2.1 INTRODUCTION

Grid computing has been widely concerned by many researchers in recent years. A considerable amount of research has been done on matching the user’s requirements with the resource capabilities and several matchmaking algorithms are proposed in the literature. This chapter deals with different proposed matchmaking approaches. The performance analysis of different matchmaking approaches has been studied. Various components used in different matchmaking frameworks have been analyzed. The limitations in various approaches have been analyzed.

2.2 CATEGORIZATION OF DIFFERENT MATCHMAKING APPROACHES

In this chapter, matchmaking is categorized into different approaches based on its distinguished features and is described in table 2.1.
<table>
<thead>
<tr>
<th>Matchmaking approaches</th>
<th>Prominent features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class ads matchmaking</td>
<td>Matchmaking is based on Class ads (advertisements)</td>
</tr>
<tr>
<td>Semantic matchmaking</td>
<td>Matchmaking is based on the semantic distance of concepts (metadata)</td>
</tr>
<tr>
<td>Ontology based matchmaking</td>
<td>Matchmaking is based on ontology (service, function, policy and application) specification.</td>
</tr>
<tr>
<td>Online resource matchmaking</td>
<td>Matchmaking is based on dynamic capabilities</td>
</tr>
<tr>
<td>Decentralized matchmaking</td>
<td>Matchmaking mechanism has matching engines at each grid site.</td>
</tr>
<tr>
<td>Security based matchmaking</td>
<td>Matchmaking mechanism focuses on security issues</td>
</tr>
<tr>
<td>Multi agent based matchmaking</td>
<td>Matchmaking is based on agents</td>
</tr>
<tr>
<td>Knowledge based matchmaking</td>
<td>Matchmaking is based on past experiences</td>
</tr>
<tr>
<td>Reservation based matchmaking</td>
<td>Matchmaking is based on advance reservation of resources</td>
</tr>
<tr>
<td>Optimization based matchmaking</td>
<td>Matchmaking is based on optimization techniques</td>
</tr>
</tbody>
</table>
2.3 CLASS ADS MATCHMAKING

2.3.1 Condor matchmaking

Condor simplified job submission by acting as a matchmaker of Class ads. Condor's Class ads were analogous to the classified advertising section of the newspaper. Sellers advertise specifics about the product they sell, hoping to attract buyers. Buyers advertise specifics about the product they wish to purchase. Both buyers and sellers list constraints that need to be satisfied. In Condor, users submitting jobs are buyers of compute resources and machine owners are sellers.

All machines in a Condor pool advertise their attributes viz. RAM memory, CPU type and speed, virtual memory size, current load average along with other static and dynamic properties. They also advertise the conditions at which they are willing to run a Condor job and the type of job they prefer. These policy attributes reflect the individual terms and preferences by which all the different owners have graciously allowed their machine to be part of the Condor pool. Machines can also be advertised that it is the only machine willing to run jobs at night and when there is no keyboard activity on the machine. When submitting a job, a job with requirements and preferences are advertised. The Class ads includes the type of machine it wishes to use. The job attributes and requirements are bundled up into a job Class ads. Condor plays the role of a matchmaker by continuously reading the entire job Class ads and the entire machine Class ads, by matching and ranking job ads with machine ads. Condor makes certain that all requirements in both Class ads are satisfied.
In condor matchmaking mechanism (Emir Imamagic 2009), users described their requirements in class ads. Class ads described attributes of request viz. storage space and library files required. Resources also described class ads describing their attributes. Condor provides numerous advanced functionalities such as job arrays, workflows support, check pointing, job migration, rescheduling and fault recovery. It enabled users, to define resource requirements, rank resources and provides mechanism for transferring files. It enabled job scheduling based on complex job requirements. A user can easily define specific requirements, such as storage space requirement and libraries requirement. In addition, a user can prefer specific set of resources to others, by giving a higher rank. This system lacks support for parallel jobs, lacks integration with GIS, and ranks were not normalized.

