The Research Monograph Series in Computing, Electrical & Communication Networks
Vol. 1 (01), January 2023, pp. 3–7

INTRODUCTION TO MOBILE GRID COMPUTING

It highlights the importance and definition of mobile grid computing. It also discusses the challenges that arise in mobile grid computing, as well as the main cause that motivates the design of the proposed work.

Grid computing originated as the next-generation parallel and distributed computing, aggregating multiple geographically dispersed heterogeneous resources to solve compute-intensive and data-intensive issues that would be impractical to run in traditional distributed computing environments. Grid is a large-scale virtual computing environment in which users can share computer resources such as files, software, computing power, and data storage among multiple businesses or users. A Grid enables constant, trustworthy, widespread, and low-cost access to computing devices, resulting in resource capacity comparable to high-end supercomputers. It uses remote collaboration, storage, and the potentially infinite power of ubiquitous computing devices to solve problems in a variety of fields, including high energy physics, earth system sciences, astronomy, Geosciences, Archaeology, Bioinformatics, biomedical science, and financial modeling.

Grid environments are made up of dedicated grids and volunteer grids. The dedicated grid is federated by computers such as database servers, application servers, and blade clusters that commit to completing a specified task. The Volunteer grid, on the other hand, is a collection of federated computers that are not entirely dedicated to a single task and are only used when they are in peak condition. These resources are used at random and infrequently. Their proprietors can connect or detach them at any time without previous notice. The desktop PCs of navy users collaborate to construct the volunteer Grid.

Aside from these, the most notable devices that arise to engage in Grid are technologically adept Mobile devices such as laptops, PDAs, sensor devices, and so on. These new gadgets with increased processing capability collaborate to build the mobile Grid, an efficient collaborative network. Figure 1.1 depicts a mobile network made up of several mobile devices.

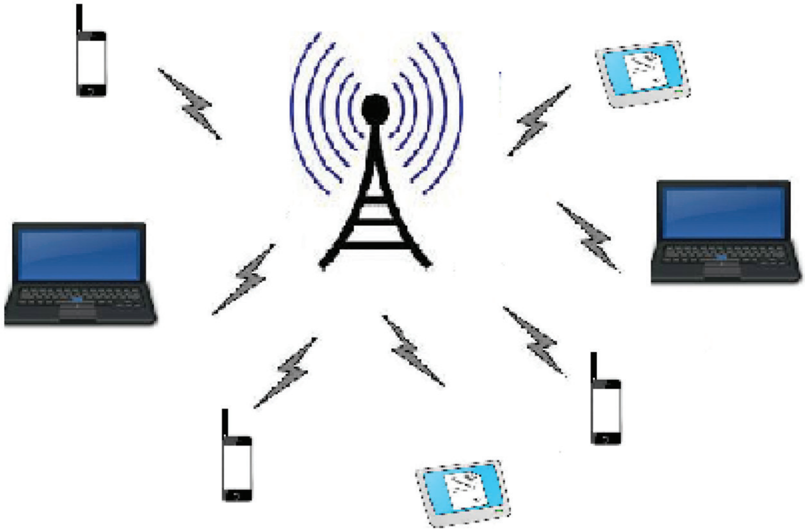

Figure 1.1. Mobile Network.

The reason behind developing the mobile grid is that as the number of mobile device users grows, devices with processors and huge memory and advanced technologies are being released at an increasing rate. Mobility, portability, and pervasiveness are also advantages of mobile devices over fixed computing resources. Furthermore, there are a large number of highly capable mobile devices that are mostly inactive in the same company network. This mobile grid's strength can be well-applied to location-restricted industries that require supportive infrastructure, such as wildfire prevention, disaster management, and e-health systems, among others.

In comparison to wired Grid, the Mobile Grid inherited from Grid has the added characteristic of delivering seamless, transparent, and efficient assistance to users and resource suppliers. The mobile grid is a self-configuring Grid system that is linked by wireless links that construct arbitrary and unpredictable topologies with underlying adhoc networks. Because of its openness, heterogeneity, and scalability, the mobile grid serves as the foundation and enabling technology for pervasive and utility computing.

The mobile grid system is based on wireless communication, with cells forming the network. Each cell is made up of several mobile devices. These mobile devices in a wireless network cell are coordinated by a central entity called the Wireless Access Point Server (WAP), which is located at the Access Point/Base Station. These WAP servers exist networked via cables or wirelessly with an edge router (proxy server) that receives jobs from a Grid Controller (GC) and returns the computing results to the GC [4]. The work scheduler is handled by the Proxy server. In order to do his work, the client submits a resource request to the Grid Controller. The Grid Controller sends the message to the proxy server. The Proxy server chooses an appropriate mobile resource and assigns the work to it. After completing the work, the resource returns the results to the user via the Grid Controller.

