A Survey of Evolutionary Heuristic for Job Scheduling using Grid Computing

Japinder Kaur
Department of computer science
Global Institute Amritsar

Mehak Naib
Ass.Prof. Department of computer science
Global Institute Amritsar

ABSTRACT

Grid resources scheduling has become a challenge in the computational Grid. This paper reviews efficient management techniques for managing resources in Grid computing which crucially depends upon the efficient mapping of the jobs to resources according to the user's requirements. The mapping of the jobs is an NP-Complete problem. There is no best solution for all grid computing system. In grid atmosphere Job and resource scheduling is one of the key research area in grid environment.

Keywords

Resource scheduling, Grid computing, Heuristic approach, Hyper Heuristic approach

1. INTRODUCTION

Grid computing systems have become popular for the resolution of large-scale complex problem from science, engineering, finance etc. The ultimate goal of Grid computing is to provide the computing facility to users like power Grid without knowing the detailed characteristics of the source. Thus Grid resource management has turn into one of the most important areas of Grid computing. Grid resource management can be defined as a process of resources to identifying requirements, applications are matched with the resources, resources are allocated and at the closing stages scheduling and monitoring the Grid resources over time in order to run Grid applications as efficiently as possible. Grid resource management system is required to perform resource management decisions which include resource provisioning as well as scheduling, although maximize the Quality of Service (QoS) metrics delivered to the clients [1]. Grid computing is supported by resource. To manage resources in Grid environment is a difficult task. In turn to achieve a job requiring large-scale computation over a distributed system, it is difficult to have a fair judgment on which dispatching method leads to best result. However, mapping self-determining task on to a heterogeneous computing (HC) suite is a well-known NP Complete Problem[2]. NP-Complete problems are often solved using heuristic methods. Heuristic approaches can be easily applied to Grid scheduling problems because Grid scheduling has various important issues such as heterogeneity of resources, energetic and autonomous atmosphere of Grid resources and finally there source providers and resource consumers have different policies for execution of their applications. Hyper-heuristic can be seen as a superior methodology, which when specified a exacting difficulty instance or a class of instances and a number of low-level heuristics, automatically produced an adequate combination of the provided components to effectively solve the given troubles [3]. This paper analyze why heuristic and meta heuristic methods are good alternatives to more traditional scheduling techniques and what make them appropriate for grid scheduling [5].This paper is ordered as follows: Section 2 discusses related work. In section 3, some scheduling problems in grid systems are discussed and phases of scheduling in grid computing as well. In section 4, we discussed different heuristic and meta heuristic approaches for scheduling in grid computing. Section 5, we presents the Analysis and comparison between various heuristic approaches.

2. RELATED WORK

Author used nature's heuristics namely Genetic Algorithm (GA), Simulated Annealing (SA) and Tabu Search (TS) for scheduling of jobs on computational Grids. Authors identify that GA works better than TS and SA for scheduling of the jobs to exact resources but hybrid heuristic algorithms perform better than GA approach as it minimizes the time required for scheduling the job [5]. Author describes the hyper heuristic based resource scheduling approaches in grid computing. Authors proposed a novel hyper-heuristic based scheduling algorithm for scheduling of jobs in Grid environment so as to minimize the cost and time by minimizing the makespan [4].Author describe the scheduling problem in grid system and different phases of scheduling in grid environment[6].

3. GRID SCHEDULING

3.1 Scheduling problems in Grid systems

The scheduling problem is one of the most studied problems in the optimization research community. However, in the Grid setting there are several characteristics that make the problem different and more challenging. Some of these problems are the following[6].

The dynamic structure of the Computational Grid: Unlike traditional distributed systems, in a Grid system the resources can fix or leave the Grid in random way. This may be just due to losing connection to the system or because their owners switch off the machine or change the operating system, etc. Given that the resources cross different organizational domains, there is no control on the resources.