2.3.2 Data aware scheduler (Stork)

Data-aware scheduler-Stork (Tevfik Kosar 2009) used the Class ad job description language to represent the data placement jobs. The Class ad language provides a very flexible and extensible data model that can be used to represent arbitrary services and constraints. This flexibility allows Stork to specify job level policies as well as global ones. Global policies that are applied to all jobs are scheduled by the same Stork server. Users can override them by specifying job level policies in job description Class ads. Stork can interact with higher level planners and workflow managers. This allows the users to schedule both CPU resources and storage resources together. The author introduced a new work-flow language capturing the data placement jobs in the work-flow as well. The enhancements made to the workflow manager allowed it to differentiate between the computational jobs and data placement jobs. The workflow manager submits computational jobs to a computational job scheduler, such as Condor or Condor-G, and the data placement jobs to Stork.
Stork also acts like an I/O control system (IOCS) between the user applications and data storage servers. It provides complete modularity and extendibility. The users can add support for their favorite storage system, data transport protocol or middleware very easily. The author proposed four different techniques to determine the order to execute the data transfer request in the queue without over committing the available storage space at the destination. They are first fit, largest fit, smallest fit and best fit. This was a very crucial feature in a system designed to work in a heterogeneous distributed environment. The author proves that concurrency and parallelism have completely opposite impacts on CPU utilization at the server side. The performance of the system is analyzed by calculating the number of jobs completed by the scheduler with the increase in time. It has been found that the Stork completes more number of jobs with increase in response time. The increase in response time is due to the interactions between the computation scheduler and the data scheduler.

2.3.3 Tree based matching

In tree based matchmaking framework (Md Rafiquil Islam 2007), the matchmaking framework was decomposed into four components, the class ad specification, the language for expressing characteristics and constraints, the semantics for evaluating the attributes of the machines, the semantics for evaluating the attributes of the jobs. The matchmaking algorithm is used by the matchmaker to create matches. The matchmaking algorithm relates the contents of the submitted Class ads and the state of the system to the matches that will be created. The matchmaking protocol defines the process through which the matched entities are notified and the information are given to them in case of a match. The claiming protocol defines the actions the matched entities take to enable discharge of service. The matchmaker creates a dynamically allocated tree, based on the attributes described in machine Class ads. The tree contains a
dummy root containing a left sub tree of busy machines and a right sub tree of idle machines.

Tree based matching reduces the searching complexity. Nodes are allocated dynamically in the trees based on the load characteristics of the nodes. The performance would be improved if the system considers the dynamic capabilities of the resource as a criterion in placing the nodes in the tree.

2.3.4 Set matching

A general-purpose resource selection framework (Chuang Liu 2002) defined a resource selection service for locating Grid resources that match user requirements. It proposed a declarative language based on a technique called set matching, which extends the Condor matchmaking framework to support both single resource and multiple resource selection. The set-matching algorithm evaluates a set-extended Class ads against a set of Class ads and returns a Class ads set that has the highest rank. It comprises of two phases. In the filtering phase, individual Classads are removed from consideration based on individual expressions in the request. In the set construction phase, the algorithm seeks to identify a Class ads set that best meets application requirements. Set-matching technique accepts user resource requests and finds a set of resources with highest rank based on resource information from a Grid Information Service. An open interface allows users to customize the resource selector by specifying an application-specific mapping module. The author performed experiments with diverse configurations many number of times and calculated the execution time. They analyzed the deviation of predicted times from the execution time during process.
As the resource pool is highly dynamic, the number of Class ads in the resource pool is dynamic. Hence the execution time of filtering phase which filters the Class ads in the resource pool is also dynamic. Thus the performance of the system is degraded with the increase of the resource in the resource pool.

2.4 SEMANTIC BASED MATCHMAKING

2.4.1 Matchmaking based on semantic distances

The matchmaking based on semantic distances (Gao Shu 2007) effectively computes the semantic distance of concepts in ontology. It was based on description, logic formalization and reasoning and allowed match ranking. The author implemented the developed prototype in the context of service discovery in the visualization domain. Their matchmaking algorithm aimed at precisely computing the semantic distance of concepts in the ontology and expressed it using a score in order to improve the quality of matchmaking. In the process of matchmaking, there is often a conflict between quality and efficiency. This algorithm aimed to get a good quality of match, by precisely calculating the semantic distance between concepts and sacrifices the efficiency. The efficiency was achieved by pre-computing the semantic distance between the advertisements and the possible requests to speed up matchmaking (Foster I 2005).

