CHAPTER 1

Introduction
CHAPTER 1

**Chapter 1: Introduction**

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Name of the Sub-Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>General</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>State of art</td>
<td>9</td>
</tr>
<tr>
<td>1.3</td>
<td>Thesis Statement</td>
<td>13</td>
</tr>
<tr>
<td>1.4</td>
<td>Objectives</td>
<td>13</td>
</tr>
<tr>
<td>1.5</td>
<td>Thesis Organization</td>
<td>13</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1 General

The distributed computing can be defined as, the use of multiple computers networked through a wide geographical areas. The Internet can be used for connecting the computers. In order to solve a single problem multiple computers are involved in solving the problems or processing the information.

Grid computing improved from Ian Foster [1-3], is concerned with "facilitated resource distribution and critical thinking in dynamic, multi-institutional virtual organizations". A Grid is a decentralized heterogeneous framework in which resources are associated with numerous organizations. Outright control is not implemented by the grid on these resources and resource administration is liable to various and different hierarchical managerial approaches. From client’s viewpoint, a Grid is a collective critical thinking environment in which one or more client tasks might be issued without the knowledge about the location and the owner of the resources.

A Grid must ensure the nature of the administration of the execution of tasks, i.e. the quality of service. With a specific goal to develop a Grid computing environment, it is important to have a typical software system with a Grid framework (i.e., a Grid middleware). Globus [4] has been effective in creating open and typical conventions and interfaces, and is the true standard in Grid world. Its real commitment to the Grid security issues is a PKI-based Grid license key, which empowers the control to access the resources.
across various organizations. Likewise, it gives set of tools and instruments to VO-scope task proposal (GRAM) and resource identification (GRIS).

Cloud computing is a group of virtualized computer resources that has a collection of different workloads and permits them to be organized and scaled-out through the quick provisioning of physical or virtual machines; provisions highly scalable, redundant, resource usage monitoring, self recovering programming models in real time to empower rebalancing of allocations when required.

Most of the enterprises are looking forward to use virtual data centers to maintain infrastructure and to decrease the cost of the hardware. This type of IT infrastructure shrinks the cost of hardware and brings the infrastructure as a service. For instance, there are many clients like to organize applications in a distributed environment, Nevertheless the cost of this organization and losing flexibility. The reason for the designing of cloud services for the users is to to share the resources and capabilities. Cloud computing will act as grid computing, regardless still there are some important considerable differences.

Cloud computing is supported by a number of large organizations like, Amazon, Yahoo, sun and google. Normal users also adapt this through the organization, Amazon is one of the key developers in maintaining the cloud. Amazon modernizes their own particular data centers, which brought about noteworthy increments in inner productivity. In 2005 Amazon's cloud computing framework
called Amazon web services was conceived. Amazon was the one of the first associations to give cloud computing facility.

When compared to cloud, grid computing is a framework that can oversee and sort out services and resources that are distributed among different domains, by utilizing protocols and interfaces and provide high quality of services to the users.

**Grid Layered Architecture**

The structural design of the grid is in the forms of layers. The grid structure consists of four layers which are listed below:

- Fabrication layer
- Grid core middleware
- Development layer
- Applications
As shown in Figure 1.1, the grid contains

- Various kinds of components.
- The resource managers present in the local machines are in the base layer. Furthermore, network resources like information sources, auxiliary stockpiles and are part of the base layer. Network protocols are managed by this layer.
- Distributed services are virtuously associated with the next layer. Security systems similar to GSI are used in this case. At the same time, QoS is also taken into consideration.
Resources impart the data productively in a distributed environment.

- The devices and languages are contributed by the developing the environment in order to create grid applications.
- The top most layer of the grid structure is the application layer. The interaction among clients and critical thinking abilities are implemented in this layer.

The solution to the below query is one of the exploration challenges in grid scheduling: In what way the scheduling of set of grid applications is to be carried out in multiple decentralized resources?

There will be numerous jobs under every application. There are many queries to be answered when the mapping between the jobs and the resources is to be made. They are as given below:

- How the plans related to scheduling are affected by the associations among tasks?
- How much impact the scheduling performance will have due to the various kinds of resources?
- How the scheduler can estimate the quality of a schedule and which models of performance need to be utilized?

Scheduling becomes more troublesome when the jobs are reliant on one another. Actually, the efforts and work done in this area till now is mostly concentrated on the scheduling of the jobs which are not dependent on one another, detachable or not tightly coupled tasks.
These methodologies regularly utilize either critical systems to make deterministic agendas (typically an ideal schedule outcome) or experimental information investigation and heuristic quest strategies to search for a worthy result.