Though mobile devices have been used successfully for job execution, the addition of these diverse portable computing and communication resources increases the scale and complexity of grid computing systems. There are other hurdles ahead, including large-scale deployment and appropriate use of parallel scientific applications. Device mobility, frequent disconnections that affect resource availability, security to data communication and mobile resources, controlling access to shared resources, the type of environment, the type of grid architecture, task processing policy, task interrelationships, and battery lifetime are some of the challenges encountered during the job execution process [3, 6]. Furthermore, mobile devices provide less dependable platforms for long-running code, resulting in application execution times that surpass the machines' mean time to failure.

We provided an exceptional composition that employs appropriate terminology and strategies to address the aforementioned mobile grid challenges and give a superior mobile grid environment.

MOTIVATION

As mobile devices proliferate, accessing and participating in the grid via mobile devices becomes more popular. Because of its open, extremely heterogeneous, and scalable character, mobile grid has emerged as the foundation and enabling technology for pervasive and utility computing.

A user requests a resource for job execution to the Grid Controller in a mobile grid computing system. The Grid controller chooses an appropriate mobile resource to execute the job and returns the outcome to the user. The Grid controller delivers the request to the proxy server in this scenario. The proxy server performs the function of a job scheduler. The scheduler is critical for job allocation and overall network performance.

An intelligent grid scheduler should optimize standard scheduling objectives such as allocating jobs to appropriate mobile devices, minimizing uncertainty in job execution, and striving to optimize scheduling objectives such as maximum throughput, response time, and balancing available resources with security requirements, as well as minimizing energy consumption. As a result, an effective task scheduler that selects the best devices within the cell for job execution is more important. A resource that generates quick results should be used for efficient grid task execution and reduced execution time. Resources having the shortest reaction time should be chosen to maximize throughput. A proper load balanced strategy should be used to avoid job execution overload at the resources.

Because the mobile Grid is becoming a less dependable platform because to its mobility, frequent disconnection, and mistakes induced by battery power, an effective fault detection and recovery technique is becoming increasingly important. Thus, for large-scale deployment and high performance of scientific applications on mobile grid systems for fostering high performance computing, a practical and robust mobile grid infrastructure that integrates components related to application monitoring, resource monitoring, performance modeling, scheduling, rescheduling techniques, and fault detection technique is extremely important [65].

As the grid becomes a more commercial resource, customers begin to demand high levels of security, as they would from any computer-based commercial instrument, in order for it to consistently give appropriate results. In the lack of security, the grid arrangement becomes insecure, allowing unauthorized users, malicious programs, and data tampering to render it useless. Mechanisms for secure authentication, authorization, data encryption, resource protection, and secure communication must be used in a secure grid environment. The Grid security system must also provide unique user identities across local and global networks, as well as manage the diversity of local resource/user security systems, trust relationships between entities, end-user key and credential management, and resource security against malicious grid user actions [57].

PROBLEM STATEMENT

The existing work scheduling method in the mobile grid context fails to distribute the job equitably throughout the client set. It takes time to validate the node's capability and optimize the job execution time. Furthermore, the current fault tolerance and security systems are not practical or efficient.

Its primary goal is to create Mobility Aware Load Balanced Scheduling. The job is divided into two categories: computing-focused jobs and communication-focused employment. It assigns computationally intensive jobs to resources with shorter round-trip time (RTT), high CPU speed, and capacity. Communication-intensive jobs are assigned to resources who are low on mobility and strong on dependability. The key benefit of this technique is that job scheduling is based on node execution capacity. The job is allocated in the cell based on the node capability, resulting in a balanced scheduling strategy.

It also creates a technique for detecting and recovering from node defects or failures. A simple pull mechanism is used for failure detection. When unrecoverable faults are identified, the job is stopped and the faulty node is proposed to be relieved of task execution. An extraordinary local node fault recovery technique is used for recoverable faults.

Existing Grid Computing security work mostly focuses on Grid service authentication and authorization. It fails to focus on harmful assaults that occur during resource communication. In this section, we provide security in the mobile grid environment, taking into account wormhole attacks and scam users. This technique employs a group key management system to provide safe data flow across grid users, as well as a Wormhole Resistance System to identify and prevent malicious Wormhole activity.