Volume 6, Issue 3, March 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Optimized Resource Selection and Scheduling for Grid

Dr. B. Radha
Asst.Professor,Department of Computer Science, Sree Saraswathi Thyagaraja College,
Pollachi, Tamilnadu, India

Abstract: Grid scheduling is the process of making scheduling decisions involving resources over many administrative domains. Resource selection is termed as resource discovery, assignment of application tasks to resources, and data staging/distribution. The Grid scheduler does not control the set of jobs submitted to it, or even know about jobs being sent to resources it considers using, so decisions that trade-off one job's access for another's cannot be made in a global sense. Makespan represents lapsed time from the first task's beginning to the end of the last scheduled task. If the Makespan is small, the utilization of the machines is high.Design space for Grid Scheduler's is usually rich. First, it is based on the objective function, a user's need to minimize or maximize a function. Second, it is also based on how job requirements, job performance models, and Grid resource models are used. The scheduler must choose carefully between different implementations of user authentication, allocation, and reservation. The objective in this study is to minimize overall job completion time or the application Makespan, the latter often being the performance feature in resource allocation study. Makespan minimization arranges tasks to level differences between each work phase's completion time. Techniques combined into heuristic approaches or in upper level multi-objective methodologies (i.e., meta-heuristics), are the lone methods to schedule when there is a high problem dimension and/or complexity. As optimization techniques, metaheuristic are stochastic algorithms trying to solve many hard optimization problems that are effective than traditional methods. So, recent scheduling problem research focused on such techniques.

Keywords: Grid Computing, Makespan, Genetic algorithm, Support vector Regression

I. INTRODUCTION

Grid is the latest evolution of developments like distributed computing, Web, peer-to-peer computing and virtualization technologies. A Grid is a structure with distributed heterogeneous resources that are offered to users. Users submit jobs which are efficiently processed using resources available in the Grid. Grid computing allows many-to-many sharing of both files and resources. Grid brings computing resources together similar to distributed computing. Unlike distributed computing, requiring operating homogeneity, Grids can be geographically distributed and of heterogeneous in nature.

Grid scheduling is the process of making scheduling decisions involving resources over many administrative domains including searching multiple administrative domains to use a single machine, scheduling single job to use multiple resources at single or multiple sites. Resource selection is the term for resource discovery, assignment of application tasks to resources, and data staging. Grid schedulers must make best-effort decisions and submit job to selected resources. Further, the Grid scheduler does not have control over the set of jobs submitted to it, or know about jobs being sent to resources it considers using, so decisions that trade-off one job's access for another's cannot be made in a global sense. This lack of control is the reason for many problems in Grid.

The operations management has to optimize the resource utilization. Techniques combined into heuristic approaches (Xhafa et al 2010) or in upper level multi-objective methodologies (i.e., meta-heuristics) (Xhafa and Abraham 2008), are efficient methods to schedule when there is a high problem dimension or complexity. In optimization techniques, meta-heuristics are stochastic algorithms trying to solve hard optimization problems which are effective than traditional methods. In recent years, meta-heuristics attracted attention from the hard optimization community as a powerful tool, as it demonstrated promising results from both experiments and practice in engineering areas. Hence current research focused on such techniques (Kołodziej and Xhafa 2012, Gonzalez, et al 2009 and Khan, 2012).

II. AIM OF THE PRESENT INVESTIGATION

This research examines meta-heuristic approaches for scheduling issues as it is a NP complete problem. This work also examines techniques for efficient resource selection from the list of available resources. The aim of resource selection and scheduling is to reduce makespan which is the completion time of the last job. Low makespan improves resource utilization, reduces bandwidth utilization and improves Quality of Service (Fibich et al 2005).The objectives of the work carried out are listed:

1. Performance Evaluation of existing Scheduling Algorithms for Grid Systems.
2. Propose a RBF Kernel Optimized Support Vector Regression Algorithm for Resource selection.
3. Propose a Hybrid Optimization for Grid Scheduling using Genetic Algorithm with Local Search.
4. Propose a Hybrid Evolutionary Algorithm for Grid Scheduling based on Genetic Algorithm and Ant Colony Optimization.

