VOL. 12, NO. 17, SEPTEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

AN ADVANCED IWD BASED HYPER-HEURISTIC WORKFLOW SCHEDULING IN COMPUTATIONAL GRID

S. Gokuldev[1], C. Sowntharya[2] and S. Manishankar[1]
[1]Department of Computer Science, Amrita VishwaVidyapeetham, Mysore Campus, Amrita University, Karnataka, India
[2]Department of Computer Science and Engineering, Karpagam College of Engineering, Coimbatore, Tamil Nadu, India
E-Mail:gokuldevs@gmail.com

ABSTRACT

Grid computing paved way for efficient utilization of resources and this is one of the important aspects in the recent research trends. In this paper, an Intelligent Water Drops (IWD) algorithm is proposed for task workflow scheduling with hyper-heuristic search for achieving better performance optimization for workflow distribution in computational grid environment. The challenge is to find the best matching resource among the available free resources for which a grid broker is used. A hyper heuristic search is performed which undergoes multiple rounds of searches for finding the best matching resource by imparting a specific point in time rule. Load balancing is achieved by computing the average computational power of all the resources and grouping the resources. This helps the system to assign various types of jobs to the resources, thus improving the computational efficiency. The performance parameters such as makespan, cost, deadline of jobs and number of searches for finding best optimal resource are considered for performance enhancement. The simulation results of Intelligent Water Drops based Hyper-heuristic algorithm is compared with few of the latest existing works and the proposed algorithm outperforms thus achieving an optimized performance.

Keywords: computational grid, hyper-heuristic search, intelligent water drops, performance optimization, resource broker, workflow scheduling.

1. INTRODUCTION

A Grid [1] is a kind of distributed computing, scalable, geographically distributed, wide area computing infrastructure consisting of grid users, grid resources and resource providers. The grid users are one who wants to submit their jobs to grid. The resources are the networking components, memory, processing power, printers etc. where the computing grid resources are organized virtually [2] which are connected together to form a collection of clusters to form a grid and the resources providers are the individual computing machines which owns the computing resources and provides them to the grid users for computations. The grid and cloud systems comprises of heterogeneous [3], [4] collection of computing resources and the schedulers are decentralized [5] to improve its efficiency whereas the clusters are homogeneous in nature. The system is applicable to execute applications based on high performance computing [6] for managing resources with multiple clusters [7] integrated together for workload analysis [8].

It is the responsibility of the resource broker [9] of the grid system to schedule the jobs to various resources by finding the best matching resource and return the processed jobs back to the users. The task of scheduling the jobs to suitable resources is done by introducing an efficient hyper-heuristic workflow scheduling algorithm which has been implemented in the resource broker. The idea is that, the resource broker may not be able to find a suitable resource by performing single round of search among the resources as the resources are dynamically being allocated to the jobs and hence a hyper heuristic search is implemented. The hyper-heuristic search performs multiple rounds of searches to find the best possible resource avoiding the communication overhead. The challenge lies with the batch of jobs submitted by one or more users; the resource broker allocates the suitable resource among the available resources by scheduling the jobs by considering the size of the job and users QoS constraints for instance the user deadline [10]. The makespan, cost, resource utilization etc. are also computed once the jobs get executed. The objective of the resource broker is that to maintain the updated status of all computing resources available for it. The task is to find the best matching resource through performing multiple searches (hyper heuristic search) with the objective of finding optimal combination. The grid model with a resource broker is depicted in Figure-1.

