Available online at http://www.journalcra.com

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

International Journal of Current Research
Vol.11, pp.228-235, December, 2010

ISSN: 0975-833X

REVIEW ARTICLE

GRID SCHEDULING ALGORITHMS: A SURVEY

Maruthanayagam, D[1]. and Uma Rani, R[2].

[1]Department of Computer Applications, Vivekanandha College of Arts Sciences for Women, Tiruchengode, Tamilnadu, India
[2]Department of Computer Science, Sri Sarada College for Women, Salem,

ARTICLE INFO

Article History:
Received 12th September, 2010
Received in revised form
15th October, 2010
Accepted 13th November, 2010
Published online 28th December, 2010

Key words:

Grid computing,
Static, dynamic, ant colony,
Multiobjective,
Simulated annealing,
Taboo,
Genetic,
Heuristic,
Pheromone,
Chromosome,
Short hop and long hop.

ABSTRACT

Thanks to advances in wide-area network technologies and the low cost of computing resources, Grid computing came into being and is currently an active research area. One motivation of Grid computing is to aggregate the power of widely distributed resources, and provide non-trivial services to users. To achieve this goal, an efficient Grid scheduling system is an essential part of the Grid. Rather than covering the whole Grid scheduling area, this survey provides a review of the subject mainly from the perspective of scheduling algorithms. In this review, the challenges for Grid scheduling are identified. The various Grid scheduling algorithms such as Static, Dynamic, Ant Colony, Multi objective Evolutionary algorithms, Simulated Annealing, Taboo search and Genetic algorithms are discussed from different points of view. Their merits and demerits are pointed out and some general issues worthy of further exploration are proposed Algorithms, Simulated Annealing, Taboo search and Genetic algorithms are discussed from different points of view.

© Copy Right, IJCR, 2010 Academic Journals. All rights reserved.

INTRODUCTION

The popularity of the Internet and the availability of powerful computers and high-speed networks as low-cost commodity components are changing the way we use computers today. Recent research on these topics has led to the emergence of a new paradigm known as *Grid computing.* To achieve the promising potentials of tremendous distributed resources, effective and efficient scheduling algorithms are fundamentally important.

***Corresponding author:** kesimaruthu@gmail.com

Unfortunately, scheduling algorithms in traditional parallel and distributed systems, which usually run on homogeneous and dedicated resources, e.g. computer clusters, cannot work well in the new circumstances. In this paper, the state of current research on scheduling algorithms for the new generation of computational environments will be surveyed and open problems will be discussed.

Local vs. Global

At the highest level, a distinction is drawn between local and global scheduling. The local scheduling discipline determines how the processes resident on a single CPU reallocated and executed; a global scheduling policy uses information about the system to allocate processes to multiple processors to optimize a system-wide performance objective. Obviously, Grid scheduling falls into the global scheduling branch.

Types of Scheduling Algorithms

- Static scheduling algorithms (SS)
- Dynamic scheduling algorithms (DS)
- Ant Colony Scheduling algorithms (ACS)
- Multi objective Evolutionary algorithms (MOEA)
- Simulated Annealing algorithms (SA)
- Taboo search algorithms (TS)
- Genetic algorithms (GA).

Static scheduling Algorithms

In the case of static scheduling, information regarding all resources in the Grid as well as all the tasks in an application is assumed to be *available by the time the application* is scheduled. Every task comprising the job is assigned once to a resource. Thus, the placement of an application is *static.*

Merits

- Estimate of the cost of the computation can be made in advance of the actual execution.
- In the case of static scheduling, information regarding all resources in the Grid as well as all the tasks in an application is assumed to be available by the time the application is scheduled (Fatos Xhafa *et al.,* 2009).
- The assignment of tasks is fixed a priori, and estimating the cost of jobs is also simplified.
- The static model allows a global view of tasks and costs.
- A major benefit of the static model is that it is easier to program from a scheduler's point of view.