Scheduling in grid systems is characteristically more convoluted than confined resource scheduling in light of the fact that it must control substantial scale of resources through administration limits. In such a vibrant distributed computing environment, resource accessibility changes drastically. Hence scheduling becomes difficult. There has been a broad study of the scheduling issues in distributed frameworks.

For many years, there are many progressions that have been attained towards minimizing the power consumption of the computers. Dynamic power management (DPM) algorithm [6] specifically puts diverse parts of the workstations to rest, in view of the heap accomplished by the segments. However, DPM is not extremely successful regarding the matter of the processor. The power consumption of the processor decreases with the varying frequency using DVS algorithm. The expense of cooling the hotness dispersed in the huge server farms and all in all for all the machines can be reduced by decreasing the power consumption. Hence, the minimization of power consumption has become a significant research area. The criticalness and need to preserve or save battery life in the perspective of compact gadgets and omnipresent gadgets is one more point for the concentration on power minimization.
Power lessening strategies have an immediate effect on the performance of the devices and on the budgetary part of business which utilizes substantial power for computing. Power consumption of a processor can be decreased in two ways: utilizing the data from the compiler and modification of assembly code in the non-compiler approach; wherein the number of processes is altered frequently with the verification of the heap on the processor. Example for the compiler-directed algorithm is the DVS algorithm. It investigates set of program sections and chooses a local core to run at a less frequency to reduce the anticipated power depletion while fulfilling a client stated comparative performance consequence.

The applications which need to be executed on the portable and handheld devices need to consume less power. Hence, power minimization plays a major in such applications. Some of the examples of such applications which are based on the natural language processing are data mining, data compression and data recovery.

1.2 State of the art

Power utilization is one of the significant discriminating issues when taking care of the requests of different applications and resource scheduling of the grid. Decreasing the execution time is one of the significant performance parameters during the process of scheduling the tasks. Execution time might be lessened by expanding clock recurrence yet that prompts more power utilization and high temperature dispersal. Performance and the power utilization have a
trade-off between them. The outline which is having knowledge about energy does not essentially reduce the power yet it is conceivable to lessening ultimate power utilization in a processor because of the delay.

DVS algorithm is a compelling strategy that lessens power utilization by bringing down CPU voltage where the CPU can back off with minor damage to performance. The high level language which is also a language of the source and is divergent from one language to the other is for which the DVS algorithm is proposed. In PERMA schema [7], DVS is functional for sections that have only one-in and one-out section for control. The paramount QoS components of the grid computing systems are the time for computation and expense. There is a trade-off between the cost and time as the time to prepare the tasks can be reduced by using additional resources which increases the cost [8]. Technology of workflow is used to represent the various tasks of the intricate applications in different views of numerous disciplines [9] [10]. The dependency of the various tasks on one another of an application is shown in the workflow. The methodology of relating and dealing the tasks which are dependent on each other with the distributed resources is referred as workflow scheduling. This process is considered to be as NP-hard Problem [11] [12].

The jobs which can be considered in batches and in parallel that have requests for various resources is taken into viewpoint for
discussing the scheduling algorithm in [13] by M. Silberstein et al. The performance is enhanced using meta-scheduling and the capacity of co-scheduling is done for applications in grid [14] by M.W. Margo et al. Meta-scheduling has the ability of making clients to request for reservations among cluster sites that are distributed.

Virtual Machines (VMs) upholds the rate of computation for the configuration of contract administration structure for a workload of numerous varieties and intuitiveness objectives for assignments to give constant assurances for VMs facilitated on a solitary server machine [15]. R. S. Montero et al [16] characterize the initialization of a VM with $k$ jobs to be exchanged from the store to the network by making use of DAG that diminishes the amount of initializations and lessen the possibilities of congestion in the network.

Implementing this system will be all the more expensive with respective to workflow, on the grounds that for each one undertaking in the work process require one VM instantiation. Batista et al [17] changes the DAG of workflows and incorporates the similar implementation settings instantiated for all assignments which makes the complete concluding time to be more extraordinary. The complete concluding time is decreased with more concentration on Workflow scheduling.

The previous information related to minimum concluding time is considered to select the resources in Optimizing Probabilistic Load Balancing Algorithm in grid computing. The response time is decreased with the help of load balancing. Neighborhood property is
associated with the hierarchical load balancing method for task load balancing for grid computing in [9]. First, the load balancing is done in this method and later higher level hierarchical balancing is performed. The model for performing the task load balancing uses the past reference about the particulars of the system as presented in [10] for dynamic load balancing in grid computing.

The primary attributes of this methodology are:

- It uses load balancing according to the level of jobs.
- It benefits the exchange of jobs locally to minimize the communication expenses.
- Distributed method with neighborhood choice making. Irrespective of the topological structure of the grid, this framework changes the structure of the grid in the form of a tree.