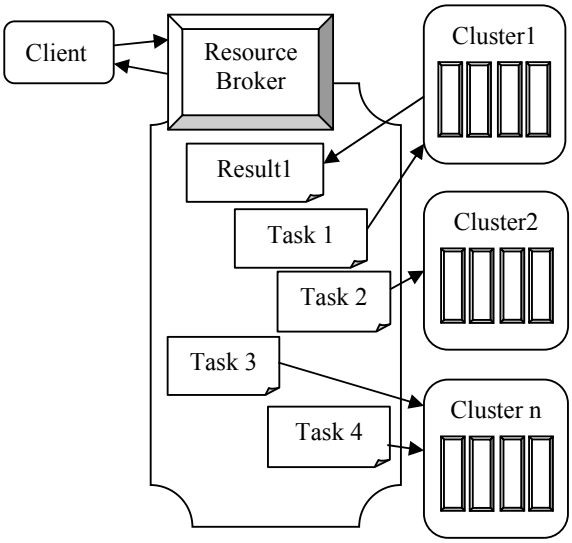

Figure-1. Grid system model.

Intelligent Water Drops algorithm is shown to be extremely efficient in finding optimal solutions in very large search spaces with in a limited amount of time. An Intelligent Water Drops based Hyper-heuristic is a generic method, finds right methodology for a given situation with an acceptable quality. The proposed algorithm is customized to suit the job-resource pattern match in the grid environment. The development of a novel scheduling approach involving the resource brokering concept incorporates the aspect of dynamicity of workflow scheduling in computational grid environment. This makes the resource broker to play a vital and crucial role thus proving its efficiency.

The work is divided in to five sections: the first section gives the introduction to the problem statement. Section 2 represents the various reviews of literature stating all related traditional and conventional scheduling strategies with heuristic or meta-heuristic search algorithms. Section 3 describes the proposed Intelligent Water Drops based Hyper Heuristic algorithm for workflow scheduling. Section 4 deals with the experimental results considering the performance parameters. Section 5 represents the conclusion and future work

2. LITERATURE SURVEY

Few resource brokering architectures along with lot of heuristic and meta-heuristic searches for performing workflow scheduling [11] are considered for the review of literature and few important among them have been described in this section.

Jia Yu & Raj Kumar Buyya (2006) addressed on budget constraint workflow scheduling [12], a meta-heuristic approach, reduces the execution time of jobs with reduced cost. The authors introduced a new genetic algorithm which has a fitness function to achieve a better time minimization with reduced budget. The workflow refinement using Markov decision process helps the genetic algorithm resulting with reduced execution time and budget. The scope is to support various service negotiation models. Dynamic runtime rescheduling can also be included considering the users QoS requirement during resource failures.

Wei-Neng Chen & Jun Zhang (2007) proposed an Ant Colony Optimization approach [13] for scheduling workflow applications. Multiple QoS constraints are considered enabling and guaranteeing quality scheduling. A meta-heuristic ACO helps to perform effective search for finding an optimal resource. The algorithm is found to be efficient in finding optimum solutions considering the entire user preferred QoS constraints [14]. The scope is to perform multiple searches to find optimum solution as the algorithm is found highly efficient for very large grid workflow applications. May not probably be much cost effective for limited workflow applications

Suraj Pandey *et al* (2010) in their work focused on scheduling the resources and to reduce the execution time of applications and the cost. A better balancing of load among the resources is also achieved in a cloud forum by proposing Particle Swarm Optimization based meta-heuristic search [15] process along with best resource selection strategy which resulted in better performance for scheduling applications. The work may be extended for increased number of tasks as well as resources. Integrating the workflow management with better heuristic search for scheduling applications may lead to better performance.

Wang Yong *et al* (2011) proposed Relative Cost - Deadline and Budget Constraint RC-DBC scheduling heuristic [16] for scheduling workflow applications. This helps to deal with sequential workflow application for dependent tasks. The task-starving degree of the resources is indicated and addressed thus achieving reduced waiting time of tasks. A better heuristic approach may be introduced considering few additional parameters to further enhance the performance.

Simrat Kaur &Sarbjeeth Singh (2012) proposed a group based scheduling [17] and compares it with various other scheduling techniques in terms of processing cost and processing time in a dynamic grid computing environment. The proposed algorithm is heuristic in nature with a group based Particle Swarm Optimization technique. The scope is to reduce the complexity of the algorithms and communication overhead. The work can be further extended to consider load balancing in local and global with various load factors.