Performance Evaluation of Grid Scheduling Algorithms

Evaluating the resource sharing effects on application performance is challenging in shared environments due to task scheduling where a remote task competes with local jobs, or other remote tasks, for resources. A task's completion time is determined by its workload and resource availability. Deploying a distributed task in a Grid computing environment after scheduling decisions still requires integrated system solutions (Wu et al 2007). Deployment involves job submission and resource monitoring. It is hard in a shared environment, where computing crosses multiple administration domains that use varied resource management policies. Task scheduling needs integrated solutions of scheduling algorithms, performance prediction and system development. This study proposes to investigate the performance of dynamic scheduling algorithm to execute different tasks.

Methodology: Random Scheduling, a simple algorithm statistically guarantees variable fraction of processor time for every process. Then it generates a random number which corresponds to a specific job. Though there is no guarantee that processes are treated equally, scheduling events frequency in a preemptive multitasking system ensures that it is really close to doing so. Random Scheduling advantages include fine grained priorities and statistical fairness.

The Dynamic Level Scheduling (DLS) algorithm (Shi and Dongarra2006) uses the Dynamic Level (DL) attribute, the difference between a node's static level and its earliest execution start time. At every scheduling step, a pair of node processor providing the highest DL value is selected; a process which is similar to that used by Earliest Time First (ETF) algorithm. But, there is a subtle difference between ETF and DLS: ETF algorithm schedules the node with minimum early execution start time with the static level being used to break ties. But the DLS algorithm in contrast, schedules nodes with the descending order static levels when the process first starts, but schedules nodes in EEST ascending order of EEST when the process nears the end.

Result: Simulations were carried out in Simgrid framework. The resources are located at different locations connected using switches. The resources are scheduled using Random and Dynamic scheduling algorithm. Simulations were conducted using 8 and 20 resources. The numbers of jobs were varied from 100-1000 and the makespan is computed.

It can be observed DLS scheduling decreases the average makespan by 2.9% compared to random scheduling for jobs in the range of 100-500. The lowest makespan is obtained when the number of jobs is 300 with decrease in makespan by 9.09% compared to random scheduling. However, it is observed that when the number of jobs increases, the performance of DLS may degrade as seen when the number of tasks is 1000. This is due to job failures that have occurred when the job is being executed.

III. RESOURCESELECTION USING REGRESSION TECHNIQUES

Grid Management Information Server (GMIS) framework provides services for discovery, monitoring and brokerage of resources in a grid network. Discovering resources is an automated process which keeps track of addition and deletion of resources. To gather the information about resources, communicator server and communicator agents are used. Resources details can be stored in a relational database on GMIS. Monitoring service monitors the resources on grid level, cluster level and node level. Using this service, grid resources are visually represented as icons on a viewer with resource status. GMIS framework allows viewing of summary information of the individual node and summary view of cluster which is comparable and analyzable. Brokerage service allocates the best possible resource or a combination of resources for a job execution based on set of attributes given by grid users. This work investigates the efficacy of Data Mining techniques for resource selection in improving the Quality of Service of the grid system.

Methodology: The Support Vector Regression(SVR) is a popular tool for function estimation problems, and it has been widely used on many real applications in the past decade, for example, time series prediction, signal processing and neural decoding (Gunter and Zhu 2005).

For a training data consisting of N pairs $(x_1, y_1)......(x_N, y_N)$ where x denotes the input patterns and y is target variable. In SVR with ε -insensitive loss function, the goal is to find a function f(x) that has at most ε -deviation from the actually obtained targets yi for all the training data, and at the same time, is as flat as possible (Jun and Oh 2007). The ε - insensitive loss function is defined as:

$$M(y, f(x,\alpha)) = L(|\,y - f(x,\alpha)\,|_z) \tag{3.1}$$

This is denoted by

$$|\,y - f(x,\alpha)\,|_z = \begin{cases} 0, \text{ if } |y - f(x,\alpha)| \le \varepsilon, \\ |\,y - f(x,\alpha)\,| - \varepsilon, \text{ otherwise} \end{cases} \tag{3.2}$$

where α is a positive constant. The loss is equal to 0 if the discrepancy between the predicted and the observed values is less than ε. The case of linear function f is described in the following.

$$f(x) = < w, x > + b \tag{3.3}$$

where, <· ,· >denotes the dot product. For SVR, the Euclidean norm $\|w\|^2$ is minimized. Formally this problem can be written as a convex optimization problem. Slack variables ξ_i, ξ_i^* are introduced to copy with otherwise infeasible constraints of the optimization problem (Wu and Chang 2012).

$$\min imize\frac{1}{2}\|w\|^2+C\sum(\xi_i+\xi_i^*)$$
$$subject\ \text{to}\begin{cases} y_i-<w,x_i>-b\le\varepsilon+\xi_i \\ <w,x_i>+b\le\varepsilon+\xi_i^* \\ \xi_i,\xi_i^*\ge 0\end{cases} \tag{3.4}$$

The constant C >0 determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated. Using a standard dualization method utilizing Lagrange multipliers, the parameters are determined.

