A Survey on Determining the Characteristics of Deadlock, Scheduling and Workflow Instances and Providing New Algorithms for the Issues Arising in Hybrid Cloud

B. Arunkumar,
Research Scholar, Dept. of CSE,
Karpagam University
Coimbatore.

T. Ravichandran, Ph.D.
Principal,
Hindustan Institute of Technology,
Coimbatore.

ABSTRACT
Effective scheduling is a key concern for the execution of performance driven cloud applications. We have analyzed the various characteristics of deadlock, its occurrences and the strategies to overcome it. We have also discussed some of the algorithms used for scheduling and workflow in hybrid and grid computing. To overcome this, we can use dynamic-critical path algorithm and Cost based workflow scheduling algorithm. The goal of Grid computing is to aggregate the power of widely distributed resources. Grid is not capable of optimizing multiple scheduling but it is possible in cloud. Cloud computing is an emerging technology that provide the users on demand service. One of the issues is the occurrence of deadlock because many users are requesting for the same resource. To improve the performance of the entire system a high degree of concurrency is obtained by running multiple instances at the same time. On the other hand, since the amount of storage is limited on most systems, deadlock due to numerous storage requests would-be a problem.

Keywords
workflows, task scheduling, instances, deadlock, dependency.

1. INTRODUCTION
In recent world time plays a major impact of finishing their job as soon as possible, so in order to provide the users with all the requested resources at their hand is also not possible. But it is possible by using some of the complex algorithms like cleanup job, progressive reservation approach and some of the scheduling algorithm to avoid some of the conflicts arising in the system. Cloud computing holds a promise to deliver large-scale utility computing services to a wide range of consumers. Through the creation of *hybrid clouds*, one can use this internal infrastructure with public cloud resources [1].Grid computing provides the solution for large-scale problems [2],[5].Various scheduling algorithm used in hybrid and grid computing. The genetic and List scheduling algorithm proved to be efficient [3]. [6] Scientific workflow system are playing a major role nowadays. They have become a tool for many applications that helps in the execution of many complex problems. [11]Basically scheduling these instances to a set of processors is the major problem which are been broadly classified into two types: job scheduling which are independent jobs can improve the systems performance only dynamically but whereas the latter one Scheduling and mapping involves allocation of multiple tasks of a single parallel program in order to minimize the completion time which is accomplished in both static and dynamic run time. Multiprocessor Scheduling has been source of challenging problem, the general problem of multiprocessor scheduling can be stated as scheduling as set of partially ordered computational task [4].In Workflow scheduling, workflow management system defines manage and executes workflows on computing resources [5].In static scheduling which is usually done during compile time the characteristics of parallel program or before the program executes.DAG can be effectively used in static scheduling [7]. Task scheduling ready to run task to a host based on information into only about the task. It describes CPOP (Critical Path on a Processor), PETS (Performance Effective Task Scheduling) algorithms [8], [10]. In this paper, we proposed dynamic critical path for effective workflow scheduling. Several works have been proposed to address scheduling problems based on users' deadline constraint. Nimrod-G [6] schedules independent tasks for parameter-sweep applications to meet users' deadline. In contrast, the scheduling algorithm developed in the paper aims to schedule tasks with certain dependencies. A market-based workflow management system [12] locates an optimal bid based on the assigned deadline of each individual task. However, in most situations users may only want to specify a deadline for the entire workflow execution. Therefore, we focus on how to assign sub deadlines of tasks to meet the overall deadline.

2. SCHEDULING ALGORITHM
Scheduling is the major task of deciding upon which task is to allocate to which resource and in which order. The major impact of deciding this order is based on some algorithms. The cost –optimal scheduling algorithm is also known as Optimal Virtual Machine Algorithm. This algorithm can minimize the cost spending in each payment plan. It is only for hosting virtual machine in a multiple cloud. It determines the cost-optimal allocation of reserved versus on-demand .The optimal algorithm attempt to schedule individual applications and tasks on a hybrid cloud [1]. The Dominant sequence clustering (DSC) algorithm, the priority of the tasks are dynamically computed as the summing up of their top level and bottom level[2].Critical path of a clustering graph is the longest path in that graph. This longest path combines both Non-Zero communication edge cost and task weights. The *top level* and *bottom level* are the sum of the computation and communication costs along the longest path from the given task. While the bottom level is statically computed at the beginning, the top level is computed incrementally during the scheduling process. Tasks are scheduled in the order of their priorities.

