International Journal of Emerging Technologies in Engineering Research (IJETER)
Volume 4, Issue 8, August (2016) www.ijeter.everscience.org

# Resource Allocation Algorithm in Distributing System Using Ant Colony Optimization-A Review

Er. Anuj Aggarwal
Research Scholar, Uttrakhand Technical University, Dehradun, Uttrakhand, India.

Dr. Rajesh Verma
Assistant Professor, Department of Computer Science, SUS College, Indri, Karnal, Haryana, India.

Dr. Ajit Singh
Associate Professor, Department of Computer Science & Engineering, BTKIT, Dwarahat, Almora , India.

**Abstract – A Resource allocation system is the central component of networking system. There are many projects which are implemented in distributed system with different architectures and services. A Network Computing System is a virtual computer formed by a networked set of heterogeneous machines that agree to share their local resources with each other followed in the implementation of real resource allocation systems for large scale network computing systems known as Grids. The resource allocation system is the back bone of grid computing system. Resources in the grid system are distributed, heterogeneous, autonomous and unpredictable. Resource allocation in the grid environment depends upon the characteristics of the tasks, machines and network connectivity. In this paper, we provides a brief overview of resource allocation in grid computing considering important factors such as types of resource allocation in grid computing, resource allocation models and comparison of various scheduling algorithm in resource allocation in grid computing.**

**Index Terms – Grid Computing, Resource Allocation, Algorithms for Resource Allocation Scheduling.**

## 1. INTRODUCTION

A Grid enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce. To build a Grid, the development and deployment of a number of services is required. They include *low-level services* such as security, information, directory, resource management (resource trading, resource allocation, quality of services) and *high-level services/tools* for application development, resource management and scheduling (resource discovery, access cost negotiation, resource selection, scheduling strategies, quality of services, and execution management) . The resource management and scheduling systems for Grid computing need to manage resources and application execution depending on resource consumers' and owners' requirements, and they need to continuously adapt to changes in the availability of resources. This requirement introduces a number of challenging issues that need to be addressed; namely, site autonomy, heterogeneous substrate, policy extensibility, resource allocation or co-allocation, online control, resource trading, and quality of service based scheduling. Among them, the two most challenging aspects of Grid computing are resource allocation and scheduling.

## 2. RELATED WORK

There have been many studies on survey of grid resource allocation in distributing system. For a dynamic resource allocation algorithm, it is unacceptable to exchange state information frequently because of the high communication overhead. In order to reduce the communication overhead Martin et al. [18] studied the effect of communication latency, overhead and bandwidth in cluster architecture for application performance. Distributed network computing environments have become a cost effective and popular choice to achieve high performance and to solve large scale computation problems. There are many types of algorithms that have been used in resource allocation in Grid computing system. Martino and Mililotti addresses the use of Genetic Algorithm (GA) and Tabu Search (TS) to solve the resource allocation problem in the dynamic environment [5]. These algorithms have their limitations like these algorithms require extra storage and processing requirement at the scheduling nodes. Grid computing involves coupled and coordinated use of geographically distributed resources for purposes such as large scale computation and distributed data analysis [17].

A new hybrid resource allocation policy can be integrated in static and dynamic allocation techniques with the objective to allocate effective node, identify the system imbalance immediately when a node becomes unable to participate in task processing. Ibarra and Kim in [15] used a combination of intelligent agents and multi-agent that work in grid load balancing area. In static grid load balancing, the iterative heuristic algorithm is better than the FCFS algorithm. Grid infrastructures are dynamic in nature in the sense of resource availability and hence a changing network topology. Resource heterogeneity and network heterogeneity also exits in Grid environment. Scheduling jobs onto resources is NP hard problem. So, we need an algorithm that consider optimization in terms of time and cost for efficient scheduling and execution. Current grid resource scheduling algorithms are mainly based on User-Directed Assignment (UDA), Minimum Completion Time (MCT), Minimum Execution Time (MET), Min-Min [5], Max-Min, heuristic algorithm, genetic algorithm (GA)[5], Ant Algorithm (AA) [6], multi-Agent, computational economy model [7]. In [19] authors proposed a dynamic load balancing algorithm which considers CPU length, CPU and memory utilization and network traffic as load metric.

