HybridSGSA: SexualGA and Simulated Annealing based Hybrid Algorithm for Grid Scheduling

[1]Bhupinder Singh, [2]Seema Bawa

GJCST Computing Classification
F.2.2, H.3.4

***Abstract*-Scheduling jobs on computational grids is a compute intensive problem. Existing methods are unable to perform the required breakthrough in terms of time and cost. A Grid scheduler must use the available resources efficiently, while satisfying competing and mutually conflicting goals. The grid workload may consist of multiple jobs, with varying resource requirements and quality-of-service constraints. In this paper A hybrid algorithm based on SexualGA and simulated annealing is proposed, implemented and tested which tries to minimize makespan and cummulative delay in meeting user specified deadline time. Simulation results show that proposed algorithm performs better than other hybrid genetic simulated annealing algorithms proposed earlier.**

Keywords-Grid Scheduling, Computational Grids, Genetic Algorithms, Simulated Annealing, SexualGA.

I. Introduction

Computational grids are wide-area (Internet-scale) distributed environments that differ from conventional distributed computing by their focus on large-scale resource sharing, innovative applications, and highperformance orientation [1]. In grid architecture, four levels of management can be distinguished: fabric, connectivity, single resource, and collective multiple resources. The fabric layer typically constitutes computational resources, storage resources, network resources, and code repositories. The connectivity layer deals with easy and secure communication by providing single sign on, delegation, integration with local security solutions, and user-based trust relationships. The single resource layer is concerned with individual resources, and the two primary classes of resource layer protocols are information protocols and management protocols. The collective multiple resources layer provides directory services, co-allocation, scheduling, and brokering services, monitoring and diagnostics services, data replication services, grid-enabled programming systems, workload management systems and collaboration frameworks (problem solving environments) etc.

The responsibilities of a resource management system on the grid includes the "discovery of available resources for an application, mapping the resources to the application subject to some performance goals and scheduling policies and loading the application to the resource in accordance with the best available schedule." Scheduling, co-allocation and brokering services of resource management system are taken care by a special component of the grid resource management system known as scheduler / broker. A scheduler system provides the interface between a user and the grid resources. Scheduling of jobs on a grid or a cluster is the task of mapping jobs to the available compute-nodes which is NP-complete problem [2] and requires lot of optimization. A scheduler is designed to satisfy one or more of the following common objectives: (a) maximizing system throughput; (b) maximizing resource utilizations; (c) maximizing economic gains; and (d) minimizing the turnaround time for an application [3].

Simulated Annealing (SA) and Genetic Algorithms (GA) had been used in solving many NP-complete problems [4]-[7]. Since grid scheduling is also a compute intensive problem, hence it is a suitable candidate for applying evolutionary techniques. This paper presents a hybrid algorithm based on SexualGA[8] and SA to optimally schedule the jobs on computational grids. The Proposed algorithm tries to satisfy the constraints imposed by resource users and resource providers. Grid scheduler is the important part of Grid Resource Management System (GRMS), which gathers the information about the resources and chooses the best resource as per the job requirements. This is followed by the actual execution of the jobs. The problem of finding the best "job-resource" pair is a compute-intensive problem and need to be formulated mathematically to find the optimal solution.

To formulate the problem, we consider mapping a set of independent user jobs $\{J_1, J_2, J_3, \ldots, J_N\}$ to a set of heterogeneous processors / resources $\{P_1,P_2,P_3,\ldots,P_M\}$ (Though we can have any type of resource that can be shared on a computational grids, but for our simulations we have considered processors as a resource.) This mapping is done with an objective of minimizing the completion time by utilizing the resources effectively, and also minimizing the delay in meeting user specified deadlines. The speed of each resource is expressed in number of cycles per unit time, and the length of each job in number of cycles. Each job J has processing requirement C_j cycles and processor P has speed of V_i cycles/second. Any job J has to be processed byP, until completion.

This paper is organized as follows. Section 2 provides the details of different components of the proposed HybridSGSA algorithm. In section 3, experimental test bed and setup details are discussed. Section 4 provides the experimental results and finally section 5 provides the conclusion describing the usefulness of the proposed algorithm in scheduling resources on computational grids.

