CHAPTER 1
INTRODUCTION

1.1 PREFACE

Grid computing is a service for sharing computer resources and data storage over the internet. As resource requirements of recent applications increased greatly, Grid systems have gained importance in the last decade. Complex and large-scale problems in science, engineering and business need more powerful computing machines because these problems are more computing-intensive and data-intensive. Grids are geographically distributed platforms with heterogeneous resources in which users can access via a single interface. Grid provides a common resource-access technology and operational services across widely distributed and dynamic virtual organizations. The nature of Grid infrastructure is to integrate large computational and storage resources, data, services and applications from different disciplines. The ability to manage Grid systems depends on the accuracy and availability of the resources like computational power, storage and networking.

As Grid techniques are growing rapidly in recent years, large-scale Grid systems appear to provide flexible, secure, coordinated resource sharing, and problem solving among dynamic virtual organizations. These systems consequently are required to manage a large amount of related resources. In particular, the shared Grid resources can vary from plain desktop systems to clusters and from storage devices to large datasets, even Grid service could be seen as an extension of Grid resource. Therefore, Grid Resource Discovery (GRD) plays a crucial role in
the whole system [15], and its discovery Models and strategies have a vital influence on the performance of Grid system.

GRID can be classified into attribution matching pattern and semantic matching pattern. In attribution matching pattern, GRID can also be classified into Centralized, Hierarchical, Peer-to-Peer (P2P)-based, and group-clustering models from the system architecture aspect. Among the models above, centralized[42,116] and hierarchical models[22,74,75,76] are easier to design and manage, whereas they prove to be inefficient as the scale of Grid system rapidly increases due to the bottleneck of servers and single point of failure. Therefore, the P2P-based model [50,18,125] which combines the P2P technology into Grid is applied widely to overcome those problems.

Grid and P2P [150] are the two most common types of resource sharing systems currently in wide use. These two systems evolved from different communities and serve different needs. Grid systems interconnect computer clusters, storage systems, instruments, and in general available infrastructure of large scientific computing centers in order to make possible the sharing of existing resources, such as CPU time, storage, equipment, data and software applications. Most Grid systems are of moderate-size; they are centrally or hierarchically administered and there are strict rules governing the availability of the participating resources. Grids are used for complex scientific applications which are time critical and they comply with strict QoS (Quality of Service) rules. Grid resources are highly dynamic and their values vary significantly over time. The resources required by applications are described by specifying a set of attributes (multi-attribute queries) like available computing power and memory. On the other hand, the most popular service provided by P2P systems is file sharing (e.g., Gnutella, KaZaA). Other applications are real time data transfer (e.g., telephony such as Skype), cycle stealing (e.g.,
SETI@Home), or collaboration (e.g., Groove). The typical participant in such systems enters from a common desktop asking to download music or video files over Internet TCP connections. Participation is highly dynamic as users can enter, depart, and rejoin the system totally unpredictable. In P2P system, most resource location queries are not attribute-dependent as in Grids but they either specify a file name or they search keyword or range queries.

### 1.2 GRID COMPUTING

In most organizations, there are large amounts of underutilized computing resources. Most desktop machines are used less than 5 percent of the time. In some organizations, even the server machines can often be relatively idle. Grid computing provides a framework for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage. The processing resources are not the only ones that may be underutilized. Often, machines may have enormous unused disk drive capacity. Grid computing, more specifically, a “data Grid”, can be used to aggregate this unused storage into a much larger virtual data store, possibly configured to achieve improved performance and reliability over that of any single machine.

Locations of resources in Grid environment is the primary necessity which permits the discovery of services across multiple administrative domains. Resource discovery is the process of locating proper resource candidates which are suitable for executing jobs within a reasonable time. The characteristics of the Grid systems make the resource discovery a time consuming process which can decrease the performance of the whole system. Resource discovery in Grid is a challenging issue because characteristics of its resources are heterogeneous, dynamic, various and autonomous.
1.2.1 Grid Architecture

The ideas of the Grid were brought together by Ian Foster, Carl Kesselman, and Steve Tuecke, widely regarded as the "fathers of the Grid". Ian Foster et al., presented [46] the various layers of the Grid architecture by using **Hourglass model**. As shown in the Figure 1.1, the narrow neck of the hourglass defines a small set of core abstractions and protocols, onto which many different high-level behaviors can be mapped, and which themselves can be mapped onto many different underlying technologies. While the base of the model represents the different underlying technologies, the top of model shows high-level behaviors that translate into services and user applications.

