ISSN No. 0976-5697

Volume 4, No. 9, July-August 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# Genuine Character Foundation of Scheduling and its Challenges in Grid Computing

Anil Kumar*, Ajay Jangra
Computer Science and Engineering Department, University Institute of Engineering and Technology,
Kurukshetra University, INDIA.
anilsaini303@gmail.com, ajay.jangra@uietkuk.org

***Abstract:*** Scheduling in Grid computing system is used to schedule tasks in such a way that perfect resources allocated to the task submitted by the user. The main aim of scheduling is that it reaches to its goal of highest possible system outturn and mapping the application needed with the available computational resources. In this paper, we will review the introduction of grid scheduling, framework of Grid scheduling, types of grid scheduling and scheduler organization's characteristics. Also here, basic foundation of grid scheduling is explained that will benefit the researchers to carry out their further work in the area of grid scheduling.

***Keywords:*** Grid scheduling, types of scheduling, scheduling approaches, scheduler organization.

## I. INTRODUCTION

Grid is a collaborative infrastructure where different administrative organizations having heterogeneous resources, are connected to each other virtually to achieve a common goal [1]. Grid computing is a kind of distributed computing but it is geographically independent and share information to solve various perplexed problems and develop large scale applications [2]. In this paper, basic foundation of grid scheduling is explained that will benefit the researchers to carry out their further work in the area of grid scheduling. Scheduling in Grid computing system is used to schedule tasks in such a way that perfect resources allocated to the task submitted by the user. The main aim of scheduling is that it reaches to its goal of highest possible system outturn and mapping the application needed with the available computing resources [3]. Grid scheduling process undergoes through a series of 3 stages for scheduling a job when multiple sites are involved in it. Stage one resource discovery, it involves a user to make a list of potential resources to use; stage two, it involves gathering information about the enlisted resources and choosing a best set and at third the user runs the job [4]. Grid computing is a most efficient computing epitome where applications, records and IT services are provided over the internet and the fastest delivering on demand solution of submitted problems. Grid computing has come out to be an interesting and beneficial way of changing the whole computing Schedulers for grid computing determine on which processing resource jobs of a workflow should be allocated. Scheduling theory for grid computing is in advance a lot of awareness with increasing popularity in this grid era. Service provider like GARUDA[5], Healthgrid[6] etc. to ensure that income are utilized to their fullest and best capacity so that resource power is not left unused.

## II. GRID SCHEDULING PROCESS

Grid is an environment in which one or more user jobs can be submitted for solving the problem without knowing the resources location or even who own that resources. Or we can say that "Grid scheduling [7] is the process of scheduling applications over Grid resources". Grid scheduler is different from local scheduler. A local scheduler can only manages a single site or cluster and usually have its own resource.

### *A. Stage 1: Resource Discovery:*

Resource discovery [8] involves selecting a set of resources to investigate in more detail by the user; in stage two information gathering. In beginning, the potential set of resources is empty and at the end, the potential set of resources contains some set that has passed a minimal feasibility requirement. It is mostly performed by users in three steps:-

a. ***Authorization filtering:*** In it as an assumption, it is taken as the user will know which resources have to be accessed by him in terms of basic service policies. A list of accessible machines or resources will be available at the end of this step.

b. ***Application Requirement Definition:*** For further process, some minimal set of job requirements must be specified by user some in order to further filter the set of feasible resources. The possible set of job requirements can be very broad and vary significantly between jobs. It may include static and dynamic details. Static details such as operating system or hardware for which a binary value of the code is available. Dynamic details such as e.g. a minimum RAM requirement, connectivity needed. Authorization filtering expected to include all information that should be specified to make sure that the job could be matched to a set of resources.