2.4.2 Two step matching algorithms

The modified semantic matchmaking policy (Neela Narayanan V 2007) matched the request with the available resources using semantic technology and reduces the average turnaround time of jobs based on user preferences. Here resource broker architecture consists of three modules: matchmaking, request handling and resource discovery. Resource Specification
Language (RSL) is used to express the resource requirement. User submits job to the resource broker with assigning weights to the individual components of the resource. The weights of the individual components of the resource can take value from 0 to 1 and the total weight of all the components of the request should be equal to 1. A two step match making algorithm is used to produce a detailed ranking. In the first step the matches are classified into four different levels: Exact, Plug-In, Subsume, and Fail. In second step, Ranking Compare function is applied to produce more detailed ranking. After analysis the author concluded that the time taken for loading ontology depends on the number of resources. The author also measured the time taken to Query and retrieve from Ontology Graph. It was found that the query time increases with the increase in the resources. This in turn increases the average response time.

The service registration process (Simone A 2006) is a process in which the service providers need to register their services for the matchmaking process. The service provider registers its service semantics in the Grid service ontology and the necessary contact details in the service registry. Service semantics comprises of a service name, a service description, service attributes (input/output) and metadata information. The service requester specifies the context semantics of the application in the application ontology. The Grid application sends out a request to the service discovery matchmaker. The request goes through the context matching module first. Additional parameters are attached to the request and forwarded to the semantic matching module. In this module the semantic match is performed. Semantic matchmaking allows a service request to be matched using the semantics (metadata) of services. Having all necessary semantic data, a service lookup is done using a service registry. This lookup information is sent back to the Grid application to be used for Grid service call. Introduction of service-oriented architectures gives users access to a large pool of resources spanning multiple security domains. The author loaded
nine different ontologies on the web and measured the performance of the system with increasing complexity of ontology. The rule complexity measurements were conducted by increasing the complexity of the rules, queries and facts and the response times were measured. From the results, the author concluded that adding complexity to the rules, queries and facts increases the precision and quality of matching. The speed of the matchmaking is not considered.

A two-level methodology (Clematis A 2010) aimed to facilitate the matchmaking process based on two integrable approaches: (1) the use of micro-benchmarks, to supply a "zero-degree" of resource description, (2) the deployment of application-driven benchmarks to get a closer insight into the behavior of resources for a class of applications under more realistic conditions. The metric chosen to characterize resources was the execution time evaluated on a reference data set and specific parameters. This second level of benchmarks profits a closer knowledge of the characteristics of the application at hand. Depending on the application domain, a job could be more computing than data intensive, and therefore it benefits from faster CPU rather than from greater memory capacity or more advanced I/O devices. Through application-driven benchmarks it was possible to assign an evaluation to resources on the basis of the indicators that were more stressed by the application. The decision of supplying a double level of benchmarks was a response to users with different degrees of confidence on the behavior of their applications.

According to this methodology, every resource was tagged with the results obtained through the two levels of benchmarks. This evaluation process played a significant role, as during the matchmaking phase, resources were selectable on performance basis, allowing a better fit of the job needs. All these issues needed effective management of resources. The author validated the
model using three metrics viz. average execution time, maximum number of waiting jobs and more loaded resource. The two level methodology finds its limitation for the user with scarce knowledge of the job.

2.4.3 Meta Data semantic matching

The meta data semantic matching (Zhen Liu 2005) combined exact matching of core layer metadata and fuzzy matching of extensional layer metadata. It took into account of semantic matching and resource instance feature matching synthetically. The resource agent is used to manage local resources. The metadata is described in two layers: Core Layer and Extensional Layer. Core layer metadata includes the resource description. It describes the global feature of resources. Extensional layer metadata is the extent of core layer metadata. It provides task-oriented or domain-oriented semantic content description of resources. Matchmaking process was improved by providing effective request routing strategy in the network. As local resources are managed by local agents, fairness in allocation is lost.

