A Hybrid Heuristic Algorithm for Task Scheduling in Grid Environment

D.Ramyachitra, Ph.D
Assistant Professor
Bharathiar University
Coimbatore

P.Suganthi
Assistant Professor
Sudharsan Engineering College
Pudukkottai

ABSTRACT

Grid computing is an emerging trend that provides a high performance computing platform to solve larger scale applications by coordinating and sharing computational power, data storage and network resources. A grid coordinates and integrates resources and users of different administrative domains inside the same company or in different countries. Task scheduling is one of the key research areas in grid computing. The goal of scheduling is to achieve highest possible system throughput and to match the application's need with the available computing resources. This paper primarily focuses on task scheduling process of Artificial Bee Colony (ABC) algorithm. The objective of this algorithm is to generate optimal solution dynamically. By this scheduling, complete the task in minimum time and use the available resources in efficient manner. The best assignment of tasks produced by ABC is selected and applies Genetic Algorithm (GA) for achieving better performance and evaluation. This heuristic algorithm provides an optimal task scheduling in heterogeneous computing environments.

Keywords

Grid Computing; Resource Sharing; Task Scheduling; Heuristic Algorithms.

1. INTRODUCTION

Grid Computing is an infrastructure that enables the integrated collaborative use of high-end computers, networks, databases and scientific instruments owned and managed by multiple organizations. Grid means blue print for a new computing infrastructure [1]. Grid applications often involve large amount of data and/or computing and often require secure resources sharing across organizational boundaries. Grid computing links potentially vast numbers of computational resources such as computers, storage devices, and scientific instruments that are distributed over a network to solve complex, massive, computation-intensive problems. The size of a grid may vary from small confined such as a network of computer workstations within a corporation to large such as public collaborations across many companies and networks.

In this paper section 1 briefly discusses about the basic concepts of Grid Computing. Section 2 briefly reviews the resource sharing and scheduling in grid computing. Section 3 provides the concepts of task scheduling and scheduling algorithms. Section 4 introduces the working principles of the Artificial Bee Colony (ABC) algorithm with GA. Section 5 gives the performance results obtained from ABC and also the assignment is functioned by GA. Finally, section 6 gives conclusions and future implementations.

The general characteristics of Grid are numerous resources; connected by heterogeneous, multi-level networks [16]; different resource management policies; geographically distributed; resources are heterogeneous; unreliable resources and environments; different security requirements and policies; owned by multiple organizations and individuals.

Grid Computing Supports *Layered architecture* which is a technique used in designing computer software, hardware, and communications in which system or network components are isolated in layers so that changes can be made in one layer without affecting the others [17]. To enable *interoperability* of heterogeneous distributed resources, a grid computing system is often divided into five layers:

The application layer which is the highest level contains applications that use the lower layers to access distributed resources. The application layer adaptively adjusts user's resource demand based on the current resource conditions. *The collective layer* is responsible for coordinating distributed resources, such as scheduling a task to analyse data received from a scientific device. *The resources layer* enables applications to request and share a resource. *The connectivity layer* carries out reliable and secure network communications between resources. *The fabric layer* accesses physical resources such as disks. The fabric layer allocates CPU, storage and bandwidth required by the upper layer.

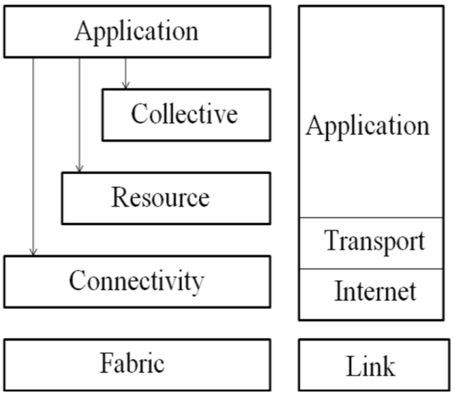

Figure 1: Grid Architecture

2. RESOURCE SHARING AND SCHEDULING

Resource sharing and coordinated problem solving in dynamic, multi-institutional virtual organizations enable integration of distributed resources using general-purpose protocols and infrastructure to achieve better-than-best-effort service. The sharing that are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource brokering strategies emerging in industry, science, and engineering [6]. The Resource layer builds on Connectivity layer communication and authentication protocols for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources. Resource layer implementations of these protocols call Fabric layer functions to access and control local resources. Resource layer protocols are concerned entirely with individual resources and hence ignore issues of global state and atomic actions across distributed collections.