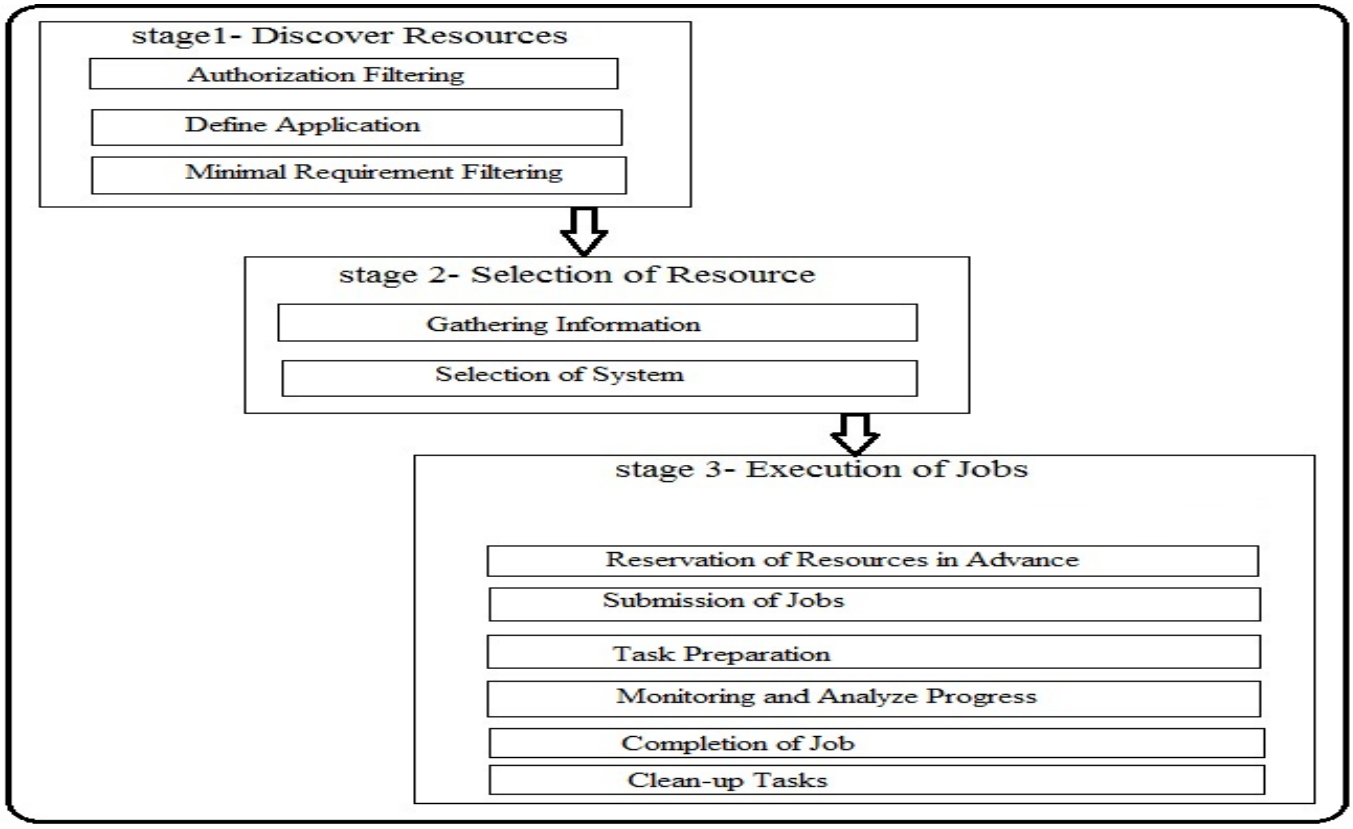


Figure 1: Grid Scheduling Process

c. ***Minimal Requirement Filtering:*** This step is to filter out the resources that do not meet the minimal job requirements. In this we include the minimal set of requirements the job has and a user accessible set of resources is given. In this step, user eliminates the resources that do not meet the job requirements after having undergoing through the list of resources. It may also be concluded as gathering more detailed information about each resource.