2.1 List Scheduling Algorithm

The List Scheduling is a large class of algorithm. The terms, task and module deals with the communications and busses indirectly. (i.e.) schedules tasks one by one until a valid schedule for all the tasks are obtained. It has User selection, Resource selection, Updating of the sets [3]. In List scheduling algorithms, first it schedules the critical users and assigns them to the right resource. A free set is implemented as a tree of intervals for improving the efficiency and sometimes only the last interval is maintained for simplicity. Genetic algorithm is developed to solve the multiprocessor scheduling problem [4]. Once the representation of the search node is done based on the order of the task, the method for initial population is generated. To evaluate the search nodes fitness functions can be used. The Genetic algorithm selects from a population of search node instead of a single node. The workflow scheduling algorithm describes Batch Mode scheduling[5] which uses min-min algorithm and also describes dependency batch mode using Hybrid Heuristics algorithm. In min-min each step computes ECT (Early Completion Time) and assigns the task on the resource which is expected to complete it earlier. The heuristic combines dependency mode and batch mode and computes the rank values of each task and ranks and finally all tasks in the decreasing order of their rank values are determined. Dynamic critical-path scheduling algorithm determines efficient mapping of tasks by calculating the critical path in the workflow task graph at every step. In DCP, First in determines the critical path of the task graph and selects the next node. Second, it rearranges the schedule on each processor dynamically (i.e.) position of the nodes in the partial schedule are not fixed. Third, select the suitable processor for a node .It suitable for a wide range of graph structures [6], [9].

2.2 Static Scheduling Algorithm

It has a large classification of algorithm based on static model known as: HU'S algorithm uses a polynomial-time algorithm to construct optimal schedules for in-tree structured DAGs with unit computations and without communication. In the scheduling process, each level of tasks is assigned to the available processors. Coffman and graham's algorithm is used for generating optimal schedules for arbitrary structured Task graphs with unit-weighted tasks and zero-weighted edges to a two-processor system. EZ algorithm (Edge-zeroing) selects clusters for merging based on edge weights. At each step, the algorithm finds the edge with the largest weight. The two clusters incident by the edge will be merged if the merging does not increase the completion time.

2.3 Workflow based Algorithm

Task based algorithm allocates task to a host based on information about the task [8]. Workflow based algorithm search for an efficient allocation for the whole workflow. The Task based algorithms makes local decisions about which job to send to which resource. In the Workflow based algorithm, local search algorithm is used for allocation of a workflow. A number of iterations are made to find the best possible mapping of jobs to resources for a given workflow. Three algorithms are used for task scheduling; HEFT algorithm uses a recursive procedure to compute the rank of task by traversing the graph upwards from the exit task. CPOP algorithm calculates the traversing task graph downwards from the entry task. PETS algorithm uses DAG structure, gives the best performance and cost metrics compare to HEFT and CPOP.

2.4 Cost based Workflow Scheduling Algorithm

The performance of any cluster of workstations can be improved when its resources are used wisely. A poor job assignment strategy can result in heavily unbalanced loads and thrashing machines, which cripples the cluster's computational power. Resources can be used more efficiently if the cluster can migrate the jobs and moving them transparently from one machine to another. However, even systems that can reassign jobs can still benefit from a carefully-chosen assignment strategy. The algorithm minimizes execution cost while meeting the deadline for delivering results. It can also adapt to the delays of service executions by rescheduling unexecuted tasks [11]. It is denoted as Deadline-MDP with two other scheduling approaches: Greedy- Cost and Deadline-Level. These two approaches are derived from the cost optimization algorithm in Nimrod-G, which is initially designed for scheduling independent tasks on Grids.

2.5 Deadlock Avoidance Algorithms

High Performance computing applications requires high speed processors and may not be dependent on one another. Another important consideration is the optimization of effective storage utilization since, given the ease with which parameter based studies can generate a large number of jobs and workflow instances, storage can often be a constraining resource. Therefore, a scheduler should both deal with potential deadlock issues due to oversubscribed resources and maximize performance, as measured by *makespan.* Parameter sweep is an important example of parallel jobs for cluster and grid based projects. The algorithm is based on two concepts:

First algorithm helps in minimizing the makespan. Second the algorithm decides on dynamically adjusted priority. The algorithm finally focuses on improving the performance by finishing the work before the order of the completion of the workflow itself. When Parameter sweeping is executed numerous times over a workflow by doing this it can easily execute multiple instances of a workflow in parallel. The main problem is that same set of resources are requested by the instance called resource competition problem, so to overcome this a new scheduling algorithm is proposed which involves supporting multiple workflow and avoid this allocating problem. Ping problem requires the allocation of multiple interacting tasks of a single parallel program in order to minimize the completion time on the parallel computer system. Static Scheduling algorithm Basically scheduling these instances to a set of processors is the major problem which are been broadly classified into two types: job scheduling which are independent jobs can improve the systems performance only dynamically but whereas the latter one Scheduling and mapping involves allocation of multiple tasks of a single parallel program in order to minimize the completion time which is accomplished in both static and dynamic run time.