Demerits

- Cost estimate based on static information is not adaptive to situations such as one of the nodes selected to perform computation fails, ***becomes isolated*** from the system due to network failure.
- Heavily loaded with jobs that its response time becomes longer than expected. Unfortunately, these situations are quite possible and beyond the capability of a traditional scheduler running static scheduling policies. To alleviate this problem, some auxiliary mechanisms such as rescheduling mechanism are introduced at the cost of overhead for task migration. Another side-effect of introducing these measures is that the gap between static scheduling and dynamic scheduling becomes less important.
-

Dynamic Scheduling Algorithm (On line Scheduling Algorithms)

In the case of dynamic scheduling, the basic idea is to perform task allocation on the fly as the application executes. This is useful when it is impossible to determine the execution time, direction of branches and number of iterations in a loop as well as in the case where jobs arrive in a real-time mode (Kun-Ming Yu and Cheng-Kwan Chen, 2008).

Merits

- In the case of dynamic scheduling, the basic idea is to perform task allocation on the fly as the application executes.Dynamic task scheduling has two major components 1. System state estimation 2. Decision making. System state estimation involves collecting state information throughout the Grid and constructing an estimate. On the basis of the estimate, decisions are made to assign a task to a selected resource. A natural way to keep the whole system health is balancing the loads of all resources. This is the advantage of dynamic over static scheduling. Another benefit is maximizing resource utilization, rather than minimizing runtime for individual jobs. If a resource is assigned too many tasks, it may invoke a balancing policy to decide whether to transfer some tasks to other resources, and which tasks to transfer.

Demerits

- Dynamic scheduling is usually applied when it is difficult to estimate the cost of applications, because jobs are coming online dynamically.
- It is impossible to determine the execution time, direction of branches and number of

iterations in a loop as well as in the case where *jobs arrive in a real-time mode.*

Ant Colony Algorithms

Dorigo M. introduced the Ant algorithm in 1996, which is *new heuristics,* predictive scheduling algorithm. It is based on the real ants. When an ant looks for food, ant deposits some amount of pheromone on the path, thus making, it is followed by a trail of this substance, shown by fig-1. If an ant tries to move from one place to another then it encounters a previously laid trail. The ant can detect the pheromone trail and decide with high probability to follow it (Dorigo *et al.,* 1991). This ant also reinforces the trail with its own pheromone. It also shown by Fig. 2. When more ants are following the trail, then the pheromone on shorter path will be increased quickly. The quantity of pheromone on every path will affect the possibility of other ants to select path. At last all the ants will choose the shortest path, shown by Fig. 3.

Merits

- In the context of Ant colony scheduling, the scheduler attempts to find optimal distribution of work units to processing nodes.
- Work units are the ants, computational power is the food, and the mapping is the path.
- More work units are sent to more powerful nodes.
- Meta-heuristic modeling the behavior of ants searching for food.
- Ants make decisions based on pheromone levels**.**
- Random searching provide potentially better routes.
- Decisions affect *pheromone* levels to influence future decisions. Initial decisions are made at random. Ants leave trail of pheromones along their path. Still random since initial trails were random.
- Due to batching of work units, server to client communication is consolidated and reduced (Marco Dorigo *et al*., 1999)

Demerits

- Decisions affect ***pheromone*** levels to influence future decisions. Initial decisions are made at random. Ants leave trail of pheromones along their path. Still random since initial trails were random.
- Due to batching of work units, server to client communication is consolidated and reduced

Multiobjective Evolutionary Algorithms (MOEA)

Merits

Only dynamic scheduling will be useful for available resources.