Rajni & Inderveer Chana (2012) addressed on Bacterial Foraging algorithm [18] for resource scheduling in grid. The algorithm helped to schedule parallel independent jobs which resulted with improved make span and cost. The search includes low level heuristic to associate the matching job with the available resources. As the grid jobs and resources are dynamic in nature, it is always better to propose a search which could contribute on both high level and low level heuristics to find optimal search solutions.

Tarun Goyal *et al* (2013) proposed an approach where the Suffrage heuristic is coupled with Genetic algorithm [19] with an objective of achieving best scheduling of resources for independent jobs in cloud environment. The algorithm achieved an improved make span by reducing the network delay. The work can be extended towards considering certain other important parameters to meet the QoS constraints considering the user deadline.

From the above review of literatures, it is very clear and evident that a perfect heuristic search algorithm is essential to find a perfectly matching resource for the job that needs to be scheduled. Since grid resources are dynamic, autonomous and are heterogeneous in nature, grid can be considered as NP-complete problem and hence requires a effective heuristic search approach and hence the idea is to propose an Intelligent Water Drops algorithm with efficient hyper-heuristic search to map with apt resource thus performing multiple searches

3. METHODOLOGY

There are few optimization techniques for heuristic searches found to be interesting and proven better solutions for the optimization problems. Their efficiency

has been proven which is mostly applicable for a limited search space and few for a larger search spaces.

3.1 Particle Swarm Optimization (PSO)

PSO [20-22] simulates the bird flocking behavior and hence helps in resolving optimization issues. Here, every single resolution is a bird within a search house particle. All particles have its own fitness values evaluated by its fitness function to get optimized and have velocities which directs the fly of particles. It is initialized with a set of random particles treated as solutions and then searches for an optimum solution by updating the generations. Every particle is updated by two best values. The first is the best resolution or fitness that has achieved initially which is represented as P-best.

The second is the one which is partially tracked by the Particle Swarm Optimizer at present among the entire population (globally) and it is represented as G-best. When the particle considers a part of the population, considered to be a topological neighbour, the optimum solution may be a native best represented by l-best.

3.2 Genetic Algorithm (GA)

Genetic Algorithm [23] is useful in providing search techniques leading to quality solutions drawn from very large search space. GA [15] combines the most effective solutions from the past performed searches with the study of recent regions of the solution space. Any solution within the search space of the problem is delineated by an individual chromosome. The algorithm incorporates a population of individuals that evolve over different generations. The quality of the individual within the population is determined by a fitness function and therefore the fitness value represents how best that individual is whereas compared to others within the population.

3.3 Suffrage Algorithm (SA)

Suffrage is a heuristic search algorithm [15] in which a task is randomly assigned to certain resource and if the resource does not accept executing it, then the task suffers. The difference between the least completion time and the second least completion time of each task is termed as franchise worth and the tasks which obtained highest franchise worth takes the precedence throughout the programming. The first and second minimum completion time measures for each job are found in the first step. The franchise value is the difference between these two worths. Appointing the task with highest franchise worth to the corresponding machine with least completion time is the second step.

3.4 Bacterial Foraging Optimization (BFO)

BFO [18] is a population based numerical improvement algorithm which supports the hunting behavior of E.coli bacterium. The behavior of E.coli is that it directs the system to hunt for food. The group of bacterium goes in search of food and decides whether to enter in to food region and then to a new region and so on for obtaining high quality nutrients. The objective is to acquire nutrients in a fashion so as to obtain maximized intake of energy per unit time. The major role playing is based on four main mechanism adapted in this algorithm such as taxis, swarming, replica and elimination dispersal event.

3.5 User Deadline based Heuristic Algorithm (UDH)

A heuristic search based algorithm which produced a better result for heuristic search problems. The system performs a search process for finding an optimal resource for processing the incoming job by considering the QoS constraints. The deadline of the job is given by the user along with the job. Higher priority is set for those jobs which has lesser deadline. An efficient search process is performed to find the optimal resources to execute the task based on the heuristic search process. The system has shown improved results.

