An Object Based Paradigm for Integration of Mobile Hosts into Grid

M. A. Maluk Mohamed
M.A.M. College of Engineering - Anna University of Technology Tiruchirappalli

This paper proposes a perspective of a novel object mobile grid as an integration of mobile computing paradigm into the grid computing paradigm along with service composition technology. It shares both advantages of powerful computation capability of grid and ubiquitous accessibility of distributed mobile system. It combines the middleware solution from services composition, resource-sharing solutions of grid computing and the anywhere, anytime resource access of mobile computing. It is a distributed system model where computational, data and other resources including the experimental devices throughout the wired and wireless networks are organized into federation, as peer to peer model. All the resources, data and services offered by the participating static and mobile hosts are virtualized as services, to enable service composition which may require both resources and services for composition of a complex service. The proposed system is realized as a shared distributed object space enabling the scalability, and helps in handling the constraints of the distributed mobile system. The benefits of this model are information processing capacity increase and service sharing, which may include services that may provide location and context sensitive information. The paper discusses number of technical considerations using the proposed framework. These include, the visualization of resources as objects, the novel object oriented model of the mobile grid and the composition of services. The performance of the proposed model is studied using simulation and emulation study.

Keywords: Mobile Grid, Surrogate Object, Service Composition, Distributed Shared Object, Naming Services, Trading Services, Virat

1. INTRODUCTION

The power of networks and user applications have evolved from centralized computing platforms to distributed collaborative computing services. Collection of network nodes can jointly achieve advanced computing goals beyond relaying packets. Collaborative user applications enable people and computers to work together more productively. Collaboration as a fundamental capability of networks represents a significant step forward, leading to grid computing. This has triggered many research work in the field of grid computing. Grid Computing is viewed as a distributed, resource sharing computing and data handling infrastructure. It pools geographically and organizationally dispersed, heterogeneous resources and provides common interface for all the resources, using standard, open, general purpose protocols and interfaces [Foster et al. 2001].

Mobile computing system brings about a new paradigm of distributed system in which communication may be achieved through wireless networks and users can compute seamlessly even as they move from one environment to another. It is apparent that mobility, affects the computational, data and transactional model, and the communication paradigm of the distributed systems. The impact of the huge growth of resource constrained devices goes beyond the networking issues such as bandwidth and connectivity, and directly affects computing, data, and service management. The recent technological advancements in computing, wireless communications, networking and electronics has embedded processing power, storage space and communication capabilities in electronic devices of day-to-day use, leading to the era of ubiquitous computing [Weiser 1993]. This trend has led to the pervasiveness [Satyanarayanan 2001], invisibility [Norman 1998] and mobility of computing hosts. These factors have made us look at mobile hosts (MHs) as both provider and consumer of services.

The advancement in technology has enabled MHs to become information and service providers by complementing or replacing static hosts (SHs). They are no longer used just in the domain of sales force and field service workers. Such mobile resources are highly essential for on-field applications that require advanced collaboration and computing. This motivates the need for merging of mobile and grid technologies, leading to mobile grid paradigm. The key idea is to integrate the capability of computational, data and service grids to form a novel mobile grid. The proposed mobile grid is built to, seamlessly and transparently, manage and bridge the requirement of the mobile and static users, and the actual service providers. Thus a MH from anywhere and at anytime, can utilize large computing power, required resources and services seamlessly. Simultaneously, the MH could also be providing location sensitive data to the grid. The mobility of the MHs also makes the hosts to interact with a changing landscape of information resources and services. These resources are not only heterogeneous but also carry context-dependent information, depending on user, location, community and other parameters. Such services may be value added, and include content provisioning.

Mobility of the participating MHs lead to intermittent connectivity, thus making the proposed mobile grid more dynamic in nature in contrast to the traditional grids. The dynamic nature in addition to other constraints of the MHs [Forman and Zahorjan 1994; Imielinski and Badrinath 1994] becomes a real challenge when integration of mobile and grid computing is done. Further, based on the key design issue of the distributed systems, all aspects related to distribution and mobility must be transparent to users. Thus, aspects regarding the current location of the MHs, accessing the service from where it is available, and transferring it to the user must be hidden by the system.

To the best of our knowledge such systems that allow users to share and manage arbitrary services, offered by both static and mobile hosts, in a transparent way does not exist. Thus the goal of the research is to design a mobile grid middleware model that provides transparency to the users, in addition to handling the key constraints of the participating MHs. Distributed object middleware provides a solution by hiding lower level network details and provides transparency to the user. However, replication and replica consistency become important issues in any internet scale distributed system, as they are essential to achieve scalability. Distributed object middlewares do not address replication issues naturally. For instance, the fault tolerance specification of CORBA [Group 2001] which addresses replication, still has serious limitations with respect to transparency and other issues [Felber and Narasimhan 2004].

In distributed shared object (DSO) systems, replication and consistency are key issues that have been addressed extensively. This has motivated us to make distributed shared object space as the basis for our system, in addition to the fact that DSO can potentially provide a higher level of abstraction compared to middleware. Thus a new object oriented mobile grid, based on virat [Srinivas and Janakiram 2006], a wide area shared object space model is proposed.

In particular the proposed model takes into consideration, issues related to mobility of the participating MHs and heterogeneity of the connecting networks. In addition, critical challenges also need to be addressed in the context of composing independent components across multiple service providers. When providers deploy service instances independently, the composed service-level path could span multiple cluster domains, affecting the availability of the composed service. This is because, inter-domain cluster path availability is very poor [Labovitz et al. 1999], and that inter-cluster route recovery can take of the order of a few minutes [Labovitz et al. 2000].

The rest of the paper is organized as follows: Section 2 gives the motivation for mobile grid along with an application. Section 3 briefs the background that is needed for the proposed model. Section 4 gives an overview of the proposed approach, that includes the architecture of the proposed object based mobile grid. Section 5 gives the mobile grid computational model. Section 6 describes the services management. Section 7 gives the two different sets of performance analysis. Section 8 discusses the related work and section 9 concludes the paper with summary and future directions.

2. MOBILE HOSTS AND THE GRID

Mobile hosts characterized by limited resources and by their high mobility are gaining popularity and their numbers are growing rapidly. Thus traditional grid applications need to expand their scope by extending the interactions to mobile hosts. However the issues related to the constrained characteristics [Forman and Zahorjan 1994; Imielinski and Badrinath 1994] of these hosts must be addressed to shield the user from possible instability, by making them transparent to the end user.

2.1 Mobile Hosts as Providers and Consumers of Services

When MHs are part of grid, in addition to the capability of the SHs, we view that the MHs participating in the mobile grid can be both 'provider' and 'consumer' of services and other resources. When MHs act as consumer of services, they can use the services and resources, that are distributed throughout the wired and wireless networks, and monitor the jobs being executed remotely.

A MH may be capable of solving a task on its own if it has enough processing power, available memory and battery charge. However in some mobile applications where the task is too big and/or resource demanding, they need to move the computational load elsewhere [Z.Li et al. 2001] or distribute across multiple hosts which may be both MHs and SHs. This also enables the MHs to solve far more complex and resource-demanding problems. But this raises the need for establishing appropriate collaborative processes between these geographically distributed tasks.

With the advancement in technology of the MHs related to computing and memory capability, we believe that MHs can act as provider of services and resources, such as the data source, result destination or computational resources. The user need not be aware of how a mobile service moves but be guaranteed that the service can accomplish the assigned task.

Mostly the data offered by the MHs will involve context and location data. This data is constantly updated and queried in real time by the network. This is mostly a network level data management problem but still these involves storage and access of data as well as dynamic replication of resources in response to the changing context and location of the host. On the other end, the general public information such as traffic, weather, stock quotes, etc. are more of read only and will involve large number of users. These information mostly may not be dependent on the location and context.

2.2 Motivating Application

With the emergence of grids as a suitable hosting technology for applications in various areas including data- and compute-intensive computing, such as online gaming, data storage and archiving, it is becoming important for MHs to be a part of grid. For example, in a bridge infrastructure maintenance system [Garrett et al. 2002]which will be highly benefited using the proposed mobile grid paradigm. The transportation systems like roads and bridges are improving in most of the developed and developing countries. It has thus become essential to improve the current mode of monitoring and maintaining these infrastructures. These facilities deteriorate with use and exposure, interact with other systems, and when disrupted cause major delays, loss of economic activity, and most importantly, loss of necessary support functions.

During construction, inspection, maintenance, and repair of such infrastructures, field-based engineers and technicians need to refer to maps, engineering drawings, databases, and other technical documents to check for the location of structures and structural elements, and to collect data related to these structures. The current means by which bridges are inspected and assessed uses a condition rating method, whereby the inspectors go to the bridge and assign a condition rating to the various elements of the bridge. The rating of the entire bridge is based on the conditions of all the elements, no matter the relative importance of the elements. This method, while easier to collect data, does not recognize that the nature and location of damage on an element, and the type of element will greatly change the reliability of the overall structure.

