International Journal of Soft Computing 2 (5): 606-611, 2007
ISSN: 1816-9503

Scheduling of Scientific Workflows Using Simulated Annealing Algorithm for Computational Grids

S. Benedict and V. Vasudevan
Software Technologies Group, TIFAC Core in Network Engineering, Anna University, Anna, India

Abstract: The Grid has emerged as an attractive platform to tackle various large scale problems, especially in science and engineering. The objective of this study is to generate an optimal schedule and evaluate heuristic algorithm namely Simulated Annealing (SA) with other scheduling algorithms for data intensive grid applications that maximizes Job completion ratio and minimizes lateness in job completion. The 2 objectives are aimed at achieving simultaneously by the scheduling mechanism. In addition, to create neighborhoods for simulated annealing, 3 Perturbation schemes, viz. pair wise exchange, insertion and random insertion perturbation are used. The results pinpoints that the proposed Simulated Annealing search mechanism performs well compared to other scheduling mechanisms on considering the workflow execution time within the deadline and lateness of individual workflows.

Key words: Grid computing, genetic algorithm, scheduling, simulated annealing

INTRODUCTION

This study addresses the problem of making sequencing and scheduling decisions for Grid scheduling problem environment. Grid is an important concept toward a computational infrastructure for resource sharing in the Internet is Grid computing. Grids are considered to provide multiple advantages to their participants (Berman *et al*., 2003). This aggregates the geographically distributed heterogeneous resources to solve large scale problems in science and engineering (Cannataro *et al*., 2002; Frey *et al*., 2002). The Grid sites accept tasks submitted from intra and inter sites. The availability of resources in Grid sites is still limited (Aloisio and Taha, 2002), although the computers have shown the growth in all extends, to run applications such as the mass spectrometric analysis in 3D rendering computations. Many applications such as data mining jobs and high energy physics experiments currently being developed by European Nuclear Research Centre (CERN) (Chia-Hung Chien *et al*., 2005; Frey *et al*., 2002), has a very large data's (Cannataro *et al*., 2002) to compute and thus considers maximization of job completion and minimization of lateness as an objective.

Grid in reality becomes a major problem owing to the challenges in security issues, protocol design, access control (Mark Baker *et al*., 2002; Shanshan Song *et al*., 2005), implementation models (Manish Parashar *et al*., 2005), resource management and grid scheduling. Following are the grid scheduling challenges (Alain Andrieux *et al*., 2004):

- Unpredictable challenges in Grid resources (i.e., Availability, accessibility and so on).
- Need for a multiple resource types for completing a job.

Advanced reservations for scheduling the workflows via a central scheduler were the traditional system. The vulnerability arises when the scheduler undergoes problems such as overloading and power failure. The overloading and the scheduler failure problem are overridden by a 2 level scheduling scheme where the first level is used for frequent small jobs and second level for large jobs is often used. The market oriented approach (Chia-Hung Chien *et al*., 2005) algorithm succeeded in the distributed scheduling of workflows, but could not appease completion of more workflows within the deadline. The success ratio of the workflows allotted for mapping the Grid sites is 30% (Chia-Hung Chien *et al*., 2005) when 30 workflows are scheduled at a time.

The problem vividly pinpoints that the distributed resource discovery and algorithms suit appropriately when the scheduling is for small jobs. On considering the large periodical jobs that are scheduled in advance, evolutionary techniques, namely, Genetic algorithm provide optimal results. The Genetic algorithm is efficient on considering a single objective.

Corresponding Author: Shajulin Benedict, Software Technologies Group, TIFAC Core in Network Engineering, Anna University, Anna, India

In this study, it is attempted to employ simulated annealing approach to implement in the Grid scheduler to solve the scheduling problem with dual objective for Grids. The evaluation of the SA with other algorithms such as First Come First Serve (FCFS) and Earliest Deadline First (EDF) are made.

Recent years have seen many efforts focused on the efficient utilization of Grid resources by Grid scheduling to appease the customer and provider satisfactions, such as Condor project (Frey *et al*., 2002), which being developed for about fifteen years, aim for high throughput. Two-level scheduling for large periodical jobs is discussed using Genetic algorithm in (Vincenzo and Marco *et al*., 2002). Here an optimal solution is selected using the parameters such as job characteristics, Grid environment characteristics and data distribution characteristics.

In Marek *et al.* (2005), scheduling two scientific workflows with the algorithm namely Heterogeneous Earliest Finish Time (HEFT) is compared with other myopic and genetic algorithms. HEFT schedules workflows by creating an order for the tasks. The above approaches work well when the number of the workflows is small.