It has been observed that finding a perfectly matching resource using heuristic and meta-heuristic for a selected job is highly tedious as these methodologies performs only one search. The best alternate is to replace these heuristics with an hyper heuristic search for obtaining best optimized results. By applying hyper heuristic technique, there are possibilities to achieve better utilization of resources however it may result with increase in time for a job to settle in a most effective resource. Moreover, this can lead to increase in cost to execute a job and a raise in hardware execution time. Clients while submitting jobs expects the minimum cost, on the other hand, the server which allocates resources should invariably reduce the makespan. Hence an attempt is made to find a best matching resource which can be allocated to jobs by proposing an effective scheduling and search optimization technique

4. INTELLIGENT WATER DROPS ALGORITHM WITH HYPER-HEURISTIC SEARCH

The proposed method imparts a specific point in time based rule which a hyper-heuristic methodology is performing multiple searches to find the best matching resources. The proposed system with hyper-heuristic search process is depicted in the Figure-2.

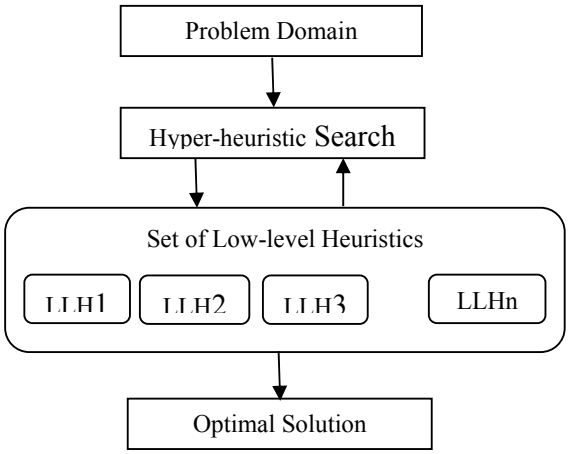

Figure-2. Hyper-heuristic search model.

The Hyper-heuristic search model performs multiple searches for the given problem domain to find the best optimal matching resource for the incoming job. The search is performed to a set of low level heuristic, where the multiple searches are performed for clusters of workstations enabling high computational power. The search process is performed at each cluster for finding the optimal resource and the optimal solution is identified.

The idea is to employ the proposed methodology wherever the user requires the submitted job with minimal completion time. The local scheduler provides roles to find specified resourcesand these jobs are executed with in the period. There may be few cases where, job category which could not be executed within the time bound would extend the time. The average computational power of all the resources is computed which can be allocated for different kinds of jobs which cannot be executed within the specified time bound. This would help the system to result with better resource utilization, thus reducing the CPU execution time. The proposed system initializes the static and dynamic parameters for the resources and then follows the steps.

Algorithm: IWD

1) Static parameters are initialized for resources
2) Dynamic parameters are initialized for resources
3) IWD's are randomly spread over the resources
4) Visited resource list of each IWD is updated
5) Step 'a)' to 'd)' is repeated for the IWD's with partial solutions:

a) For the IWD residing in resource 'i': the next resource 'j' is chosen,
b) Execution time is updated for each IWD moving from node 'i' to node 'j'
c) Makespan is computed
d) Makespan is updated

6) Minimized Execution time is identified, among the execution time of IWD
7) Iteration number is incremented
8) Optimal resource is obtained through identifying minimized execution time.

Description: Once the resources are initialized both statically and dynamically, the Intelligent Water Drops are randomly spread over the resources. The associated water drops with in a coverage area forms the cluster and the resources which are within the coverage area are associated to that cluster. Similarly multiple clusters are formed which holds a pool of computing resources. The search starts from any computing resources and the visited resource list of each intelligent water drop is updated. For the intelligent water drop residing in resource 'i', the next resource 'j' is chosen and the execution time of each intelligent water drop moving from node 'i' to node 'j' is updated. The makespan is also computed and updated. The minimum of all execution time is identified among all the execution time listed. The number of iterations to find the best optimal resources is also noted. Through this the optimal resource is obtained by identifying the minimized execution time and this resource is the best matching resource for executing this job.