The above mentioned kind of work environment requires collaboration at work. This allow synchronous and asynchronous real time seamless interactions between individuals who define common objectives and work actively and effectively to achieve these common goals, participating in agreed business processes. It requires seamless interaction within a working environment with context-aware location-based services. Thus leading to an interconnected world, where human needs are at the center. Workers will interact throughout this new world anywhere, anytime, via mobile hosts surrounded by an intelligent environment. Adopting efficient methods using emerging computing technologies, such as mobile computers and data mining technology, can significantly save costs of conventional maintenance strategies. It also helps in determining the location and amount of damage along all the elements so that the relationship of the damage to the reliability of the elements and to the bridge can be assessed.

Consider a scenario where a team of bridge inspectors are inspecting a large highway bridge. Some inspectors are visually inspecting the more accessible parts of the structure. They rapidly record what they are seeing using a speech interface to their data collection mobile host. They compare the current condition to that reported during previous inspections by viewing any of the previous inspections reports stored in a data grids. The comparison that may be most appropriate for the inspector using the mobile host, may be resource intensive and can be performed on demand computations in the other portion of the grid. Some inspectors are querying sensor systems embedded in, or attached to, specific locations on the structure using a mobile host for interfacing with these various sensors. The inspectors are able to access the loading conditions and corresponding structural responses, and any changes in local material properties, of the structure since the last inspection period. Other inspectors are operating mobile hosts from a remote position somewhere on the bridge. They are able to move these hosts to different locations so as to view and sense the condition of the underside of the bridge superstructure. They are also able to control and interpret the information being provided by these hosts. Finally, other inspectors are using advanced and sophisticated hosts such as laser scanning, ground-penetrating radar and other technologies to inspect certain elements of the structure, such as the bridge deck. As these inspectors are collecting and viewing the results collected from these various sensor systems, they are evaluating the conditions of the structure and the need for additional tests, with the help of decision support. Hence, it is essential that the data is manipulated on high-performance machines that are part of the data Grid forming the back-end instead of the mobile hosts that are the front-end.

3. BACKGROUND

This section presents information necessary to illustrate how our proposed mobile grid model is built by describing Virat [Srinivas and Janakiram 2006], a distributed object space paradigm.

The key idea of Virat model is that it supports wide area shared object space and provides a higher level of abstraction to application developers. Virat uses a novel mechanism for handling failures and provides a data centric concurrency control mechanism to realize various consistency models.

It is non-trivial to extend existing Distributed Shared Memory (DSM) systems to the Internet scale. The additional messaging latencies, non-deterministic message deliveries and failures pose challenges. Specifically, DSMs need to use some form of check-pointing to recover from failures. Coordinated check-pointing involves synchronization between the various nodes to ensure consistent global states. Most existing DSMs use coordinated check-pointing which would inhibit their scalability [Sultan et al. 2002]. Further, existing DSMs either use centralized mechanisms or some form of group communication (multicasting or broadcasting) for object location. This is another factor that affects their scalability. In this context, we have endeavoured to build a large scale DSM that handles these issues elegantly.

Developing infrastructure support for large scale distributed applications is an arduous task, especially over a Wide Area Network (WAN). Node failures, network failures as well as non-deterministic message latencies pose challenges. Further, various object interaction styles have to be supported so that applications can be architected easily. In this context, we have developed Virat, a wide area shared object space[1] that addresses these issues and provides a higher level of

[1] A shared object space such as Linda [Carriero and Gelenter 1989] enables sharing at the level of application

abstraction to application developers.

Virat supports middleware services such as naming and trading as well as replica object management. Virat uses an independent check pointing and lazy reconstruction mechanism to handle failures of object repositories. The object repositories (one per cluster) are responsible for cluster level management of replicas. Communication between the object repositories themselves is through a peer-to-peer protocol. This is useful for locating objects across clusters.

Virat provides a data centric Concurrency Control (CC) mechanism to realize various consistency models. Virat has been extended to a fully typed shared event space, facilitating publish-subscribe kind of interactions and large scale event notifications. A shared service space has also been built over the shared object abstraction of Virat. This allows services to be discovered at runtime and composed, as well as dynamically reconfigured. Virat has been implemented using J2EE over an Institute wide network and its scalability is being tested over a WAN.

4. AN OBJECT BASED MOBILE GRID ARCHITECTURE

The proposed mobile grid is organized as a cluster of clusters as illustrated in figure 1. Each cluster may refer to a logical group, which includes both SH and MH. The MHs are handled by the mobile support station (MSS) of the traditional cellular system. Each cluster is coordinated with a designated static host acting as a cluster head (CH). The CH manages all the resources and services which within their cluster. Depending on MSS load conditions, MSS and CH may be configured on same host or in different host. The CH coordinate among the other neighbouring CHs in a peer-to-peer fashion.

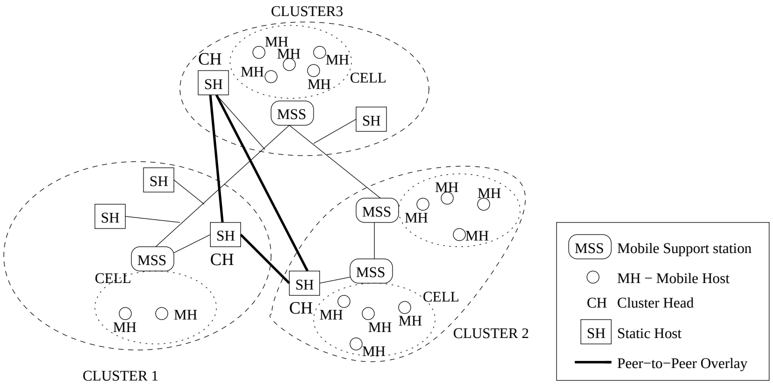

Figure. 1: Mobile Grid Architecture Clustered as P2P overlay

The proposed architecture deals with considering each participating MH as a unique object, called surrogate object (SO). The SO acts as a representative of the mobile host in the wired network. This helps to maintain transparency to the instability of wireless communication. The SO can remain active, maintaining information regarding the current state and plays an active role on behalf of the MH. When a MH enters a cell it sends a control signal to the corresponding MSS, which consists of the address of the MSS from where the mobile node is entering. In case when the old MSS address is null it indicates that the node had entered the system for the first time and thus creates the surrogate object encapsulating the mobile device completely. The object reference of the created surrogate object is sent to its MH for its future interaction with the SO. In addition to the SO, to have a unified design, all other entities, including the SHs are also represented by independent objects. Each object has one or more interfaces, with each interface consisting of one or more methods, representing the services offered by the node.

objects, unlike traditional Distributed Shared Memory (DSM) systems that share pages [Stumm and Zhou 1990].

By encapsulating the participating nodes, in distributed objects, the grid is transformed from a collection of nodes, offering and consuming services, into an object space of distributed objects. Thus objects are the building blocks of our mobile grid. The proposed mobile grid structured as distributed objects is shown in figure 2.

Abstracting the resource constrained MH into SO, helps to handle most of critical issues associated with the MH. The key features of the surrogate object model are as follows:

- ✦ It provides an elegant solution for handling the asymmetry, due to wired and wireless bandwidth. The surrogate object model is a customizable approach, meaning that host specific and application specific constraints can be enforced. This would be difficult to achieve without the surrogate object model.
- ✦ It also provides an ideal placeholder for MH location information, thus solving the location management problem in the system.
- ✦ It acts as a data source for handling data dissemination to provide mobile data access, in both *server-push* and *client-pull* models [Jing et al. 1999].
- ✦ The SO can also cache MH specific data and reduce the response times for many client queries. It also supports disconnected operations of the MHs by buffering client requests or using the cached data to handle them, and also helps to save task results temporarily when MH is disconnected.
- ✦ It provides optimal utilization of wireless bandwidth, as the SO knows the current network connectivity and other constraints of its corresponding host.

The proposed approach significantly help in realizing a distributed and decentralized infrastructure of SOs that work on behalf of the participating hosts and are hosted by the wired network. With SO the MH movements does not affect the service provisioning, as the entire state of the host is maintained. The model helps in achieving the properties of dynamicity, asynchronocity, autonomy and security.

The proposed infrastructure supports dynamicity by letting new services modify and extend the wired network infrastructure, thus helping in adapting services based on the requirements of the mobile grid users. The surrogate objects can also play the role of data processors, by directly offering the required data and accepting the data that need to be delivered to the corresponding mobile node. Mobile computing can also take advantage of asynchronicity and autonomy between user requests and their execution. For instance, wireless connectivity impose strict constraints on available bandwidth and communication reliability and force mobile nodes to minimize their connection time.