![Figure 1.1: Model of Grid Architecture](image)

The **fabric layer** provides the resources to which the shared access is controlled by the Grid protocols. The resources normally include physical and logical entities. Physical entities are
resources like storage systems, catalogs, servers, and network resources. The resource may be a logical entity like distributed file system, computer cluster or distributed computer pool, and database systems to store structured data. The Grid mechanism normally permits the capability for the resource management, which involves discovery and control. The **connectivity layer** defines core communications and authentication protocols required for Grid specific network transactions. These protocols enable the exchange of data between fabric layer resources. The **resource layer**, based on the connectivity and authentication protocols, controls the access resources.

The **collective layer** deals with the directory brokering services, scheduling services, data replication services, and diagnostics/monitoring services. These services are not associated with any one specific resource but focus on interactions across resources. The programming models and tools define and invoke the collective layer functions. This is the layer that glues all the resources together in expedient exchange. The top layer, **User Applications**, comprises the user applications that operate within a virtual organization (VO) environment.

The components in each layer share common characteristics but can build on new capabilities and behaviors provided by the lower layer. This model demonstrates the flexibility with which Grid Architecture can be extended and evolved. This is the precise reason why Grid architecture is taking shape and form based on the guidelines developed by the visionaries and Grid standard forums.
1.2.2 Globus toolkit

Globus is an open source Grid software that addresses the most challenging problems in distributed resources sharing. The Globus Toolkit includes software services and libraries for distributed security, resource management, monitoring and discovery, and data management.

The Globus Toolkit [53] is a software toolkit, developed by The Globus Alliance, which we can use to program Grid-based applications. The toolkit, first and foremost, includes quite a few high-level services that we can use to build Grid applications.

Figure 1.2: Relationship between OGSA, Globus Toolkit 4, WSRF, and Web Services
These services, in fact, meet most of the abstract requirements set forth in Open Grid Service Architecture (OGSA). In other words, the Globus Toolkit includes a resource monitoring and discovery service, a job submission infrastructure, a security infrastructure, and data management services etc., The Relationship between OGSA, Globus Toolkit 4, WSRF, and Web Services is shown in Figure 1.2. Web Services Resource Framework (WSRF) is a specification developed by OASIS. WSRF specifies how we can make our Web Services stateful, along with adding a lot of other cool features. It is important to note that WSRF is a joint effort by the Grid and Web Services communities, so it fits perfectly inside the whole Web Services Architecture.

There are three main components in the Globus Toolkit:

- Resource management
- Information services
- Data management

The technologies used to realize these three components include Grid Resource Allocation Management (GRAM), Monitoring and Discovery Service (MDS), and Grid File Transfer Protocol (GridFTP). All these components utilize the Grid Security Infrastructure (GSI) protocol for security at the connection layer.

The resource management component provides support for resource allocation, Submitting jobs, remotely running executable files and receiving results, Managing job status and progress. Globus Toolkit does not have its own job scheduler to find available resources and automatically sends jobs to suitable machines. Instead, it provides the tools and interfaces needed to implement schedulers and is often used with third-party schedulers.
The information services component provides support for collecting information in the Grid and for querying this information, based on the Lightweight Directory Access Protocol (LDAP). The data management pyramid provides support to transfer files among machines in the Grid and for the management of these transfers.

Globus defines a single, unified access mechanism for this wide range of information, called the Metacomputing Directory Service (MDS) [42]. Building on the data representation and application programming interface defined by the Lightweight Directory Access Protocol (LDAP), MDS defines a framework in which can be represented information of interest in distributed computing applications.

1.2.3 Virtual Organization

In Grid computing, a virtual organization (VO) refers to a dynamic set of individuals or institutions defined around a set of resource-sharing rules and conditions. All these virtual organizations share some commonality among them, including common concerns and requirements, but may vary in size, scope, duration, sociology, and structure.