2.6 DAG Scheduling Algorithm

The main objective of this algorithm is to reduce the overall program's finish time which is otherwise called the makespan or schedule length. But unbounded number of cluster (UNC), scheduling algorithms are some algorithms that consider the availability of unlimited number of processors while bounded number of processors (BNP) scheduling algorithms work on considering only limited number of processors. The advantage of using UNC based algorithm is that it takes advantage of using more processors inorder to reduce the makespan but it is accomplished only if it obtains post processing step which is not required in the BNP algorithms. Most of these algorithms basically employ the following three-step approaches:

- First deciding upon new priorities of all unscheduled nodes;
- Secondly selecting a particular node that has the highest priority for scheduling;
- Finally allocate the node to one of the processors available which allows the earliest start-time.

[13] Recent days issues of optimizing disk usage and scheduling large scale scientific workflows are emerging as a major problem. One of the solutions is that decreasing the amount of space a workflow will occupy during its execution, this is done by removing the data files which are no longer needed called the cleanup job, by which means this method helps in reducing the overall execution time and so the performance is greatly increased. Basically the algorithm follow three steps which involves identifying the available resources and allocate the task to resource that has the highest finish time and finally the cleanup job is done to remove the extra files.

2.7 Heterogeneous Master-Slave Platforms

Difficulty arises when independent tasks depend upon files and difficulty may arises from the fact that some files may well be shared by several tasks are scheduled on heterogeneous clusters. These files are the input to the master whose work is to decide upon sending which file to which slave and in which order. Many slaves are ready to receive the work so by this mean the execution time is decreased

2.8 Extension to the Banker's Algorithm to Avoid Deadlock

[14] A new extension to the banker's algorithm that helps to avoid the deadlock. They have proposed a quadratic time algorithm that decomposes each flow graph into nested family of regions. This algorithm will allocate the resources before their control is released. During system calls each process at runtime enters a new region and applies the original banker's algorithm which has the potential to achieve better resource utilization by testing whether the system is in safe state or not by using the information available in "localized approximate maximum claims". [15] Recent developments are facing a major impact over improving the utilization of system resources more effectively and hence reduce the cost to users. In any operating system of this type the problem of deadlock is a major concern. A system will become complex if it has different resource types will surely result in the deadly embracing problem (deadlock).based on the studies made following are the situations deadlock occurs.

- A particular task claims the control of resource they require exclusively known as mutual exclusion.
- A single task will request for more resources even though it has some resource which is termed as Wait for condition.
- It is clear that no resource should be forcibly removed before the completion of the task it is meant for, known as No preemption condition.
- A chain is created where a single task holding one or more resources which are been requested by next task, called as Circular wait condition.

All the above conditions are facing a major issue in the field of scientific applications, which can be overcome if the system follows the basic ways as such said by Havender [4]

- Each of the tasks in the workflow can request only one time. This condition overcomes the drawback of Wait for condition.
- If a task is holding a resource it cannot request for more resource. It is a must that the task should release and request. By this rule the No preemption condition is wrecked.
- Similarly the order of execution of the tasks should be strictly maintained.

An alternative search algorithm that is more flexible only if the number of safe sequences is relatively small. This approach will maintain a list of safe sequences and updating them periodically. It makes a clear vision that only the first element of the safe sequence will decide whether the request can be granted or not. Even though as the number of safe sequences are increasing the storage and updating problems make this approach less efficient and the transition between the methods may be mandatory. But even though complete deadlock is avoided it is a must that for the sake of efficiency it is important to prevent the system from entering into the state of "near deadlock" in which progress can be made only by granting request from one task at a time. Even though there are many techniques to avoid deadlocks it cannot be fully omitted.

2.9 Transparent Dataflow Detection

A workflow will consists of a set of control or data dependent jobs to carry out a well defined scientific process. It is mandatory that a job has to finish its process before the second job starts its execution which is termed as control flow. Similarly a job can start its process only if it has all necessary input data. The former problem can be resolved with two algorithms namely[16].

- Dataflow based aggregate requests (DAR)

  This algorithm is designed to improve the inter-instance concurrency. Here storage allocation and de-allocation of workflow instances are considered as a single unit. The banker's algorithm concept is been integrated in a way that maximum claim of the dynamically grouping up the jobs that have not yet completed its execution.

- Dataflow based topological ordering (DTO)

  This algorithm is designed in a way to improve intra-instance concurrency. Here storage allocation de-allocation of each job is considered as a single unit. Each job is localized to requested job for safety check which is to avoid occurrence of deadlock.

The latter problem of collecting the dataflow information is resolved by a new technique called WaFs i.e., workflow aware file system which is based on a client/server architecture that integrates the file systems and batch schedulers. A new Versioned Namespace Manager (VNM) is built on top of an existing file system to collect the dataflow information of each instance and provide service to the client. WaFs is designed such a way that it can automatically collect the dataflow information in real applications. They have discussed about the value of dataflow information in workflow scheduling.