[1]*Bhupinder Singh, Module Lead, Design Tools Department, Ciena India Pvt. Ltd. Gurgaon (Haryana), India.*
(bsaini@ciena.com)
[2]*Seema Bawa, Professor, Computer Science and Engineering Department, Thapar University Patiala. (Punjab), India .(seema@thapar.edu)*

II. Proposed HybridSGSA Algorithm

A. Objective / Fitness Function

The Objective / fitness function is used to differentiate between high and low quality solutions. It is defined in terms of objectives of the problem in hand. For grid scheduling problem, the users have goal of satisfying the deadlines of jobs submitted by them whereas the resource providers like to minimize the makespan. The schedule S is evaluated on the basis of fitness function, which is defined as.

$$\text{Fitness, } F(S) = 1/(\omega * MS + \theta * T_{delay}) \quad (1)$$

Where MS denotes the makespan of the schedule and T_{delay} denotes the cumulative delay in meeting deadlines. ω and θ are the weights to prioritize the components of the fitness function as per the needs of the resource provider and job users. The components MS and T_{delay} of the fitness function are defined as follows:

Let $t_{end}(i)$ and $t_{dline}(i)$ be tunaround time and deadline time for i_{th} job,

then $t_{delay}(i) = t_{end}(i) - t_{dline}(i)$ (2)

{t_{delay} is delay in meeting deadline for i_{th} job, and $t_{delay} = 0$ if $t_{end} < t_{dline}$ }

and $T_{delay} = \Sigma\, t_{delay}(i)$ for $1<i<N$ (3)

Makespan of schedule S is defined as

$MS = Max(t_{processing}(k))$ for $1<k<M$ (4)

{where $t_{processing}(k)$ is the time in which processor P_k will complete processing jobs assigned to it.}

B. Initial Temperature and Annealing Schedule

Initial temperature, T, is set to a value such that the acceptance probability for the chromosomes of the population is very high i.e. close to 1.0. The annealing schedule is generated by multiplying the current temperature by a constant less than 0, which is known as the cooling rate,ϒ. The probability of accepting new solution in the reproduction (crossover and mutation) is implemented as A function of temperature, T. So initially the new off-springs are accepted with a high probability, where as, in the later stage their probability of being accepted is lowered as the temperature decreases. Temperature, T, is decreased at cooling rate,ϒ, in each iteration and is given by Equation 5.

$T_i = \Upsilon * T_{i-1}$ {where i is the iteration number} (5)

C. SexualGA

SexualGA [8] is an enhanced selection scheme that tries to mimic the sexual selection in human beings. In that context sexual selection is described as the concept of male vigor and female choice, meaning that male individuals try to spread their gene material as wide as possible and female individuals are more selective by choosing rather above average fit males to guarantee a high survival probability of their off-springs. So it seems to be preferable not to use identical selection mechanisms for male and female population members also in GAs. Inspired by this view of sexual selection, a new selection mechanism for Gas (SexualGA) is presented by introducing two different selection operators, one for the selection of male and one for the selection of female individuals.

D. Chromosome Representation

Schedule, S, of independent jobs is represented by A chromosome. Chromosome is basically a $N\cdot$_ _1 vector, where the position i $(0 < i < N)$ represents the job / task and the entry at position i $(s[i])$ is the processor / resource to which the job has been assigned. For example, in Fig.1, $S[1] = P_2$, means job J_1 is assigned to Processor P_2.

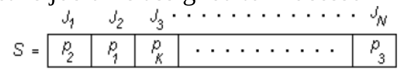

Fig.1. Chromosome representation

E. Crossover

Crossover operation is performed to generate new offsprings for the next generation. Two parents for the crossover are selected using the SexualGA based selection scheme and two-point crossover is performed. Using the vector representation for the chromosomes as shown in Fig.1, crossover operation is similar to exchanging the contents of two vectors for all positions before first crossover point and after the second crossover point. After the generation of new off-springs, change in the energy, δE, _and their probability of acceptance, P, is computed using the Equation 6 and Equation 7 respectively.

$$\delta E = f_{offspring} - f_{Parent}. \quad (6)$$

Where $f_{parent} = Max\,(f_{Parent1}, f_{Parent2})$

$f_{offspring}$ = Fitness of the new offspring

$$P = exp\,(-\delta E / T) \quad (7)$$

and $P = 1$, if $\delta E > 0$

After the finding the probability of acceptance, a uniform random number, ⌠, is generated between 0 and 1. New offspring is accepted if, $P >=$ ⌠, otherwise the new offspring is rejected.

F. Mutation

Mutation in the vector representation is considered as moving a job from one resource to another or swapping of two Jobs. For moving a job from one resource to another, two random numbers l and k are generated such that $S[l] = k$, where $0<l<N$ and $0<k<M$. After applying the mutation, the change in energy, δE, is computed using Equation 6 and probability of acceptance, P, is computed using Equation 7. If, $P >$ ⌠, (a uniform random number between 0 and 1), the new offspring is added to the next population, otherwise it is rejected.

G. Exit criteria

An exit criterion specifies when to stop further exploration, and accept the solution. Proposed algorithm uses two exit criteria.

i. Standard deviation of the fitness value of the individuals in the population and
ii. No change in the elite chromosomes after 50 iterations.