The SO encapsulates all the characteristics and properties of a host fully, such as the computing power, memory availability, bandwidth, and all other resources and services associated with the host. The representation of the characteristics of the hosts in the SO is made as attributes, methods and subobjects. The attributes of the SO includes the computing capability of the node, the memory capability and the bandwidth of the medium by which the node is connected. The methods and the subobjects of the SO represents the services and other resources that are offered by the node. In addition each SO encapsulates the security policy and agreement for each of the services that is associated with that node, which will specify how and by who the service may be used.

The SO model does not assume continuous network connectivity, rather it requires wireless connectivity to exist only for the time needed to place SOs from MHs to the wired network. SOs are autonomous and can carry on services even after the corresponding MHs disconnect. Thus the model helps in permitting mobile grid user to continue using services from SO with the available state information maintained, even when the MH is physically not reachable. They can then return service results and deliver the data to the host once it reconnects. The paradigm is made to support security by incorporating the policies within the SOs regarding the authority to access the services and objects.

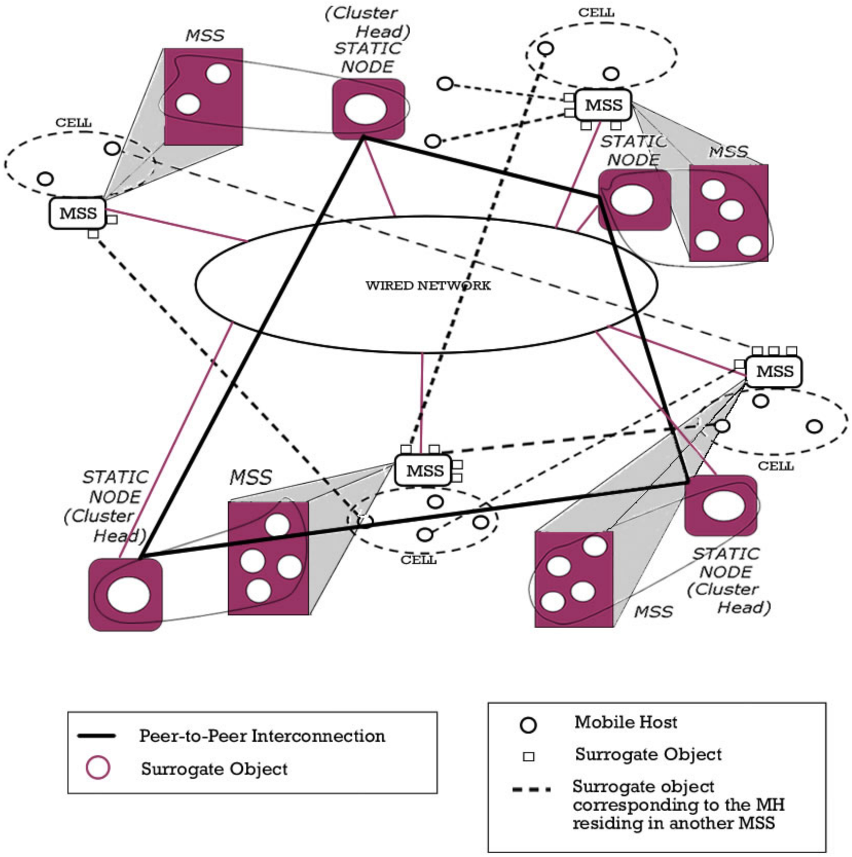
MSS
(Cluster Head) STATIC NODE
CELL
MSS
CELL
MSS
STATIC NODE
MSS
WIRED NETWORK
MSS
CELL
STATIC NODE (Cluster Head)
MSS
MSS
CELL
STATIC NODE (Cluster Head)
MSS
Peer-to-Peer Interconnection
Surrogate Object
Mobile Host
Surrogate Object
Surrogate object corresponding to the MH residing in another MSS

Figure. 2: Mobile Grid Architecture with Surrogate Objects

When the MH acts as an information service provider, contacting the host directly will lead to consumption of the constrained resources like battery and bandwidth. In this case it would suffice if the corresponding SO is contacted to get the information. In addition the SO's can be replicated to avoid the congestion in the network and to improve scalability of the system. With the SO being fully autonomous, users can access services even if the node disconnects because the SO delivers the results upon reconnection.

5. MOBILE GRID COMPUTATIONAL MODEL

Surrogate objects are physically distributed, spreaded out over multiple address spaces, providing view of distributed shared object (DSO) space, as an analogy to distributed shared memory. Thus in the proposed model the processes interaction and communication is actually through the distributed shared objects. The DSO is built over the participating mobile and static hosts objects, using Virat [Srinivas and Janakiram 2006]. When a MH wants to participate in a mobile grid, it registers itself with an MSS. The MSS assigns a unique identification for the MH namely, the Mobile Host Identifier (MHID) and passes the information to the underlying middleware about the entry of a new MH to the system. The middleware creates an object corresponding to the MH and assigns a unique Surrogate Object Identifier (SOID), and gives the DSM run time object instance to the MH. The DSM run time object essentially acts as a mediator between the client and the DSM services.

5.1 Horse Power Factor and Metrics for the Services

As each participating host may have different services capabilities (service includes all resources including memory, processing power, etc.), it is essential to allocate tasks to the SHs and MHs based on their capabilities. The task may involve either a computing task or data which needs memory space for storage, etc. Thus the metrics associated with each service is calculated and advertised in the trading service along with the registration of the services. To incorporate this,each service is allocated integers which refers to the metric, which is dynamic. The metric in case of processing power service is called as Horse Power Factor (HPF) [Joshi and Ram 1999; Mohamed et al. 2005], which is a measure of the computing power of a machine, load on the machine and the network bandwidth of the communication channel. Machines in the network are normalized by a benchmark program to obtain a relative index of the machine, which is a static factor. The dynamic HPF of a machine is obtained using this static relative index, the load on the machine and the communication bandwidth with which the machine is connected to the network. This dynamic factor is normalized as a factor that represents the number of entities that it could compute. When a host has HPF 'h', then 'h' computing entities are allocated to the host. For example, if hosts A and B have h_1 and h_2 as their respective HPFs, then the time taken by A to compute h_1 amount of a task is approximately equal to the time taken by B to compute h_2 amount of the same task. Similarly metrics related to other services, such as memory are calculated. Abstracting the heterogeneity in this way makes it viable for parallel processing on unevenly loaded heterogeneous machines.

5.2 Directory Services

The proposed model virtualizes all the resources and services offered by the participating nodes as services, to ensure uniformity during service composition, which may need both resources and services. The proposed mobile grid architecture which is a collection of non overlapping clusters, also provides directory service, namely trading service, to register resources, services and other properties, while providing efficient search facility to the clients, for locating the servers. The trading service provides a mechanism to support the offering and discovery of instances of services of particular types. Traders mediate between service providers and service requesters, thus providing a loosely coupled architecture where the binding between provider and requester is established at run-time and can change. The architecture provides local service management within the cluster in the CH. Global service management is done by grouping the information from the CHs through peer to peer interaction among CHs.

Every cluster supports trading service in the CH, for storing, managing and making available data about all the servers in that cluster. This provides the bindings between components (objects providing services) be discovered or rediscovered at run time. This helps in overall system scalability. A provider that wishes to offer its services to the mobile grid users, registers their SO, with the trading service by providing details of its services with their properties and capabilities.

The required service offered by the object is identified by the functionality of the service. The functionality of the service is mapped by a directory service namely, trading service to the SOID, that offers the required service. To locate the actual location of the object referred by SOID, the naming service is used to extract out the object reference. If the required service is not available within that cluster then peer-to-peer search is done among the cluster heads. Thus the design also takes care of the close proximity of the services. Before selecting the required service, from the same kind of services available from more then one server.

5.3 Mobile Node Location Management

Mobility is a natural process in mobile computing scenario [Forman and Zahorjan 1994; Imielinski and Badrinath 1994], as the user of the mobile device has the advantage to move. This leads to changing the point of attachment of the device with the wired network. Thus maintaining the location information of mobile device in the mobile grid model becomes one of the critical issue the handled. Many techniques had been proposed from up-to-date information maintenance to no information maintaining, based on the compromise on availability, precision and other properties [Pitoura and Samaras 2001].

In the proposed model it is not necessary that information of the current location of the mobile devices needs to be maintained. Instead maintaining the current location of the corresponding surrogate object is done, which is equivalent to maintaining the location of the mobile device itself. The benefit of the technique is that the surrogate object is not made to move with every move of its mobile device. Thus, after some point of time the mobile device might have moved to a far off place with respect to the surrogate object. This leads to increased latency and heavy traffic in the wired network, when there is a need to maintain consistency between current state of the MH and its associated surrogate object. This is because the messages need to travel more hops as they are physically separated out by a long distance. Hence, it will be advisable to keep the MH and its associated surrogate object as near as possible. However, migrating the surrogate object from one MSS to another with the movement of the mobile device from one cell to the other will also be inefficient. There will also be cases where the MH may keep moving back and forth between cells. In such cases swapping the surrogate object between the MSSs will be highly inefficient. Thus, both the cases of never moving the surrogate objects or always moving the surrogate objects are ineffective. Instead the surrogate object is moved to the next location only when the mobile device has moved 'n' hops away from the current location.