- Jobs and resources are arranged in an ascending order (and dynamically updated) according to the job lengths and processor speeds.
- To schedule Shortest Job on the Fastest Resource (SJFR).
- To schedule Longest Job also on the Fastest Resource (LJFR) because of Minimizing completion time both jobs and the average job finishes quickly.
- Pareto dominance concept is used in order to compare 2 solutions. jobs allocation between machines is uniform and is preferred (Chitra and Venkatesan, 2010). This means, there will not be *idle machines* as well as *overloaded machines.*
- In MOEA fundamental criterion is that of minimizing the makespan, a secondary criterion is to minimize the flow time which gives minimizing the sum of completion times of all the tasks
- Three well known global optimization techniques namely simulated annealing (SA), genetic algorithm (GA) and particle swarm optimization (PSO**).** Even though the MOEA approach obtained better results for the considered test problems compare above mentioned algorithms (David A. Van Veldhuizen, 2006).
- The most common approaches of a multiobjective optimization problem use the concept of Pareto dominance as defined below:
- *Definition* (*Pareto dominance*)

- Consider a maximization problem. Let x, y be two decision vectors (solutions) from the definition domain.
- Solution x *dominate* y (also written as x f y) if and only if the following conditions are fulfilled:
- *(i)* $fi(x)$ _ $fi(y)$; $\forall i = 1, 2, \ldots, n$;
- *(ii)* $\exists j$ $\{1, 2, \ldots, \in n\}$: $fj(x) > fj(y)$.
- That is, a feasible vector x is Pareto optimal if no feasible vector y can increase_some criterion without causing a simultaneous decrease in at least one other criterion. *Multiobjective evolutionary algorithms* can yield a whole set of potential solutions, which are all optimal in some sense.

In the proposed multiobjective approach, the solution is represented as a string of length equal to the number of jobs. The value corresponding to each position i in the string represent the machine to which job i was allocated. Consider we have 10 jobs and 3 machines (David and Van Veldhuizen, 2006).

Then a chromosome and the job allocation can be represented as follows Fig. 4.

1 2 3 2 1 1 3 2 1 3
Machine 1 Job 1 Job 5 Job 6 Job 9
Machine 2 Job 2 Job 4 Job 8
Machine 3 Job 3 Job 7 Job 10

The Pareto dominance concept is used in order to compare 2 solutions. The one which dominates is preferred. In case of non dominance, the solution whose jobs allocation between machines is uniform is preferred (Dorigo *et al.,* 1991). This means, there will not be idle machines as well as overloaded machines.

Demerits

The main challenge in a multiobjective optimization environment is to minimize__the distance of the generated solutions to the Pareto set and to maximize the diversity of the developed Pareto set. Scheduling problem involves simultaneous optimization of several *objectives* including completon time, resource utilization, Qos metrics, costs, reliability factors etc. So, by its nature, it is a multi objective optimization problem. All the existing approaches for dealing with grid scheduling problem transform this problem into a single objective problem. To obtain diversification special care has to be taken in the selection process. MOEA approach obtained better results for the considered test problems but more conclusions could be drawn only after extensive validation using bigger problem sizes and more objectives etc.

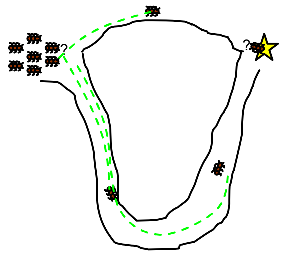

Fig.1 Ants try to move from one place to another.

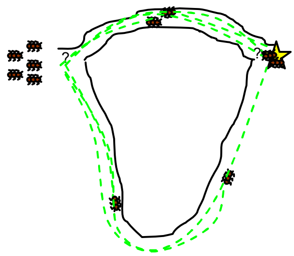

Fig. 2. Ants are reinforcing the trail with its own pheromone.

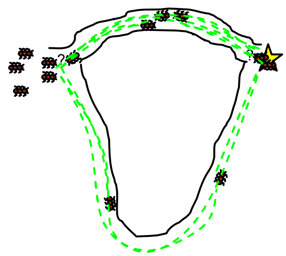

Fig. 3. Ants choose the shortest path.