### *B. Stage 2: System Selection:*

In this stage a group of possible resources is given all of which meet the minimum requirements for the job, a single resource (or single resource set) must be selected on the basis to best suited for the job. This is generally done in two steps [8]:

a. ***Gathering Information (QUERY):*** In the order to make the best possible match of resources for jobs a user must need to gather dynamic information for the question or queries arise about the resources. As application and resources may in questions so there is a need of different information. For example: the case of finding the best single resource to run a job. If the machine has queues then user might want to know the load on the various machine(s) and queue lengths. In addition to this, there is significant role to play by physical characteristics and software requirements, just liking checking that is their enough space for data on the disk etc. then location/connectivity issues are there that is the machine close enough to the stored data. All of these issues increase with the increase in the resources or for multiple resources.

b. ***Select the system to run on:*** After analyzing the information that is gathered in previous steps, on the basis of which a decision is made that which resource or the set of resources should the user, submit for the job. In this step, it can be done in various ways. Note that this does not give a situation to view execution, where a job is submitted to multiple resources and when one begins to run the other submissions is cancelled.

### *C. Stage 3: Run the Job:*

In this stage of scheduling is for running a job. This involves a series of steps [8]:

a. ***Make an Advance Reservation (Optional):*** In this step, in the manner to make the best use of all given resources, system and part, they will have to be reserved in advance. However depending on the resource, this may be done with mechanical means which is opposed to human means, this can be easy or hard to do and the cost factor may or may not expire the reservations.

b. ***Submit Job to Resources:*** In this the application are allocated with the resources. This may include as easy as running a single command or as complicated as running a ordered number of scripts, and May or may not include setup or staging.

c. ***Preparation Tasks:*** This stage may include setup, reservation claiming, or other operations needed to ready the resource to run the application. One of the first attempts was made by America's National Aeronautics and Space Agency (NASA) of writing a scheduler to run over multiple machines was considered unsuccessful because it did not address the need to stage files automatically.

d. ***Monitor Progress:*** This stage is used by the user to monitor the progress of their application depending on the application and its running time.

*e.* ***Find out if Job is done:*** This stage is check whether the job is finished or not. When the job is finished, the user needs to be notified it.

*f.* ***Completion Tasks:*** This stage is undergoes an analysis after a job is run, the user may need to retrieve files from that various resources in order to do analysis on the results, break down the environment and remove temporary settings etc.

## III. COMPARISON OF GRID SCHEDULING ALGORITHMS WITH PARALLEL AND DISTRIBUTED APPROACHES

Distributed approach compared with Centralized approach: In dynamic scheduling approach, the responsibility for performing global scheduling decisions might be concerned with one centralized common scheduler, or can be shared by more than one distributed schedulers. The centralized approach has the benefit of ease of its implementation, but it has deficiency of lack of scalability, fault tolerance and probability of acting as a performance chokepoint.

In [9], Casavant et al described a hierarchical assortment for scheduling algorithms in a common general-purpose parallel and distributed computing systems. Since Grid is an efficient system, where scheduling algorithms in Grid can be treated as a subset of scheduling algorithms for parallel and distributed computing. From the top to the bottom, this subset can be identified as follows [10].

Local scheduling vs. Global scheduling: At the top level of hierarchical assortment, the local scheduling scheme shows how the jobs submitted to a single CPU and executed by the processor. On the other side, global scheduling scheme collect updated status about the grid resources to allocate jobs to multiple processors to achieve minimum completion time objective.

Static approach vs. Dynamic approach: The next respective level in the hierarchy (under the Global scheduling) gives options to choose one out of static approach and dynamic scheduling. Available options show that at which moment the scheduling decisions are made. In the case of static scheduling, information concerned with all resources in the Grid as well as all the processes in an application is assumed to be available by the time, the application is scheduled. But, in the case of dynamic scheduling, the basic idea is to perform task allocation is vary as the application executes. This is helpful when it is impossible to analyze and 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 environment. These variances introduced forms of dynamism into the running process [11]. Both static and dynamic Scheduling are openly and widely taken in Grid computing.

Optimal scheme vs. Sub-optimal scheme: when schedulers know updated status of all available grid resources and the processes, an optimal assignment could be done based on some pre-specified function, such as minimum make span and maximum resource utilization. But due to the NP-Complete nature of scheduling algorithms and the difficulty in Grid scenarios to make appropriate and reasonable assumptions which are needed to justify the optimality of an algorithm, current status of research tends to find suboptimal solutions, which can be proceed into the following two general categories.