Surrogate object migration needs to be handled at the distributed shared object runtime level, as the surrogate object resides in the wired portion of the network. Existing middlewares such as CORBA do not handle object migration efficiently. The granularity of migration cannot be at the level of individual objects, as this affects the scalability of the implementation repository [Henning 1998]. We have developed an object migration mechanism based on the concept of message filters to allow individual objects to migrate freely in a distributed system [Janakiram and Srinivas 2002]. A message filter which is always static is attached to each surrogate object. The filter has the same interface as the surrogate object in addition to certain special methods for plugging/unplugging the filter and to update the location of the object. The filter maintains the current location of the surrogate object and is responsible for redirecting client requests to the current location. The filter also handles the migration window, i.e., the period during which the surrogate object is migrating. Once the client gets the new object reference from the filter, it caches the new location and the rest of the communication is directly routed to the current location of the server. Full details of the object migration mechanism is beyond the scope of this paper and is available in [Janakiram and Srinivas 2002].

5.4 Message Delivery and Node Mobility Management

Mobile connectivity is highly variable in performance and reliability. The wireless communication channels used by the MHs also have a lower bandwidth than the wired channels. Mobility implies that a MH changes its location. Thus, the location management for the targeted MH becomes an indispensable task of any application that runs on the distributed mobile system. This complicated issue could be handled effectively without affecting the delivery using our proposed model. The sender of a message targeted to an MH need not bother about whether the MH is in motion or out of coverage region. All that needs to be done is to deliver the message to the surrogate object which is residing in the static portion of the network. The surrogate object takes opportune time to deliver the message to its MH considering the availability of the wireless bandwidth and the traffic on the network.

Mobile hosts often get disconnected from the rest of the system due to mobility or by switching into doze mode to save power. Disconnected operations is a regular feature in mobile computing and is distinct from failure [Kistler and Satyanarayanan 1992]. Disconnected operation due to doze mode is voluntary in nature and a mobile host can be required to execute a disconnection protocol before its detachment. Thus, the algorithms have to accommodate such voluntary disconnections and make progress during the disconnection of mobile hosts. This is done by caching the current state of the MH in its place holder namely the surrogate object. However in case of disconnection due to mobility the proposed surrogate object model could help in continuing with the execution of the application using the available information cached in the surrogate object.

5.5 Mobile Grid as Data Grid

In addition to the CPU resources the second most common resource used in a grid is data storage. Sharing of data in the form of files or databases can also be done using the proposed Mobile grid, thus behaving as a data grid in addition to the computing grid. A "data grid" can expand data capabilities in several ways. First, files or databases can seamlessly span many systems and thus have larger capacities than on any single system. Such spanning can improve data transfer rates through the use of stripping techniques [Ferreira et al. 2002]. However using the memory of the participating mobile devices for storing data will not be much reliable with respect to the getting data whenever required. This is because the mobile device are prone to getting disconnected from the network due to mobility. But still the memory of the mobile devices can be used for the data to be duplicated throughout the mobile grid to serve as a backup, which is one of the essential requirement.

6. SERVICES MANAGEMENT

Early grid middleware emphasized low-level resource management aspects. However, now the grid computing has adopted the notion of *services* as a basic building block for large-scale scientific computations. Among the most challenging problems posed by grids are service discovery and coordination, that is, how to find and orchestrate a set of services operating on heterogeneous resources under different administrative control in order to solve a given problem. Mobile grids are even more dynamic than a traditional grid, services may join and leave the system at any time by their own will or due to mobility or due to faults. Current grid frameworks do not adequately support dynamic service discovery. Heterogeneity is mostly addressed by creating a common (XML-based) protocol to achieve interoperability.

One of the focus of our mobile grid paradigm is moved from CPU sharing to defining computational resources and functionality as composable services, which also includes services offered by the MHs. With services as basic building blocks, it becomes possible to rapidly build grid applications by composing these services at a high level of abstraction. Service Composition [Ser] allows different autonomous services to be combined in such a way that a new service with a different functionality is created. Service composition allows highly modular and independent services to be developed and then use all of them to create a much larger and more complex service. Service composition is intimately linked to service description, as well as service discovery. In order to have an automatic composition of services, it is necessary to know beforehand what services are available and their functionality. User preferences, rules set by providers should also be taken into account. The users location and the technological means available at that moment will also influence service composition. The proposed mobile grid also supports the most optimal partitioning strategy based on the available resources.

The MHs in the mobile grid add new challenge to service composition. A user may request a service from a host, and then move to another host which has more limited functionality and is not able to display the services results. Also, the user may move between different access points, so if the service requested is location dependent, its result will be invalid. Therefore, service composition must handle possible errors, eventually restarting the service composition process with the new parameters, so that the result will be useful for the user. The layered architecture for the service composition infrastructure of our mobile grid is as shown in figure 3.

The Resource Virtualization layer, namely the distributed object space represents the mechanisms through which services express themselves and concentrates primarily on the use of resource sharing in the grid. Resource Virtualization is thought of as an abstraction of some defined func-

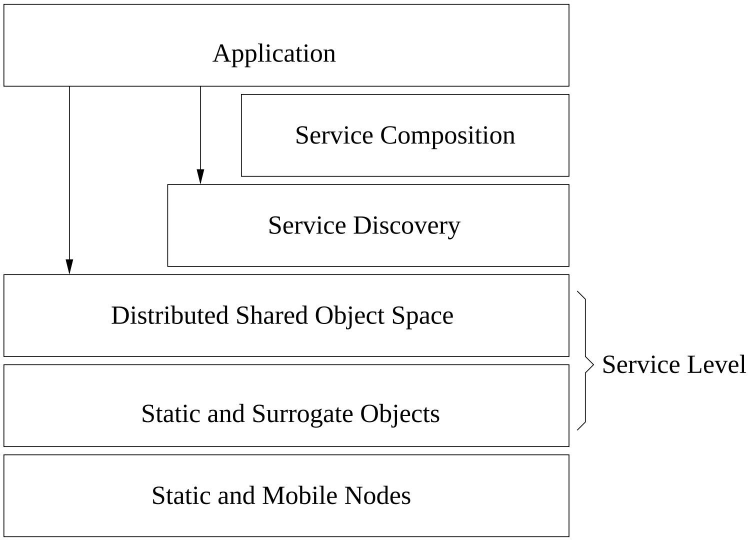

Figure. 3: Layered Reference Architecture for Service Composition in Mobile Grid

tionality and its public exposure as a service through an interface that applications and resource managers can remotely invoke. We consider the service to be a virtualized software functional component. Services can be advertised and discovered using directories and inspection. Once discovered, an invoking entity can bind to the selected service and start communicating with its externally visible functions, through platform independent protocols. Each such virtualized component can be abstracted, discovered and bound to. In the same way this can be extended to virtualize hardware resources. Thus a planetary-sized pool of composable hardware and software resources can be envisioned.

Discovery is the fundamental component, as the system must find a service before it could use it. Traditional systems often implicitly discover or fix it at configuration or compile time. In case of mobile grids which has to accommodate both mobile and static devices with dynamic aggregated environments, it should be able to support a flexible service advertisement and discovery mechanism. Applications can discover services based on their functionality, characteristics, cost or location. Dynamic discovery enables devices to adaptively cooperate and extend functionality.

Dynamic configuration and service composition depends upon the capability to bind components at runtime, in contrast to the binding at design time. Runtime binding can be implemented with the assistance of service discovery mechanisms. This helps in decoupling application design from the detailed awareness of the underlying system configuration and connectivity. Dynamic configuration facilitates application portability across a wide range of platforms and networks configurations. Runtime binding also enables load balancing and improved reliability. Service Composition deals with control and management of the aggregated set of services for completing a task. It also includes the communication and synchronization necessary for coordination and collation of partial results. On completion of a task, the system releases the services back to the pool for allocation to other users.

6.1 Service Description

Service description are needed to describe the services provided by various components of the mobile grid. The providers of services need to publish the characteristics and requirements of the services. The service descriptions are defined by the providers and sent to the cluster head for registration. Once the services are published, a discovery protocol is needed to map the services to the application requests. The notion of grid service [Foster et al. 2001] is extended to the proposed mobile grids. The mobile nature of some of the participating components makes it challenging to provide for discovery mechanisms across virtual organizations.

Two important features considered in our service model are: (i) it is a semi-structured. No specific schema is required so that the system can work naturally in a heterogeneous environment, and (ii) it allows providers to express constraints on the services they are willing to serve. There are two main parts in the description, namely, attributes and constraints. The attributes part includes characteristics of a resource, such as location, CPU usage, and free memory. The values of some characteristics are dynamic and change over time, for example the amount of free memory, free storage space, and CPU usage. The values of dynamic characteristics can be obtained via a daemon process running on the resource. The constraints part includes constraint expressions defined by the service provider for the allocation of this service and also includes the monetary processing cost charged.