The works related to (Halpern and Moses, 1990; Rodrigo Real *et al*., 2003), considers scheduling using learning algorithm based approach, requiring more database for efficient scheduling.

Our approach, using SA based scheduling, is efficient for scheduling.

PROBLEM DESCRIPTION

Figure 1 shows the Inter-Grid architecture described with a three node Grid environment example. The goal of the Inter-Grid Scheduler architecture is to find the allocation sequence of workflows on each Grid site.

This architecture is studied in the Software Technologies Lab of TIFAC Core in Network Engineering. Four major entities are involved in this architecture:

- The Grid users submit their request for job completion in the form of workflows to the local grid managers.
- All the tasks should be received by the Grid managers and the decision for the scheduling is made on deploying the request to the Intra-Grid schedulers.
- The Intra-Grid schedulers have the updated information of the grid resources that are idle during time‘t’. This information is frequently updated. The smaller jobs can be scheduled within the deadline by the Intra-Grid schedulers where the scheduling is often dynamic.

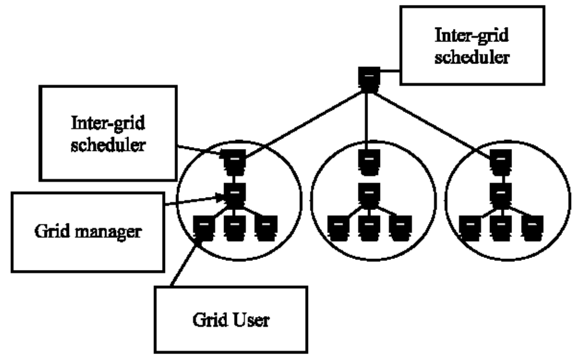

Fig. 1: Inter-grid scheduler architecture

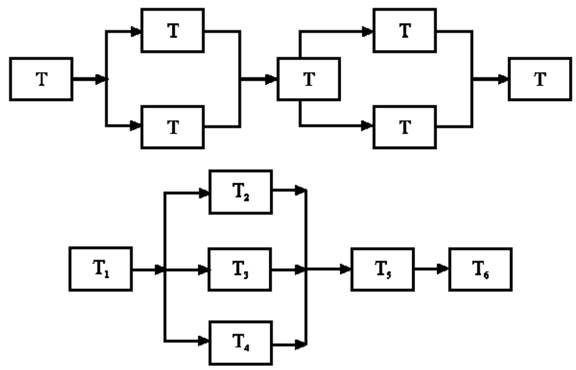

Fig. 2: DAG for W1 and W5

- For data intensive applications where the jobs are larger require the necessity of the resources world wide. At that moment, there is a necessity of Inter-Grid schedulers which is static often.

The workflow allocation strategy in a Grid environment differs as in Hamscher *et al.*(2000). The goal of the Inter-Grid scheduler is to receive the request from different intra-grid schedulers and make an optimistic scheduling such that it accommodates many workflows completing within its deadline.

The simulator developed is to show the performance of the SA compared to the other traditional algorithms. The FCFS map tasks to the idle Grid sites based on first task arrival to serve first. The EDF algorithm executes the tasks whose absolute deadline is the earliest. Hence it estimates the execution deadline of the individual workflow for any standalone system and schedules such that the workflows that require greater completion time is served first. In EDF, the task priorities are not fixed but change depending on the closeness of their absolute deadline.

The setting of the experiment consists of workflows with following assumptions:

Table 1: Experimental work flows

S.N	Work flow	Tasks/(duration)	Grid sites
1	W1	$T_1(2),T_2(4),T_3(3),T_4(1),T_5(3),T_6(2),T_7(3)$.	Any
2	W2	$T_1(3),T_2(2),T_3(3),T_4(4),T_5(3),T_6(1)$.	Any
3	W3	$T_1(2),T_2(4),T_3(1),T_4(2),T_5(3),T_6(4),T_7(5)$.	Any
4	W4	$T_1(2),T_2(3),T_3(1),T_4(4),T_5(5),T_6(1),T_7(2), T_8(3)$.	Any
5	W5	$T_1(1),T_2(3),T_3(2),T_4(4),T_5(5),T_6(6)$.	Any
6	W6	$T_1(1),T_2(3),T_3(1),T_4(2),T_5(4),T_6(2)$.	Any
7	W7	$T_1(1),T_2(4),T_3(2),T_4(5),T_5(3),T_6(2)$.	Any
8	W8	$T_1(1),T_2(3),T_3(2),T_4(3),T_5(1),T_6(1),T_7(2), T_8(3)$.	Any
9	W9	$T_1(3),T_2(3),T_3(4),T_4(6),T_5(2),T_6(4),T_7(2),T_8(1),T_9(5)$.	Any
10	W10	$T_1(4),T_2(5),T_3(4),T_4(4),T_5(5),T_6(1),T_7(2)$.	Any

- Each workflow received in the Inter-Grid scheduler consists of a set of Tasks T_1, T_2, T_3 and so on. The task in each workflow is a Directed Acyclic Graph (DAG) model.
- The output from a task can be transferred to other tasks as per the DAG graph model and its transmission time is negligible.
- At any time, a task can be executed only on a Grid site which is reported to the Inter-Grid scheduler as idle via Intra-Grid scheduler.