2.5 ONTOLOGY BASED MATCHING

Ontology is defined as a set-of-concept-definitions or specification of a conceptualization. Ontology based matching (Azzurra Ragone 2009) used separate ontology for request and service. Ontology translator translated various formats into one. Matchmaking was done by using the terms defined in ontologies. It was based on static information. In this system, users should express their preferences manually by calculating the ranking function and there were no application level descriptions in request ontology.
The semantic discovery process (Shahid Mahmood 2008) contained a semantic knowledge base, in the form of domain, functional and policy ontologies of resources description. Three types of ontologies were formed from the resource description which was saved into the knowledge base. They were domain, functional and policy ontologies. Domain ontology comprises the concepts of different resources. It contains information of two different types of resources which are software and hardware. Functional ontology consists of functional information of Grid services. Usage policies have the information about the list of users allowed to use the resources, the time these resources will be available, the conditions and the functions under which the users can perform with them. The author proved that the system (Semantic with Usage Policies) has more relevant resources since it not only uses semantics but also enforces resource usage policies during matchmaking process. As precomputation of semantic distance between ontologies were not considered, the speed of the matching would be low.

2.6 DYNAMIC RESOURCE MATCHING

2.6.1 Online resource matching

Online resource matching for heterogeneous grid environments (Vijay 2005) took into account the dynamically changing conditions on resources and request. Resource state repository (RSR) stored the real time status of all resources. Each job required resources from various grid sites. It used integer programming to solve the problem of heterogeneity. Each job had one dependency on each resource type. The job requirements consist of a set of dependencies and constraints. Each requirement was a dependency on a resource instance of a particular resource type. By associating constraints with each dependency, a job may specify the minimum criteria for selecting a resource instance of a particular resource type. Here jobs restrict one dependency per
resource type, but they do not have any restrictions on the number of different resource types. A job may depend upon the number of constraints, resource type as long as the constraints are independent. In addition to the hard constraints for selecting resources, a job may also have resource preferences among the qualifying resource instances. Thus this system lacks fairness and preemption.

In Interoperating grids through delegated matchmaking (Alexandru Iosup 2007), delegation chain has been formed. It focused on balancing loads at grid sites. The author analyzed the performance of the system by varying the number of jobs submitted at a time in a batch and measured the time between the submission of a batch of jobs and the return of matched results i.e., the response time. It has been found that the speed of the matching is degraded when more delegation chain is formed.

2.6.2 GRADSolve RPC system

In GrADSolve RPC system (Vadhiyar S 2003), the resources used for the execution of parallel application were chosen dynamically based on the load characteristics of the machines. The system significantly reduced the overhead associated with data movement in RPC systems. It worked based on the changing load characteristics of resources. The migration framework implemented by this system is aimed at performance-oriented Grid systems and implements tightly coupled policies for both suspension and migration of executing applications. The suspension and migration policies considered both load changes on systems as well as the remaining execution times of applications, thereby taking into account both system load and application characteristics. This improves the response times for individual applications.
The resource selection is based only on the load characteristics of the machine and the remaining execution times of applications. Hence the job’s successful completion till the end is not assured.

2.7 DECENTRALIZED MATCHMAKING

2.7.1 Matchmaking engine overlay (MEO)

In Grid resource discovery approach based on matchmaking engine overlay (Yang Liu 2007), matchmaking engine was running on the super node of every grid site. Matching was done to evaluate the similarity of two entities. Every entity has its particular characteristics, and every characteristic can be denoted as vector \((name, value)\), in which name is a string to identify the characteristic, and value is a constant or an expression with return value. By forming matchmaking engines into an unstructured P2P overlay network, grid users or applications can discover resources by sending requests to corresponding matchmaking engine, thus matched resources can be found within the same organization or cross organizations. Searching suitable matchmaking engine adds complexity to the process. The author proved that quality performance increases with the increase of number of matchmaking engines and efficiency decreases with the increase of number of matchmaking engines.

