User Deadline Based Job Scheduling in Grid Computing

S.Gokul Dev, R.Lalith Kumar
Department of computer science and engineering, SNS College of engineering, Coimbatore-641035, Tamilnadu.

Abstract
Grid computing could be a style of distributed computing that co-ordinates and provides the ability of resource sharing over varied geographical locations. Resource planning in Grid computing is an advanced task owing to the heterogeneous and dynamic nature of the resources. Bacterial foraging has recently emerged as a world optimization algorithmic program for distributed optimization and management. This paper proposes the job of the bacterial foraging optimization technique for Grid resource planning. A completely unique bacterial foraging based mostly hyper-heuristic resource planning algorithmic program has been designed to effectively schedule the roles on the market resources during a Grid environment. The performance of the planned algorithmic program has been evaluated with the present common Meta heuristics based scheduling algorithms through the GridSim toolkit. The experimental results show that the planned algorithmic program outperforms the present algorithms by minimizing price and makespan of user applications submitted to the Grid. Although this algorithm performs well there are fewer disadvantages that gives less results and that can be improved by using deadline based hyper heuristic method which gives much better results than the Bacterial Foraging Optimization (BFO).

Keywords:
Grid computing, Job scheduling, Meta heuristic, Hyper heuristic, User deadline.

1. Introduction

Distributed computing utilizes a network of the many computers, each accomplishing some of associate degree overall task, to attain a computational result far more quickly than with one pc. Distributed computing will be defined in many different ways in which. In distributed computing the task is split up into smaller chunks and performed by the many computers owned by the final public. Distributed computing utilizes a network of the many computers, each accomplishing some of associate degree overall task, to attain a computational result far more quickly than with one pc. In addition to a higher level of computing power, distributed computing also permits many users to move and connect overtly.

Computers spend a lot of their time doing nothing. With the recent popularity of the Internet, distributed computing has become standard. The technology behind distributed computing is old, where it is usually referred to as parallel computing. Distributed computing takes identical principle a step further, transfer in computers owned by the final public over the globe. Standard comes include SETI, distributed.net, GIMPS and plenty of others. The key issue here is that you simply area unit victimization computing power that you simply don't own. These computers area unit owned and controlled by people, United Nations agency you would not essentially trust. This primer makes an attempt to indicate the basics of running a distributed computing project and avoiding the pitfalls. Grid computing could be a kind of distributed computing that involves coordinative and sharing computing, application, data, storage, or network resources across dynamic and geographically distributed organizations.

Grid technologies promise to tackle complex computational issues. Grid computing permits the virtualization of distributed computing and data resources such as processing, network information measure and storage capacity to form a single system image, granting users and applications seamless access to large IT capabilities. At its core, grid computing relies on associate degree open set of standards and protocols e.g., open Grid Services Architecture (OGSA) that change communication across heterogeneous, geographically distributed environments. When you deploy a grid, it will be to meet a collection of customer necessities. To raise match grid computing capabilities to those necessities, it is useful to keep in mind the reasons for victimization grid computing. This section describes the foremost vital capabilities of grid computing.

2. Types Of Grid Computing

Grid has been divided into a number of types, on the basis of their use:

2.1 Computational Grid

These grids provide secure access to large pool of shared processing power suitable for top outturn applications and computation intensive computing.

2.2 Data Grid

Information grids provide an infrastructure to support information storage, information discovery; information handling, information publication, and information manipulation of enormous volumes of information truly keep in varied heterogeneous databases and file systems.

Manuscript received March 5, 2015
Manuscript revised March 20, 2015

2.3 Collaboration Grid

With the arrival of internet, there has been an exaggerated demand for higher collaboration. Such advanced collaboration is possible by virtualization of the grid. As an example, persons from different firms during a virtual enterprise will work on different parts of a CAD project without even disclosing their proprietary technologies.

2.4 Network Grid

A Network Grid provides fault-tolerant and superior communication services. Each grid node works as an information router between 2 communication points, providing data-caching and alternative facilities to hurry up the communications between such points.