An Approximate estimation compared with a Heuristic program: The approximated estimation algorithms use formal computational models, but instead of searching the complete solution space for an optimal solution, they are agreed when a status of solution "good " is found. In the case where a metric is available for assessing a solution, this technique can be used to decrease the time taken to find an acceptable schedule. The factors which describe whether this approach is beneficial to include availability of a function to assess a solution.

a. Time needed to assess a relative solution.
b. Ability to measure the value of an optimal solution according to some heuristics or metric.
c. Weed out unwanted solution space with the help of available mechanism smartly.
d. Those algorithms comes into suboptimal category, more adaptive to the Grid scenarios, which provide the most real time presumptions of prior information about relative process and system attributes, called heuristic . It also resolve the scheduling problem which cannot give optimal answers but only demand the most appropriate and reasonable amount of cost and other system resources to perform their function. The assessment of such kind of solution is usually based on experiments in the real time environment or on simulation.

## IV. TYPES OF GRID SCHEDULING

A number of efforts have been made in attempting to design scheduling systems for Grids, each of which having its unique characters. A comprehensive set of various types of scheduling [10] are:-

### *A. Knowledge of Application:*

*a.* ***Application Level Scheduling:*** The application level scheduling scheme is based on making as much as possible use of knowledge of applications. The scheduling results in custom schedulers in which each application attempting to maximize application performance. The performance is measured as runtime or speedup, with little regard to overall system performance. The complexity arise in application level scheduling is with the order of the applications considered. Example of Application Level scheduling is Apple system that uses the application level scheduling scheme.

*b.* ***Resource Level Scheduling:*** In the Resource level scheduling there is not much use of knowledge of Grid applications. In this scheme, neither the resource requirements nor provide application characteristics are specified by the application. Basically, by use of scheduling scheme a scavenging Grid aims at leveraging the idle computing power. Example is Condor [12], which uses resource level scheduling.

### *B. Inter-Job Dependency:*

Given an application, the constituent jobs may either be dependent or independent. The mapping of algorithms for a set of dependent jobs differs significantly from those for a set of independent jobs. DAG is usually used to represent an application with a set of jobs. There is much more

complication in mapping algorithms for a set of dependent jobs.

### *C. Information Service:*

The state information of all the resources in a Grid system is determined by the scheduler through the information service before making a scheduling decision. Different scheduling systems have been constructed having quite different structures to provide information services.

- ***a. Centralized:*** Under a centralized scheme, the state information is maintained by a single centralized entity. The centralized entity traverses periodically through every resource to get the most of date of state information and thus the information is stored in its storage, where it will be waiting for queries issued by the schedulers. A centralized scheme introduces the risk of single point of failure thus it is not scalable. Furthermore, sometimes a centralized entity may become the performance bottleneck when the entity cannot afford a large number of resources and possible queries.
- ***b. Decentralized:*** Around this method, every resource has its duty for maintaining its current local information status and providing solution of problems that are submitted by different clients. A decentralized scheme may not be reliable due to the large amount of queries. However the decentralized scheme is more efficient and reliable because it removed the risk of single point of failure by scattering the responsibility equally to each resource. The problem of large scale should be carefully considered. Example of decentralized scheme is Network Weather Service.
- ***c. Hybrid:*** By the use of hybrid method, resources are divided into several groups based on different categories. A centralized scheme is applied within each group and respective category. Therefore each group of resources has its representative entity; which has the control over the information of all resources in its group. But over the groups, the decentralized scheme is the most practical method for large scale Grids in real.

### *D. Scheduler Organization:*

The Grid Scheduler organization can be maintained into three categories: centralized, decentralized & hierarchical.