The mobile grid services are maintained using the trading service repository. Each logical cluster of the mobile grid maintains a service directory in the CH, which acts as a central point of services registry for that cluster. Any participating node that wishes to offer its services makes its information available in the service directory by providing details of its services and other relevant properties like security policies. The repository maintains the details of the services and its associated SOID (or the OID in case of SH), which provides the service on behalf of the actual host.

6.2 Service Discovery

In order for an entity to cooperate with others in its vicinity or use the services available on the fixed grid, it needs to discover them. This problem of ”service discovery” has recently been explored elsewhere as well in the context of distributed systems. Thus there is a need to develop mechanisms which allow for components to describe themselves (at a semantic level) and their ”requirements”, as well as for other components to locate them. The existing systems such as Jini, Salutation, UPnP, Service Location Protocol [Lee and Helal 2002], Ninja [Czerwinski et al. 1999], and UDDI [udd] provide for networked entities to advertise their functionality. However, these systems are either based on a language (Java/Jini), or describe services entirely in syntactic terms as interface descriptions. This not only limits interoperability, but forces a client to know a-priori how to describe a service it needs in terms of an interface. Moreover, they return ”exact” matches and can only handle equality constraints. This leads to a loss of expressive power in the component description. When we look at service discovery in case on mobile grids the case is even worse and makes it clearly inadequate.

In the proposed model, we make a distinction between local service management and global service management. Local service management deals with managing services within a domain or a logical cluster(say, cell). In other words, if two processes are in the same cell, the directory service will consider them as being at the same location. Global service management deals with grouping the information from cells into larger units, and making that information available to clients. In particular, global service management deals with globally tracking services, and providing efficient search facilities for clients, regardless of a clients location.

The local service management is done in cluster head of each logical cluster. The basic components that are required to manage the services within the cluster includes the information database, scheduler and an execution monitor. The information database contains the information that describes the services (includes resources and services). The scheduler queries the information database and based on the result and knowledge of the application, computes a mapping of objects to services. Using this binding the scheduler contacts the object and confirms the schedule. The entire process is monitored by the monitor.

The service is associated with the SOID, which depending on the client request is retrieved using inverted index technique from the trading service repository maintained in each CH. The CHs are connected in a peer to peer fashion, and inorder to handle scalability of the system, Distributed Hash Table (DHT) functionality is used [Stoica et al. 2003; Rowstron and Druschel 2001]. The DHT nodes form an service overlay network with each node having several other nodes as neighbours. When a route(SOID) is issued, the function routes through the overlay network to the node responsible for that key (namely SOID).

6.3 Service Composition

Given a certain ordering of several sub tasks that may be executed to derive the result of a complex request, the problem is how these heterogeneous tasks can be integrated and executed in environments where there is a combination of resource-rich and resource-poor devices interconnected to each other by wired or wireless communication channels.

Web services is a new solution for dynamic business interactions over the internet. At present, the technological infrastructure for Web Services is structured around three major standards, namely WSDL (Web Service Definition Language), UDDI (Universal Description, Discovery, and Integration), and SOAP (Simple Object Access Protocol) [Curbera et al. 2002]. These technologies aim at supporting the definition of Web services, their advertisement to the users, and their binding for activating purposes. However the proposed model looks at even more challenging problem of composing services dynamically, on demand [Benatallah et al. 2003]. In particular, when a functionality that cannot be realized by the existing services is required, the existing services can be combined together to fulfill the request.

Service Composition refers to the construction of complex services from primitive ones, thus providing rapid and flexible creation of new services. Fixed broker-based or fixed central-entity based service composition techniques cannot be applied as a solution to compose services on the fly as the clients can be mobile. The client may also use a service which may be mobile and connected through wireless network. This is due to the fact that such approaches presuppose the existence of a persistent central entity or coordinator in the clients neighbourhood, which is reliable and constantly available on request. Thus service composition can be defined as a dynamic integration of multiple services available in an mobile grid environment in response to a request from a client.

Services Composition provisioning is a very active area of research and development. However, very little has been done regarding the provisioning of services in hybrid environments where services may be hosted on mobile or fixed computing resources. To optimize service provisioning in mobile environments, several important issues need to be considered. Two of the more critical issue that the proposed model handles are related to handling disconnections during service execution and context-sensitive service execution planning. In a mobile environment, disconnections may be frequent (e.g., disconnection due to the problems of battery or communication costs, disconnection due to the fact that devices can change location). As a consequence, dealing with user or service device disconnection during service execution is a critical issue. In addition to criteria such as monetary cost and execution time, service execution planning should consider the location of requesters of services, capabilities of computing resources on which services will be executed (e.g., CPU, bandwidth), and so on. It is a necessity to enable the system to adapt itself to different computing and user's requirements. Therefore, the identification of the resources on which the execution of the service takes place is very important.

The surrogate objects representing the mobile devices helps in handling the above mentioned issues on behalf of the mobile devices as addressed in previous sections. They are autonomous, make decisions, and interact with the other devices in the system. The object architecture offers interesting features for service provisioning in mobile grid environments. For example, the surrogate object may be used to collect execution results during disconnection of the mobile device, and returns these results to the user upon re-connection. However, the proposed composition technique solves the problem of read only kind of services, and does not much discusses on the services were live updates are to be carried out.

The required services are tried, if they are individually available, or composition of services based on the the required functionalities are selected during each execution of the composite service. Based on a set of service quality criteria such as execution cost, resource reliability, etc., the execution of the composition is done using the optimal computing resources on which services will be executed.

A client and a service provider are not distinguished. Composite requests may be sent out from multiple hosts. A node initializing a composite request can itself participate in another composition. The client of each composite request forwards the request to its associated CH, which manages the discovery, integration and execution of the composite request. The composition of service is done using the compositional language Soma [Janakiram et al. 2006]. The critical challenge related to availability of the composed service-level path across multiple cluster domains in an wide area network is handled by the peer to peer organization of the cluster head. This ensures the fault tolerance of the system.

7. PERFORMANCE ANALYSIS

The performance of the proposed mobile grid model is studied at two levels, namely to study the performance of the new surrogate model and to study the grid model which uses the surrogate object. A simulator that suits the requirements of our model was required. The most common freely available network simulators ns-2 [McCanne and Floyd 1995] and GloMoSim [Zeng et al. 1998], do not support cellular network architecture. Thus two different set up was made to evaluate the two set of performances.

7.1 Surrogate Model

The first was a simulated model of the cellular network. Query execution scenario was studied over both the surrogate and without surrogate object models. Throughout this section, the algorithm without the surrogate object model is referred to as the old model and the one with the surrogate object model is named as the new model. Typical performance studies include actual number of packets lost for different packet loss probabilities in both old and new models; proportionate increase in the query time for different packet loss probabilities in both the models; impact of movement of the surrogate object in terms of packet loss and message traffic.

7.1.1 *Comparison of Old and New Models.* Figure 4 shows the comparison of the query latencies in the caching application over both the old and new models. In each graph, the query time variance is plotted against simulation time for different packet loss probabilities. The study shows that as the packet loss probability increases, the query time increases. However, in the new model, this increase is not significant compared to the same in the old model. Further, it can be observed that the maximum query time with zero packet loss probability is much higher in the old model, compared to the maximum query time in the new model, even with 5% packet loss probability. Figure 5 shows the percentage increase in query time as packet loss probability in the wireless network increases for both the old and the new models.

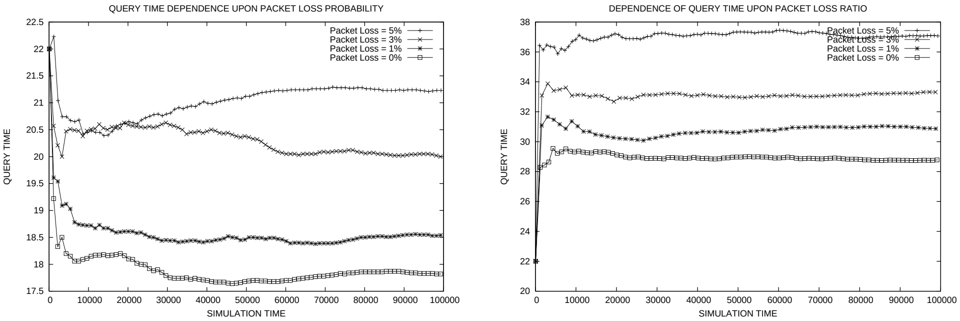

Figure. 4: Comparison of Query Latencies in the New and Old Models

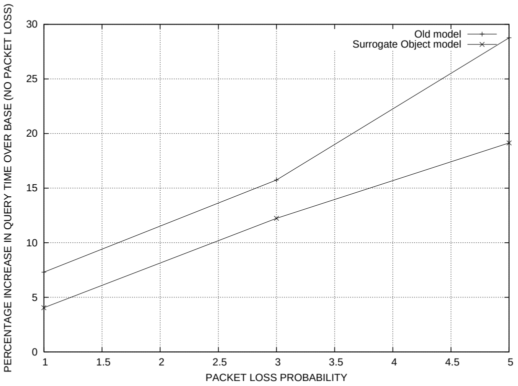

Figure. 5: Percentage Increase in Query Time with Increasing Packet Loss Probability