Although [6][14] presented scheduling algorithms based on ant algorithm, but the authors just analyzed ant algorithm based classified task scheduling method and ACA based scheduling without considering about the price attribute. In this paper, price factor and time factor are taken into consideration, and the objective is how to save total execution time and cost by studying the various factors of resource allocation algorithms.

## 3. RESOURCE ALLOCATION SCHEDULING

In paper [24] author proposed that ant algorithm can be improved using some form of local search algorithm. Local search algorithm can be applied to the output of the ant algorithm to find the optimal resource to schedule a job. Author used Move-Top, Move-Minimum Completion Time Job First and Move-Maximum Completion Time Job First local search methods. The job completion time is the only main input for the proposed algorithm. The factors such as CPU workload, communication delay, QoS are not considered. After experimentation it is found that the ant colony algorithm with local search algorithms performs 30% better than the algorithm without local search.

In paper [25] author proposed an ant colony algorithm for dynamic job scheduling in Grid environment. The next resource selection depends on the pheromone value and the transition probability. Author improved the existing ant colony algorithm and tried to minimize the total tardiness time of the job. Author considered that the initial pheromone value depends on current and the expected tardiness time of the job. Two pheromone updating rules are used local update rule and global update rule. Transition probability is used to select the next resource for the job in which the heuristic desirability of the assignment of the job on a machine is inversely proportional to the completion time of the job. In last author compared the performance of various job schedulers and dispatching rules for Grid environment like FCFS, METDD, MTERDW in ACO and it is found that proposed ACO performs 17% better than others. The different cost measures for the algorithm such as makespan time, Grid efficiency and job error ratio, job workflow may be considered for the future work.

In paper [26] author described an ant colony optimization algorithm in combination with local and tabu search. The author suggested that in ant colony algorithm the ants build their solutions using both information encoded in the pheromone trail also specific information in the form of the heuristic. The pheromone value updation rule is taken from the Max-Min algorithm in which the pheromone is only updated by the bets ant. And for heuristic information the Min-Min heuristic is used, which suggests that the heuristic value of a particular job should be directly proportional to the minimum completion time of the job. The local search algorithm is applied to each of the solutions built by the ants before the pheromone updation stage to take ant solution to its local optimum. The tabu search algorithm performs number of trails or iterations on the solution build by the ant colony algorithm and after each iteration the solution gets improved. In last the author compared the performance of Min-Min algorithm with the Min-Min+local Search and Min-Min+Tabu Search and ACO and found that the proposed algorithms perform better than the Min-Min.

In paper [27] the author proposed heuristic scheduling algorithm that works in two phases. In the first phase heuristic begins with the set of all unmapped tasks then the set of minimum completion time is found like Min-Min heuristic. In the second phase there are two choices based on the threshold value (represents the percentage balance of load in the system). The idea is to select the pair of machine and task from the first phase so that the machine can execute its corresponding task effectively with a lower execution time in comparison with other machines and by comparing workload in the second phase. Proposed algorithm has the asymptotic complexity of $O(mn^2)$. The author compared the performance of proposed heuristic scheduling algorithm and various existing algorithms using the benchmark simulation model implemented in MATLAB Environment for distributed heterogeneous computing systems. The results show that the makespan and resource utilization is improved. The limitation of the proposed algorithm is that it does not consider the factors such as CPU workload, communication delay etc.

In paper [28] an improvement in the ant colony algorithm is proposed to reduce the makespan and to converge towards the optimal solutions in a very faster manner. The algorithm proposed used the basic ant pheromone updating rule and modified transition updating rule. The proposed algorithm uses a new pheromone updating rule and probability matrix calculation formula in order to increase the efficiency the existing ant colony algorithm. The proposed algorithm is simulated with large data set and results show the improved makespan.

In paper [29] the author proposed an ant colony algorithm with the modification in existing ant pheromone updating rule proposed by Marco and Dorgio in 1992 in his Ph.D. thesis. The proposed algorithm takes in to consideration the free time of resources and the execution time of the jobs to achieve better resource utilization and better scheduling. The proposed and the existing algorithm are simulated with GridSim toolkit with different number of tasks and it is found that the existing algorithm takes less time to execute the tasks.