VO is created to run specific (group of) tasks and may include multiple specially created Grid Services. VO is created on the base of the business agreement between participating organizations and individuals each of which contributes their specific resources (computers, services, people, etc.). The agreement defines all resources and services available to VO members and conditions on which these resources and services are provided and used. VO like real organization may contain all basic services required to run typical organization but these services “physically” and administratively may be run by member organizations on behalf of the VO. The examples of VO are: members of a large international long-term collaboration in high
energy physics; or group of organizations participating in severe weather simulation and prediction; virtual laboratory involving a group of specialists using remotely located unique analytical equipment (e.g., electronic microscope, or mass-spectrometer) for analysis a few samples.

1.2.4 Grid Resource Discovery

Grid Resource Discovery (GRD) can be defined as searching and locating resources across multiple administrative domains. A GRD mechanism returns a list of locations which satisfied the requirements and forms a resource service chain to complete the task. Grid resources are characterized into two types namely static and dynamic. Static attributes refer to resource characteristics that do not change frequently, such as Operating System, CPU speed of a computing resource. Dynamic attributes are associated with fast changing characteristics, such as CPU load and free memory.

In Grids, resource discovery is usually guided by centralized servers. For example, in Globus Toolkit [53], users can get a node’s resource information either by directly querying a server application running on the specific node, or querying dedicated information servers that retrieve and publish resource information of the organization. Techniques of how to associate information servers and construct an efficient, scalable network of directory servers are not specified. To discover resources in more dynamic, large-scale, and distributed Grid environments, P2P techniques have been used (e.g., [16, 62]). Flooding is the predominant search method in unstructured P2P networks. This method, though simple, does not scale well in terms of message overhead. An efficient approach is the use of distributed hash tables (DHTs) [113, 118, 128, 163], which has been shown to be scalable and efficient. However, a missing
feature is the inherent support for expressive queries. Research on the Semantic Web project has recently gained much attention for its knowledge integration vision. Its focus is to exploit the power of semantic technologies to aid in information representation, retrieval and aggregation over the web. Most of the Semantic Web projects use the standard RDF language [28, 97] to describe data.

1.2.5 Types of Grid Resource Discovery

Grid resource discovery process uses different classes of systems like centralized, hierarchical and agent based systems. Even though the first two methods have the advantages of Open Grid Service Architecture (OGSA), they suffer from scalability, reliability and false-positive problems. But agent based systems are attractive in Grid systems because of their autonomy properties. They have capabilities to determine new migration sites according to their migration policies for the distribution of resource discovery queries and so researchers adopted Peer-to-Peer (P2P) technology in Grid environment to solve these problems.

P2P systems are mainly divided into Structured P2P networks, Unstructured P2P networks and Super-Peer systems.

1.2.5.1 Hierarchical Systems

In these approaches, queries are processed hierarchically since servers have been organized hierarchically so that each server is responsible for partitions of resource information. These approaches ensure low search delay due to the use of multi layer architecture. The hierarchical model shows good performance of small and medium sized Grids. The hierarchical model can be efficient when it is used to compute aggregate values, that is, global information
including average load, amount of available memory, location and distribution of hotspots and so on.

In this model, database is maintained in hierarchically located servers. The main idea behind this model is to organize the index servers into a hierarchical topology such as Globus Toolkit4 version which exploits the functionalities of index services. Often, Grid systems are using hierarchical systems. MDS-4 (Monitoring and Discovery Service) follows Globus toolkit which is a good example that maintains hierarchical information services holding the information about each node that belongs to the Grid. The hierarchical resource discovery model is suitable for small and medium scale Grids and it is impractical in a large multi-institutional Grid because of the bottleneck at the tree root, leak of autonomy and load imbalance.

1.2.5.2 Centralized Resource Discovery

The main idea of centralized Grid resource discovery model is that Grid resource information is stored in centralized databases. It can easily manage all the resources. Because of its centralized nature, it supports complex queries such as multi-attribute and range query. When the types and amount of resources become large, all queries and operations have to go through the centralized server. If a single system in a network fails, ultimately the whole system ceases to function. Therefore, the centralized architecture is not suitable for resource discovery in a distributed Grid environment. It also compromises both scalability and reliability in addition to the issues with regard to robustness and fault tolerance. Condor middleware adopts a centralized mechanism named “matchmaker” to match the advertisement with resource requester and resource providers. Centralized resource management mechanism such as Globus MDS is simple, easy to maintain and to be widely adopted in many applications.
1.2.5.3 P2P based Grid Resource Discovery

Peer-to-peer systems and applications are distributed systems without any centralized control or hierarchical organization, where the software running at each node is equivalent in functionality. Design issues of Centralized and Hierarchical approaches are

- Highly prone to a single point of failure
- Lacks scalability
- High network communication cost at links leading to the information server (i.e., network bottleneck, congestion)
- The machine running the information services might lack the required computational power required to serve a large number of resource queries and updates.

