Performance Based Hypercube Interconnection Structure for Better Grid Resource Discovery

M. Shanthi and Anthony Irudhayaraj

Abstract—**This paper focuses on the benefits derived from an extensive use of an interconnection structure for resource discovery and develop an algorithm which will help the user to discover the available resources among nodes in an efficient manner. The main objective of this research is to enable users to find out the available resources at the earliest to solve the problems. Most current implementations of grids have a centralized or hierarchical architecture with little attention to fault tolerance and scalability. Processing (resource discovery) must achieve performance in an environment which is heterogeneous, dynamic and shared by other users with competing resource demands. The proposed algorithm overcome the challenges like single point of failure , number of hops and network diameter.**

Index Terms—**Fault tolerenat hybecube, grid, processing resource discovery,scalable.**

I. Introduction

All Grid systems, which are composed of autonomous domains, are open and dynamic. In such systems, there are usually a large number of users, the users are changeable, and different domains have their own policies. Current technology either does not accommodate the range of resource types or does not provide the flexibility and control on sharing relationships needed to establish Virtual Organizations. The Grid infrastructure can benefit many applications, including collaborative engineering, data exploration, high-throughput computing, and distributed supercomputing. Computing Grids are basically networks that pool resources -- CPU cycles, storage or data from many different machines and apply them to one complex application. The grid can Parallel CPU capacity, Exploit underutilized resource, scalable Applications, Resource Balancing, Reliability and management

The grid resource discovery problem can be defined as the problem of matching a query for resources, described in terms of required characteristics, to a set of resources that meet the expressed requirements. The problem is complicated by the fact that some resource information (e.g., CPU load or available storage) changes dynamically. Resource discovery techniques maintain the resource attribute and status information in a distributed database and differ in the way they update, organize, or maintain the distributed database. Techniques that are used are fault tolerant and highly scalable.

Manuscript received September 10, 2012; revised October 15, 2012.

M. Shanthi is with the Velammal Engineering College,Chennai-66: (sha_karnan@gmail.com).

Anthony Irudhayaraj is with AVIT,Chennai, India.

Matching the needs of an application with available resources is one of the basic and key aspects of a Grid system. Resource Discovery is systematic process of determining which grid resource is the best candidate to complete a job with following trade-offs.

- In shortest amount of time.
- With most efficient use of resources.
- At minimum cost

Generally, resource discovery schemes maintain and query a Resource status database. One of the key motivations for constructing Grids is to provide application-level connectivity among the various machines so that resources and services supported by the individual systems can be shared in a Global fashion. To enable such sharing, it is necessary for the Grid architecture to support several services and resource discovery is one of them. In a Grid system, the resource discovery service may operate in conjunction with the resource management service. When a client requests service, along with the request it presents a set of attributes that should be satisfied by a candidate resource. The resource discovery process may be responsible for generating a set of best possible candidates for the given set of attributes. For scalable implementations, it is essential to organize the status databases in a distributed fashion. With a distributed organization for the status databases, the queries can be executed very efficiently but the updates to the databases may be costly.

Hypercube is one of the most important structures which are regular, symmetric, shorter diameter and having fault tolerant properties . Hypercube has a strong point that it can easily provide a communication network system required in applications of all kinds since it is node and edge symmetrical and has a simple reflexive system. Any n dimensional cube can be divided into two n - 1 dimensional cube. Many other topologies, such as ring, mesh, and tree, can be mapped into the hypercube topology. It is rich in connection; a message can be transferred from one node to all the other nodes in a total of n steps in in-cube. Extensive research efforts have been focused on hypercube design aspects and hypercube applications. Focus is needed to have such architectures that can accommodate large numbers of processors as needed at minimal cost.

In a Grid environment, it is hard and even impossible to perform scheduler performance evaluation in a repeatable and controllable manner as resources and users are distributed across multiple organizations with their own policies. To overcome this limitation, we have developed a Java-based discrete-event Grid simulation toolkit called GridSim. The toolkit supports modeling and simulation of heterogeneous Grid resources (both time- and space-shared),

DOI: 10.7763/IJMO.2012.V2.221

users and application models. It provides primitives for creation of application tasks, mapping of tasks to resources, and their management.