For experimental purpose the following DAG workflows as given in Table 1 are considered.

The duration for each task in any workflow and the required grid resources are given in the Table 1. The tasks taken for experiment have its predecessors and successors, such as T_1 follows T_2 or T_2, T_3 are parallel computation once when the task T1 is executed.

The workflow model for W1, W5 are shown in Fig. 2:

THE SIMULATED ANNEALING (SA)

In recent years, much attention has been devoted to heuristics that are applicable in particular for solving combinational optimization problems. In this section, the SA heuristic to solve grid scheduling in the m-grid sites problem is described. For more literatures it is advisory to refer to Van Laarboven and Aarts (1987).

The SA algorithm: The SA approach is based on the ideas from statistical mechanics and motivated by an analogy to the behavior of a heat bath. It can be viewed simply as an enhanced version of the familiar techniques of local optimization or iterative improvement, in which an initial solution is repeatedly improved by making small local perturbations until no further improvements.

The algorithm begins with an initial solution and at high temperature 'T'. A second point is created by using a perturbation scheme. The difference in the function values (delta) at these two points is calculated. If the second one has a smaller function value, that is accepted, otherwise it is accepted with a probability exp (-delta /T).

The algorithm is terminated when a sufficiently small temperature is reached or a small enough change in function values is found.

The Inter-Grid scheduling for optimized scheduling, with the decision variable being the sequence of workflows to be fed into the Inter-Grid scheduler and the constraint being the sequence of operations within a workflow with any available grid sites in the Grid environment, is as under:

The general scheme of SA algorithm: The structure of the SA is as shown:

Step 1: Get an initial solution S

Step 2: Set an initial temperature, T>0

Step 3: While not frozen do the following:
Step 3.1. Do the following n times:
Step 3.1.1. Sample a neighbour S' from S
Step 3.1.2. If delta<=0
Then set S=S'
Else set S=S'
with probability of exp(-delta / T)
Step 3.2. Set T=r*T, where r is the reduction factor

Step 4: Return S.

Perturbation schemes: The three perturbation techniques used in this paper are as follows (Van Laarboven and Aarts, 1987):

- Pair wise exchanges
- Insertion
- Random Insertion Perturbation Scheme (RIPS)

Pair wise exchanges: The proposed perturbation scheme namely the pairwise exchange can be best explained with an example. Consider the workflow sequence 1,2,3,4,5 as a seed sequence and that the integers i,j ($i,j<=W_n$) are randomly generated. Suppose in the first instance i=1 and j=4, the pair wise exchange technique will generate the new sequence 4,2,3,1,5; which is the result of exchanging the first and fourth integers in the starting sequence. In this case, the same sequence is generated by the pair wise exchange scheme as when i<j.

Considering ten workflow sequences, the effect of pair wise exchange is shown as below:

Before Pair wise exchange
W1, W2, W3, W4, W5, W6, W7, W8, W9, W10

After Pair wise exchange
W4, W2, W3, W1, W5, W6, W7, W8, W9, W10

Insertion technique: Consider the sequence 1,2,3,4,5 as a seed sequence and that the i,j (i,j<= W_n) are randomly generated. Suppose in the first instance i=1 and j=4. The insertion technique will generate a new sequence 2,3,4,1,5. i.e. the result of inserting the first integer in the sequence in the 4th position. If however, i=4 and j=1, the insertion technique will generate the sequence 4,1,2,3,5. i.e, the 4th integer has been moved to the first position. In this case the same sequence is generated, the movement of the integers (workflows) is forwards or backwards as i<j or i>j.

Before Insertion technique
W1, W2, W3, W4, W5, W6, W7, W8, W9, W10

After Insertion technique
W2, W3, W4, W1, W5, W6, W7, W8, W9, W10

Random Insertion Perturbation Scheme (RIPS): The perturbation scheme, RIPS, can be best explained by a simple example. Consider a seed sequence S={1,2,3,4,5}. The digit in the first position can be inserted at any position to its right. Hence, the digit in the first position is inserted at any position between 2 and W_n (for example W_n=5) and a random number generation between 2 and W_n is used to select the job position. Suppose the selected position is 3, first digit is inserted in position 3, yielding a new sequence, {2,3,1,4,5}.