Recently, proposals for decentralizing a (Grid Resource Information Service) GRIS have gained significant momentum. The decentralization of GRIS can overcome the issues related to current centralized and hierarchical organizations. Depending on the overlay networks, the P2P based Grid resource discovery model can be classified into unstructured, structured and super-peer-based Grid resource discovery model.

(i) Unstructured P2P Grid Resource Discovery

Unstructured network is formed by randomly joining peers and has no fixed strategy. In this model, resource descriptions are published by peers without a global pattern or structure. The process is to find a suitable resource from the vast network by sending queries to all peers. Flooding method is used and it finds a matched resource, if there exists the corresponding resource. Unstructured P2P Grid resource discovery model originates from the P2P system
Gnutella. Several request forwarding strategies like (time-to-live) TTL-redressed strategy, Random walks strategy and routing indices strategy are proposed to handle the traffic and reduce the response time simultaneously. The TTL indicates the number of hops away from its source a query should propagate. A small TTL value can reduce the network coverage, defined as the percentage of network nodes that receive a query; thus a file may fail to be located although present in the system. While a random walk is usually the less costly alternative in terms of the number of messages sent per query, the flooding approach is more robust and has better response times. Generally, unstructured overlays have loose guarantees for resource discovery, and it is possible that a file is not found although it exists in the network.

(ii) Structured P2P Grid Resource Discovery

In structured systems, resources are assigned to hosts with a well-specified strategy, for example through a Distributed Hash Table (DHT), through which the correct location of a resource on the network is obtained as the result of a hash function.

Structured P2P Grid resource discovery model constructs a rigid overlay network and establishes a strict resource index mechanism. Structured peer-to-peer systems follow DHT based routing algorithms. Chord [128] and Tapestry [163] are examples for DHT based protocols. The overlay network uses virtual topology, which is created and maintained by routing messages between nodes. These systems allow applications to locate resources that are available in the network in a limited number of hops and offer decentralized control and fault tolerance. In DHT (key, value) pairs are stored and mapping from key to value is distributed among the nodes. The distribution makes DHT scalable for a large number of nodes. A node can efficiently retrieve the value associated with a given key by performing a DHT lookup. The structured P2P methods [112] ensure the scalability of the system by involving all the nodes in the query
processing. This ensures that all nodes in the Grid will have the equal load. The structured P2P networks are based on DHT, which uses structured hash algorithm for hashing resources and node ids in the same space. The most well known DHT based protocols are Chord, Tapstry, CAN [113] and Pastry [118]. Among these, Chord algorithm is efficient and easy to design and implement. Due to its simplicity, scalability and high efficiency, Chord lookup protocol has been widely researched and applied in Grid environment especially in resource discovery.

(iii) Super Peer based Grid Resource Discovery Model

The super-peer model has been proposed [18] to build the information system of Grids. This model is naturally appropriate for Grids, as a large-scale Grid can be viewed as a network composed of small-scale, proprietary systems, also referred to as Grid Organizations. Within each Organization, the nodes that have the largest capabilities act as super-peers, while the other nodes use super-peers to access the Grid. Super-peers are interconnected with each other, thus forming a P2P network at a higher layer. Super-peer systems have been proposed to achieve a balance between the inherent efficiency of centralized networks, the autonomy, load balancing and fault-tolerant features offered by P2P networks. In such systems, a "super-peer" node can act as a centralized resource for a limited number of regular nodes, in a fashion similar to a current Grid system, whereas the super-peer overlay network enables distributed computing on a much larger scale. The performance of super-peer networks is evaluated, and rules of thumb are given for an efficient design of such networks. The objective is to enhance the performance of search operations and at the same time to limit bandwidth and processing load.

Super-Peer-based Grid resource discovery model is considered to be more scalable than unstructured P2P model, but super-peers may lead to bottlenecks when the number of requests in
Grid system is large. In this case, this model is confronted with single point of failures. Since super-peers maintain the resource information of its own and other nodes in the same VO (Virtual Organization), it supports range queries and multi-attribute queries easily.