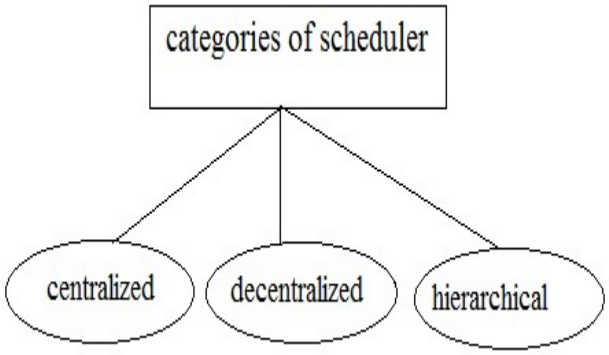


Figure 2.2 Scheduler Organization Types

- ***a. Centralized:*** In this method, the centralized scheduler received all users applications, which require various types of resources. The centralized scheduler has applied a queue system for holding all the pending applications. When the applications are submitted to the scheduler by the user, it may not be scheduled in first time attempt. Instead it will be put in the queuing system and applied a scheduling algorithm and then wait for scheduling of user's applications and resource allocation. Hardily, each site is not able to maintain its queue and even do not able to perform any scheduling decisions. A site only gets jobs from the scheduler and executes them. The centralized scheme is not very efficient and scalable with increasing large amount of required resources. The central scheduler may prove to be a chokepoint in some environment and situation.
- ***b. Decentralized:*** Decentralized scheme distributes the responsibility of scheduling to every. Each site in the Grid acts as both a scheduler and a computational resource. User applications submitted to the local Grid scheduler where the applications originate. The local scheduler is responsible for scheduling its local applications.
- ***c. Hierarchical:*** In the hierarchical method, various levels of schedulers co-operate with each other and share the scheduling process to achieve better results. The higher-level schedulers organize the resources of heavy sets and lower level schedulers manage smaller sets of resources. A higher-level scheduler has no direct control of a resource if there is one lower-level scheduler between the higher–level scheduler and the given resource. A higher-level scheduler just able to take into account the capability of the set of resources managed by a lower-level scheduler as a whole entity, and utilizes the capability through invoking the lower-level scheduler. Compared with the centralized scheduling, hierarchical scheduling addresses the scalability and the problem of single point failure and more over it also retains some of the good features of the centralized scheme.

### *E. Rescheduling:*

When the scheduling of an application was completed, after that the resulted performance of the application may not provide satisfaction regarding the desired performance due to the dynamic nature of the resources. It may be profitable to reschedule the application during execution to maintain good performance. At the minimum, an adequate Grid scheduler should acknowledge the resource failure and resend lost work to a live computational resource. In summary, rescheduling is done to guarantee the job's completion and performance goal's achievement.

## V. GRID SCHEDULING APPROACHES

The scheduling policy determines how the scheduling should be performed. The performance goal defined in the scheduling policy plays a particularly important role in a Grid scheduling system. According to the various performance objectives, the scheduling systems can be classified into three categories [13]:

### *A. Application Centric:*

Scheduling systems sometimes try to favor the performance of individual applications when it falls in the application centric category. Application centric scheduling systems sought to have performance goals including minimizing execution time; maximizing the speed etc. Example an application-centric scheduling system will feat a greedy mapping algorithm, which produces the best

performance by allocates the application to the resources without considering the rest of pending applications.

### *B. System-Centric:*

A system centric scheduling system is concerned with the overall performance of the set of applications and the whole Grid system. The system centric policy typically desired performance goals which include resource utilization, system throughput and average application response time.

### *C. Economy Based:*

An idea of market economy was introduced by the economy based scheduling system. Under this scheme, scheduling decisions are made based on the economy model. The economy model defines that each application is having the desired QoS, such as execution time and deadline and the cost that the application will pay for the desired QoS; each resource is specified by its cost and the capacity. For each application, economy based scheduling system wants to get as higher QoS as possible within the budget constraint. For each resource, it wants to obtain profit as much as possible by keeping itself busy. Nimrod-G [14] is a scheduling system which exploits idea of economy mechanism.