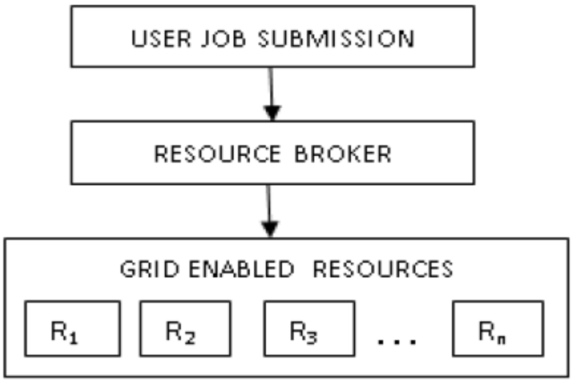

Figure 2: Grid Resource Sharing

Two primary classes of Resource layer protocols can be distinguished:
1. Information protocols are used to obtain information about the structure and state of a resource, for example, its configuration, current load, and usage policy (e.g., cost).
2. Management protocols are used to negotiate access to a shared resource, specifying, for example, resource requirements including advanced reservation and quality of service and the operations to be performed, such as process creation, or data access.

2.1 Scheduling

Scheduling is the process of deciding how to utilize the available resources for the tasks [3]. Scheduling tasks is one of the major processes in distributed computing and other related techniques. Scheduling refers to the way in which processes are assigned to run on the available processors. This assignment activity is carried out by software known as a scheduler and dispatcher. The Grid system is hierarchical two level systems, user level and resource level [8]. In user level, Grid user submits jobs to scheduler onto the grid environment. The scheduler obtains the information about the resources and its characteristics and sends jobs to the selected resources for execution.

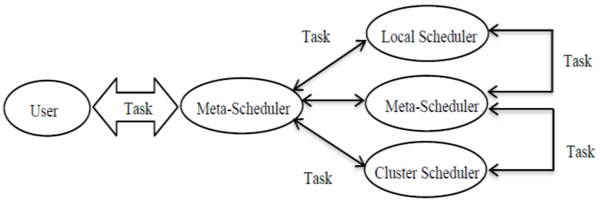

Figure 3: Grid Scheduling

In the above figure the user send the tasks to the meta-scheduler which handles all scheduling process in Grid environment. It knows all information about the resource availability and capability of resources. Hence it finds out the best allocation of tasks to the resources for the execution of the task.

2.2 Resource Scheduling

Resource scheduling is the process of mapping tasks into specific available physical resources, trying to minimize some cost function specified by the user. Task is a single unit of the application that can be independently assigned to a resource for execution. A task scheduler is a software application that is in charge of unattended background executions, commonly known for historical reasons as batch processing. Effective computation and task scheduling is rapidly becoming one of the main challenges in grid computing and is seen as being vital for its success.

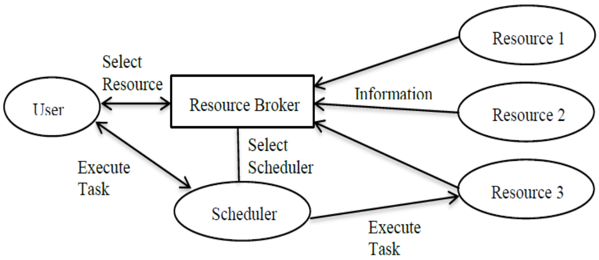

Figure 4: Grid Resource Scheduling

Today's task schedulers typically provide a graphical user interface and a single point of control for definition and monitoring of background executions in a distributed network of computers. Increasingly task schedulers are required to coordinate the integration of real-time business activities with traditional background IT processing, across different operating system platforms and business application environments [4].