2.7.2 Market oriented grid

In market oriented Grid (Rajkumar Buyya 2001), there are markets, resource consumers, resource providers and their interactions. The author proposed a decentralized computation market with multiple markets and numerous consumers and providers spread across the grid environment. Consumers and providers announce their desire to buy or sell compute power from the market. As part of expressing their desire to contribute to the market or
to benefit from the market, they register with the market. When resource providers register with the market, they obtain a "Market Resource Agent" from the market and deploy their resource. Consumers obtain a "Market Resource Broker" from the market. These agents help synchronize and maintain the flow of interaction among the three entities. The intent of the Market Resource Agent is to update the Market with the latest information about the resource provider and to accept, deploy and launch the job. The intent of the Market Resource Broker is to help the consumer to find an appropriate provider based on the information provided by the Market. The resource information provided by Market agents is maintained in the database for providing Market Information Services (MIS).

The author proved that the market resource agent helped in increasing the number of successful jobs. However the popular resources become bottleneck for the system.

2.7.3 Tree based matching

Rendezvous node tree and CAN-based resource matching (Jik-Soo Kim 2006) introduced randomness into the system by mapping names to nodes through hash functions. A job is inserted into the system by using its constraints as coordinates, and defining the owner of the resulting zone as the owner of the job. The candidate nodes are drawn from the owners of neighboring zones, such that each candidate is at least as capable as the original owner in all dimensions. The mapping process uses only the hashing functions to match the job with resources which will increase the time. The performance metrics evaluated are matchmaking cost (the number of messages required for finding candidate run nodes by the owner node of a job), wait time (the amount of time between the injection of the job and the time at which the job starts running), and average queue length, which is the length of the non-preemptive run queue seen by a job.
when it is finally assigned to a run node. The candidate node tree has to be formed using hash functions for each run. This increases the response time.

### 2.7.4 Bid based matching

The XMAP framework (Carmela Comito 2009) was a decentralized network of semantically related schemas that enabled the formulation of queries over heterogeneous, distributed data sources. It considered two types of queries viz. reformulated queries and concurrent queries. XMAP abstracts from the underlying network infrastructure. It was modeled as a number of autonomous nodes holding information, which are linked to other nodes by mappings. XMAP is composed of a collection of 'N' nodes, which are logically bound to XML data sources. Data sources employ the XML data model and each source defines its own XML schema. Each node also holds a collection of mappings from its local schema to other foreign schemas. Each node knows a list of other nodes (called neighbors). These nodes are connected to each other through declarative mappings rules. Thanks to such mappings, it is possible to translate queries from a source to another. The author experimented and concluded that increasing the number of query increases the response times.

A set of resource selection heuristics was proposed (Chien-Min Wanga 2010) to minimize the turnaround time in a non-reserved bidding based grid environment, while considering the level of information about competing jobs revealed by providers. To avoid single point of failure, bidding provided alternative means of resource allocation in distributed systems, by depicting the abstract process of the bidding model. A resource requester started a bidding process by sending a set of call-for-proposal (CFP) requests, which contain job requirements, to resource providers. Based on their resource utilization and policies, the providers decide whether or not to participate in the bidding process.
If they join the bidding process, they return bids that describe the states of their resources to the requester. Finally, the requester evaluated and ranked the collected bids based on its selection strategy and submitted the job to the provider that proposed the best-ranked bid. The advantages of this resource selection heuristics are: it dealt with resource selection issue in an online non-reserved bidding-based grid system that focused on minimizing the turnaround time, it addressed the issue of probabilistic and deterministic resource selection heuristics, it considered various levels of information about competing jobs, it examined the impact of cooperative requesters and non-cooperative requesters on the performance of grid resource selection. The author also analyzed the effect of increasing the number of requestors with the average turnaround time which negatively affects the average turnaround time.

The bidding model has advantages over the condor matchmaking model like scalability i.e., resource allocations between providers and requesters in the bidding model were fully distributed without the intervention of a centralized matchmaker/broker.