The IWD algorithm provides good quality solutions using average values. The jobs which could not match with the existing computing resources for execution, shows that job requires resources with higher computational power and hence the grouping of resources takes place by computing the average computational power of the resources of each cluster. Thus the makespan is reduced for the jobs executed. Since the average computational power is computed by grouping the resources, the optimal utilization of computing power with the available resources ensures the reduced cost. The IWD algorithm has fast convergence when compared to other methods because of its quick search process with intelligent water drops and thus provided optimal solution and proves performance optimization.

5. EXPERIMENTAL ANALYSIS

The initial simulation setup is designed with five clusters holding the corresponding resources which are listed in the below Table-1.

Table-1. Number of clusters, CPU'S and service rates.

No. of clusters	No. of CPUs	Service rate
C1	88	1.93
C2	97	2.1
C3	106	1.9
C4	110	1.5
C5	112	2.2

The Intelligent Water Drops algorithm is tested with varying workloads provided with input parameters such as number of jobs, number of resources, number of searches to be performed and the deadline of the job to be executed. The simulation result obtained is shown in Figure-3.

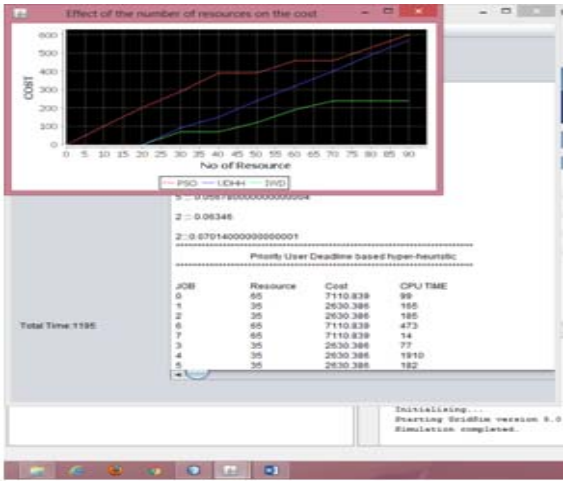

Figure-3. Execution time of IWD.

The simulation is tested with varying workloads by assigning the required number of jobs to the grid environment.

Table-2. Execution time of IWD with Hyper-heuristic.

Jobs	Execution time (milliseconds)	Cost
80	403	400
160	411	360
320	540	220
640	1427	430
1280	1656	480
2560	1887	390
5120	1517	330
10240	1092	495
20480	1350	430
40960	783	345
81920	899	495
161440	1195	230

As the system has a middleware to route the jobs to the corresponding cluster holding the pool of computing resources, the jobs are assigned to the perfect matching resources by the resource broker performing multiple searches visiting all idle resources. The simulation results are obtained for varying workloads and are represented in Table-2.

The simulation work is tested with 100 computing resources with deadline given as 1000 and the number of searches is set to 2. The corresponding execution time for varying incoming jobs is experimented.

6. CONCLUSION AND FUTURE SCOPE

The Intelligent Water Drops (IWD) algorithm is proposed for scheduling the resources to the incoming job by identifying the best matching resources among the available idle resources. The proposed algorithm works based on performing hyper-heuristic search thus promoting multiple rounds of searches for finding the apt resources by imparting a specific point in time rules. Since the incoming workload is scheduled to the idle or resources with less workload, a better load balancing is also achieved. It also enables the system to compute the average computational power and to group the resources for assigning higher loads. The system proves a better stability to deal with heavy loads by grouping those computing resources with improved efficiency. The overall performance of the system is compared with various other systems and the proposed proved the improved efficiency.

The future scope may include methodologies for an improved system performance considering various QoS parameters.

REFERENCES

[1] Buyya R. and Venugopal S. 2005. A gentle introduction to grid computing and technologies. Database. 2, p.R3.