2.5 Utility Grid

This is often the ultimate kind of the Grid, during which not only information and computation cycles are shared however code or just regarding any resource is shared. The main services provided through utility grids are code and special equipment's. As an example, the applications will be run on one machine and all the users will send their information to be processed to that machine and receive the result back.

3. Related Works

[1] Describes the uses of gridsim and the functions in which they can be executed and most of the work in implementation is described in this package. [2][3] Are survey papers which help understand the concepts of grid computing and the scheduling strategies. It also describes the difference between grid and various other computing concepts. Various algorithms are compared and their results are graphed and the optimal solution is found. [4] Implements the conception of particle swarm optimization that suggests Meta heuristic search. It selects the optimum solution in an exceedingly single search based on the fitness value by repeating those values into classes like population test and global test. [5] Introduces an idea in genetic rule that is once more a Meta heuristic like choice, cross over, fitness and mutation. [6] Planned suffrage algorithm within which the roles are assigned to the resources and if the roles cannot be executed in that resource than that jobs suffers pretty much. Suffrage value calculated is employed to perform the task. [7] Planned an idea of bacterial foraging optimization which supports the fitness functions. The roles are allotted to the resources of which solely minimum fitness worth relies, deleting the utmost fitness value.

4. Existing Reference Algorithms

4.1 Particle Swarm Optimization (PSO)

Particle Swarm optimization (PSO) simulates the behaviors of bird flocking. PSO learned from the state of affairs and used it to resolve the optimization issues. In PSO, each single resolution may be a "bird" within the search house, called as "particle". All of particles have fitness values which square measure evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem house by following the current optimum particles. PSO is initialized with a bunch of random particles (solutions) then searches for optima by updating generations.

Each particle is updated by following 2 "best" values. The primary one is that the best resolution (fitness) it has achieved up to now. This worth is called p best. Another "best" worth that is half-track by the particle swarm optimizer is that the best worth, obtained up to now by any particle within the population. This best worth may be a global best and called g best. When a particle takes a part of the population as its topological neighbors, the simplest worth may be a native best and is called l best.

4.2 Genetic Algorithm (GA)

Genetic algorithmic rule primarily based meta-heuristics follow the Darwin's natural selection law i.e. only the fittest will survive. GA a population-based meta-heuristic, was created by John Netherlands and produces consequent generation with the techniques inspired by evolutionary biology, like inheritance, mutation, crossover, and selection. GA considers associate degrees as an organism; therefore better the standard of the answer higher is the survival chance, through crossover (also called recombination) and mutation. GA will escape from the local best to search for the global best. In this paper, we have a tendency to propose a genetic algorithmic rule for job planning to address the heterogeneousness of security mechanism in a procedure grid.

4.3 Suffrage Algorithm (SA)

Suffrage is that a task should be appointed to a certain resource and if it doesn't attend that resource, it will suffer the foremost. For each task, its franchise worth is defined because the difference between its best Minimum Completion Time (MCT) and its second-best MCT. Tasks with high franchise worth take precedence throughout programing. In franchise, the minimum and second minimum completion time for each job square measure found in first step. The difference between these 2 worth's is defined as franchise value. In second step, the task with maximum franchise worth is appointed to corresponding machine with minimum completion time.

4.4 Bacterial Foraging Optimization (BFO)

The Bacteria Foraging Optimization (BFO) algorithmic program was planned by Passino et al. It's a population-based numerical improvement algorithmic program supported hunting behavior of E. coli bacterium. E.coli bacterium features a system that directs its behavior in food hunting. In the hunting theory, the objective of the animal is to look and acquire nutrients during a fashion that energy intake per unit time (E/T) is maximized. Hunting is a process during which a group of bacterium moves in search of food during a region, they decide whether to enter into an attainable food region and seek for a new food region so on to get a high quality of nutrients. The Bacteria Foraging Optimization process consists of 4 main mechanisms: taxis, Swarming, replica and Elimination-dispersal event.