Consider the digit in the second position of sequence S and chose randomly two positions for its insertion. Note that this digit can be inserted at any position between (2+1)th to nth position (i.e., a position to its right) and between 1st and (2-1)th position (i.e. a position to its left). Suppose position 1 is selected to the left and position 4 is selected to the right of digit 3, the new sequences thus generated are {2,1,3,4,5} and {1,3,4,2,5}. Similarly, for the digits in positions 3 and 4, we select two positions randomly, one to the right and one to the left and obtain the resulting sequences {1,3,2,4,5}, {1,2,4,3,5}, {4,1,2,3,5} and {1,2,3,5,4}.

For the digit in position 5, only one position is randomly selected towards the left of the digit, i.e. between positions 1 and (W_n-1). Let the randomly selected position be 2 and the resulting sequence be {1,5,2,3,4}. Therefore, RIPS generates 2*(W_n-1) sequences.

The best solution obtained in SA depends on the SA operators such as initial temperature, final temperature, reduction factor and number of iterations to be performed at a particular temperature. The initial temperature and temperature reduction factor are fixed such that a reasonable number of iterations can be performed before the algorithm freezes. An initial temperature of 300 and a freezing temperature of 30 are chosen and the temperature reduction factor is fixed at 0.75.

Combined Simultaneous Objectives (CSO): The two objectives namely minimizing lateness of executing individual workflows 'F1' and maximizing number of workflows completed with in deadline 'F2' are combined using the weighting factors after every solution move among the neighborhood for SA. The *CSO* is represented as given by the formula given:

Minimize CSO

$$\text{CSO} = \left(\left(0.5\times \text{F1}\right) + \left(0.5\times\left(1\div \text{F2}\right)\right)\right) \quad (1)$$

The individual objective functions F1 and F2 are given by the formulae 2 and 3.

F1 = Minimizing lateness of executing individual workflows:

$$\text{F1} = \sum_{i=1}^{m}(tw_i - dtw_i) : \text{Only when } tw_i \geq dtw_i \quad (2)$$

F2 = Maximizing number of work flows completed with in dead line

$$\text{F2} = \sum_{i=1}^{m} xw_i$$

$$\text{where } xw_i \quad = 1 \quad \forall tw_i \leq dtw_i$$
$$= 0 \quad \text{Otherwise}$$

RESULTS AND DISCUSSION

This optimal schedule for the Inter-Grid computation of workflows is obtained by the procedure using SA. This is compared with sequences obtained by different scheduling rules viz. FCFS and EDF. For the experimental problem the workflow sequences obtained by SA gives minimum lateness for each workflow received in Inter-Grid schedulers and maximize the job completion ratio. This is performed as a multiple objective satisfaction criteria during the same search time.

The parameters used in the SA algorithms are given in the Table 2:

The maximum number of workflows completed by different algorithms such as FCFS, EDF and the different SA perturbation schemes are shown in Fig. 3. It is found that the JCR is better for SAI than other schemes.

Table 2: Parameters used in algorithm

SA Parameters	
Initial temperature	300
Temperature reduction factor	0.75
Freezing temperature	30
Number of iterations to be performed at a particular temperature	Number of workflows

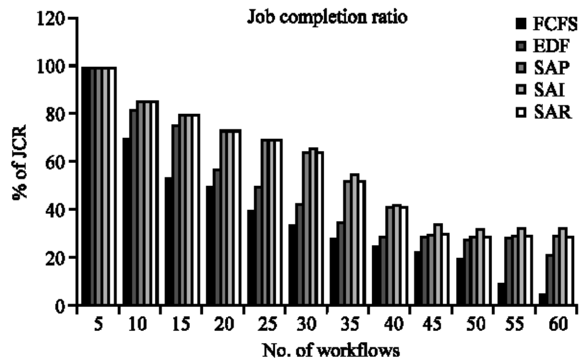

Fig. 3: Job completion ratio

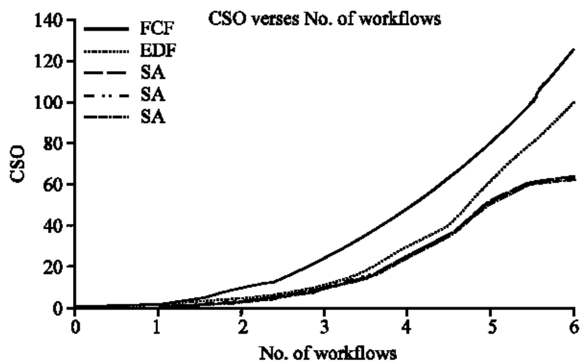

Fig. 4: Performance of different scheduling algorithms

The CSO get minimized when the number of the workflow increases, using various SA perturbation schemes. The result for the illustration of the minimization of the CSO of the scheduling scheme is shown in Fig. 4.

