009). A scheduler or dispatcher arranges the assigned jobs into an appropriate sequence to run on available CPUs.

As job scheduling is a multiobjective problem in computational grids. This study is proposed with two following scheduling algorithms to solve scheduling problems in grid environment.

ACO for grid scheduling: The ant colony optimization is a probabilistic method to solve computational problems by reducing it to achieve good paths through graphs. It was aimed to find optimal path and was named as ant system, based on the behaviour of ants seeking a path between their nests to source of food. The ant colony is used to produce optimal shortest round-trip to link a series of nodes. This procedure is inspired by ants to find the shortest path from nest to food based on their pheromone value. The higher the pheromone value in a path is shorter than a smaller pheromone value. The pheromone trail starts to evaporate thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. Shakerian *et al*. (2011) introduces job scheduling to choose best job whereas by Kousalya and Balasubramanie (2008a, b), the ACO for job scheduling is easier to allocate resources by updating previous data optimally. If graph may change dynamically it has an advantage over stimulated annealing as one of the scheduling algorithm in grid to adapt several changes by using ACO approach.

Edge selection: An ant is a simple computational agent in the Ant Colony Optimization algorithm. It iteratively constructs a solution for the problem to produce solution states. At each iteration of the algorithm, each ant moves from a state a to state b corresponding to a more complete intermediate solution. Thus, each ant n computes a set of feasible expansions to its current state in each iteration, moves to one of these in probability. For ant n, the probability ρ_{ab}^{m} of moving from state a to state b depends on the combination of two values, viz., the attractiveness η_{ab} of the move as computed by some heuristic indicating the a priori desirability of that move and the trail level τ_{ab} of the move indicating how proficient it has been in the past to make that particular move. The trail level represents a posteriori indication of the desirability of that move. Trails are updated usually when all ants have completed their solution increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively. In general, the mth ant moves from state a to state b with probability:

$$p_{ab}^{m} = \frac{(\tau_{ab}^{\alpha})(\eta_{ab}^{\beta})}{\sum(\tau_{ab}^{\alpha})(\eta_{ab}^{\beta})}$$

Where:

τ_{ab} = The amount of pheromone deposited for transition from state a to b

α and β = The respective adjustable weights

Consequently, the routing preferences of ants can be altered by selecting different values of α and β. If, α and β ants paths with higher pheromone concentrations and a higher value of directs ants to paths with more optimistic heuristic values. In general, different values of and are suitable to be applied at different states of a network. A lower value of is generally preferred when pheromone concentration along paths may not necessarily reflect their optimality. Examples of such situations include the initial stage after a network reboots and when there are frequent and abrupt changes in network status due to either link (or node) failure or introduction of new paths (nodes). However, as a network stabilizes, a higher value of is preferred favor $0 \leq \alpha$ is a parameter to control the influence of τ_{ab}, η_{ab} is the desirability of state transition ab and $\beta \geq 1$ is a parameter to control the influence of η_{ab}.

Pheromone update: When all the ants have completed a solution, the trails are updated by:

$$\tau_{ab} \leftarrow (1-\rho)\tau_{ab} + \sum_{m} \Delta\tau_{ab}^{m}$$

Where:

τ_{ab} = The amount of pheromone deposited for a state transition ab

ρ = The pheromone evaporation coefficient

$\Delta\tau_{ab}^{m}$ = The amount of pheromone deposited by mth ant, typically given for TSP problem by:

$$\Delta\tau_{ab}^{m} = \begin{cases} K/T_{m} & \text{if any ant T uses curve ab in its path} \\ 0 & \text{otherwise} \end{cases}$$

Where:

T_{m} = The cost or fluctuation rate of the mth ant's tour

K = Constant

The better solution ant's tour is the more pheromone is received by elements belonging to the tour. In general, elements which are used by many ants and which are contained in better solution will receive more pheromone and therefore are also more likely to be chosen in future iterations of the algorithm.

PSO for grid scheduling: PSO is an Adaptive algorithm finds the best solution for a problem as a point on

n-dimensional space. It works based on particles speed and can be operated by two operators: velocity update and position update. These can be generated by accelerating towards the particles and its speed is to be calculated at its each iteration by assuming the distance of previous information based on current velocity and the distance from the global best position. According to Kousalya and Balasubramanie (2008a, b), PSO technique keeps record where they have success at a point. PSO is a metaheuristic approach makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions (Selvarani and Sadhasivam, 2010). The PSO generally used in two ways as followed:

Parameter selection: The choice of PSO parameters can have a large impact on optimization performance. It selects parameters that yield good performance. It is to be minimized to form a hyper-surface of dimensionality same as that of the parameters to be optimized variables depending on the particular problem. A search is to be assumed depends on extensive parameters. Where a solution based hyper-surface would need fewer particles and lesser iterations. This is analogous to another realistic situation of flocks searching for a good source, compared to another terrain where there are very few others where it becomes easy to search for particle and lesser number of individuals and iterations will suffice. Meta-optimization is tuned by using another overlaying optimizer for PSO parameters.