The high heterogeneity of resources: In Grid systems, computational resources could be very disparate in their computing capacity, ranging from laptops, desktops, clusters, supercomputers and even small computational devices. Current Grid infrastructures are not yet much versatile but heterogeneity is among most important features in any Grid system.

The high heterogeneity of jobs: Jobs arriving at any Grid system are diverse and heterogeneous in terms of their computational needs. For instance, they could be computing intensive or data intensive; some jobs could be full applications having many specifications and others could be just atomic tasks. Significantly, in common the Grid system will not be aware of the type of tasks arriving in the system.

The high heterogeneity of interconnection networks: Grid resources are connected through the Internet using different interconnection networks. Transmission costs will often be very important in the overall Grid performance and hence smart ways to cope with the heterogeneity of interconnection networks is necessary.

The large scale of the Grid system: Grid systems are expected to be vast level. likewise, the quantity of jobs, tasks or applications submitted to the Grid over time could be large as well. Therefore, the efficient management of resources and planning of jobs will require the use of different types of scheduling: schedulers, local schedulers, and their possible hierarchical combinations to achieve scalability.

The existence of local policies on resources: Again, due to the different ownership of the resources, individual cannot suppose full control over the Grid resources. Companies might have unexpected computational needs and may decide to reduce their contribution to the Grid. Other policies on access, accessible storage etc. are also to be taken into account.

The job-resource requirements: When scheduling, the Current Grid schedulers assume full availability and compatibility of resources. In actual situations, however, various restrictions and/or incompatibilities could be derived from job and resource specifications.

Security: This characteristic, which is non-existent in classical scheduling, is a significant concern in Grid scheduling. At this point, the security can be seen as a two-fold objective: on the one hand, a task, job or application could have security requirements as well as, on the other side, the Grid nodes can have their own security requirements.

3.2 Phases of scheduling in Grids

In turn to achieve the scheduling process, the Grid scheduler has to follow a series of steps which could be classified into five blocks[5]: (1) Preparation and information gathering on tasks submitted to the Grid; (2) Resource selection; (3) Computation of the planning of tasks to selected resources; (4) Task (job or application) allocation according to the planning (the mapping of tasks to selected resources); and (5) Monitoring of task completion.

Preparation and information gathering: Grid schedulers have access to the information on available resources and tasks through the Grid Information Service. Moreover, the scheduler will be informed about updated information (according to the scheduling mode) on jobs and resources.

Resource selection: Not all resources could be candidates for the allocation of tasks. Therefore, the selection process is carried out based on job requirements and resource uniqueness. The collection process, will depend on the scheduling method. For example, if in a batch mode, tasks will to be allocated, a pool of as many as possible candidate resources will be identified out of the set of all available resources. The selected resources are then used to compute the mapping that meets the optimization criteria.

Calculation of the planning of tasks: In this phase the planning is computed.

Allocation of Task: Planning is prepared effective in this phase: tasks are allocated to the selected resources according to the planning.

Task execution monitoring: Once the allocation is done, the monitoring will inform about the execution progress as well as possible failures of jobs, which depending on the scheduling policy will be rescheduled or migrated to other resources.

3.3 Heuristic and meta-heuristic approaches for scheduling in Grids

From the exposition in the previous sections, it is clear that the Grid scheduling problem is really demanding. Dealing with the several constraint and optimization criteria in a dynamic environment is very complex and computational hard. Meta-heuristic approach are certainly considered the de facto approach. This paper point out the main reasons that explain the strength of meta-heuristic approaches for designing efficient Grid schedulers.

Meta-heuristics are well understood: Meta-heuristics have been studied for a large number of optimization problems, from theoretical, practical and experimental perspective. Definitely, the known study, product and experience with meta-heuristic approaches are a good starting point for designing Meta heuristic- based Grid schedulers.

No need for optimal solutions: In the Grid scheduling problem, for most practical applications good quality planning of jobs would suffice rather than searching for optimality. In fact, in the highly dynamic Grid atmosphere, it is not probable to still describe the optimality of scheduling, as it is defined in combinatorial optimization. This is so due to the fact that Grid schedulers run as lengthy as the Grid system exists and thus the performance is measured not only for particular applications but also in the extended run. It is recognized that meta-heuristics are able to compute in a short time high-quality feasible solutions. Therefore, in such situation meta-heuristics are among best candidates to cope with Grid scheduling.