II. Modeling and Simulation

The GridSim toolkit supports modeling and simulation of a wide range of heterogeneous resources, such as single or multiprocessors, shared and distributed memory machines such as PCs, workstations, SMPs, and clusters with different capabilities and configurations. It can be used for modeling and simulation of application scheduling on various classes of parallel and distributed computing systems such as clusters, Grids, and P2P networks. The resources in clusters are located in a single administrative domain and managed by a single entity, whereas in Grid and P2P systems, resources are geographically distributed across multiple administrative domains with their own management policies and goals. Another key difference between cluster and Grid/P2P systems arises from the way application scheduling is performed.

III. Facilities

The GridSim toolkit provides facilities for the modeling and simulation of resources and network connectivity with different capabilities, configurations, and domains. It supports primitives for application composition, information services for resource discovery, and interfaces for assigning application tasks to resources and managing their execution. These features can be used to simulate resource brokers or Grid schedulers for evaluating performance of scheduling algorithms or heuristics.

IV. Netbeans IDE 6.8

NetBeans refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, Scala and Clojure. The Net Beans IDE is written in Java and runs everywhere where a JVM is installed, including Windows, Mac OS, Linux, and Solaris. A JDK is required for Java development functionality, but is not required for development in other programming languages.

V. Literature Survey

The main objective of employing resource discovery is to produce an optimized algorithm that covers all the working nodes within minimum amount of time. Literature survey provides enough facts that help in giving a strong base for the proper construction of the proposed system. It deals with less scalability, reliability, single point failure, fault tolerance and less network diameter. There are number of algorithms and approaches are available for resource discovery. Some of the resource discovery approaches found in the literature. Muthucumaru Maheswaran [Data Dissemination for Resource Discovery],Damandeep Kaur [Web-services based Grids],Seyed Mehdi Fattahi.Distributed[ACS Algorithm] ,Anju Sharma [Optimized Resource Discovery Algorithm] ,Jia-lin Jiao [improved Hilbert-Chord resources discovery algorithm] , JI Xiao-bo,[Hierarchy-based model],Libing Wu [Probe Feedback Mechanism] , Ami T [Improving Semantic Matching of Grid Resources Using Refined Ontology with Complement Class]. Anand Padmanabhan self-organized grouping(SOG) and Youli Min [Message Routing Algorithm].

Centralized resource discovery schemes such as Napster and Globus original. MDS implementation is based around a cluster of central servers hosting a directory of resources. These introduce issues such as scalability, since the entire index must be stored on a single cluster. Centralized schemes can be vulnerable to attack, such as Denial-of-Service attacks, because there is a single point of failure. Distributed solutions such as Gnutella discover resources by broadcasting queries to all connected nodes.

Anju Sharma and Seema Bawa [1] analyzed various resource discovery algorithm like Peer-to-Peer approach, Request forwarding approach and compared performance factors like scalability, reliability and adaptability. Antonia Gallardo, Luis Diaz de Cerio and Kana Sanjeevan [2] discussed about the features being used in their work which is able to handle dynamic attributes by using Hypercube topologyBuyya R and Murshed M [3] allowed modeling of heterogeneous types of resources and capability can be defined (in the form of MIPS (Million Instructions Per Second) as per SPEC (Standard Performance Evaluation Corporation) It supports simulation of both static and dynamic schedulers, There is no limit on the number of jobs that can be submitted to a resource. HaoRen Zhiying and Wang ZhongLiu[4] proposed new approach for DGIS that imposes a deterministic P2P shape based on hypercube topology, which allows for very efficient query broadcasting. Iamnitchi and Ian Foster[5]-[6]proposed the set of request forwarding algorithms in a fully decentralized architecture, designed to accommodate heterogeneity and dynamism.

Sujoy Basu and Sujatha Banerjee[7] described about multi-attribute range queries to be performed efficiently in a distributed manner Yan Ma[8] ,introduced small-world cluster into hierarchical grid in which intra-cluster adopts centralized management and cluster center nodes form small-world network. The architecture strikes a balance between high efficiency of total centralized management and good scalability of absolute distributed disposal. John B. Oladosu Titilayo M. Ajulo, Kazeem O. Adepoju 'their work enabled online real time access to academic records for the purpose of higher education using VB.net deployed on an already existing test bed (AccessGrid).