Neighbourhood and topologies: PSO is basically trapped in a local minimum. This premature convergence can be avoided by the best known position 1 of a sub-swarm around the particle that is moved. Such a sub-swarm can be a geometrical one can be assumed as the d nearest particles or a set of particles which do not depend on any distance. Thus, a best local PSO variant is found. An information link between each particle and its neighbours is generated to the set of these links building a graph for communication network, known as topology of the PSO variant. The topology is not necessarily fixed and can be adaptive. Social topology commonly used in ring in which each particle has just two neighbours.

When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle and the resulting algorithm is referred to as the gbest PSO. When smaller neighborhoods are used, the algorithm is generally referred to as the lbest PSO (Hu *et al*., 2009). The performance of each that varies depending on the optimization problem (Fidanova and Durchova, 2006), each particle in the swarm isrepresented by the following characteristics:

- ai→The current position of the particle
- vi→The current velocity of the particle
- bi→The personal best position of the particle

The personal best position of particle i is the best position visited by particle i so far. There are two versions for keeping the neighbours' best vector, namely lbest and gbest. In the local version, each particle keeps track of the best vector gbest attained by its local topological neighborhood of particles. For the global version, the best vector gbest is determinedby any particles in the entire swarm. Hence, the gbest Model is a special case of the lbest Model. During each PSO iteration, particle i adjusts its velocity v_{ij} and position vectorparticle ij through each dimension j with random multipliers, the personal best vector ($pbest_{ij}$) and the swarm'sbest vector ($gbest_{ij}$, if the global version is adopted). If global version is adopted, the Eq. 1 and 2 are used:

$$v_{ij} = w \times v_{ij} + c_1 \times randl(pbest_{ij}\ particle_{ij}) \quad c_2 \times rand2(gbest_j\text{-}particle_{ij}) \tag{1}$$

$$Particle_{ij} = Particle_{ij} + v_{ij} \tag{2}$$

If local version is adopted, then the following Eq. 3 and 4 are used:

$$v_{ij} = w \times v_{ij} + c_1 \times r \text{ and } 1((pb_{ij}\text{-}particle_{ij}) + c_2 \times r \text{ and } 2(lb_{ij}\text{-}particle_{ij}) \tag{3}$$

$$Particle_{ij} = Particle_{ij} + v_{ij} \tag{4}$$

Where:

c_1 and c_2 = The cognitive coefficients
rand1 and rand2 = Random real numbers

Equation 1-4 show explicitly that a particle always apply its own and other particles' successes as the strategy to adjust its trajectory. In Eq. 1, a time-varying coefficient which is called inertial weight (Sim and Sun, 2003) decreases linearly from a maximum when the optimization starts to a minimum toward the end. It has the effect of changing the scale of exploration from global to localized during the entire course.

COMPARISON BETWEEN ACO AND PSO

In this study, the comparison between ant colony optimization and particle swarm optimization is being compared based on its throughput measure according to grid system. As ACO Method does not use controlled global schedulers to finish the allocation of resources

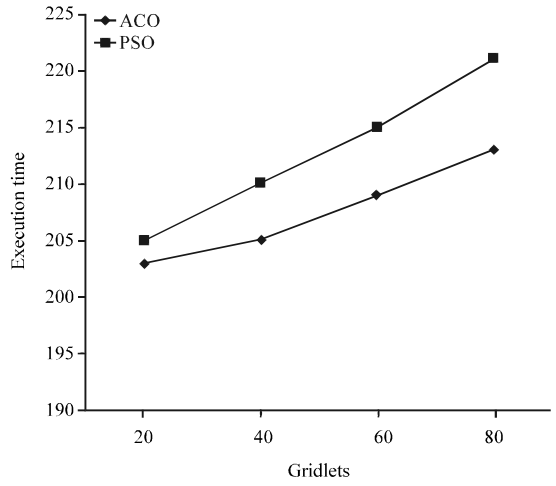

Fig. 1: ACO and PSO performance comparison based on simulation time

Table 1: Experimental results of ACO and PSO for probability makespan

No. of iteration in matrice	Values
Probabilistic makespan of ACO	
[0][0]	818
[0][1]	652
[0][2]	563
[0][3]	506
[1][0]	491
[1][1]	404
[1][2]	348
[1][3]	343
[2][0]	309
[2][1]	280
[2][2]	176
[2][3]	163
Probabilistic makespan of PSO	
[0][0]	799
[0][1]	550
[0][2]	490
[0][3]	476
[1][0]	359
[1][1]	350
[1][2]	332
[1][3]	312
[2][0]	275
[2][1]	230
[2][2]	141
[2][3]	132

within a given time. Whereas in PSO is best ever for its low computational cost and high throughput during run time. It helps to find the optimal solution efficiently.