Efficient solutions in short time: Research work on meta heuristics has by large tried to find ways to avoid getting stuck in local optima and ensure convergence to suboptimal or optimal solutions. That allows one to tune the convergence speed. For instance, in Genetic Algorithms, by choosing appropriate genetic operators one can achieve a very fast convergence of the algorithm to local optima. Similarly, in the Tabu Search method, one can work with just short-term memory in combination with an intensification procedure to produce high-quality feasible solutions in a very short time. This feature of meta-heuristics is very useful for Grid schedulers in which we might want to have a very fast reduction in makespan, flow time and other parameters.

Dealing with multi-objective nature: Meta-heuristics have proven to efficiently solve not only single-objective optimization problems but also multi-objective optimization problems.

Appropriateness for periodic and batch scheduling: Periodic scheduling is a particular case of Grid scheduling. when companies and users submit their applications to the Grid System periodically, it arise frequently. In this case resource provisioning can be done in the Grid infrastructures and, which is more important in our context, there are no strong time restrictions.

Hybridization with other approaches: Meta-heuristics can be easily hybridized with other approaches. This is useful to make Grid schedulers to better respond to concrete Grid types, exact types of applications, etc. The hybridization in general can produce better solutions than those delivered by single approaches.

Designing robust Grid schedulers: The changeability of the Grid environment over time is among the factors that directly influences the performance of the Grid scheduler. A robust scheduler should deliver high-quality planning even under frequent changes of the characteristics of the Grid communications. Verification in meta-heuristics literature exists that in general meta-heuristics are robust approaches.

In the following subsections this paper briefly review the most important heuristic approaches and the benefits of using them for the Grid scheduling[10].

4. LOCAL SEARCH-BASED HEURISTIC APPROACHES

Local search heuristic approaches is a family of methods that explore the solution space by starting at an initial solution, and constructs a path in solution space during the search process. Local search heuristic approaches improve solutions through neighborhood search. The main objective of this local search based heuristic approach is to gain feasibility as soon as possible. They have been applied successfully to many industrial problems and performance of local search based heuristic approaches depending on construction of neighborhood.

Tabu Search: Tabu Search (TS) is a high level heuristic procedure for solving optimization problems. Tabu search is a meta-heuristic that guides a local search procedure to explore the solution space beyond local optimality [5].

Hill Climbing: Hill Climbing (HC) is a graph search algorithm where the current path is extended with a successor node which is closer to the solution than the end of the current path. Hill climbing is logical and beneficial especially in situations where the search space is of simple nature with no more than a single maxima or minima.

Simulated Annealing: SA is an iterative technique that considers only one possible solution (mapping) for each meta task at a time [8]. This solution uses the same representation as the chromosome for the genetic algorithm. The initial implementation of SA was evaluated and then it was modified and refined to give a better final version. SA uses a procedure that probabilistically allows poorer solutions to be accepted in an attempt to obtain a better search of the solution space.

Genetic Algorithms: Genetic Algorithm (GA) Genetic algorithms are playing an increasingly important role in studies of complex adaptive systems, range from adaptive agents in economic theory to the use of machine learning techniques in the design of complex devices such as aircraft turbines and integrated circuits [14]. GA is a famous stochastic optimization algorithm which uses biologically inspired techniques such as genetic inheritance, ordinary selection, alteration and sexual imitation (recombination, or crossover). Genetic algorithms are useful heuristics to find a near optimal solution in large search spaces [15]. In GA, a point in search space is represented by a set of parameters and these parameters are known as genes and a set of genes is known as string or a chromosome. A fitness function must be devised for each problem to be solved. Each chromosome is assigned a fitness value that indicates how closely it satisfies the desired objective. Given a exacting genetic material, the strength function returns only numerical fitness or stature of assessment, which will establish the capacity of the individual, which that chromosome represents. A set of chromosomes is called population. imitation is an additional critical feature of GAs where two individuals selected from the population are allowed to mate to create offspring, which will contain the next production. Having chosen two parents, their chromosome is recombined, usually the mechanisms of intersect and alteration. Mutation provides a small amount of casual search, and helps to make sure that no point in the search space has a zero probability of being examine. If the GA has been properly implemented, the inhabitants will develop over succeeding generations so that the fitness of the best and the average individual in each generation increases towards the global optimum.