1.3 GRID AND P2P

Nowadays P2P system is the latest research topic in a largely distributed system. Since P2P based network approach may overcome the limitations of hierarchical and centralized methods, P2P techniques are especially used in resource discovery process. The self-organization, scalability and dynamicity are the inspiring features of P2P systems. Through these features, P2P network is achieving High Throughput Computing (HTC). HTC system can be defined as a platform to execute a large number of jobs per unit of time. These jobs are independent, meaning that they can be submitted by different users, and that jobs submitted to the system can be executed in any order.

Grid and P2P are both resource sharing systems and harnessing of resources across multiple administrative domains is their main goal. The two main characteristics of these two systems are dynamic behavior and heterogeneity of the involved components. Even though the goals of Grid and P2P are the same, they use different approaches towards resource discovery. The first is that P2P systems are designed to share files among peers but Grids deal with complicated set of resources. Another difference is that the dynamism of P2P systems comes from both nodes and resources but in Grids, nodes are connected to network in a relatively static manner and the dynamism comes from the fast-changing status of resources. Due to the techniques used in Grid and P2P systems, both are mutually benefited.
The complementary nature of the strengths and weaknesses of the two approaches suggests that the interests of the two communities are likely to grow closer over time [44]. Because of the scalability nature of Grid systems, centralized management will become inefficient and other mechanisms are considered. P2P systems will open up to more sophisticated applications and they will have to support more complex queries and different Qos levels. An era of group based resource-aware communications is opened by P2P systems for many new applications and services. New P2P service scenarios can be envisioned by adding together location information, presence information in the end-to-end communication path and local resources. So the market is driving P2P services based on IP into wireless mobile devices. VoIP, Messaging, file sharing, navigation systems and multi-party video conferencing are a few examples of such services. Nowadays, the research and development community agrees that the adoption of the Peer-to-Peer paradigm could favor Grid scalability. Grids and P2P systems share several features and can profitably be integrated, bringing benefits to both the fields resulting in future convergence. The introduction of the web services resource framework is particularly useful for this convergence trend, since WSRC Grid services can support P2P interactions between hosts belonging to different domains.

1.4 CHORD PROTOCOL

Chord, CAN, Tapstry and Pastry are the original distributed hash table protocols. Chord protocol is proposed by Ion Stocia, Robert Morris, David Karger, Frans Kaashoek and Hari Balakrishnan, and was developed at MIT (Massachusetts Institute of Technology), Cambridge. In Chord ring, each machine acting as a Chord server has a unique 160-bit Chord node identifier, produced by a SHA hash of the node's IP address. Chord is a structured P2P routing protocol.
based on DHT, with the features of fully distributed, scalability, load balancing and stability. A distributed hash table stores key-value pairs by assigning keys to different nodes. A node’s identifier is calculated by hashing the node’s IP address while a key identifier is produced by hashing the key. In the Chord ring, the keys and the node identifier are arranged on an identifier circle of $2^m$ where $m$ is the number of bits in the identifiers. The identifiers on the Chord ring are ranged from 0 to $2^m$-1. Identifiers are ordered in an identifier circle modulo $2^m$.

Each node has a successor and a predecessor. The successor to a node (or key) is denoted by successor $(k)$ and is the next node in the identifier circle in a clock-wise direction. The predecessor is counter-clockwise. In the steady state, in an N-node system, each node maintains information only about $O(\log N)$ other nodes, and resolves all lookups via $O(\log N)$ messages to other nodes. Key $k$ is assigned to the first node whose identifier is equal to or follows in the identifier space. This node is the successor node of key $k$, denoted by successor $(k)$. The Chord protocol supports just one operation: given a key, it will store the key’s value. Queries are transferred across the Ring via successor pointers and half of the circle is searched every time. Each node maintains a routing table which contains addresses of $m$ successor nodes. When a node joins and leaves the system, maximum number of messages needed is $O(\log 2 N)$ in most cases. Fingers are arranged in such a way that the distance to the queried ID can at least be halved with every hop. A node’s $i^{th}$ finger is the first node succeeding the node ID plus $2i$. The $i^{th}$ finger = successor (node ID +2i) Lookups are routed clockwise through the network using finger table. The lookup starts with the closest finger and the process is continued till it gets the result.