In paper [30] the author suggested that if in an ant colony algorithm for task scheduling in Grid computing the tasks are always scheduled to the resource with the high possibility, then the load on the resource may be increased and the job may be kept waiting in the queue for the resource to be free even though the other resources are free. So, the algorithm can be modified in such a way that if the difference between possibility of the resource selected for execution of a task using ant algorithm and the possibility of the resource with the highest possibility is less than certain threshold, then the task will be scheduled to the resource selected by the ant algorithm. Otherwise, the scheduler selects another resource and above procedure will be repeated. The proposed algorithm is implemented through GridSim toolkit and compared with the existing ant algorithm with different number of resources and tasks and it is found that performance is increased in terms of low processing time and low cost of execution. The algorithm does not consider that there can be hierarchy of cluster of resources and jobs can be grouped according to their properties or resource requirement.

In paper [31] the author proposed a hybrid ant colony optimization scheduling algorithm. The proposed algorithm uses a new pheromone updating rule and probability matrix. The experimental tests are conducted on high task high machine, high task low machine, low task high machine and low task low machine with heterogeneous large data set. The author proposed that the increasing in the level of pheromone level will give the better results.

In paper [32] author proposed an algorithm for distributed systems, shared asynchronously by both remote and local user. The author consider that there will be multiple nodes corresponding to one machine .Then the performance of the proposed algorithm is compared with the ant colony algorithm in online mode and it is found that the proposed algorithm performs better.

In paper [33] the author proposed an ant colony algorithm to achieve the QoS. In this paper author proposed a two new versions of pheromone updating rule, one in which pheromone value is inversely proportional to the execution time of the particular task on a selected resource and in second the pheromone value is inversely proportional to the total make span of the schedule and also the transition probability calculation formula is modified. Both improvements show that overall makespan and flow time is improved.

## 4. EXISTING RESOURCE ALLOCATION APPROACHES

Various approaches have been designed to schedule the jobs in computational Grid. The most popular are:

Opportunistic load balancing (OLB)

Without considering the job’s execution time, it assigns a job to the earliest free machine. If more than one machine is free then it assigns the job in arbitrary order to the processor. This scheduling mechanism runs faster. The advantage of this method is that it keeps almost all machines busy. Yet it does not assure load balance.

Minimum Execution time (MET)

The minimum execution time or MET assigns each job to the machine that has the minimum expected execution time. It does not consider the availability of the machine and the current load of the machine. The resources in Grid system have different computing power. Allocating all the smallest tasks to the same fastest resource redundantly creates an imbalance condition among machines. Hence solution is static.

Minimum Completion Time (MCT)

The algorithm calculates the completion time for a job on all machines by adding the machine’s availability time and the expected execution time of the job on the machine. The machine with the minimum completion time for the job is selected. The MCT considers only one job at a time. This causes that particular machine may have the best expected execution time for any other job. The drawback of MCT is that it takes long time to calculate minimum completion time for a job.

Max-Min

Max-min begins with a set of all unmapped tasks. The completion time for each job on each machine is calculated. The machine that has the minimum completion time for each job is selected. From the set, the algorithm maps the job with the overall maximum completion time to the machine. Again the above process is repeated with the remaining unmapped tasks. Similar to Min-min, Max min also considers all unmapped tasks at a time.

Min-Min

Min-min algorithm starts with a set of all unmapped tasks. The completion time for each job on each machine is calculated. The machine that has the minimum completion time for each job is selected. Then the job with the overall minimum completion time is selected and mapped to the machine. Again, this process is repeated with the remaining unmapped tasks.

Compared to MCT, Min-min considers all unmapped tasks at a time. The drawback of Min-Min is that, too many jobs are assigned to a single node. This leads to overloading and response time of the job is not assured.