4.1 Population-based heuristic approaches

Population-based heuristics is a large family of methods that have shown their efficiency for solving combinatorial optimization troubles. Population-based process usually requires large running times if suboptimal or optimal solutions are to be found.

Memetic Algorithm Memetic Algorithm (MA) is an extension of genetic algorithm. Memetic algorithms are evolutionary algorithms that can be applied on a local search process to refine solutions for solid troubles. Memetic algorithms are the issue of concentrated scientific research and have been successfully applied to a multitude of real-world problems ranging from the construction of optimal university exam timetables, to the forecast of protein structures and the optimal design of space-craft trajectories.

Ant Colony Optimization: The real power of ants resides in their colony brain. The self-organization of those individuals is very similar to the organization found in brain-like structures. Like neurons, ants use mainly chemical agents to communicate. One ant releases a molecule of pheromone that will influence the behavior of other ants. Ant algorithms are often compared with other evolutionary approaches such as Genetic Algorithms, Evolutionary Programming and Simulated Annealing. It is important to remember that Ant algorithms are non-deterministic and rely on heuristics to approximate to a sub-optimal solution in cases where the number of combinations is extremely huge and is impossible to calculate using a deterministic algorithm

Particle Swarm Optimization: Particle Swarm Optimization (PSO) is a method for performing numerical optimization without explicit knowledge of the gradient of the problem to be optimized.It simulates the process of a swarm of nature predatory. It has the enhanced capacity of comprehensive searching and has been successfully applied to many areas. A flock or swarm of particles is randomly generated. Initially, each particle position represents a possible solution point in the problem space. The fitness value of each particle is evaluated by the objective function to be optimized. All particles remembers the coordinates of the better solution (gbest) achieved so far. The coordinates of current global best (pbest) are also stored.

4.2 Hybrid heuristic approaches

Hybrid strategies have been constructed to exploit the meta heuristic techniques. To get a better result of genetic algorithm, it has been hybridized with local search methods as tabu search and simulated annealing etc. The major advantage of parallel hybrids implemented on shared-memory parallel architectures is their simplicity.

For instance, MAs combine an evolutionary search with a local search. However, hybridization among different meta-heuristics has been shown to be effective for many problems by outperforming single methods. However, hybrid meta-heuristics have been less explored for the problem. Author addressed the hybridization of GA, SA and TS heuristics; GA+SA hybridization is expected to have a better convergence than a pure GA search and GA+TS could improve the efficiency of the GA. In these hybridizations a heuristic capable to deal with a population of solutions, such as a GA, is combined with local search heuristics that deal with only one solution at a time.

Hyper-heuristics are methods that lead the search, at a superior level as compared to meta-heuristics. Hyper-heuristics have proven to be effective for scheduling and timetabling[3]. The Hyper heuristic is a high level algorithm. A hyper-heuristic can be seen as a high level methodology which when given a particular problem instance or a class of instances and a number of low-level heuristics, manually produces an adequate combination of the provided components to effectively solve the given problems.

Advantages

- Hyper-heuristics operate in a space of heuristics, choosing and applying one low-level heuristic from a given set at each decision point.
- Hyper-heuristics do not need knowledge of each low level heuristic.
- Hyper-heuristics are robustness and re-applicability heuristics.

Disadvantages

- For many hyper-heuristics, a significant amount of parameter tuning is required in order to find good parameter settings for a given problem.
- A huge number of difficult instances may be essential for training and testing of the method in order to accumulate enough knowledge to make the right choice of low-level heuristics.