After the fitness function for all the chromosomes in the population is calculated, standard deviation is computed. If the standard deviation reaches a sufficiently low threshold value or there is no change in the elite chromosomes after 50 iterations, solution is said to converge.

H. Building New Population

New population is build from previous population through the crossover, mutation and elitism. This process is executed as follows. X_c% of the chromosomes, for the new population, is constructed using the crossover process as described earlier. X_m% of the chromosomes is constructed by applying the mutation operator. After the application of crossover and mutation, the rest of the chromosomes are taken out of the existing population by making them identical to existing population's best chromosomes, so as to make the size of new population equal to the existing one.

III. EXPERIMENTAL TEST BED AND SETUP DETAILS

Experimental test bed has been implemented to perform the experiments and to test and evaluate the proposed algorithm. This test bed uses the simulation model as used by Braun et. al. [9] in their study, so that results can be compared.

A. Grid Test Bed

Grid Test bed provides the actual grid like environment with a simple web based user interface for submission of jobs, registering and de-registering of resources and administrative tasks. Fig.2. shows the high level diagram of Grid Test Bed environment

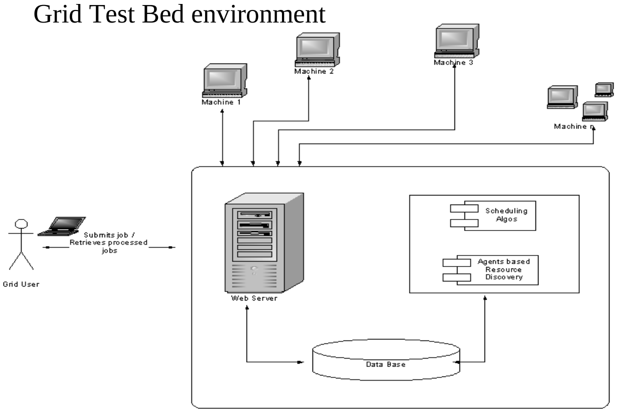

Fig.2. High level view of Grid Test Bed used for experiments

A web server listens to the user requests provided to the system by web interface. The information provided by the user is stored in the database. Users for the system can be grid users or resource owners. Grid users send the job requests for the jobs to be executed. Job characteristics are stored in the database. Grid users also specify the deadline time for the jobs. This is the time before which the job must be executed. Similarly resource owners use web based interface to register and de-register the resources. Whole information about the resources is also maintained in the database. Scheduling algorithm component provides the implementation of proposed HybridSGSA algorithm. Resource discovery is done using the ACO based resource discovery algorithm as discussed in [10]. Proposed algorithm work on the information present in the database about various jobs and resources and update the information as soon as the job's processing is finished.

B. Simulation Model

Grid test bed uses the simulation model as used in [9]. Since the results for the most of existing scheduling heuristics are already available, hence it is possible to compare the performance of proposed algorithm with the existing scheduling heuristics. Details of the simulation model are discussed in [9].

A completion time (CT) matrix is computed based for the job completion times on different resources. The prediction system in the grid test bed uses this completion time matrix to predict the execution time of a job on a resource. CT is an $N \cdot M$ matrix, which contains an entry for every jobresource pair. This entry specifies the completion time for A job on that particular resource.

In the actual grid systems state estimation is generally done based on the task profiling and analytical benchmarking. In this experimental Grid Test Bed, a pre-computed CT matrix is used, instead of actual task profiling.

A row in the CT matrix contains the completion time for A given job on each resource, where as a column consists of the completion time of every job on a given resource. Hence, for a job J_i and a resource R_j, an entry of the CT matrix contains the completion time of job J_i on resource R_j. Completion time matrix can be consistent, partially consistent or inconsistent that corresponds to the homogeneous, semi-heterogeneous, heterogeneous environments respectively. Since grid is highly dynamic and heterogeneous environment, hence only partially consistent and inconsistent matrices are considered for the experiments. Furthermore the heterogeneity can be considered as machine or task heterogeneity. CT matrix can be varied based on low machine or task heterogeneity and high machine or task heterogeneity. Taking into consideration that grid is highly heterogeneous environment; so low and high machine heterogeneity and high task heterogeneity CT matrix are considered in experiments.

IV. EXPERIMENTAL RESULTS

This section discusses the experimental details and results of proposed HybridSGSA algorithm and its comparison with existing algorithms. The objective function of proposed algorithm mainly focuses on minimizing two Quality of Service (QoS) parameters: 1. Makespan 2. Cummulative time delay in meeting user specified deadlines. Since existing results were available for minimizing the makespan only, so the results are compared with SA and GSA based algorithms from [9], by taking $\theta = 0$ in the fitness function $F(S)$. To support the algorithm proposed in this paper, we have performed several experiments taking different scenarios into consideration. Value of the different parameters used in experiments is given in Table 1.