5. Architecture

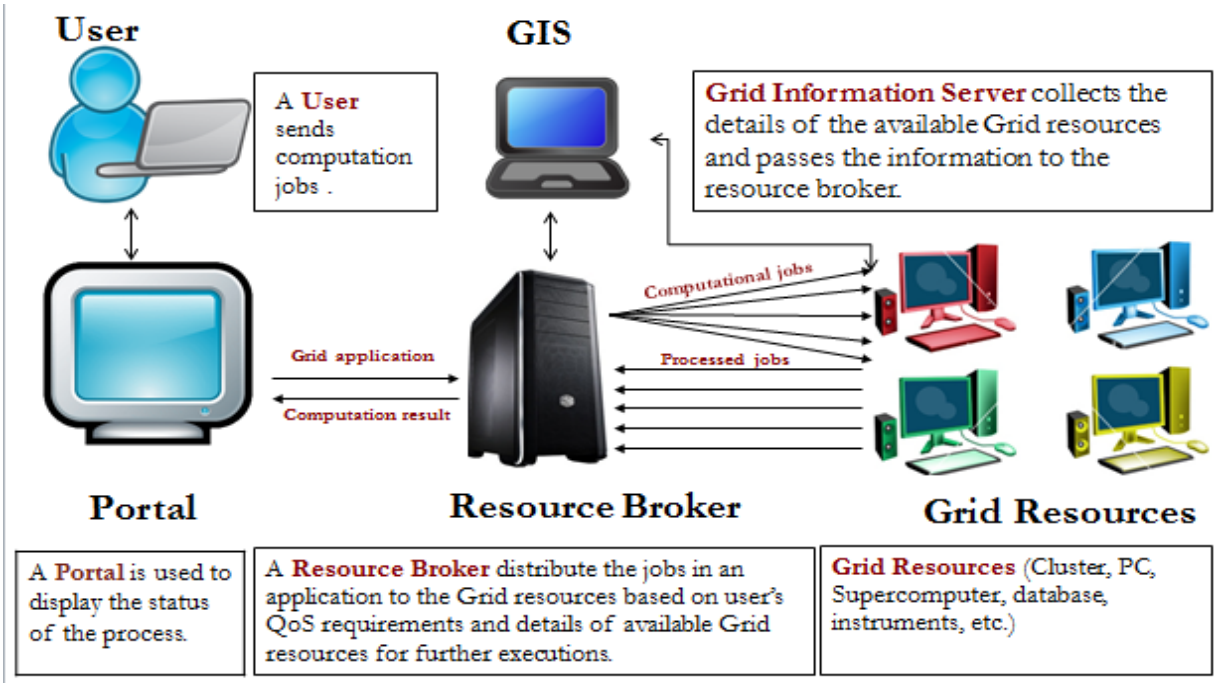

Fig.1 Architecture for grid scheduling

The programing in grid environment where the roles are allocated to the resources is shown within the fig. at first the user provides the input of variety of jobs and therefore the variety of resources to the portal which is used to show the standing of the programing process. In order of any system failure or shy resource the user is warned by the portal. The portal sends data to the resource broker which performs the programing process by obtaining the information regarding the resources current standing from Grid information Server (GIS) which periodically updates its information regarding the resources. Once the information's are gathered the resource broker sends the machine jobs to the resources the roles are processed and therefore the computed results are sent back to the broker who again sends the result to the portal from which the user gets the result.

Once job is processed the GIS is updated and therefore the resource broker allocates the resource to ensuing job. During this environment the hyper heuristic method is implemented in which the resource broker performs multiple gathering of data from the GIS and therefore the best one is chosen for the work.

6.Problems Of The Existing System

The problem of finding the most effective resources for a selected job is incredibly tedious particularly in meta heuristic technique where just one search is performed however this is overcomed in hyper heuristic technique where multiple searches are done that chooses the most effective resources for a job therefore improves the resource utilization. By Hyper heuristic technique, the resource utilization may be achieved nice however the time to settle on the most effective resource is drastically raised therefore it also allocates the job with a resource utilizing high information measure. This can raise the

price to execute a job and also raises the hardware execution time.