One of the most important features of structured P2P protocols is keeping the logical overlay structure. The basic Chord protocol has a simple stabilization mechanism in which each node periodically sends a stabilization message to its successor. If any node leaves the network,
it just has to inform both its successor and predecessor to update its neighbour list. Similarly
node joining also informs the corresponding neighbors and stabilization is processed.

![Finger Table Diagram]

**Figure 1.3: Chord Ring with 2^6 Nodes**

Figure 1.3 shows the Chord Ring with 64 Nodes. N1, N8, N14, N21, N32, N38, N42, N48, N51 and N56 are active nodes. Finger table of node N8 is given along with the Chord ring.

**1.5 LIMITATIONS OF EXISTING METHODS**

Traditionally, the data to be gathered by the resource discovery mechanism is statically
defined. In a highly heterogeneous and dynamic environment such as a Grid, statically defined
searches are usually inappropriate. Peer-to-Peer approach is a suitable technique for Grid
resource discovery because it is a term used to describe the current trend towards utilizing the
full resource available within a large distributed network. Drawbacks of existing Chord Protocol
in resource discovery processes are inefficient information in node's finger table and variation
between logical address and physical address.
1.6 PROPOSED METHODS

The two main drawbacks of existing Chord Protocol in resource discovery process are listed below:

1. Information given in node's finger table is not enough to locate the resources quickly.

2. Variation between logical address and physical address, i.e., node's logical ID is independent of its physical location bringing tremendous delay to network routing.

In this method, we concentrated on these two problems and the entire Resource Discovery process is depicted in Figure 1.4. As given in the Figure 1.4, DHT based Structured P2P Chord protocol is modified in applying four different systems to overcome the drawbacks of Chord protocol. The researcher developed the four different systems for quick Grid Resource Discovery using P2P Chord Protocol which resulted in efficient resource discovery. The results of the new implementations are tested in NSC-SE simulator and performance analysis is given at the end of the corresponding chapters.

The proposed algorithms are listed below:

(i) RVN-Chord (Recently Visited Node history based Chord)

(ii) FS-Chord (Feature Selection based Chord)

(iii) FZ-Chord (Fuzzy Classification based Chord)

(iv) Geo-Chord (Geographical location based Chord)
Figure 1.4: Work flow of the Proposed Grid Resource Discovery Process
i. RVN-Chord (Recently Visited Node history based Chord)

In RVN-Chord [108], Recently Visited Node’s ID (i.e., IP address) is included in the first column of the original Chord’s finger table. Every new lookup uses this ID to find the successor of the key and the search is minimal if the key matches with RVN.ID. In RVN-Chord, the researcher applied SHA1 hash function to map key and node id as in the original Chord. The modified finger table stores the Recently Visited Node id which reserved the key of the previous lookup. This id will be stored in all the nodes of the Chord ring. This is identified by the variable RVN.id which is the first column entry of any finger table. This RVN.id is similar to the use of Recent Document list in our computer file system. By using the recent profile, the next search is made easier and we find the document’s location quickly if the search matches with the recent document. Compared to the base Chord procedure, this type of lookup will reduce the number of hops, messages and communication time.

ii. FS-Chord (Feature Selection based Chord)

The FS-Chord [102] algorithm is based on the application of feature selection technique in Chord protocol for the improvement of resource discovery process. The aim of this method is to consider the valuable features of resources of each node, based on the individual resource’s strength. To extract meaningful features of nodes, frequency of queries posted on a particular resource is calculated. Based on the strength and resource count of each node, nodes are taken for comparison. A Priori algorithm is applied and dominant resource sets are selected among the nodes. This analysis is made on all the resources of nodes in the Chord ring. Nodes with greater number of weighted resources and nodes with unique resources are finally selected for searching.
resources for current discovery process. The use of Resource management table will reduce the searching time of resources.

iii. FZ-Chord (Fuzzy Classification based Chord)

In Chord algorithm [128], Single Chord ring forms logical overlay network which takes maximum of $m$ hops to locate the resource. Fuzzy classification is applied in this work, which classifies the elements into a fuzzy set and its membership function is defined by the truth value of a fuzzy propositional function. The Chord is constructed initially with $2^m$ nodes. The ring is divided into three rings according to the basic Fuzzy-rule. To reduce the number of hops and communication time, FZ-Chord algorithm has been proposed which takes less number of hops and messages[106] [107].

iv. Geo-Chord (Geographical location based Chord)