### 2.7.5 Fair share scheduling

The Fair share system (Elmroth E 2005) preserved local site autonomy and enforced locally and globally scoped share policies, allowing local resource capacity as well as global grid capacity to be logically divided across different groups of users. This policy model is hierarchical and sub policy definition is delegated. There is no need for a central coordinator as policies are enforced collectively by the resource schedulers. Share enforcement is addressed by the algorithm that calculates simple priority values, thus simplifying integration with local schedulers, which can remain unaware of the hierarchical share policy structure.
The author concluded that the Fair share scheduling improves load balancing. As the local schedulers are not aware of the share policy structure, wrong allocations might be possible.

2.7.6 Backfilling

Backfilling (Daniel C 2009) was an effective strategy for scheduling parallel tasks efficiently. In systems of parallel tasks, the job at the head of the queue is often delayed because of a lack of available resources. However, there are often less demanding jobs in the queue which could be executed earlier without delaying the job at the head. If the resources for the job at the head of the queue are expected to become available at time ‘t’, then the backfilling strategy allows a job with expected length ‘l’, where l < t, to be executed immediately, rather than waiting, assuming its required resources are available. This effectively fills the idle bubbles in the schedule and results in a shorter overall completion time. But the queue has both local and remote jobs together and it did not discriminate the two. The authors also concluded that as the task length increases the number of task completed decreases.

New scheduling algorithms and selection strategies for the resource management problems (Volker Hamscher 2000) combined different jobs and machine models. They recommended FCFS for central pool jobs. But FCFS is not efficient since it increases the waiting time. The distributed scheduling algorithms (Vijay Subramani 2002) used multiple simultaneous requests at different sites and considered the response times of jobs in evaluating the performance of grid scheduling strategies. Trace-based simulations show that the use of multiple simultaneous requests provides significant performance benefits. This scheme can also be adapted to provide priority to local jobs, without much loss of performance. But this affects remote jobs.
2.8 SECURITY SERVICES IN MATCHMAKING

The security services in matchmaking (M. Altunay 2005) checked trust policies of both grid users and resources without requiring policy disclosures. It mutually evaluated evidence-based authorization policies. Trust-based service selection occurs after the discovery and matchmaking stages. The trust-worthiest service among the discovered services was selected. This system integrated mutual authentication and authorization checks into the matchmaking process. The author analyzed the system by increasing the number of registered services and measured the average execution time.

In Mutual Trust Evaluation (Altunay M 2005), service-oriented grids have set the stage for a paradigm shift in classical Access Control Systems (ACS). The security design of grid systems protects resources from malicious users. Since users can access a limited set of designated resources, protecting users from malicious resources is not a consideration. This approach is viable given that users implicitly trust the organization with which the grid and its resources are associated. Moreover, users can only gain access to the resources within the organization.

2.9 MULTI-AGENTS IN MATCHMAKING

The intelligent agent-based framework (Francisco García-Sanchez 2009) combined intelligent agents and semantic web services. In traditional system, the users and resource providers are bound to a single broker. To overcome the limitations of a single broker system, a scalable and fault-tolerant topology of specialized brokers was proposed in which the users and resource providers were not bound to any particular broker. They were free to choose the broker according to their choice. The topology was designed in such a way that
each broker is specialized in one type of resource. Agents can be useful as stand-alone entities that are delegated particular tasks on behalf of a user. However, in majority of cases, agents exist in environments that contain other agents, constituting multi-agent systems (MAS). A MAS can be seen as a system consisting of a group of agents who can potentially interact with each other. MAS presented several advantages over isolated agents, such as reliability and robustness, modularity and scalability, adaptivity, concurrency, parallelism and dynamism.

The author discussed that when a group of individual agents form a MAS, the presence of a mechanism to coordinate such a group as well as a communication language becomes necessary. In particular, agents can either cooperate (if they have the same global objective) or compete (when they have different yet conflictive objectives). Along with this, the system distinguishes between distributed problem solving systems (constituted by agents explicitly designed to cooperatively achieve a given goal) and open systems (agents developed by different people to achieve possibly different objectives). For cooperative agents, the most usual cooperation mechanisms are: organizational structures, (centralized and distributed) multi-agent planning and contract nets. Competitive agents need a negotiation mechanism. Coalition formation, market mechanism, bargaining theory, vote, auctions and task assignment between two agents are different negotiation mechanisms.