[2] Foster I., Kesselman C. and Tuecke S. 2001. The anatomy of the grid: Enabling scalable virtual organizations. International journal of high performance computing applications. 15(3): 200-222.

[3] Ekmecic I., Tartalja I. and Milutinovic V. 1996. A survey of heterogeneous computing: concepts and systems. Proceedings of the IEEE. 84(8): 1127-1144.

[4] Fard H.M., Prodan R., Barrionuevo J.J.D. and Fahringer T. 2012. A multi-objective approach for workflow scheduling in heterogeneous environments. In: Proceedings of the 12thIEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 300-309). IEEE Computer Society.

[5] Vahdat-Nejad H. and Zamanifar K. 2009. A New Randomized Algorithm for Handling Scheduling Conflicts in Grids. Advances in Electrical and Computer Engineering. 9(3): 22-26.

[6] He L., Jarvis S.A., Spooner D.P., Chen X. and Nudd G.R. 2004. Hybrid performance-based workload management for multiclusters and grids. IEE Proceedings-Software. 151(5): 224-231.

[7] N. Indrajith, Manishankar S. and Monika B. R. 2016. Alpha scheduler approach to enhance security and high performance in cluster environment. Journal of Theoretical and Applied Information Technology. 87(1).

[8] Medernach E. 2005. Workload analysis of a cluster in a grid environment. In Workshop on Job Scheduling Strategies for Parallel Processing (pp. 36-61). Springer Berlin Heidelberg.

[9] Afgan E. 2004. Role of the Resource Broker in the Grid. In Proceedings of the 42nd annual southeast regional conference (pp. 299-300). ACM.

[10] Abrishami S., Naghibzadeh M. and Epema D.H. 2013. Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds. Future Generation Computer Systems. 29(1): pp.158-169.

[11] Singh L. and Singh S. 2013. A survey of workflow scheduling algorithms and research issues. International Journal of Computer Applications. 74(15).

[12] Yu J. and Buyya R. 2006. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. Scientific Programming. 14(3-4): 217-230.

[13] Chen W.N., Zhang J. and Yu Y. 2007. Workflow scheduling in grids: an ant colony optimization approach. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 3308-3315). IEEE.

[14] Chen W.N. and Zhang J. 2009. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 39(1): 29-43.

[15] Pandey,S., Wu L., Guru S.M. and Buyya R. 2010. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In Advanced information networking and applications (AINA), 2010 24th IEEE international conference on (pp. 400-407). IEEE.

[16] Wang Y., Bahati R.M. and Bauer M.A. 2011. A novel deadline and budget constrained scheduling heuristics for computational grids. Journal of Central South University of Technology. 18(2): 465-472.

[17] Kaur S. and Singh S. 2012. Comparative analysis of job grouping based scheduling strategies in grid computing. International Journal of Computer Applications. 43(15): 28-35.

[18] Chana I. 2013. Bacterial foraging based hyper-heuristic for resource scheduling in grid computing. Future Generation Computer Systems. 29(3): 751-762.

[19] Goyal T. and Agrawal A. 2013. Host scheduling algorithm using genetic algorithm in cloud computing environment. International Journal of Research in Engineering & Technology (IJRET). 1: 7-12.

[20] Tao F., Zhao D., Hu Y. and Zhou Z. 2008. Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. IEEE Transactions on industrial informatics. 4(4): 315-327.

[21] Tao Q., Chang H., Yi Y., G, C. and Yu Y. 2009. QoS constrained grid workflow scheduling optimization based on a novel PSO algorithm. In Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on (pp. 153-159). IEEE.

[22] Wu Z., Ni Z., Gu L. and Liu X. 2010. A revised discrete particle swarm optimization for cloud workflow scheduling. In Computational Intelligence and Security (CIS), 2010 International Conference on (pp. 184-188). IEEE.

[23] Talukder A.K.M., Kirley M. and Buyya R. 2009. Multiobjective differential evolution for scheduling workflow applications on global grids. Concurrency and Computation: Practice and Experience. 21(13): 1742-1756.