Due to the use of logical overlay network, Chord suffers from high routing latency and low efficiency in data lookup. Geo-Chord [103] is based on geographical location of nodes which form the subrings of regions. Based on the Euclidian distance among nodes, neighbors are identified and many Region-Rings are formed. After grouping the nodes according to their physical location, sub-rings are formed based on Euclidean distance among nodes. When a node joins the overlay, Geo-Chord introduces other nodes within the same region by binding it to the Region-ring. Due to its lack of physical network topology information, base Chord suffers from low routing efficiency, and ineffective use of bandwidth in data lookup. But Geo-chord alleviates this problem and reduces the required number of hops, messages and communication time.
1.7 FRAMEWORK OF THE THESIS

The thesis is organized into nine chapters as explained below. The first chapter is preliminary in nature and all the proposed algorithms are dealt in it.

Chapter 2: Literature Review

This chapter presents the review of literature relevant to Grid Resource Discovery (GRD) techniques. Different types of GRD mechanisms like Centralized, Hierarchical, Agent based systems and Peer-to-Peer systems have been discussed in this chapter. Structured and unstructured P2P resource discovery mechanisms are also discussed here to clearly identify the correct discovery procedure which will be suitable for Grid Computing environment. This chapter also reviewed various Chord algorithms related to recent history of nodes, feature selection, Fuzzy logic and geographical location based resource discovery procedures. This chapter also reviews various real-life applications of Grid Computing. The content of this chapter has been submitted [104] for publication.

Chapter 3: Chord Protocol in Grid Resource Discovery

This chapter mainly includes the study of Chord P2P protocol and its variations. The Chord finger table with redundant values is also discussed. A comparative study of various chord algorithms is made and the content of this chapter is published in [105].

Chapter 4: Grid Resource Discovery based on Recently Visited Node

The performance of P2P Chord like algorithm can be analyzed in terms of three metrics: the size of the finger table of every node, the number of hops a request needs to travel in the worst case, and the average number of hops. AB-Chord [157], MF-Chord [162], BNN-chord [39]
algorithms have been studied in this chapter. The proposed algorithm RVN-Chord is presented in this chapter which needs less number of hops for resource discovery. A comparative study is also made based on finger table modification. The content of this chapter has been selected for publication in [100] [108].

**Chapter 5: Grid Resource Discovery through Feature Selection (FS-Chord)**

FS-Chord [102] algorithm is proposed in this chapter which uses the feature selection technique to select the nodes with more number of dominant resources. Based on this selection, Chord ring is restructured which has only selected node among $2^m$ nodes. Every new lookup searches for the resources only in the dominant nodes and locate the keys quickly. TB-Chord and A-Chord algorithms are also studied to discuss the discovery of resources and eliminating redundant information in Chord Finger table. One part of the contents of this chapter has been published in [101] and the rest of the content is submitted to [102] for publication and is currently under review.

**Chapter 6: Grid Resource Discovery using Fuzzy logic**

The required theoretical concepts of Fuzzy-logic are studied in this chapter [37, 87]. In FS-Chord [102] the average memory consumed by each lookup process is more. To overcome the above inconvenience, FZ-Chord is proposed which uses Fuzzy classification which classifies elements into a fuzzy set and its membership function is defined by the truth value of a fuzzy propositional function. The proposed algorithm has been compared with Chord and RVN-Chord [108]. The contents of this chapter has been published in [106, 107].
Chapter 7: Geographical location based resource discovery in Grid Computing

Scalability, load balancing, good performance, simplicity are the main features of the Chord P2P protocol. Despite these favourable characteristics, the Chord protocol has the disadvantages such as the nodes in the network do not consider physical location of nodes that are in their geographical range. Geo-Chord is proposed which segments the Chord Ring into multiple Region-Rings based on their physical location. Sub-rings are formed based on Euclidean distance among nodes. The content of this chapter has been submitted to [103] for publication.

Chapter 8: Applications of Grid Computing in Video Rendering

A brief introduction to video rendering is provided in this chapter and its related techniques are studied. A comparative analysis is also made in this chapter. In this chapter, the application of Grid Computing in video rendering is discussed with suitable demonstrations.

Chapter 9: Conclusion

The thesis concludes with this chapter, with a summary of the key findings from the research work carried out and contributions of this research are set. More promising directions for further research work are also hinted at the end of the thesis.