EXPERIMENTAL RESULTS

The experimental simulation is implemented by evaluating the parameters to find the efficient throughput measure. It can be done by analysing the probability makespan values. The values are shown in the in Table 1. From comparison purposes, ACS was run using 10, 20, 30, 40, 50, 60, 70, 80 and 80 ants such that the number ofants used by ACS is equal to the sum of ants of all colonies in ACO and PSO. Here, researchers use 4 gridlets and 3 resources and the results of the experiments are shown in Table 1.

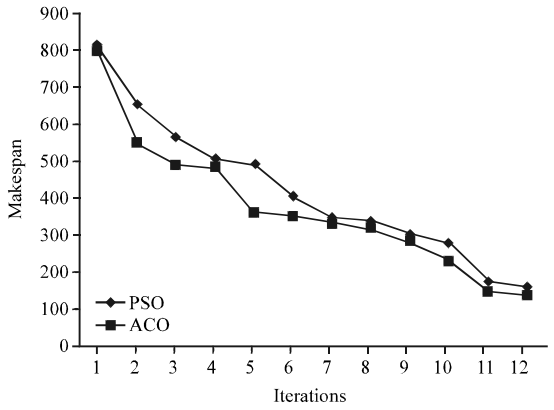

Fig. 2: ACO and PSO performance comparison

Probabilistic makespan: Figure 1 shows the study of minimum simulation time for particle swarm optimization and Ant Colony algorithm. These algorithms are compared with the performance of minimum stimulated time than traditional process. According to, groups of ants construct local pheromone of their own using some expected execution time matrix in parallel (Shakerian *et al.*, 2011). This method is best suited and used for tacking problems with large number of data sets. This performance measure produces a fast rate better optimization makespan.

Figure 2 defines a method to find the solution for the problem of resource with total execution times which equal to the makespan, to move or swap a set of jobs from a processor to another resource with minimum makespan. According to Shakerian *et al.* (2011), this search is performed on each problem processor until an optimal solution is found based on global pheromone.

CONCLUSION

The study on comparison of ACO and PSO has been presented in this study by analysing the optimization methods of each algorithm. Both optimization techniques are assigned with a specific task to allocate resources within minimum execution time by analysing the makespan to measure the throughput. PSO is considered as best optimization with low computational cost. The Stimulated Annealing Method is used as global optimization so search the optimal solution compared with ACO.

REFERENCES

Berman, F., G. Fox and A.J.G. Hey, 2003. Grid Computing: Making the Global Infrastructure a Reality. John Wiley and Sons, London, UK., ISBN-13: 9780470853191, Pages: 1012.

Chang, R.S., J.S. Chang and P.S. Lin, 2009. An ant algorithm for balanced job scheduling in grids. Future Generat. Comput. Syst., 25: 20-27.

Fidanova, S. and M. Durchova, 2006. Ant algorithm for grid scheduling problem. Proceedings of the 5th International Conference on Large-Scale Scientific Computing, June 6-10, 2005, Sozopol, Bulgaria, pp: 405-412.

Foster, I. and C. Kesselman, 2003. The Grid 2: Blueprint for a New Computing Infrastructure. 2nd Edn., Morgan Kaufmann, UK., ISBN-13: 9780080521534, Pages: 748.

Hu, X.H., J.C. Ouyang, Z.H. Yang and Z.H. Chen, 2009. An IPSO algorithm for grid task scheduling based on satisfaction rate. Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics, August 26-27, 2009, Hangzhou, Zhejiang, China, pp: 262-265.

Kousalya, K. and P. Balasubramanie, 2008a. Ant algorithm for grid scheduling powered by local search. Int. J. Open Problems Compt. Math., 1: 222-240.

Kousalya, K. and P. Balasubramanie, 2008b. An enhanced ant algorithm for grid scheduling problem. Int. J. Comput. Sci. Network Secur., 8: 262-271.

Parsa, S. and R. Entezari-Maleki, 2009. RASA: A new grid task scheduling algorithm. Int. J. Digital Content Technol. Appli., 3: 152-160.

Schopf, J.M., 2004. Ten Actions when Grid Scheduling: The User as a Grid Scheduler. In: Grid Resource Management: State of the Art and Future Trends, Nabrzyski, J., J.M. Schopf and J. Weglarz (Eds.). Chapter 2, Kluwer Academic Publisher, Norwell, MA., USA., ISBN-13: 9781402075759, pp: 15-24.

Selvarani, S. and G.S. Sadhasivam, 2010. Improved job-grouping based PSO algorithm for task scheduling in grid computing. Int. J. Eng. Sci. Technol., 2: 4687-4695.

Shakerian, R., S.H. Kamali, M. Hedayati and M. Alipour, 2011. Comparative study of ant colony optimization and particle swarm optimization for grid scheduling. J. Math. Comput. Sci., 2: 469-474.

Sim, K.M. and W.H. Sun, 2003. Multiple ant colony optimization for load balancing. Proceedings of the 4th International Conference on Intelligent Data Engineering and Automated Learning, March 21-23, 2003, Hong Kong, China, pp: 467-471.