The distinct nodes are loaded based on the overall average value using the distributed partial random sampling technique which is presented in honey-bee foraging. Enhanced load balancing can be achieved by limited renovating to associate related services and dynamic system restructuring. The task of load balancing is to allocate services vigorously to normalize the fluctuating requests of the client irrespective of the rise or fall in web server’s requests. The separate queues are maintained by each VM and servers are formed into groups with respect to virtual machines. The profit or reward is
computed based on the excellence that the bees display in their waggle dance when a request from the queue is processed by each server.

1.3 Thesis Statement

To develop power aware architecture for scheduling the tasks and various optimization strategies to minimize the power utilization of resources, execution time of tasks and enhance Quality of Service in grid environment.

1.4 Objectives

- To propose an architecture that includes distribution, optimization, estimation of power consumption, and scheduling for the grid environment.
- To propose an optimal strategy to minimize power utilization of resources without affecting the execution time.
- To propose an optimized technique to perform effective load balancing to the virtual machines based on fuzzy bee

1.5 Thesis Organization

The thesis is organized as shown in figure 1.2 and described as follows:
Chapter 2: Literature Survey

The study in the area of grid computing is presented. The study is concentrated to analyze the frameworks which are presented in order to reduce the power utilization without affecting the execution time. The optimization techniques based on the honey bee colony to be applied for task scheduling and resource allocation strategies are also reviewed in this chapter.

Chapter 3: Power Estimator and Reducer for Multi-Core Architectures (PERMA) Frame Work

The PERMA framework is proposed to reduce the power utilization and execution time in case of both pre-emptive and non
pre-emptive jobs in the grid environment. The power estimator and reducer are used to deliver the information related to the power consumed by the program and recommend the appropriate frequency levels in order to decrease the power utilization for the different basic blocks. The analytic approach is also presented in this chapter. The analysis of the execution time is carried out at the software abstraction layer in sequential and parallel implementation. The execution time is evaluated by deploying the PERMA-G framework for an application in the Grid.

Chapter 4: Optimized PERMA-G based Workflow Scheduling

The power utilization is reduced by considering the execution time in the workflow scheduling using the PERMA framework is proposed as OPSA-G algorithm. The mapping of the available task T to the available resource R is made by using the OPSA-G algorithm. The task T is divided into subtasks like t and resource R like r. Now the subtask t is allocated to the part of resource r in R and this assignment is done based on the time of completion of the original parent task. The scheduling of the total available tasks to the available resources is based on the workflow of the tasks. The threshold is evaluated based on the difference between the cost and the time which is used to optimize the process of scheduling. The completion time is noticeably reduced with the minimum utilization of VMs which reduces the power utilization using this approach. The scheduling heuristics for the optimizations in multi-core based grid
systems are developed based on execution time and power utilization using PERMA-G. The three different tasks and resource scheduling algorithms with and without honey Bee optimization technique based on the PERMA framework is proposed in this chapter.

Chapter 5: Power Aware Based Workflow Model Optimization Using Fuzzy Bee Colony Method

The searching of honey-bee, a natural phenomenon is applied to the method of distributed biased random sampling in order to maintain the load of a distinct node through an evaluation of the closest overall average quantity. Finally, related services associated through limited rewiring are evaluated as a process of improving the load balancing through dynamic system reorganization. The active resources are allocated to regulate the fluctuating requirements of the customer with the increase or decrease in the demands of the web servers with respect to load balancing. The virtual machines maintain separate queues for resources and the servers are formed into groups. The profit or reward is computed by the server when a request is processed and the evaluation is based on the quality of the waggle dance of the bees. The issues related to the power utilization and the execution time of the task in virtual machines is solved by employing the fuzzy bee colony optimization technique in the PERMA-G framework. The effective load balancing is done in VMs using PERMA-G which is used to reduce the power utilization and the execution time in a VM. A model depicted in this chapter is used to determine
the way to minimize the power utilization without affecting the execution time and is tested in the grid environment. Load balancing can be done effectively based on the distance from the resource.

Chapter 6: Fuzzy Bee Colony Optimization for Resource Allocation in Grid Computing

The resource scheduling proposed in this chapter minimizes the number of preemptions of tasks. The unavailable periods in fine grained cycle sharing systems in the combination of local and global tasks are analyzed in the prediction method. The available resource is selected vigorously by bearing in mind. The suitable number of VMs according to the proposed algorithm. The grid system with dynamic resource allocation for the jobs and the estimation of the number of VMs available is proposed, developed and tested in this chapter. The available resources are ranked depending on the performance that can be attained by approximating the pre-eminent probable set of properties about each resource. Later, the resource allocation engine is built to select and associate the best suitable resource for the available tasks in grid dynamically. The resource is selected based on the power utilization factor.