## 5. ANT COLONY BASED SCHEDULING ALGORITHMS

Ant colony optimization (ACO) was first introduced by Marco Dorigo as his Ph.D. thesis and was used to solve the TSP problem. ACO was inspired by ant's behavior in finding the shortest path between their nests to food source. Many varieties of ants deposit a chemical pheromone trail as they move about their environment, and they are also able to detect and follow pheromone trails that they may encounter. With time, as the amount of pheromone in the shortest path between the nest and food source increases, the number of ants attracted to the shortest path also increases. This cycle continues until most of the ants choose the shortest path. As this work is a cooperative one and none of the ants could find the shortest path separately, ACO algorithm can be categorized as a swarm intelligent algorithm. Various types of ACO algorithm are presented. Each of them has some special properties, i.e. Ant Colony System (ACS), Max-Min Ant System (MMAS). Fast Ant System (FANT), Max-Min Ant System is based on the basic ACO algorithm but considers low and upper bound values and limits the pheromone range to be between these values. Defining those values, lets MMAS avoid ants to converge too soon in some ranges. In Fast Ant System version of ACO just one ant participate in each iteration search and also there is no pheromone evaporation rule in this version. Hence the ant algorithm is suited for usage in Grid computing task scheduling. In the Grid environment, the algorithm can carry out a new task scheduling by experience, depending on the result in the previous task scheduling. In the Grid computing environment, this type of scheduling is very much helpful.

## 6. CONCLUSION

In this paper, a review of resource allocation system in grid computing is presented. Various types of Resource allocation in grid computing and various models for resource allocation in grids computing are discussed. Various scheduling algorithm in grid computing have been analyzed. A comparison on various parameters like architecture type, environment type (Heterogeneous or Homogeneous), response time, and resource utilization is done on different types of job scheduling.

## REFERENCES

[1] Foster I., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of High Performance Computing Applications,vol.15(3), pp.230-235, 2012.

[2] Baker M., Buyya R., and Laforenza D., "Grids and Grid Technologies for Wide-Area Distributed Computing", Software: Practice and Experience, vol. 32(15), pp. 1437-1466 , Dec 2009.

[3] Berman F., Hey A. J. G., and Fox G., "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons, 2007.

[4] Foster I. and Kesselman C., "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 2008.

[5] Gibbins H., Nadiminti K., Beeson B., Chhabra R., Smith B., and Rajkumar Buyya, "The Australian BioGrid Portal: Empowering the Molecular Docking Research Community", APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch (APAC 2005), Sept. 2005.

[6] Krauter K., R. Buyya R., and Maheswaran M., "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing", International Journal of Software: Practice and Experience, vol. 32, pp. 135-164,2002.

[7] Abramson D, Giddy J, Kotler L. "High performance parametric modeling with nimrod/g: Killer application for the global grid?" Proceedings of the 14th International Parallel and Distributed Processing Symposium ( IPDPS 2000 ), pp. 520–528, Apr 2007.

[8] Buyya R, Abramson D, Giddy J. "Nimrod/G: An architecture for a resource management and scheduling system in a global computational Grid". Proceedings of the International Conference on High Performance Computing in Asia–Pacific Region (HPC Asia 2000), 2009.

[9] Hoscheck W, Jaen-Martinez J, Samar A, Stockinger H, Stockinger K. "Data management in an international data Grid project". Proceedings 1st IEEE/ACM International Workshop on Grid Computing (Grid 2000), pp. 77–90, Dec 2011.

[10] Chervenak A, Foster I, Kesselman C, Salisbury C, Tuecke S. "The data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets." Journal of Network and Computer Applications, vol. 23(3), pp. 187–200.

[11] Nahrstedt K, Chu H, Narayan S. "QoS-aware resource management for distributed multimedia". Journal of High-Speed Networking (Special Issue on Multimedia Networking); vol. 7(3), pp. 227–255, Dec. 2010.

[12] The Globus Project, http://globus.org

[13] Baker M, Fox G. "Metacomputing: Harnessing informal supercomputers". High Performance Cluster Computing: Architectures and Systems, vol. 1, Buyya R (ed.). Prentice-Hall: Englewood Cliffs, NJ, 2013.

[14] Buyya R, Giddy J, Abramson D. "A case for economy grid architecture for service-oriented Grid computing". 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), in Conjunction with IPDPS 2001, San Francisco, CA, Apr. 2001. IEEE Computer Society Press: Los Alamitos, CA, 2001.