Agent Communication Languages have been developed so that agents can carry out negotiation processes. The author concluded that the cooperative agents improve the performance than the competitive agents.
2.10 KNOWLEDGE BASED MATCHMAKING

The knowledge based matching (Xin Bai 2005) provided a matching service for an intelligent grid. Here each request was characterized by a set of properties. Each property was a tuple (name, value). Matching service used matching algorithm to evaluate a matching function which computed matching degree. The matchmaking framework included a resource specification component, a request specification component, and matchmaking algorithms. A request specification included a matchmaking function and possibly two additional constraints, a cardinality threshold and a matching degree threshold. The cardinality threshold specifies how many resources are expected to be returned by the matchmaking service. The matching degree threshold specifies the least matching degree of one of the resources returned by the service. The matchmaking service executes a matchmaking algorithm for each request sent by the requester. The input of the algorithm was the request and the grid resource instances are stored in the knowledge base of the matchmaking service. The matchmaking algorithm evaluated the request function in the context of each resource instance in the knowledge base. The output of the algorithm was a number of grid resources ranked according to their matching degrees. ‘n’ denoted the cardinality threshold specified by the request. The matchmaking algorithm returned the grid resources that have the n largest matching degrees to the requester.

The author conducted performance studies regarding the variation in the response time with the size of the knowledge base. He concluded that the knowledge base access time is the major contributor to the response time. He also indicated that the complexity of the result has little effect on the efficiency of the algorithm.
2.11 RESERVATION BASED MATCHING

The reservation based matching (Seymour K 2005) introduced a new, 3-layered negotiation protocol for advance reservation of the Grid resources. It models resource allocation as an on-line strip packing problem and introduces a new mechanism that optimizes resource utilization and QoS constraints, while generating the contention-free solutions. This mechanism supports open reservations to deal with the dynamic Grid and provides a practical solution for agreement enforcement. The algorithms for supporting advanced reservation of resources (Smith W 2000) in super-computing scheduling systems allow users to request resources from scheduling systems at specific times.

The Globus Architecture for Reservation and Allocation (GARA) (Foster I 1999) treated both reservations and computational elements such as processes, network flows, and memory blocks as first class entities, allowing them to be created, monitored, and managed independently and uniformly. It simplified management of heterogeneous resource types by defining uniform mechanisms for computers, networks, disk, memory, and other resources. Layering on these standard mechanisms, GARA enabled the construction of application-level co-reservation and co-allocation libraries. GARA implementation supports three different resource types— parallel computers, individual CPUs under control of the Dynamic Soft Real-Time scheduler, and Integrated Services networks. These provide performance results that quantify the costs of these techniques. The author concluded that the wait times of applications submitted to the queue increases when reservations are supported.
2.12 MATCHMAKING USING OPTIMIZATION TECHNIQUES

The Grid scheduling system based on Ant Colony Optimization (ACO) (Gustavo Sousa Pavani 2009) was capable of co-scheduling of both computational and optical network resources. The author presented the comparison of traditional publish-and-subscribe grid systems with an extension of the ACO-based system and provided important benchmark comparisons. Since there was a need to dynamically provide light paths, they used Routing and Wavelength Assignment (RWA) algorithm to work closely with the Grid scheduler. The use of a Generalized Multi-Protocol Label Switching (GMPLS) control plane, combined with the RWA algorithm, provided a standardized way to manage the optical network. Ant-based RWA algorithms have been shown to outperform good conventional algorithms in dynamic, Telecom-oriented networks where requests are made for light paths connecting specific source-destination pairs of nodes. In Grid environments, where requests are made for connectivity to a resource to be discovered in the network, ant-based algorithms seemed to be even more appropriate, as they naturally combine the solutions to the discovery and routing problems.

2.13 CONCLUSION

In this chapter 2, the fundamental concepts of matchmaking are presented and its associated approaches are classified into different categories based on its distinguished features. The descriptions of each matchmaking approaches are presented. Techniques used in various frameworks are analyzed. Exhaustive study was made on the approaches to improve the quality of matching. Limitations in various approaches are analyzed.