The hyper heuristic technique maybe smart for low number if inputs however when the jobs and resources are high, then the execution runs out of time and also grid system is usually utilized for a colossal processing of jobs and they don't persist with small works. The client that provides the request needs to do its work with minimum cost however the server that allocates its resources to the job needs to minimize the makespan. By using hyper heuristic technique, mostly the servers are benefited than the client. Therefore the hyper heuristic technique tries to beat the disadvantage by calculative the fitness price and therefore the higher values are eliminated and only] y the lower values are thought-about. On the other hand the time consumption cannot be reduced. This can be overcome by using the point in time primarily based hyper heuristic technique.

7. Proposed Algorithm

We have bestowed a rule referred to as point in time based mostly Hyper Heuristic methodology. In this rule a replacement idea referred to as the point in time is employed, wherever the user requests the utmost completion time for his job and servers apportion the roles to the resources specified, those jobs are executed among the period. There could also be cases once bound jobs cannot be executed among that time amount, thus those jobs can cross the limit fixed for a number of extents. This methodology can improve both resource utilization and cut back the C.P.U. execution time.

7.1 Procedural Steps

1. Each resource to be regular for application's execution has a unique id.
2. Jobs are executed independently.
3. Arrival of jobs for execution of application is random and jobs are placed in a queue of unexpected jobs.
4. The computing capacity/speed of the resources is measured in multiple instructions Per Second (MIPS) as per the standard Performance analysis Corporation (SPEC) benchmark.
5. The processing requirement of job is measured in Million instructions (MI).

6. Execution time for each job on a resource is obtained from the ETC matrix. No. of jobs ∗ no. of resources provides the dimensions of the matrix and its parts square measure defined as ETC (j, r). Rows of the ETC matrix demonstrate the calculable execution time for job on each resource and the columns demonstrate the calculable execution time for a selected resource. ETC (j, r) is that the expected execution time of job terrorist organization and the resource r.

8. Tools Used

The GridSim toolkit provides a comprehensive facility for simulation of various categories of heterogeneous resources, clients, applications, resource brokers and schedulers. It will be used to simulate application schedulers for single or multiple body domains distributed computing systems like clusters and grids. Application schedulers in grid surroundings referred to as resource brokers; perform resource discovery, choice and aggregation of a various set of distributed resources for a private user. That means, every user has his own private resource broker and thus it will be targeted to optimize for the wants and objectives of its owner, whereas schedulers, managing resources like clusters in a very single body domain, have complete control over the policy used for allocation of resources, which suggests that all users need to submit their jobs to the central hardware which can be targeted to perform global optimization like higher system utilization and overall user satisfaction betting on resource allocation policy or optimize for prime priority users.

9. Experimental Results And Comparison

9.1 Requirements

The parameters and their values required for the scheduling process are,

Table 1 Parameter requirements

PARAMETERS	VALUES
Gridlet size	100000
Gridlet length	42000000
Cost per second	1
No. of routers assigned	2
No. of GIS assigned	3
No. of grid users	7

The fitness function is defined by the amount of instructions transferred per second and the gridlets are the packages that contain all the information about the job and the resources. The processing rate of gridlets is measured in Million Instructions (MI). The number of resources and the jobs are user defined. The file size is not limited; they can be of any size.

The various existing algorithms are compared by defining the number of resources and jobs to be 5 and their results are tabulated.