Parameter Name	Value	Description
T	1	Initial Temperature
Υ	0.9	Cooling Rate
X_c	0.75	Cross Over Probability
X_m	0.01	Mutation Probability
ThresholdSD	1.0E-06	Threshold Standard Deviation

Table 1: Parameters and their values used in experiments

As per the simulation model, completion time (CT) matrix can be of three types: consistent, partially- consistent and inconsistent matrices. These can further be varied based on low and high task and machine heterogeneity. As grid is highly dynamic environment where any machines many leave or join at any time, so we have only considered partially consistent and inconsistent CT matrices with low and high machine heterogeneity and high task heterogeneity. Also we have considered $512 \cdot 16$ matrices as given in [9], i.e. $N = 512$ (number of jobs) and $M = 16$ (number of resources). Experiments have been performed for following type of CT matrices: 1) inconsistent matrix with high machine heterogeneity (IC_HM), 2) inconsistent matrix with low machine heterogeneity (IC_LM), 3) partially consistent matrix with high machine heterogeneity (PC_HM) and 4) partially consistent matrix with low machine heterogeneity (PC_LM). Comparison results are presented in graph as shown in Fig. 3. It is established from the experimental results that proposed HybridSGSA algorithm performs better as compared to existing SA and GSA based algorithms.

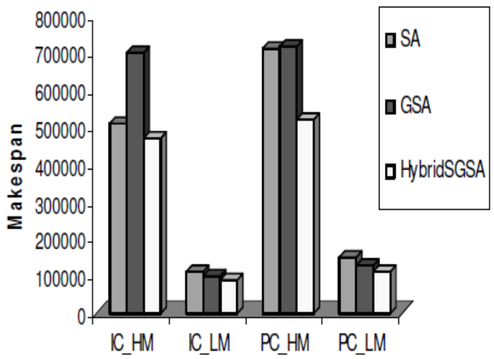

Fig 3: Comparison of SA, GSA and Proposed HybridSGSA

V. Conclusion

In this paper, we have looked at the problem of Grid scheduling and proposed, implemented and tested a new hybrid algorithm (HybridSGSA) based on SexualGA and SA. GA and SA both independently are valid approaches toward problem solving. GA can begin with a population of solutions in parallel, it has poor convergence properties. SA, on the other hand, has better convergence properties but cannot exploit parallelism. Furthermore SexualGA based selection scheme can better represent the concept of male vigor and female choice to more closely mimic the sexual selection in the human beings. Hence, it provides the advantage of not loosing the genetic diversity that result in better control over the selection pressure. The combined effect of all these factors makes proposed algorithm very effective for finding an optimal or near optimal solution quickly when search space is very large and dynamic as in case of computational grids.

VI. References

1) Foster, I. and Kesselman, C. (1998) Computational Grids, The Grid: Blueprint for a New Computing Infrastructure, Morgan- Kaufman, San Fransisco.
2) Abraham, A., Buyya R. and Nath, B.(2000) Nature's Heuristics for Scheduling Jobs on Computational Grids, The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), India.
3) Wright, D. (2001) Cheap Cycles from the Desktop to the Dedicated Cluster: Combining Opportunistic and Dedicated Scheduling with Condor, Proceeding of HPC Revolu-tion 01, Illinois.
4) Yao, X. (1995) A New Simulated Annealing Algorithm, International Journal of Computer Mathematics, Vol 56, Pp 161-168.
5) Goldberg, D. (1989) Genetic Algorithms in Search, Optimization and machine learning, Addison-Wesley Publishing Company, Inc.
6) Bawa, S. and Sharma, G.K. (2005) Search space pruning for faster test generation based on parallel and adaptive GA. Progress in VLSI Design & Test. Elite publishing, Pp 444-448.
7) Bawa, S. (2004) Parallel and Distributed Algorithms for VLSI Test Generation" Ph.D Thesis. Thapar University, Patiala , India.
8) Affenzeller. M. (2005) SexualGA: Gender-Specific Selection for Genetic Algorithms, Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics. In: International Institute of Informatics and Systemics, 4, Pp 76-81.
9) Braun, T. D., Siegel, H. J. and Beck, N. (2001) A Comparison of Eleven Static Heuristics for Mapping A Class of Independent Tasks onto Heterogeneous Distributed Computing Systems, Journal of Parallel and Distributed Computing.
10) Singh, B. and Bawa, S. Grid Resource Discovery using Natural Mobile Agents, International Journal of Computer Science and Network Security. Vol 8. No. 11, Pp 411-415