2.3 Scheduling Algorithms

A scheduling algorithm is the method by which processes or data flows are given access to system resources for example processor time, communications bandwidth likewise [9]. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking which is execute more than one process at a time and multiplexing which is transmit multiple flows simultaneously. The characteristics of the tasks, machines and network connectivity play an important role in grid computing and scheduling has to be done carefully. Some popular algorithms for task scheduling in grid environment are Genetic Algorithm (GA), Tabu Search (TS), Fuzzy Logic (FL) techniques, Simulated Annealing (SA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) algorithm, etc.

2.4 Existing Task Scheduling Algorithms

Based on the ACO and ABC algorithms the following algorithm is developed for task scheduling to suitable resources in grid environment. The algorithm is as follows:

```
Algorithm The framework of a task scheduling algorithm.
input: Task T, Number of Resources n
InitializeTasks($T$ )
s ←NULL
for $i$ = 1, . . . , $n$ do (for tasks)
    for $i$ = 1, . . . , $n$ do (for resources)
        while termination conditions not met do
            s ←FindResource($n$)
            if s is a valid solution then
                s ←AllocateTask(T)
            else
                goto FindResource($n$)
            end if
        end while
    end for
end for

Output: The best-so-far solutions
```

2.5 An Artificial Bee Colony (ABC) Metaheuristic

An *Artificial Bee Colony (ABC) algorithm* is the popular method for scheduling tasks in distributed environment by using metaheuristic optimization technique. *Heuristic* means *find* or *discover* which refers to experience-based techniques for problem solving, learning, and discovery. Heuristic methods are used to *speed up the process* of finding a satisfactory solution, where an exhaustive search is impractical [4]. Metaheuristics are used for combinatorial optimization in which an optimal solution is sought over a discrete search-space. *Metaheuristic* designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

3. ARTIFICIAL BEE COLONY (ABC) ALGORITHM

Artificial Bee Colony (ABC) algorithm is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm. In common ABC algorithm is used for optimizing multivariable functions [13]. In ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts. It is assumed that there is only one artificial employed bee for each food source. That is, the number of employed bees in the colony is equal to the number of food sources around the hive. Employed bees go to their food source and come back to hive and dance on this area. The employed bee whose food source has been discarded becomes a scout and starts to search for finding a new food source. Onlookers watch the dances of employed bees and choose food sources depending on dances. The main steps of the algorithm are given below:

- INITIALIZE
- REPEAT
  1. Place the food sources on the employed bee's memory;
     Determines neighbor source, then evaluates its nectar amount and dances in the hive;
  2. Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source;
     After choosing a neighbor around that, she evaluates its nectar amount;
     Place the onlooker bees on the food sources in the memory;
  3. Send the scouts to the search area for discovering new food sources; The best food source found so far is registered.
- UNTIL (requirements are met)

In ABC which is a population based algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality of the associated solution.

The minimum food searching time evaluated by the following formula:

$$\text{Process Time} = \frac{n_i}{\sum_{i=1}^{n} (n_i)}$$

3.1. Applications of ABC

Several applications of ABC includes Biological simulation, Genetic Algorithm Improvement, Continuous Optimization, Travelling Salesman Problem (TSP), Ride-Matching Problem, Dynamic Allocation of Internet Service, Telecommunication Network Routing, Large Scale Precise, Job Shop Scheduling [14].