9.2 Screen Shots of Bacterial Foraging Algorithm

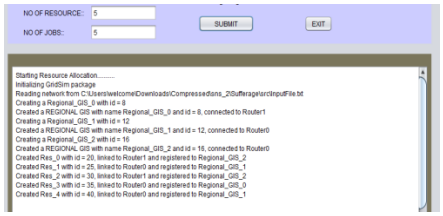

Fig. 2 Resource allocation and package initialliation

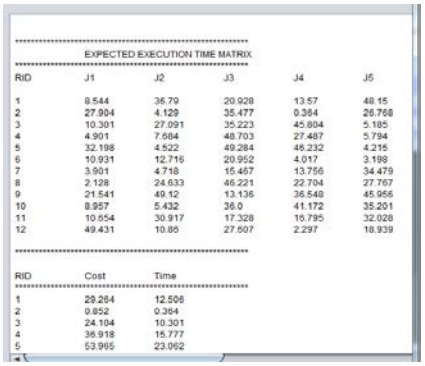

Fig. 3 Calculating expected execution time and best execution time

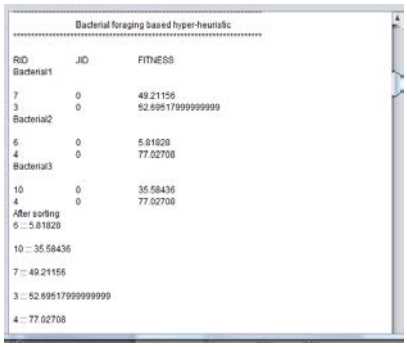

Fig. 4 Various searches performed with job 0 on resources and sorting those results

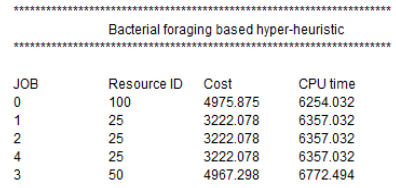

Fig.5 Calculating cost and CPU time

9.3 Screen shorts of User Deadline Algorithm

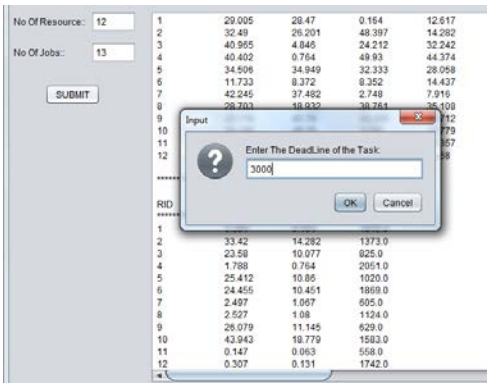

Fig.6 User Deadline value as input

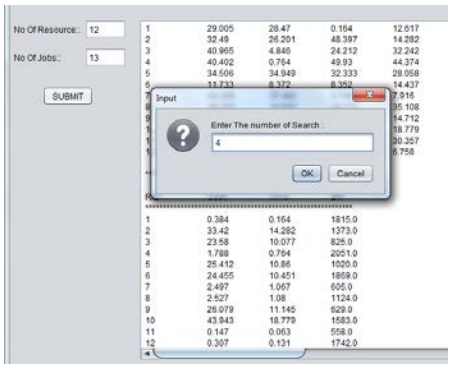

Fig.7 Number of Searchs to be done

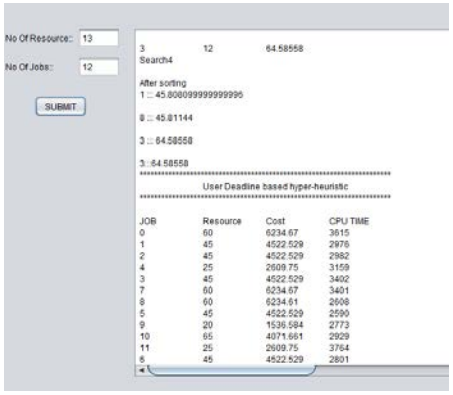

Fig.8 calculating cost and CPU time for User Deadline Algorithm

9.4 CPU Time

Table 2 Time Comparison Table

PARAMETER	GA	SA	PSO	BFOHH	UDHH
J_1	7264	7316	7178	6411	3408
J_2	7156	7316	7717	6423	2666

J_3	8228	7934	7717	7171	3119
J_4	8255	8188	7801	7181	3700
J_5	9264	8972	7655	6312	2977

9.5 Time Graph

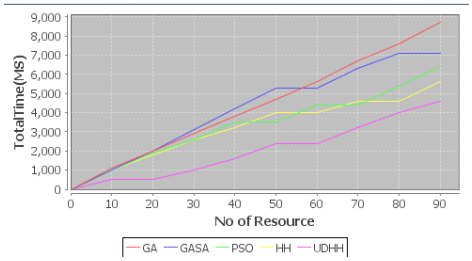

Fig.9 Makespan comparison result

9.6 Cost

Table 3 Cost Comparison Table

PARAMETER	GA	SA	PSO	BFOHH	UDHH
J_1	6585	203794	8928	5864	4080
J_2	6585	150819	7815	3818	6511
J_3	7123	181416	8155	5161	3429
J_4	4792	181416	5762	4827	5391
J_5	7895	169187	9282	6177	5427