Proposed Improvement in SVR regression by Optimization of the RBF kernel

Parameter selection is crucial since SVR algorithm is very sensitive to the adequate choice of parameter values and affects the accuracy of prediction. In SVR, parameters regularization constant C, and coefficients of SVR kernel itself, e.g. kernel width σ in RBF impact prediction performance. The regularization parameter C determines the trade-off cost between minimizing the training error and minimizing model complexity, which will reduce the generalization capability when it was set too small or excessive. And σ reflects correlation of the support vector, which also determines both the generalization capability and the prediction accuracy.

Selecting cost parameter is NP hard. In this work, it is proposed to use a novel Hybrid Genetic Algorithm (GA) to find the ideal C parameter. The proposed algorithm incorporates a local search algorithm for faster convergence. Hybridization can be an extremely effective way of improving the performance and effectiveness of GA. The most common form of hybridization is to couple GAs with local search techniques and to incorporate domain-specific knowledge into the search process. In this work, it is proposed to incorporate a local search operator into the GA by applying the operator to each member of the population after every generation. This hybridization is carried out in order to produce stronger results than the individual approaches.

The proposed improvement uses notion of state space neighbourhood with a new objective function and is described by

State space S: the set of possible states that can be reached during the search.

Neighbourhood N(s): the set of states, neighbours which can be reached from the state, s in one step.

Objective function f(s): A value that represents the quality of the state, s. The optimal value of the function is achieved when s is a solution (Ekelin and Olovsson 1996).

Performance Analysis: To test the proposed technique, the specifications of the SPEC dataset were used for the resources.Simulations were conducted with the 8 and 20 resources selected using the proposed technique. The jobs used in the previous experimental setup were used and the makespan computed.

From the results it is observed that the average decrease in makespan using DLS compared to random scheduling is 2.99%. The proposed resource selection technique decreased the overall average makespan by 1.65% (across both the technique) compared to the average makespan obtained without resource selection. Similarly, when the simulation was carried out for 600-1000 jobs DLS based scheduling decreased the makespan by 1.35% compared to random scheduling and because of the proposed resource selection technique the average makespan decreased by 2.08%.

IV. AN HYBRID METAHEURISTIC SCHEDULING ALGORITHM

This study considers scheduling as a single objective optimization problem, where makespan is minimized. Makespan, the finishing time of latest task, is defined as

$$\min_S \max\left\{F_j : j\in jobs\right\} \tag{4.1}$$

where Fj denotes the finishing time of job j in schedule S. It is advantageous to define a machine's completion time for a given schedule, as this indicates when the machine will finalize processing of jobs assigned earlier as well as those already planned. Makespan represents lapsed time from start of first task to end of last scheduled task. Makespan reduction arranges tasks to level differences between work phase completion time. This work proposed a hybrid GA with local search incorporating ACO for grid scheduling to optimize makespan (Kousalya and Balasubramanie 2009).

Methodology: Local search is a metaheuristic method for solving computationally hard optimization problems. Local search can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate solutions. Local search algorithms move from solution to solution in the space of candidate solutions by applying local changes, until a solution deemed optimal is found or a time bound is elapsed. Local search is the basis of many methods for combinatorial optimization problems.

Local search is a simple iterative method for finding good approximate solutions. The idea is that like of trial and error method. Consider an instance of a combinatorial optimization problem defined by the pair (S, g) where S is the set of all feasible solutions and g is the objective function that maps each element s in S to a real value. The goal is to find the solution s in S that minimizes the objective function g. The problem is stated as:

$$\min g(s),\ s\in S \tag{4.2}$$

A neighbourhood N for the problem instance (S, g) is given by a mapping from S to its powerset:

$$N: S\rightarrow 2^s \tag{4.3}$$

N(s) is called the neighbourhood of s and contains all the solutions that can be reached from s by a single move. The meaning of the move here is that of an operator which transforms one solution to another with small modifications. A solution x is called a local minimum of g with respect to the neighborhood N iff:

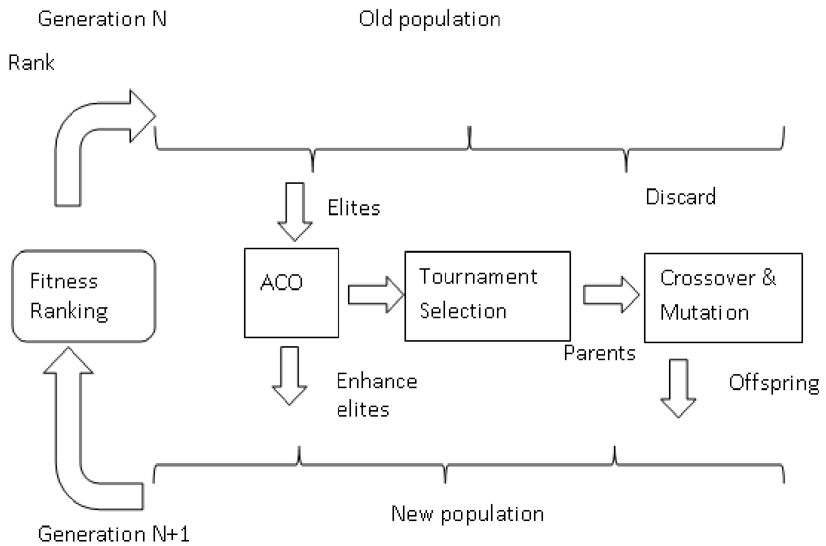
$g(x) \leq g(y),\ y \in N(x)$ (4.4)

Local search is the procedure of minimizing the cost function g in a number of successive steps in each of which the current solution x is being replaced by a solution y such that:

$g(y) < g(x),\ y \in N(x)$ (4.5)

Local search begins with an arbitrary solution and ends up in a local minimum. There are many different ways to conduct local search (Voudouris and Tsang 2010). Generally, the computational complexity of a local search procedure depends on the size of the neighborhood set and also the time needed to evaluate a move. The larger the neighborhood set, the longer the time needed to search it, the better the local minima.

The architecture of the proposed hybrid algorithm based on Genetic Algorithm and Ant Colony Optimization is shown in figure 4.1.



1	****	Ps/2	Ps/2+1	****	Ps
Ind.?	****	Ind.?	Ind.?	****	Ind.?

Figure 4.1 Block diagram of the proposed architecture

Both GA and ACO in the proposed algorithm work with the same population. Initially, Ps individuals who form the population are generated randomly. They can be considered chromosomes in GA, or as solutions in ACO. Learning parameters, like in ACO, and mutation probability should also be assigned earlier. After initialization, new next generation individuals are created by enhancement, crossover, and mutation operations.

Result Analysis: Simulations were undertaken using proposed hybrid grid scheduling algorithm. Table 4.1 is the Makespan 20 Top ranked resources.

TABLE 4.1 MAKESPAN WITH 20 TOP RANKED RESOURCES

Number of jobs	GA	Hybrid GA	GA with SVR resource selection	Hybrid GA with SVR resource selection	GA with optimized SVR kernel for resource selection	Hybrid GA with optimized SVR kernel for resource selection
100	34.8	35.04	34.27	34.43	33.63	33.87
200	73.59	75.98	72.38	74.61	71.01	73.36
300	108.18	119.18	106.37	117.02	104.33	115.04
400	142.41	144.51	139.85	141.35	137.16	138.71
500	174.4	178.03	171.23	174.02	167.89	170.73
600	211.79	216.94	207.85	211.89	203.73	207.8
700	247.13	251.5	242.46	245.36	237.44	240.5
800	283.35	284.92	277.63	277.91	271.27	272.21
900	312.87	318.1	306.39	309.92	299.34	303.47
1000	350.29	345.43	342.9	336.52	334.98	329.42

The makespan of hybrid GA with optimized SVR kernel for resource selection decreases by 2.67% than GA, by 3.34% than hybrid GA, by 1.17% than GA with SVR resource selection, by 1.63% than hybrid GA with SVR resource selection, by 0.71% than GA with optimized SVR kernel for resource selection with 100 jobs. The makespan of hybrid GA with optimized SVR kernel for resource selection decreases by 5.96% than GA, by 4.63% than hybrid GA, by 3.93% than GA with SVR resource selection, by 2.11% than hybrid GA with SVR resource selection, by 1.66% than GA with optimized SVR kernel for resource selection with 1000 jobs.