5. CONCLUSION AND FUTURE WORK

Besides the many aspects and facets of the Grid scheduling problem presented in the previous sections, there still remain other issues to be considered. This paper briefly mentions some of them here. In this survey consider hyper heuristic is near optimal solution but there are still chances of improvement by using other techniques like Bacterial Foraging Optimization. Security is an important aspect to be considered in Grid scheduling. Moreover, the objective is to reduce the possible overhead to the Grid scheduler that would introduce a secure scheduling approach. Most current Grid approaches are task-oriented or resource-oriented approaches. However, with the ever-increasing complexity of large-scale problems in which both tasks and data are to be scheduled, an integrated scheduling approach that would optimize allocation of both the task and the data is required.

In Future, this work can be done by using Bacterial Foraging Optimal algorithm. Further significant issues are associated to data-aware development. Generally existing Grid approach are task-oriented or resource-oriented approach.

6. REFERENCES

[1] Foster, I. and Kesselman, C.,"The Grid: Blueprint for a Future Computing Infrastructure", Morgan Kaufmann Publishers, USA, 2004.

[2] Lee, A. and Parashar, M., Senior member of IEEE," A Survey of Job Scheduling and Resource Management in Grid Computing", 2008.

[3] Buyya, R. and Venugopal, S.,"A Gentle Introduction to Grid Computing and Technologies" Computer Society of India, july 2005.

[4] Bhuyan, P., Sharma, R., Soni, V.K. and Mishra, M.K.," A Survey of Job Scheduling and Resource Management in Grid Computing", World Academy of Science and Engg And Tech,2010.

[5] Burke,E.K., Hyde,M., Kendall, G., Ochoa, G., Ozcan,E. and Qu, R., "Hyper-heuristics: A survey of the State of the Art",Technical report, University of Nottingham, 2009.

[6] Xhafa, F. and Abraham, A., "Computational models and heuristic methods for Grid scheduling problems" Future Generation Computer System 2010.

[7] Abraham,A., Buyya,R. and Nath,B., "Nature's Heuristics for Scheduling Jobs on Computational Grids". The 8th IEEE Conference on Advanced Computing and Communications, Cochin, India, 2000.

[8] Abraham, I., Aron, R. and Chnna, I.," Hyper-Heuristic Based Resource Scheduling in Grid Environment", IEEE International Conference on Systems, 2013.

[9] Aron, R. and Channa, I.," Bacterial foraging based hyper-heuristic for resource scheduling in Grid computing", Department of Computer Science and Engineering, Thapar University, Patiala, India, September 2012.

[10] Aron, R. and Chnna, I., "Grid scheduling heuristic methods: State of the Art", ISSN 2150-7988 Volume6 (2014).

[11] Liu, H., Abraham, A. and Hassanien, A.E., "Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm", Future Generation Computer Systems, 2010.

[12] Pooranian, Z., Harounabadi, A. and Hedayat, N., "New hybrid Algorithm For Task Scheduling in Grid Computing to Decrease missed Task ", world academy of Science, Engg and Tech,2011.

[13] Passino, K.M.,"Biomimicry of Bacterial Foraging for Distributed Optimization and Control",IEEE Control and System Magazine, 2002.

[14] Garg, S., Konugurthi, P.and Buyya, R.," A linear programming driven genetic algorithm for meta scheduling on utility Grids", in:16th International Conference on Advanced Computing and Communication, ADCOM 2008, IEEE Press, New York, USA, 2008.

[15] Hu, M. and Verravalli, B., Senior Member of IEEE, "Requirement-Aware scheduling of Bag-of-tasks applications on Grid with Dynamic Resilience", 2013.

[16] Carretero, J. and Xhafa,F. and abraham,A. "Genetic Algorithm based Schedulers for Grid Computing Systems", International Journal of Innovative Computing, Information and Control, Vol 3, No.6, 2007.