9.7 Cost Graph

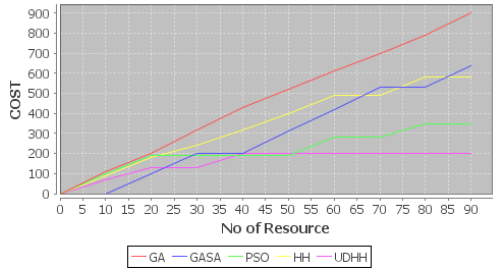

Fig.10 Cost comparison result

Thus from the various results derived we can conclude that hyper heuristic gives much better results than the other algorithms since it uses multiple search strategies. Even the cost is reduced drastically and the bandwidth occupied is also very less. They transfer bits at a very high speed when compared to the other algorithms. PSO ranks the second position in terms of both cost and time. The rest of the algorithms like the genetic algorithm (GA), suffrage algorithm (SA) are far behind and there is no proper resource allocation to the jobs.

Conclusion

Grid resource scheduling rule outperforms the hybrid heuristics in all the cases. The projected rule not solely minimizes cost however it conjointly minimizes the makespan. Thus from the results of various scheduling algorithms are compared and Bacterial algorithm is found to be the best one and this can be further enhanced by using user deadline hyper heuristic where the user specifies the time to complete the jobs. The algorithm for User Deadline based Hyper Heuristic gives the best result than the Bacterial Foraging Optimization where the users request to complete the process within their specified time. This can be verified from the table and graph generated.

References

[1] R.Buyya, M.Murshed,"Gridsim: a tool kit for the modeling and simulation of distributed resource management and scheduling for grid computing concurrency and computation", 2002.

[2] G.Jaspher W. Kathrine, Mansoor Ilaghi U, "job scheduling algorithms in grid computing survey", International journal of engineering research and technology (IJERT), ISSN: 2278-0181, vol. issue 7, September 2012.

[3] Rakesh Sharma, Vishnukant Soni, Manoj Kumar Mishra, Prachet Bhuyan, "A survey of job scheduling and resource management in grid computing", world academy of science, engineering and technology 40 2010.

[4] T.R. Srinivasan, R. Shanmugalakshmi, "Neural approach for resource selection with PSO for grid scheduling", International Journal of computer application volume no 11, September 2012.

[5] R. Kashyap, D.P. Vidyarthi, "Security driven scheduling model for computational grid using Genetic Algorithm", WCECS 2011, October 19-21, 2011, San Francisco, USA.

[6] Kamali gupta, manpreet singh, "Heuristic based task scheduling in grid", International journal of engineering and technology (IJET), ISSN: 0975-4024, vol. 4no 4 Aug-Sep 2012.

[7] Rajni, Inderveer Chana, "Bacterial foraging based Hyper-heuristic for resource scheduling in grid computing", future generation computer systems, Elsevier, 14 September 2012.

S. Gokuldev obtained his Bachelor's degree and Master's degree in Computer Science from Bharathiar University in the year 1999 and 2003, He obtained his Master of Philosophy in Computer Science from Bharathidasan University in 2004, Master of Engineering in Software Engineering from Anna University, Chennai in 2008 and currently pursuing PhD in Anna University, Chennai. He has around ten years of academic and teaching experience. Presently

working as an Associate Professor in Department of Computer Science & Engineering, SNS College of Engineering, Coimbatore. His areas of interest are Distributed and Grid Computing. He had published around 6 papers in international journals and more than 15 papers in few reputed International and National level conferences

Lalith Kumar is pursuing his bachelor's degree in computer science and engineering from SNS college of Engineering, affiliated to Anna University, India. His area of interest lies in grid and cloud computing and scheduling in grid computing. His research interest includes heuristic algorithms in grid scheduling.