7.1.2 *Effect of Surrogate Object Migration.* One of the key advantages of the surrogate object model is that it is free to be migrated to any node in the wired network. This flexibility is desirable for various reasons, including failure recovery, load balancing and network latency reduction. Figure 6 shows the effect of surrogate object migration on the query latency for various migration frequencies. It can be seen that if the surrogate object is migrated every time the MH moves, the query latencies are considerably high. This is due to the increase in the percentage of time spent in migration itself. But as the move frequency is reduced (the surrogate object moves only once for every 'n' moves made by the MH to various cells), the query time improves. Surprisingly, further changes in the move frequency seems to have almost no impact on the query time. This means that for any 'n' other than 1, the query time is the same as if the surrogate object is static, never migrated. The reason for this behaviour is that query response time is more dependent on the nearness of the surrogate object to the originating MSS and the queries are generated randomly from different parts of the network. A closer study of the network traffic generated as a result of surrogate object migration reveals the reason for providing migration freedom for the surrogate object.

Figure 7 shows the network traffic in terms of number of messages exchanged versus simulation time for various move frequencies. It can be seen that if the surrogate moves with every movement of the MH, the traffic is an order of magnitude higher. As the move frequency reduces, the traffic generated is much lesser. But the frequency value (other than 1) does not affect the network traffic. This implies that it is not a very good strategy to move the surrogate object every time the MH moves. This would be the case if, instead of using the surrogate object model, a similar effect is to be achieved using data structures at the MSS. These data structures would have to be moved with every handoff.

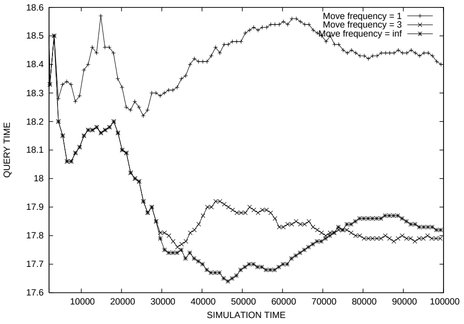

Figure. 6: SO migration: Query Time

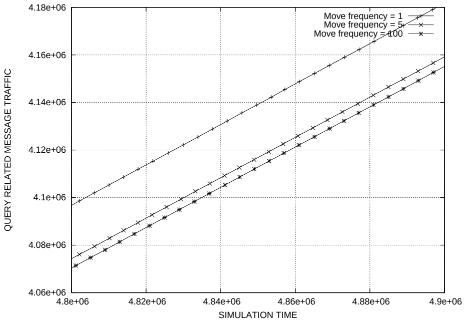

Figure. 7: SO migration: Packet Loss

7.2 Mobile Grid using Surrogate Object

The problem to determine the area covered under the curve was considered to study the performance of the proposed mobile grid. The standard technique to solve this problem, is to make the entire area into smaller slices. The area of each slice which will be a trapezoid is calculated. The sum of areas of all the slices is the area covered under the curve. The increase in number of slices, increases the computation.

The experimental model of the mobile grid was a simulation over the actual implementation. The mobile grid was built using the distributed shared object space, namely the virat [Srinivas and Janakiram 2006]. Fifty heterogeneous machines, each having memory from 128MB to 512MB, processing speed from 2 to 2.4GHz, from the institute wide network was considered. In order to realize a cellular model, seven nodes were designated as MSS, and all the remaining nodes were assumed to be MHs of the cellular network. The MHs were divided into seven groups, and the MHs in each group were made to communicate with any of the other nodes only through there corresponding MSSs. Inorder to realize a mobile scenario the communication from the MH designated nodes were delayed, to correspond, to wireless bandwidth. The nodes designated as MHs dynamically changed there groups to realize the mobility pattern of the MHs. The cell permanency time was assumed to be 2 seconds. The out of coverage of the MHs were also simulated dynamically, by making the MHs not to be a member of any of the MSSs. Each MSS and the MHs within there corresponding cell were assumed to be a cluster. The MSS designated node was the cluster head.

Some of the participating nodes were loaded using a program called cpuhog [Libenzi 2001]. Apart from loading the processor, cpuhog also hogs the memory resource. The purpose of this tool is to create a set of processes that are going to load the run queue with different cache drain values. The loading was done to simulate, that the capability of the participating MHs with respect to the computing and memory are different (as the capability of laptop, palmPC, etc. may vary). The capability effects was done by using cpuhog with memory of 1MB, and multiple half-load processes in the processor run-queue. Thus using this technique the 43 participating MHs, were grouped into four different sets. 11 machines without any load (Type 1), another 11 machines were loaded with one half-load (Type 2), another 11 machines were loaded with two half-load (Type 3)and another 10 machines were loaded with three half-load (Type 4).

Each MH has a client and a daemon. The clients are used to submit tasks to the MHs for distributed processing. The daemons are the computing entities at the MHs, that execute a part of the submitted task concurrently with other daemons. The amount of computation and data for each participating MHs was given based on the capability of each MH. The experimentation was conducted considering a granularity of 1,00,000 slices. Table I shows the time taken and the speedup attained using the grid with various combinations of MHs participating. The speedup was calculated taking the average of the time taken when executed with single different type of MH. The experimentation was done without mobility of the nodes and with dynamic mobility of all the 43 participating node (including the node which initialized the application).

8. RELATED WORKS

A computational Grid is a set of hardware and software resources that provide seamless, dependable, and pervasive access to high-end computational capabilities. It started with its roots within the field of high-performance parallel computing. By enabling the use of teraflop computers and petabyte storage systems interconnected by gigabit networks, the grid allows scientists to explore new avenues of research. Three different approaches have emerged within the last decade that provide alternatives to the massively parallel processor platform. Networks of Workstations (NOW) [Anderson et al. 1995; Joshi and Ram 1999] take advantage of clusters of uniprocessor workstations connected via a network to provide high performance at low cost. For example, Beowulf systems [Sterling et al. 1995] look to leverage low-cost, high-performance Linux PCs with commodity networking. Additionally, Condor [Litzkow et al. 1988] provides the capability

Table I: Results with Tasks Submitted Based on the Capability of the MH

		Without Mobility		With Mobility	
#MH/s	**Type of MH**	**Time Taken (sec)**	**Speedup**	**Time Taken (sec)**	**Speedup**
1	1	633		659	
1	2	717		723	
1	3	803		822	
1	4	911		937	
4	1 - 1	279	2.745	302	2.600
	2 - 1				
	3 - 1				
	4 - 1				
8	1 - 2	197	3.889	233	3.370
	2 - 2				
	3 - 2				
	4 - 2				
16	1 - 4	113	6.778	144	5.453
	2 - 4				
	3 - 4				
	4 - 4				
32	1 - 8	77	9.948	93	8.443
	2 - 8				
	3 - 8				
	4 - 8				
42	1 - 10	49	15.632	62	12.665
	2 - 11				
	3 - 11				
	4 - 10				

to share processing jobs across a Unix NOW to achieve load-balancing.

At a much larger scale of distribution, meta computing [Smarr and Catlett 1992; Grimshaw et al. 1997] links geographically diverse supercomputing resources via a high-speed network. This conglomeration of gigaFLOP-capable centers into a teraFLOP-capable virtual computer can yield vastly increased performance for applications that can take advantage of this architecture. Work using this architecture focused on computationally-intensive tasks that could be naturally distributed, such as dynamic macro molecular visualisation.

The third approach, grid computing, emerged directly from the meta computing concept but has now morphed into a resource-sharing paradigm analogous to the current peer-to-peer concept. Much pioneering work in grid computing was done with the Legion [Natrajan et al. 2001] and Globus [Foster and Kesselman 1997; Foster et al. 2001] research efforts. Globus has emerged as the middleware standard for a number of different grid projects and provides a 4-layer stack to control hardware, communications, resource sharing, and collective coordination.

Most of the current grid architecture and algorithms do not consider mobile computing environment, since mobile devices were not seriously considered as valid computing resources or interfaces in grid communities. To the best of our knowledge no other project has the same broad scope as that of us. Research efforts such as [Gonzlez-Castao et al. 2002; Migliardi et al. 2002], had mainly tried justifying the adaptability of there already proposed grid to mobile grid. [Gonzlez-Castao et al. 2002] proposes a hierarchical design methodology for grid access from handheld devices. It has also come out with a classification on the internet-enabled devices in a hierarchical way, which in no way leads to any model of the grid. Further, they had extended their earlier grid model using HTTP to provide access to the grid for PDAs and cell phones. [Migliardi et al. 2002] describes the issues related to providing mobile access to computational, data, and service Grids. It presents efforts to enhance computational and service Grids to handle mobile access. However it lacks a model specific to mobile grids which can address common problems such as connection failures, service discovery, bandwidth management etc. that are challenging in mobile environments.