As shown, the CSO increases abrubtly for the scheduling schemes with FCFS and EDF algorithms. Albeit, the SA reveals better results in respect to CSO compared with other scheduling algorithms, there is minor differentiation among the three SA perturbation schemes. However, it manifests the cruciality of the SA algorithm since it is obvious that the lateness by other algorithms is unaffordable. On suitably choosing the Temperature and reduction factor, the algorithm outperforms for its better execution time within deadline.

CONCLUSION

In this research simultaneous optimization of dual objective of an Inter-Grid Scheduler is addressed. We have developed a SA Algorithm to generate nearer-to-optimum schedules. The SA procedure is tested with an example problem environment and the results are compared with that of other available scheduling procedures. It is found SA is more effective than the other algorithms.

In the near future we combine simulated annealing along with the Tabu method to increase the efficiency. Similarly, the ant colony properties can be included for scalability in the existing algorithm. The procedure can also suitably be modified and applied to any kind of Grid scheduling with different problem environment and optimizing any number of objectives concurrently.

REFERENCES

Alain Andrieux *et al.*, 2004. Open Issues for Grid scheduling. http://www.nesc.ac.uk/ technicalpapers/ UKeS-2004-03.

Aloisio, G. and D. Taha, 2002. Grid computing: Toward a new computing Infrastructure. Future generations computer system, 18: 32-34.

Berman, F., G. Fox and T. Hey, 2002. Grid Computing: Makiing the Global Infrastructure a Reality. Wiley Pub., pp: 110-115.

Cannataro, M., D. Talia and P. Trunfio, 2002. Distributed data mining on the grid, Future Generation Comput. Sys., 18: 1101-1112.

Chia-Hung Chien, Paul Hsueh-Mtn Chang and Von-Wun Soo, 2005. Market Oriented Multiple Resource Scheduling in Grid Computing Environments, Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA). IEEE. Comput. Soc., pp: 867-872.

Frey, J., T. Tannenbaum, I. Foster, M. Livny and S. Thecke, 2002. Condor-G: A Computation Management Agent for Multi-Instituitional Grids, J. Cluster Computing, 5: 237-246.

Halpern, J.Y. and Y. Moses, 1990, Knowledge and common knowledge in a distributed environment, J. ACM, 37: 549-587.

Hamscher, V., U. Schwiegelshohn, A. Streit and R. Yahyapour, 2000. Evaluation of Job Scheduling strategies for Grid computing, GRID 2000 First IEEE/ACM International workshop, Proc. Springer Pub., pp: 191-202.

Lublin, U. and D.G. Feitelson, 2003. The workload on parallel supercomputers: Modeling the characteristics of the rigid jobs. Academic Press Inc, 63: 1105-1112.

Manish Parashar and C. James Browne, 2005. Conceptual and Implementation models for the grids, Proc. IEEE., 93: 653-668.

Marek Wieczoek, Radu Prodan and Thoas Fahringer, 2005. Scheduling of Scientific Workflows in the ASKALON Grid Enivironment, SIGMOD. Rec., 34: 56-62.

Mark Baker, Rajkumar Buyya and Domenico Laforenza, 2002. Grids and Grid Technologies for Wide-Area Distributed Computing, Software: Practice and Experience (SPE). J., Wiley Press, pp: 1437-1466.

Rodrigo Real, Adenauer Yamin, Luciano da Silva, Gustavo Frainer, Iara Augustin, Jorge Barbosa and Geyer, 2003. Resource Scheduling on Grid: Handling Uncertainty. Proc. 4th Int. Workshop on Grid Comput., pp: 205-207.

Shanshan Song, Kai Hwang and Yu-Kwong Kwok, 2005. Trusted Grid computing with security binding and Trust Integration, J. Grid Computing.

Vincenzo Di Martino and Marco Mililotti, 2002. Scheduling in a Grid computing environment using Genetic Algorithms, Proceedings of the Parallel and Distributed Processing Symposium, IEEE.

Van Laarhovan, P.J.M. and E.H.L. Aarts, 1987. Simulated annealing: Theory and applications, Dordrecht, Kluwer Academic, pp: 25-125.