4. A HYBRID HEURISTIC TASK SCHEDULING ALGORITHM

The proposed algorithm is the implementation of ABC algorithm with the key concept of GA. This algorithm implements various allocation techniques of tasks to the available resources. The resource matrix is given as input to the algorithm. Based on the size of the matrix the random process time assigned to the available resources. The probabilities of execution time calculated for each and every machine. Then allocate the task to the resource which executes the task in least time. Likewise all tasks allocated to the resources. After the allocation the makespan will be calculated. The Single Shift Neighborhood (SSN), Double Shift Neighborhood (DSN) techniques will be applied on that scheduling and the respective makespans will be calculated. Then the makespans are compared for choosing the best scheduling. On the selected allocation the GA process will be applied for performance evaluation of the algorithm. GA is a population based algorithm which provides optimal solution. The result gives optimal allocation and its makespan values with associated probabilities. The following flow diagram shows the system process:

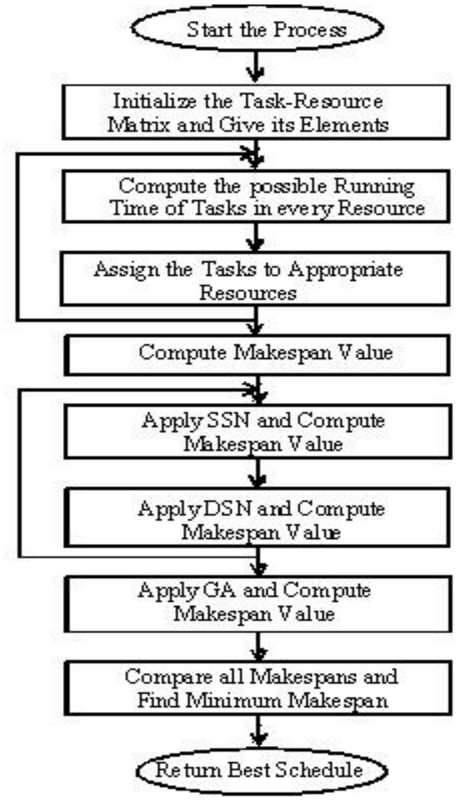
Start the Process
Initialize the Task-Resource Matrix and Give its Elements
Compute the possible Running Time of Tasks in every Resource
Assign the Tasks to Appropriate Resources
Compute Makespan Value
Apply SSN and Compute Makespan Value
Apply DSN and Compute Makespan Value
Apply GA and Compute Makespan Value
Compare all Makespans and Find Minimum Makespan
Return Best Schedule

Figure 5: Flow diagram of Heuristic Algorithm

5. PERFORMANCE EVALUATION

The ABC is a meta-heuristic optimization; hence it speeds up the process. It gives its own benefits on its performance and attributes. It reduces the turnaround time and response time. The resources in Grid were efficiently utilized. It achieves high system throughput. The following table shows the results of ABC algorithm makespan values of SSN, DSN and GA.

Table 1. Calculated Makepsan Values

Resource Matrix	SSN	DSN	GA	Makespan
10 x 2	192	147	1025	147
10 x 3	419	124	592	124
10 x 4	178	83	304	83
10 x 5	104	68	443	68

The above table shows that the ABC alone will provide better performance than applying neighbourhood structures. Then the probabilities of these allocations also show that the best solution given by ABC:

Table 2. Calculated Probabilities

Resource Matrix	SSN Probability	DSN Probability	Minimum Makespan Probability
10 x 2	0.3070	0.3859	0.3070
10 x 3	0.2141	0.6063	0.1794
10 x 4	0.3322	0.4050	0.2626
10 x 5	0.3207	0.3584	0.3207

The following graph shows the results of the hybrid task scheduling algorithm. The algorithm produces optimal solution on general allocation and DSN than SSN and GA.