Other projects such as [Clarke and Humphrey 2002; Phan et al. 2002; Park et al. 2003; Agarwal et al. 2004] had visioned to integrate these two emerging techniques of mobile and grid computing. However, they do not elaborate on how the mobile hosts may be incorporated in the current grid architecture. Most of them had focussed, only towards using the mobile devices as a front-end to grid, by accessing the grid.

[Clarke and Humphrey 2002], provides a insight on the idea of how legion project can be extended to use mobile devices as first-class compute and data servers. The work describes the unique capabilities of the Legion Grid computing infra structure that will directly support the mobile grid requirement. But legion object is characterized by inflexibility which affects considerably the mechanisms of failure recovery and load balancing. Further, the critical issues such as wireless bandwidth, mobility, etc. associated with the MHs are not addressed.

Leech project [Phan et al. 2002] describes the challenge of harvesting the Internet-connected wireless MHs such as PDAs and laptops to be beneficially used within the computational grid. They had also identified the research challenges that are likely to arise from this problem and had proposed their vision of a potential architectural solution. They had come out with a proxy based architecture, where the proxy is called as interlocutor. The interlocutor is a MH that represents to the grid the other participating MHs which are within the cluster. The architecture is based on wireless adhoc network, with the interlocutor acting as a bridging point to the grid. In contrast, the proposed surrogate object based model provides a fine grained architecture with static host, mobile host(MH), cluster head and mobile support station and the MH may be substituted by an surrogate object(SO). SO provides better solution for handling asymmetry, disconnected operations and location management with optimal utilization of bandwidth. Leech architecture had not also addressed the mobility and other issues associated with the MHs. Above all, Leech project provides a simple directory service and proposed model introduces better trading services with metric called horse power factor advertised for each service. Service discovery is vests in the hands of minions in Leech whereas proposed model makes clear boundary between local and global service by providing distinct management policy.

Our surrogate object model caches the current state of the MH in its place holder namely the surrogate object. In case of disconnection due to mobility our model continues with the execution of the application using the available information cached in the surrogate object. DCOS [Konanki and Butt 2007] supports disconnected operation using distributed proxies but fails to address its implications in a large-scale resource sharing environment like grid.

Similar to leech project, [dos S. Lima et al. 2005] proposes a mobile grid based on wireless adhoc network. However the model provides a decentralized approach that can support applications, in purely adhoc networks. As the adhoc networks are mostly seen as emergency kind of network [Royer and Toh 1999], it cannot be called as mobile grid, just because of sharing computing power among the mobile nodes in the adhoc network. Infact these adhoc networks are only meant for specific application purpose. This architecture as such does not provide service description and also trust management and privacy. Experimental Analysis has been carried out without taking into account voluntary and involuntary disconnections. In addition the paper also states that significant work has already been done towards mobile grids, and cited two earlier publications, namely [Kurkovsky et al. 2004; Yamin et al. 2003]. Both the cited paper do not brief about any model for the integration of the mobile and grid computing. Specifically [Kurkovsky et al. 2004], discusses about direct extension of the mobile devices into grid paradigm, considering mobile devices similar to that of a static node. [Yamin et al. 2003] proposes support infrastructure for the distributed mobile applications implementation with behavior adaptive in pervasive computing environment. Both the work [Kurkovsky et al. 2004; Yamin et al. 2003] does not address any of the issues of the participating mobile nodes including the mobility issue.

[Park et al. 2003] discusses on the extension of grid computing systems in mobile computing environments, where mobile devices can be effectively incorporated into the grid either as service recipients or as more valuable service providers. The work tries to identify what would be the newly required services in such a mobile/grid integrated architecture, based on the present grid architecture. The work focuses only on disconnected operation issue and they had come out with job scheduling algorithm for such a restricted mobile grid, which does not address other critical issues.

[Agarwal et al. 2004] has introduced the wireless grid and has come out with a general classification based on the possible architectures. They had come out with classifications based on architecture and usage patterns. However, the work does not address any of the design or architectural issues.

[Bruneo et al. 2003] comes out with an architecture using mobile agent technology, where the mobile agents are used as a communication primitive. The work discusses with the focus on MHs using the grid and not as a member of the grid. Further, this work assumes higher network bandwidth availability which is not prevalent in a mobile environment.

[Hwang and Aravamudham 2004] comes out with a middleware architecture, called scalable inter-grid network adaptation layers (Signal), to integrates mobile devices with existing grid platforms to conduct peer-to-peer operations through proxy-based systems. The architecture also deals mainly with the network and QoS adaptation between wireless devices and the network infrastructure. It is an extention of [Phan et al. 2002] and uses OGSA's extension to Web services technologies. The model does not address critical issues like mobility of the MHs and is more towards MHs which are in wireless adhoc networks.

[Ghosh et al. 2004] proposes a game theoretic framework to implement the pricing model for addressing the load balancing issues in mobile grid. It also details a work load allocation scheme based on the pricing model. But even when a proper mobile grid design does not exist, this work seems to be on the other end with the assumption that a mobile grid already exists. The focus is also more towards WAP based mobile grid.

There are some research focusing on designing and developing the Mobile Grid. However they are in a preliminary form and are yet to come out with results. The K*Grid project [Kpr] are working to design and implement mobile grid platform which is based on PDA and wireless LAN technology. As of now they had only come out with a study on mobile grid technologies, which includes the analysis of wireless mobile networks, devices and technologies and the requirements for mobile grid. The Akogrimo project [Ako] is aiming to architect and prototype a blueprint of a next generation grid based on open grid services architecture (OGSA) [ogs] which exploits and closely co-operates with evolving mobile internet infrastructures based on IPv6. The concept of the project is to evaluate the derived Mobile Grid through testbeds that are chosen based on existing evolving applications from the domain of e-Health, e-Learning and Crisis management.

9. CONCLUSIONS

The paper highlighted the viability of a mobile grid that enables sharing of scarce resources to perform a computationally-intensive task. A mobile grid is seen as a hybrid environment that consists of various types of computing resources, fixed ones such as desktops and mobile ones such as palmtops, PDA, laptops, etc. The backbone of the proposed framework is an object based architecture, where the object represents the participating entities. We have also described the design of the proposed novel computing paradigm using the object based model, which also supports the dynamic composition and management of services. The main motivation for developing the mobile grid system was to assist the mobile users with the power of a super computer at any time and any where.

Service Composition architectures will become very important with the increasing growth of e-services. The proliferation of mobile devices during these years have started seeing the availability of services on different mobile devices. However the conventional approach is that mobile devices will be merely web browsers without significant computational capabilities unto themselves. This paper is proposed with an envision that mobile devices (embedded devices) can be capable to deliver much greater capabilities for grid computing. This has seen to the integration of mobile devices into the grid to form a new paradigm namely the mobile grid. Thus this paper proposes a new service composition middleware architecture for mobile grid. The paper also provides the design issues and principles that have been considered to build the layered architecture.

As a future work the proposed mobile grid can be extended to integrate wireless sensor networks.

REFERENCES

Akogrimo. http://www.akogrimo.org.

K*project. http://gridcenter.or.kr/MobileGrid/index.php.

Open grid services architecture. http://www.globus.org/ogsa/.

Service composition.

Universal description, discovery, and integration (uddi) protocol.

Agarwal, A., Norman, D. O., and Gupta, A. 2004. Wireless grids: Approaches, architectures, and technical challenges. Working Paper, Massachusetts Institute of Technology (MIT), Sloan School of Management.

Anderson, T., Culler, D., Patterson, D., and the NOW Team. 1995. A case for networks of workstations(now). *IEEE Micro 15,* 1 (Feb), 54–64.

Benatallah, B., Dumas, M., Fauvet, M.-C., and Rabhi, F. A. 2003. *Patterns and skeletons for parallel and distributed computing.* Springer-Verlag, London, UK, Chapter Towards patterns of web services composition, 265–296.

Bruneo, D., Scarpa, M., Zaia, A., and Puliafito, A. 2003. Communication Paradigms for Mobile Grid Users. In *Proceedings of the third IEEE/ACM International Symposium on Cluster Computing and the Grid (CC-GRID'03).* IEEE Computer Society.

Carriero, N. and Gelenter, D. 1989. Linda in context. *Communications of the ACM 32,* 4, 444–458.

Clarke, B. P. and Humphrey, M. 2002. Beyond the "device as portal": Meeting the requirements of wireless and mobile devices in the legion grid computing system. In *Proceedings of the International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (WPIM 2002).* Fort Lauderdale, Florida, USA.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. 2002. Unraveling the web services web: An introduction to soap, wsdl, and uddi. *IEEE Internet Computing 6,* 2 (March/April), 86–93.