Simulated Annealing (SA)

SA is a search technique based on the physical process of annealing, which is the thermal process of obtaining low-energy crystalline states of a solid. At the beginning, the *temperature is* increased to melt the solid. If the temperature is slowly decreased, particles of the melted solid arrange themselves locally, in a stable ground state of a solid. SA theory states that if temperature is lowered sufficiently slowly, the solid will reach thermal equilibrium, which is an optimal state. By analogy, the thermal equilibrium is an optimal task-machine mapping (optimization goal), the temperature is the total completion time of a mapping (cost function), and the change of temperature is the process of mapping change. If the next temperature is higher, which means a worse mapping, the next state is accepted with certain probability (Asim YarKhan and Jack Dongarra, 2002). This is because the acceptance of some worse states provides away to escape local optimality which occurs often in local search.

A procedure implemented here figure 1 is as follows. The first mapping is generated from a uniform random distribution. The mapping is mutated in the same manner as the GA, and the new makespan is evaluated. If the new makespan is better, the new mapping replaces the old one. If the new makespan is worse (larger), a uniform random number $z \in [0, 1)$ is selected. Then, z is compared with y, Where

$$y = \frac{1}{1 + e^{\left(\frac{\text{old makespan-new makespan}}{\text{temperature}}\right)}}$$

Fig. 5. New Makespan is evaluated.

If $z>y$, the new (poorer) mapping is accepted; otherwise it is rejected, and the old mapping is kept. So, as the system temperature cools, it is more difficult for poorer solutions to be accepted.

Table-1: Comparision of above stated Algorithms

S.no	*Static*	*Dynamic*	*Ant Colony*	*Multi Objective*
1.	Estimate of the cost of the computation can be made in advance of the actual execution.	The basic idea is to perform task allocation on the fly as the application executes	The scheduler attempts to find optimal distribution of work units to processing nodes	To schedule Shortest Job on the Fastest Resource (SJFR)
2.	The assignment of tasks is fixed a priori, and estimating the cost of jobs is also simplified	It has two major components 1. System state estimation 2.decision making.	Work units are the ants, computational power is the food, and the mapping is the path	To obtain diversification special care has to be taken in the selection process.
3.	Heavily loaded with jobs that its response time becomes longer than expected.	It is impossible to determine the execution time, direction of branches and number of iterations in a loop	Initial decisions are made at random.	All the existing approaches for dealing with grid scheduling problem transform this problem into a single objective problem.

S.no	*Simulated Annealing*	*Taboo Search*	*Genetic*
1	*SA is a search technique based on the physical process of annealing, which is the thermal process.*	***TS** is a mathematical optimization method, belonging to the class of local search techniques*	*This is an evolutionary technique for large space search.*
2.	if temperature is lowered sufficiently slowly, the solid will reach thermal equilibrium, which is an optimal state	**TS** algorithm is the best choice for small time schedule creation	It used techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover
3.	As the system temperature cools, it is more difficult for poorer solutions to be accepted.	Taboo Search solution is forbidden (prohibited) if it is obtained by applying a taboo action to the current solution.	one open problem is how to use dynamically predicted performance information to help the speed of convergence

This is reasonable because when the temperature is lower, there is less possibility to find a better solution starting from another poorer one. After each mutation, the system temperature is reduced to 90% of its current value (Xiaohong Kong et al., 2009). (This percentage is defined as the cooling rate.) This completes one iteration of SA. The heuristic stops when there is no change in the makespan for a certain number of iterations or the system temperature approaches zero.

Merits

SA is a search technique based on the physical process of annealing, which is the thermal process of obtaining low-energy crystalline states of a solid. SA theory states that if temperature is lowered sufficiently slowly, the solid will reach thermal equilibrium, which is an optimal state. The thermal equilibrium is an optimal task-machine mapping (optimization goal), the temperature is the total completion time of a mapping (cost function), and the change of temperature is the process of mapping change.

Demerits

As the system temperature cools, it is more difficult for poorer solutions to be accepted. When the temperature is lower, there is less possibility to find a better solution starting from another poorer one. After each mutation, the system temperature is reduced to 90% of its current value. (This percentage is defined as the cooling rate.)