V. CONCLUSION AND FUTURE WORK

Conclusion

Software components make up Grid schedulers to compute task mapping to Grid resources through multiple criteria and Grid environment configurations. Scheduling's goals include achieving high performance computing and high throughput. The former is through execution time reduction for each job. It is usually utilized for parallel processing. The performance of a grid under Random scheduling and Dynamic scheduling was investigated.

In the second stage, the efficacy of SVR to select the optimal resources for the scheduled jobs is evaluated. Performance evaluation of resource selection using SVR with proposed optimization. Simulation results demonstrated the efficiency of resource selection for the grid scheduling problem.

With network technology development, grid computing, solving large scale complex problems becomes a focus technology. Task scheduling is challenging in grid computing which is a NP-Complete problem. Conventional methods for optimizations are deterministic, fast providing exact answers but often being stuck in local optima. So, another approach is required when conventional procedures are not applicable to modern heuristic as they are general purpose optimization algorithms. Simulation results prove the proposed GA's effectiveness when combined with local search to schedule Grids.

To avoid premature GA convergence, due to interference from mutation and genetic drift, sharing and crowding decrease the amount of duplicate schemata in the population. A hybrid GA incorporating ACO for grid scheduling was proposed. The approach aimed to generate an optimal schedule to complete jobs within minimum makespan.

Future Work

The proposed techniques can be investigated for hierarchically constructed grid and can also be investigated for scalable grid architecture. Cloud is emerging as a popular mechanism for high speed, large data computation. Further investigations can be carried out using the proposed algorithm in cloud.

REFERENCES

[1] Gunter, L., & Zhu, J. (2005). Computing the solution path for the regularized support vector regression. Ann Arbor, 1001, 48109.

[2] Voudouris, C., Tsang, E. P., & Alsheddy, A. (2010). Guided local search (pp. 321-361). Springer US.

[3] Xhafa, F., & Abraham, A. (2008). Meta-heuristics for grid scheduling problems. Metaheuristics for Scheduling in Distributed Computing Environments, 1-37.

[4] Kołodziej, J., & Xhafa, F. (2012). Integration of task abortion and security requirements in GA-based meta-heuristics for independent batch grid scheduling. Computers & Mathematics with Applications, 63(2), 350-364.

[5] Xhafa, F., Gonzalez, J., Dahal, K., & Abraham, A. (2009). A GA (TS) hybrid algorithm for scheduling in computational grids. Hybrid Artificial Intelligence Systems, 285-292.

[6] Khan, S. U. (2012). Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment. Information Sciences.

[7] Xhafa, F., & Abraham, A. (2010). Computational models and heuristic methods for Grid scheduling problems. Future Generation Computer Systems, 26(4), 608-621.

[8] Jun, S. H., & Oh, K. W. (2007). An Evolutionary Statistical Learning Theory.International Journal of Computational Intelligence, 3(3).

[9] Wu, J. L., & Chang, P. C. (2012). A trend-based segmentation method and the support vector regression for financial time series forecasting. Mathematical Problems in Engineering, 2012.

[10] Fibich, P., Matyska, L., & Rudová, H. (2005). Model of grid scheduling problem.Exploring Planning and Scheduling for Web Services, Grid and Autonomic Computing, 17-24.

[11] K. Kousalya and P. Balasubramanie,"To Improve Ant Algorithm's Grid Scheduling Using Local Search". (HTTP://WWW.IJCC.US), VOL. 7, NO. 4, DECEMBER 2009.

[12] Hu, C., Wu, M., Liu, G., & Xie, W. (2007, August). QoS scheduling algorithm based on hybrid particle swarm optimization strategy for grid workflow. In Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on (pp. 330-337). IEEE.

[13] Shi, Z., & Dongarra, J. J. (2006). Scheduling workflow applications on processors with different capabilities. Future generation computer systems, 22(6), 665-675.

AUTHOR BIOGRAPHY

Dr. B. Radha received B.Sc Degree in Chemistry in 2001, MCA Degree in Computer Applications in 2004. She is currently working as a Assistant Professor in the Department of Computer Applications, Sri Ramakrishna Engineering College, Coimbatore She received her Ph.D from Anna University in the year 2015 .Her research interest includes Resource allocation in Grid Computing and Simulation. She has published two technical papers in National conferences. one paper in the International journal and International Conferences she is a life time member of ISTE.