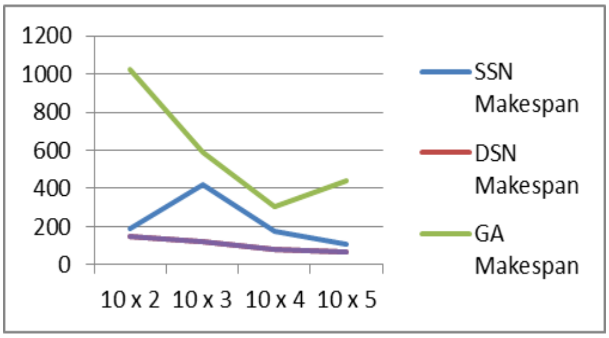

Figure 6: Makespan Evaluation

6. CONCLUSION

Grid computing presents a new trend in distributed systems and Internet computing for coordinating large-scale heterogeneous resources sharing and problem solving in dynamic organizations. Grid computing allow the challenges such as security, computational economy, uniform access, system management, resource discovery, resource allocation and scheduling, data locality, and network management. Grid scheduling is the process of scheduling applications over Grid resources. Grid Scheduling in the heterogeneous and dynamic nature of grid resources continues to be a tedious task. The paper is proposed with the packages of task scheduling, scheduling algorithms and a hybrid task scheduling algorithms with various factors. The heuristic algorithm has implemented on the base of ABC with GA. By using this algorithm a task is scheduled as optimally in Grid or any heterogeneous environment. In future this algorithm is implemented in a grid environment for effective task scheduling.

7. REFERENCES

[1] Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual organizations", International Journal of Supercomputer Applications, 15(2001).

[2] Ian Foster "What is the Grid? A Three Point Checklist" Argonne National Laboratory & University of Chicago.

[3] Peter Gradwell "Overview of Grid Scheduling Systems" Department of Computer Science, University of Bath.

[4] Fatos Xhafa, Ajith Abraham, “Computational models and heuristic methods for Grid scheduling problems”, Future Generation Computer Systems 26 (2010) 608_621.

[5] T. Casavant and J. Kuhl, ”A Taxonomy of Scheduling in General Pupose Distributed Computing Systems”, IEEE Trans. on Software Engineering Vol. 14, No. 2, PP 141-154, February 1988.

[6] M. Arora, S. K. Das, R. Biswas, “A Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments”, in Proc. Of International Conference on Parallel Processing Workshops (ICPPW’02), pp.:499-505, Vancouver, British Columbia Cananda, August 2002.

[7] L. Lee, C. Liang, H. Chang, “An adaptive task scheduling system for Grid Computing”, Proceedings of the Sixth IEEE international Conference on Computer and information Technology, CIT'06, 2006, p. 57.

[8] J.Yu, R.Buyya and K.Ramamohanrao, “Workflow Scheduling Algorithms for Grid Computing, Meta. for Sched. in Distri. Comp. Envi.”, SCI 146, springerlink.com, pp. 173–214, 2008.

[9] O. H. Ibarra and C. E. Kim, “Heuristic algorithms for scheduling independent tasks on non-identical processors”, J. Assoc. Compute. Mach. 24, 2 (Apr. 1977), 280_289.

[10] M. Dorigo, C. Blum, “Ant colony optimization theory: A survey “, Theoretical Computer Science 344 (2–3) (2005) 243–278.

[11] M.Dorigo, “Ant colony optimization”, http://www.aco-metaheuristic.org.

[12] M. Dorigo, L.M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem”, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.

[13] Dervis Karaboga, Bahriye Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, 12 February 2007 / Published online: 13 April 2007 © Springer Science+Business Media B.V. 2007.

[14] Dr.K.Vivekanandan, D.Ramyachitra, B.Anbu, “Artificial Bee Colony Algorithm For Grid Scheduling”, Journal of Convergence Information Technology, Volume6, Number7, July 2011.

[15] Jin Xu, Albert Y.S. Lam, and Victor O.K. Li, “Chemical Reaction Optimization for Task Scheduling in Grid Computing”, IEEE Transactions On Parallel And Distributed Systems.

[16] www.buyya.com

[17] www.gridcomputing.com

[18] www.gridforum.org