[15] De Roure, D. Baker, M., Jennings, N. R. and Shadbolt, N. "The evolution of the Grid". In Grid Computing - Making the Global Infrastructure a Reality, John Wiley and Sons Ltd. ISBN 0470853190, 2003, pp. 65-100,2003.

[16] Buyya R, Abramson D, Giddy J. "Economy driven resource management architecture for computational power grids". International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2000), Las Vegas, NV, 2000. CSREA Press: Athens, GA, 2012.

[17] Buyya R., "Economic-based distributed resource management and scheduling for Grid computing". PhD Thesis, Monash University, Melbourne, Australia, Apr. 2002.

[18] Siriluck Lorpunmanee, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai, "An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment," International Journal of Computer and Information Science and Engineering, vol. 1(4), pp. 207-214, 2007.

[19] Bing Qi Chunhui Zhao, "Ant Algorithm Based Load Balancing for Network Sessions," in Proceedings of 3rd International Conference on Natural Computation, vol. 4, pp. 771-775, 2007.

[20] Ruchir Shah, Bhardwaj Veeravalli, and M. Misra, "On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments," Institute of Electrical and Electronics Engineer Transaction On Parallel And Distributed System, vol. 18(12), pp. 1675-1686, December 2007.

[21] B. Yagoubi, Y. Slimani, "Task Load Balancing Strategy for Grid Computing," Journal of Computer Science vol. 3 (3), pp.186-194, 2007.

[22] K.Shahu Chatrapati , K.Purna Chand, Y.Ranjith Kumar , Dr.A.Vinaya Babu, "A Novel Approach for Effective Dynamic Load Balancing by using Tendensive Weight," International Journal of Computer Science and Network Security, vol. 8(6), pp. 42-48, June 2008.
[23] C. Chen, K.C. Tsai., "The server reassignment problem for load balancing in structured P2P systems," IEEE Transaction on Parallel Distributed Systems, vol. 19(2), pp. 234–246, 2008.
[24] C. Sosa, and A.S. Grimshaw, "Bringing the Grid Home," in Proceedings of 9th IEEE/ACM International Conference on Grid Computing, pp. 152-159, September 2008.
[25] Qingyang Meng, Jianzhong Qiao, Jun Liu, and Shukuan Lin, "A Dynamic Load Balancing Method Based on Stability Analysis," in Proceedings of International Symposium on Computer Science and Computational Technology (ISCSCT'08), pp. 404-408, December 2008.
[26] Parveen Jain, Daya Gupta, "An Algorithm for Dynamic Load Balancing in Distributed Systems with Multiple Supporting Nodes by Exploiting the Interrupt Service," in Proceedings of International Journal of Recent Trends in Engineering, vol.1(1), pp. 232-236, May 2009.
[27] Casey, L.M., "Decentralized Scheduling", The Australian Computer Journal, vol. 13(2), pp. 58- 63, February 2011.
[28] Casauant, T.L., and Kuhl, J. G., "A Tamnomy of Scheduling in General-Purpose Distributed Computing Systems," Institute of Electrical and Electronics Engineer Transaction on Software Engineering, vol. 14 (1 l), pp. 1578-88, November 1988.
[29] Livny, M., and M. Melman, "Load Balancing in Homogeneous Broadcast Distributed Systems," in Proceedings of the ACM Computer Network: Performance Symposium, pp. 47-55, Apr. 1982.
[30] Elisha T. O. Opiyo, Erick Ayienga, Katherine Getao, & William Okello-Odongo. "Dynamic Resource Allocation: A comparison of First Price Sealed Bid and Vickrey Auctions". International Journal of Computing and ICT Research, Special Issue vol. 1(1) ,Oct 2008.
[31] J. Gomoluch and M. Schroeder. "Market-based resource allocation for grid computing: A model and simulation". In Proc. of the 1st International Workshop on Middleware for Grid Computing, pp. 211–218 ,Jun. 2009.
[32] Buyya R. and Murshed M., "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing", Concurrency and Computation: Practice and Experience, vol. 14(13), pp. 1175-1220 , Nov. 2010.
[33] David Karger and Matthias Ruhl, "New Algorithms for Load Balancing in Peer-to-Peer Systems," Tech. Rep. MIT-LCS-TR-911, MIT LCS, July 2011.