Czerwinski, S. E., Zhao, B. Y., Hodes, T. D., Joseph, A. D., and Katz, R. H. 1999. An architecture for a secure service discovery service. In *Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99).* Seattle, WA, 24–35.

dos S. Lima, L., Gomes, A. T. A., Ziviani, A., Endler, M., Soares, L. F. G., and Schulze, B. 2005. Peer-to-peer resource discovery in mobile grids. In *Proceedings of the 3rd International Workshop on Middleware for Grid Computing MGC05, France.* 1–6.

Felber, P. and Narasimhan, P. 2004. Experiences, Strategies, and Challenges in Building Fault-Tolerant CORBA Systems. *IEEE Transaction on Computers 53,* 5 (May), 497–511.

Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., MasanobuTakagi, Bing-Wo, R., Amir, A., Murakawa, R., Hernandez, O., Magowan, J., and Bieberstein, N. 2002. *Introduction to Grid Computing with Globus.* IBM Red Book.

Forman, G. H. and Zahorjan, J. 1994. The challenges of mobile computing. *IEEE Computer 27,* 6 (April), 38–47.

Foster, I. and Kesselman, C. 1997. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications 11,* 2.

Foster, I., Kesselman, C., and Tuecke, S. 2001. The anatomy of the grid: Enabling scalable virtual organizations. *International J. Supercomputer Applications 15,* 3.

Garrett, J. H., Burgy, C., Reinhardt, J., and Sunkpho, J. 2002. An overview of the research in mobile/wearable computer-aided engineering systems in the advanced infrastructure systems laboratory at carnegie mellon university. *Bauen mit Computern 2002, Bonn, Germany. VDI Verlag GmbH, Duesseldorf, Germany.*

Ghosh, P., Roy, N., Das, S. K., and Basu, K. 2004. A game theory based pricing strategy for job allocation in mobile grids. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04).*

Gonzlez-Castao, F. J., Vales-Alonso, J., Livny, M., Costa-Montenegro, E., and Anido-Rifn, L. 2002. Condor grid computing from mobile handheld devices. *ACM SIGMOBILE Mobile Computing Communications Review 6,* 2, 18–27.

Grimshaw, A., Wulf, W., and the Legion team. 1997. The legion vision of a worldwide virtual computer. *Communications of the ACM 40,* 1, 39–45.

Group, O. M. 2001. Fault Tolerant CORBA (Final Adopted Specification). formal/01-12-29.

Henning, M. 1998. Binding, migration and scalability in corba. *Communications of the ACM 41,* 10, 62–71.

Hwang, J. and Aravamudham, P. 2004. Middleware Services for P2P Computing in Wireless Grid Network. *IEEE Internet Computing*, 40–46.

Imielinski, T. and Badrinath, B. R. 1994. Wireless mobile computing: Challenges in data management. *Communications of the ACM 37,* 10 (October), 18–28.

Janakiram, D. and Srinivas, A. V. 2002. Object migration in corba. *Journal of the CSI 32,* 1, 18–27.

Janakiram, D., Venkateswarlu, R., Srinivas, A. V., and Kumar, A. U. 2006. Soma: A Compositional Language for Distributed Systems. *To appear in ACM SIGPLAN Notices*.

Jing, J., Helal, A. S., and Elmagarmid, A. 1999. Client-server computing in mobile environments. *ACM Computing Surveys 31,* 2, 117–157.

Joshi, R. K. and Ram, D. J. 1999. Anonymous remote computing: A paradigm for parallel programming on interconnected workstations. *IEEE Transactions on Software Engineering 25,* 1, 75–90.

Kistler, J. J. and Satyanarayanan, M. 1992. Disconnected operation in the coda file system. *ACM Transactions on Computer Systems 10,* 1, 3–25.

Konanki, P. and Butt, A. R. 2007. On supporting disconnected operation in grid computing. In *Poster in IEEE International Conference on High Performance Computing (HiPC 2007), Goa, India*.

Kurkovsky, S., Bhagyavati, Ray, A., and Yang, M. 2004. Modeling a grid-based problem solving environment for mobile devices. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC04)*.

Labovitz, C., Ahuja, A., Bose, A., and Jahanian, F. 2000. Delayed internet routing convergence. In *ACM SIGCOMM*. 175–187.

Labovitz, C., Ahuja, A., and Jahanian, F. 1999. Experimental study of internet stability and backbone failures. In *FTCS '99: Proceedings of the Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing*. 278–285.

Lee, C. and Helal, S. 2002. Protocols for serice discovery in dynamic and mobile networks. *International Journal of Computer Research 11,* 1, 1–12.

Libenzi, D. 2001. CPUHOG - A kernel scheduler latency tester, Free Software Foundation, Inc., Boston, MA, USA.

Litzkow, M., Livny, M., and Mutka, M. W. 1988. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computer Systems*. IEEE, 104–111.

McCanne, S. and Floyd, S. 1995. ns-network simulator.

Migliardi, M., Maheswaran, M., Maniymaran, B., Card, P., and Azzedin, F. 2002. Mobile interfaces to computational, data, and service grid systems. *ACM SIGMOBILE Mobile Computing Communications Review 6,* 4, 71–73.

Mohamed, M. M., Srinivas, A. V., and Janakiram, D. 2005. Moset:an anonymous remote mobile cluster computing paradigm. *To appear in Special Issue on Design and Performance of Networks for Super, Cluster and Grid Computing in the Journal of Parallel and Distributed Computing(JPDC)*.

Natrajan, A., Nguyen-Tuong, A., Humphrey, M. A., and Grimshaw, A. S. 2001. The legion grid portal. *Grid Computing Environments 2001, Special Issue of Concurrency and Computation: Practice and Experience 14,* 13-15, 1365–1394.

Norman, D. A. 1998. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer is so Complex, and Information Appliances are the solution*. MIT Press, Cambridge.

Park, S.-M., Ko, Y.-B., and Kim, J.-H. 2003. Disconnected operation service in mobile grid computing. In *Proceedings of the 1st International Conference on Service Oriented Computing (ICSOC 2003)*. Trento, Italy, 499–513.

Phan, T., Huang, L., and Dulan, C. 2002. Challenge: Integrating mobile wireless devices into the computational grid. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking 2002 (MOBICOM'02)*. Atlanta, Georgia, USA, 271–278.

Pitoura, E. and Samaras, G. 2001. Locating objects in mobile computing. *Knowledge and Data Engineering 13,* 4, 571–592.

Rowstron, A. I. T. and Druschel, P. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*. 329–350.

Royer, E. and Toh, C.-K. 1999. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications 17,* 8 (April), 46–55.

Satyanarayanan, M. 2001. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 10–17.

Smarr, L. and Catlett, C. E. 1992. Metacomputing. *Communications of the ACM 35,* 6 (June), 44–52.

Srinivas, A. V. and Janakiram, D. 2006. Scaling a shared object space to the internet: Case study of virat. *To appear in Journal of Object Technlogy*.

Sterling, T., Savarese, D., Becker, D. J., Dorband, J. E., Ranawake, U. A., and Packer, C. V. 1995. Beowulf: A parallel workstation for scientific computation. In *Proceedings of the 24th International Conference on Parallel Processing*. Oconomowoc, WI, I:11–14.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., and Balakrishnan, H. 2003. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transansactions on Networking 11,* 1, 17–32.

Stumm, M. and Zhou, S. 1990. Algorithms implementing distributed shared memory. *IEEE Computer 23,* 5, 54–64.

Sultan, F., Nguyen, T. D., and Iftode, L. 2002. Lazy garbage collection of recovery state for fault-tolerant distributed shared memory. *IEEE Transactions on Parallel and Distributed Systems 13,* 10, 1085–1098.

Weiser, M. 1993. Hot topics: Ubiquitous computing. *IEEE Computer*.

Yamin, A., Barbosa, J., Augustin, I., Silva, L., Real, R., Geyer, C., and Cavalheiro, G. 2003. Towards merging context-aware, mobile and grid computing. *International Journal of High Performance Computing Applications (jHPCA) 17,* 2, 191–203.

Zeng, X., Bagrodia, R., and Gerla, M. 1998. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Workshop on Parallel and Distributed Simulation*. 154–161.

Z.Li, C.Wang, and R.Xu. 2001. Computation offloading to save energy on handheld devices: A partition scheme. In *Proceedings of International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*. 238–246.

M.A. Maluk Mohamed obtained the Ph.D. degree from the Indian Institute of Technology Madras in 2006, Masters in Engineering from National Institute of Technology, Tiruchirappalli in 1995 and Bachelors in Engineering from the Bharathidasan University in 1993. He is currently a professor in the Department of Computer Science and Engineering, M.A.M. College of Engineering (MAMCE), India. He coordinates research activities for the System Software Group at MAMCE. His research interests include distributed computing, grid computing, distributed mobile systems, wireless sensor networks, software engineering and distributed databases. He is a member of the ACM, IEEE, ISA, IARCS and life member of the Computer Society of India.