## VI. SUMMARY OF CURRENT APPROACHES TO GRID SCHEDULING

The given table briefs the properties of widely used the Grid Resource Management Systems with emphasis on their scheduling attributes [15]:

Table2.1 Summary of Recent Approaches in Grid Scheduling [15]

| System | Grid Type | Resources | Scheduling Approach |
|---|---|---|---|
| Condor | Computational Flat | Extensible schema model, hybrid namespace, no QoS, network directory store, centralized queries discovery. | Centralized Scheduler |
| Globus | Multiple Hierarchical | Extensible schema model, hierarchical namespace, soft QoS, network directory store, distributed queries discovery. | Decentralized Scheduler infrastructure, scheduling provided by external schedulers like Nimrod/G |
| NetSolve | Computational Hierarchical | Extensible schema model, hierarchical namespace, soft QoS, distributed queries discovery | Decentralized Scheduler, fixed application oriented policy |
| Nimrod/G | High-Throughput Hierarchical | Extensible schema model, hierarchical namespace, relational network directory data store, soft QoS, distributed queries discovery | Hierarchical decentralized Scheduler, predictive pricing models, fixed application oriented policy |

## VII. CHALLENGES OF SCHEDULING IN GRID COMPUTING SYSTEM

Besides the characteristics of Grid Environment the evaluation of scheduling algorithms is tedious with the factor of performance.

The difficulty in the task due to factors is as follow:

a. Presumptions of performance are difficult because it is extremely hard to analyze and predict end to end internet performance.
b. There is a great diversity exhibited by internet when end to end performance observed and thus using different algorithms work more effectively for different topologies and also for different time periods on same topology.
c. There are schedulers that must provide capabilities for areas such as (but not limited to):
    a) Advanced resource reservation.
    b) Service-level agreement validation and enforcement.
    c) Job and resource policy management and enforcement for best turnaround times within the allowable budget constraints.
    d) Monitoring job executions and status.
    e) Rescheduling and corrective actions of partial failover situations.

## VIII. CONCLUSION

Grid computing is a most efficient computing epitome where applications, records and IT services are provided over the internet and the fastest delivering on demand solution of submitted problems. Grid computing has come out to be an interesting and beneficial way of changing the whole computing Schedulers for grid computing determine on which processing resource jobs of a workflow should be allocated. Comparison between parallel, distributed and grid scheduling approaches explore more deeply about scheduling process under specified condition. Also here, basic foundation of grid scheduling has been explained that will benefit the researchers to carry out their further work in the area of grid scheduling.

In future, researcher can work on above mentioned challenges to provide reliable and dynamic scheduling approach.

## IX. REFERENCES

[1] Bart Jacob, Michael Brown, Kentaro Fukui, Nihar Trivedi, "Introduction to Grid Computing," December 2005

[2] Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing applications, Vol. 15, No. 3, pp. 200-222, 2001.

[3] Archit Kulshrestha and Gabrielle Allen, "Service Oriented Architecture for Job Submission and Management on Grid Computing Resources", 2009 IEEE.

[4] Yanmin ZHU, A Survey of Grid Scheduling, Department of Computer Science Hong Kong University of Science and Technology.

[5] GARUDA Inroduction and its objective" http://www.garudaindia.in/html/about_garuda.aspx".

[6] Healthgrid at VECC "http://www.isgtw.org/event/healthgrid-and-iwsg-life-2012".

[7] Yan Liu, "Grid Scheduling", Department of Computer Science, University of Iowa.

[8] Jarek Nabrzyski, Jennifer M. Schopf & Jan Weglarz, Grid Resource Management– State of the art and Future trends, Kluwer Academic Publisher.

[9] T. Casavant, and J. Kuhl, A Taxonomy of Scheduling in General-purpose Distributed Computing Systems, in IEEE Trans. on Software Engineering Vol. 14, No.2, pp. 141--154, February 1988.

[10] Fangpeng Dong and Selim G. Akl, "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems", Technical Report No. 2006-504, School of Computing, Queen's University, Kingston, Ontario, January 2006.

[11] H. El-Rewini, T. Lewis, and H..Ali, Task Scheduling in Parallel and Distributed Systems, ISBN: 0130992356, PTR Prentice Hall, 1994.

[12] http://www.cs.wisc.edu/condor/

[13] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Principle Seventh Edition".

[14] www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm

[15] Klaus Krauter, Rajkumar Buyya and Muthucumaru Maheswaran, Taxonomy and survey of Grid Resource Management Systems for Distributed Computing. Software Practice and Experience Softw. Pract. Exper. 2002.