2.10 Data Flow Detection
Computing world is meeting a major issue of deciding when to complete the job and goto the next job. But these jobs are meeting a major problem known as "filename conflicts" between concurrent workflows running parallel in a system, which may result in unsafe state condition. A two benefit algorithm namely WaFs is found to overcome some of the above mentioned strategy [17]. First to maximize the concurrency of the workflow instance separate namespaces are constructed on a per-instance basis. Secondly the trade-offs is made between the makespan and storage overhead by exploiting the inferred dataflow information. So it is made possible to provide new scheduling policies and Versioned Namespace (VNS), Overwrite-safe concurrency (OSC) and hybrids by WaFs.

3. CONCLUSION
In this paper we have discussed the different type of task scheduling algorithms their classifications and the way to resolve the system when deadlock arises. Dynamic Critical-Path Scheduling Algorithm gives high performance compared to other Scheduling Algorithms. We have also described the different strategies that can be used to solve the occurrences of deadlocks and various algorithms.

4. REFERENCES
[1] T. Werner, "Target gene identification from expression array data by promoter analysis," Biomolecular Engineering, vol. 17, pp. 87–94, 2001.

[2] D. Szafron, P. Lu, R. Greiner, D. Wishart, B. Poulin, R. Eisner, Z. Lu, J. Anvik, C. Macdonell, A. Fyshe, and D. Meeuwis, "Proteome analyst: Custom predictions with explanations in a webbased tool for high-throughput proteome annotations," Nucleic Acids Research, vol. 32, pp. W365–W371, 7 2004, http://webdocs. cs.ualberta.ca/∼bioinfo/PA/.

[3] GROMACS, http://www.gromacs.org.

[4] M. Schmidt, K. Baldridge, J. Boatz, S. Elbert,M. Gordon, J. Jensen, S. Koseki, N. Matsunaga, and J. Montgomery, "The general atomic and molecular electronic structure system," Journal of Computational Chemistry, vol. 14, pp. 1347–1363, 1993,http://www.msg.ameslab.gov/GAMESS/GAMESS. html.

[5] B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the kepler system," Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, 2005.

[6] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-science: An overview of workflow system features and capabilities," Future Gener. Comput. Syst., vol. 25, no. 5, pp. 528– 540, May 2009.

[7] K. Srimanotham and V. Muangsin, "Scheduling workflow-based parameter-sweep applications with best-intermediate-result-first heuristic," in Cluster Computing, IEEE International Conference on. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 1–6.

[8] S. Smanchat, M. Indrawan, S. Ling, C. Enticott, and D. Abramson, "Scheduling multiple parameter sweep workflow instances on the grid," in Proceedings of the 2009 Fifth IEEE International Conference on e-Science, ser. E-SCIENCE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 300–306.

[9] D. Abramson, B. Bethwaite, C. Enticott, S. Garic, and T. Peachey, "Parameter exploration in science and engineering using manytask computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 960–973, 2011.

[10] A. Mandal, K. Kennedy, C. Koelbel, B. Liu, and L. Johnsson, "Scheduling strategies for mapping application workflows onto the grid," in Proceedings of the 14th International Symposium on High Performance Distributed Computing (HPDC), Research Triangle Park, NC, USA, July 2005, pp. 125–134.

[11] Y.-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," ACM Computing Survey, vol. 31, no. 4, pp. 406–471, September 1999.

[12] M. Wieczorek, M. Siddiqui, A. Villazon, R. Prodan, and T. Fahringer, "Applying advance reservation to increase predictability of workflow execution on the grid," in Proceedings of the 2nd International Conference on e-Science and Grid Computing, Dec. 2006, pp. 82–82

[13] A. Ramakrishnan, G. Singh, H. Zhao, E. Deelman, R. Sakellariou, K. Vahi, K. Blackburn, D. Mayers, and M. Samidi, "Scheduling data-intensive workflows onto storage-constrained distributed resources," in Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid, 2007, pp. 401–409

[14] S.-D. Lang, "An extended banker's algorithm for deadlock avoidance," IEEE Transactions on Software Engineering, vol. 25, no. 3, pp. 428–432, May/June 1999.

[15] A. Shoshani and E. Coffman, "Sequencing tasks in multi-process, multiple resource systems to avoid deadlocks," in Proceedings of the 11th Annual Symposium on Switching and Automata Theory,October 1970, pp. 225–233

[16] Y. Wang, "Transparent dataflow detection and use in workflow scheduling: Concurrency and deadlock avoidance," 2008, PhD thesis, University of Alberta, Canada.

[17] Y. Wang and P. Lu, "Dataflow detection and applications to workflow scheduling," Concurrency and Computation: Practice and Experience, vol. 23, no. 11, pp. 1261–1283, 2011.