Taboo Search (TS)

Taboo search is also a solution space search. It keeps track of the regions which have already been searched. The new search need not repeat a search near this area. TS is a meta-strategy for guiding known heuristics to overcome local optimality and has now become an established optimization approach that is rapidly spreading to many new fields (Fayad *et al.,* 2007). TS is implemented beginning with a random mapping as the initial solution, generated from a uniform distribution. To manipulate the current solution and move through the solution space, a short hop is performed. The intuitive purpose of a short hop is to find the nearest local minimum solution within the solution space . The basic_procedure to perform a short hop is to consider, for each possible pair of tasks, each possible pair of machine assignments, while the other assignments are unchanged. If the new makespan is an improvement, the new solution is saved, replacing the current solution. The short hop procedure ends when (1) every pair-wise remapping combination has been exhausted with no improvement, or (2) the limit on the total number of successful hops (limit hops) is reached. When the short hop procedure ends, the final mapping from the local solution space search is added to the tabu list. The tabu list is a method of keeping track of the regions of the solution space that have already been searched. Next, a new random mapping is generated, and it must differ from each mapping in the tabu list by at least half of the machine assignments (a successful long hop) (Benedict and Vasudevan, 2008). The intuitive purpose of a long hop is to move to a new region of the solution space that has not already been searched. After each successful long hop, the short hop procedure is repeated. The stopping criterion for the entire heuristic is when the sum of the total number of successful short hops and successful long hops equals *limithops*. Then, the best mapping from the tabu list is the final answer.

Merits

- TS is a mathematical optimization method, belonging to the class of local search techniques. It is rapidly spreading to many new fields.
- TS enhance the performance of a local search method by using memory structures.
- Once a potential solution (schedule) has been determined, it is marked as *taboo* so that the algorithm does not visit that possibility repeatedly.
- This is usually obtained by keeping track of the last solutions in term and of the action.
- When an action is performed it is considered taboo for the next *T* iterations, where *T* is the taboo status length.
- TS algorithm is the best choice for small time schedule creation.

- TS gives the best schedule depends upon simulation model implementation and testing experiments.

Demerits

Taboo Search solution is forbidden (prohibited) if it is obtained by applying a taboo action to the current solution.

Genetic Algorithm (GA)

The Genetic Algorithm **(**GA**)** is one of the best methods to search the large solution space. This method operates on the population of chromosomes for a given problem. First it generates the initial population randomly. The initial population may be generated by any other heuristic algorithm; if the population is generated by Min-Min then it is called seeding the population with Min-Min. This genetic algorithm randomly selects chromosomes. Crossover is the process of swapping certain sub-sequences in the selected chromosomes. Mutation is the process of replacing certain sub-sequences with some task-mapping choices new to the current population. Both crossover and mutation are done randomly **(**Yang Gao *et al.,* 2005). After crossover and mutation, a new population is generated. Then this new population is evaluated, and the process starts over again until some stopping criteria are met. The stopping criteria can be, for example, 1) no improvement in recent evaluations; 2) all chromosomes converge to the same mapping; 3) a cost bound is met. For its simplicity, GA is the most popular nature's heuristic used in algorithms for optimization problems.

The general procedure of GA search is as follows

Population generation: A population is a set of chromosomes and each represents a possible solution, which is a mapping sequence between tasks and machines. (Seeding the population with a Min-min chromosome).

Chromosome evaluation: Each chromosome is associated with a fitness value, which is the makespan of the task-machine mapping this chromosome represents.

- **Crossover and Mutation operation:** Crossover operation selects a random pair of chromosomes and chooses a random point in the first chromosome. Crossover exchanges machine assignments between corresponding tasks. Mutation randomly selects a chromosome, then randomly selects a task and randomly reassigns it to a new machine. Finally, the chromosomes from this modified population are evaluated again. This completes one iteration of the GA. The GA stops when a predefined number of evolutions is reached.

Merits

- This is an evolutionary technique for large space search.
- It is a search technique used in computing to find exact or approximate solutions to optimization and search problems.
- This algorithm is also known as evolutionary computation Algorithm.
- It used techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

Demerits

- In a genetic scheduling algorithm, one open problem is how to use dynamically predicted performance
- information to help the speed of convergence (as the current genetic scheduling algorithms are static).

CONCLUSION

In this paper, we mainly discussed the merits and demerits of various scheduling algorithms. Although task scheduling in parallel and distributed systems has been intensively studied, new challenges in Grid environments still make it an interesting topic, and many research projects are underway. Through our survey on current scheduling algorithms working in the Grid computing scenario, we can find that heterogeneity; dynamism, computation and data separation are the primary challenges concerned by current research on this topic. We also find that the evolution of Grid infrastructures, e.g., supports for complex application models such as DAG, resource information services and job migration frameworks, provides an opportunity to implement sophisticated scheduling algorithms.

- In addition to enhancements to classic scheduling algorithms, new methodologies are
- applied, such as the adaptive application-level scheduling, Grid economic models and nature's
- heuristics. Due to the characteristics of Grid computing and the complex nature of the scheduling problem itself, there are still many open issues for Grid scheduling, relating to scheduling architectures, protocols, models etc. However, we only focus on those questions about algorithms.

REFERENCES

Asim Yar Khan and Jack J. Dongarra, 2002."Experiments with Scheduling Using Simulated Annealing in a Grid Environment", Lecture Notes in Computer Science,, 2002, Volume 2536/2002, 232-242, DOI: 10.1007/3-540-36133-2_21

Benedict, S. and V. Vasudevan, 2008. "Improving scheduling of scientific workflows using tabu search for computational grids". *Inform. Technol. J.,* 7: 91-97.

Chitra, P. and Venkatesan, P. 2010. "Multiobjective evolutionary computation algorithms for solving task scheduling problem on heterogeneous systems", International Journal of Knowledge-based and Intelligent Engineering Systems archive, 14, Issue 1 (January 2010).

David A. and Van Veldhuizen, 2006. "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art", Summer 2000, Vol. 8, No. 2, Pages 125-147,Posted Online March 13.

Dorigo , M., Maniezzo, V. and A. Colorni, "AntSystem: AnAutocatalytic Optimizing Process", Technical Report, 91:106,1991.

Fatos Xhafa, Leonard Barolli, Juan Antonio Gonzalez and Pawel Jura, 2009. "A Static Benchmarking for Grid Scheduling Problems," waina, pp.170-175, 2009 International Conference on Advanced Information Networking and Applications Workshops.

Fayad, C., Garibaldi, J.M. and Ouelhadj, D. 2007. "Fuzzy Grid Scheduling Using Tabu Search", Nottingham Univ., Nottingham fuzzy systems conference,. FUZZ-IEEE 2007,IEEE International Issue Date 23-26 July 2007.

Kun-Ming Yu and Cheng-Kwan Chen, 2008. "An Evolution-Based Dynamic Scheduling Algorithm in Grid Computing Environment," ISDA, Eighth International Conference on Intelligent Systems Design and Applications, 1, pp.450-455,

Marco Dorigo and Gianni Di Caro , Luca M. Gambardella, 1999. Ant Algorithms for Discrete Optimization", Research Paper,1999.

Xiaohong Kong, Xiqu Chen, Wei Zhang, Guanjun Liu and Hongju Ji, 2009. "A Dynamic Simulated Annealing Algorithm with Self-adaptive Technique for Grid Scheduling," gcis, vol. 1, pp.129-133, 2009 WRI Global Congress on Intelligent Systems.

Yang Gao, Hongqiang Rongand and Joshua Zhexue, 2005. "Adaptive grid job scheduling with genetic algorithms" future Generation Computer Systems,Volume 1, 1 January 2005, pages 151-161.

****