VI. Motivation

Resources are of highly diverse types (computers, data, services, instruments, storage space) and characteristics (e.g. operating system). Industrial processes require real time resource discovery algorithms, keeping the search time intact. In the past several years, the number and variety of resources available on the Internet have increased dramatically. With this increase, many new systems have been developed that allow users to search for and access these resources. One of main challenges is the heterogeneity that results from vast range of technologies both hardware and software encompassed by the grid. The motivation behind this paper is to explore the resource discovery mechanisms, which is suitable with the network environments that is the resource discovery should find out the preferred resources quickly and return the result with minimum time. In this paper focus is made to get the resource discovery efficiency with less system cost in the network. In order to discover resources in network environment, resource discovery algorithm is required to support user requests for resources, provide in-time resources information, and still serve for user's applications even some resources happen to be fault. A routing algorithm specifies how a message selects a path to cross from source to destination, and has great impact on network performance.

The motivation of using the Hypercube is twofold. Firstly, it provides a data structure with high connectivity. This makes it highly robust to failures. Secondly, the overlay distance between two nodes in the Hypercube is encoded in the hamming distance between their binary labels. The hypercube satisfies almost all the problem of scalability, partition ability and fault tolerance. The other feature which make the hypercube useful for distributed computation are short diameter, homogeneity of structure embedding and mapping capabilities. Topology like ring mesh tree could be embedded into the hypercube space. The potential of hypercube can be improved through better routing algorithm and hardware improvement.

VII. Proposed Work

The main objectives of this research is to design a new alternate computationally simple resource discovery algorithm in hypercube based structure which is suitable for real time applications. In this research, a new algorithm is proposed to meet the challenges like single point of failure, taking number hops, not reaching all the alive nodes in single trace. The paper presents a simple and effective topology for the concurrent discovery of the resources in the neighbour nodes of heterogeneous systems. Optimizing systems around specific workloads is an increasingly attractive means to improve performance.

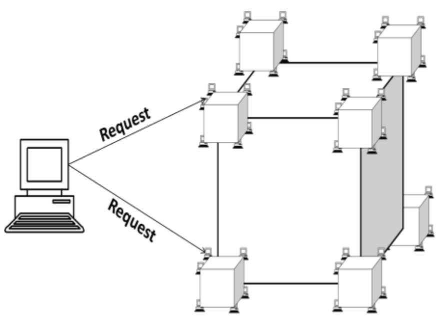

Fig. 1. Hypercube structure

Generally hypercube has n number of edges connected to each node. Our proposed method has one more edge added to each node, by making provision to have one more edge from each node connecting a node which is at exactly two levels below. The diameter of a network is the maximum internodes distance i.e. it is the maximum number of links that must be traversed to send a message to any node along a shortest path.

A broadcast operation is initiated by a single processor, the source. The source wants to send a request message to all other nodes in the system. The distance between two nodes in an undirected graph is the number of hops of a minimum path between the nodes. Each node in the hypercube is capable of achieving the following three tasks.

Process 1. Receive request . Responsible for getting the request from neighbour node

Process 2. Search for the resource : Specify whether the requested resource is available in the node.

Process 3 Requests forwarding. Listen and receive requests from users, forward requests to appropriate neighbors based on the accumulated experience.

When a user requests for a resource, the search procedure is executed at the home node. If the request is not satisfied, the search is moved to the next neighbour node.

Step1: Starts searching the home node
Step2 : If not satisfied(required information
not available in the home node)
Then Searching is carried out in
the neighbour node
Check the status of each neighbour node
If any neighbor is non alive
Start searching from the extended node
Else
Start searching from the neighbour node
Step-3 : Repeat step-2 until all nodes gets exhausted

The purpose of reconfiguring is to reduce depth (level) of the search .

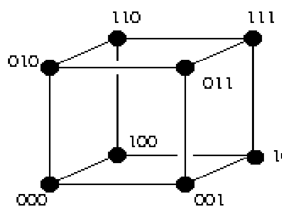

Fig. 2. Hypercube for n=3

Assume n=3 .The search starts from 000 . if request is not satisfied then look for the neighbour nodes (001,010and 100). That is change by 1 bit. If any node is non alive search can be conducted through extended link(2 bit changes) . The strategy is based on the distance between the nodes in the hypercube. The algorithm is trying to decrease the distance between communicating nodes in the hypercube. Normally for broadcasting the following figure, seven edges and 4 levels are needed. This can be reduced by using extended link concept.

GridSim provides a comprehensive facility for creating different classes of heterogeneous resources that can be aggregated using resource brokers for solving compute and data intensive applications. In this section, we present the simulation results using the GridSim simulator [3] The algorithm has been implemented for three dimensional system. This algorithm has been evaluated in comparison with the existing algorithm. In order to evaluate the algorithm, following test cases are considered.

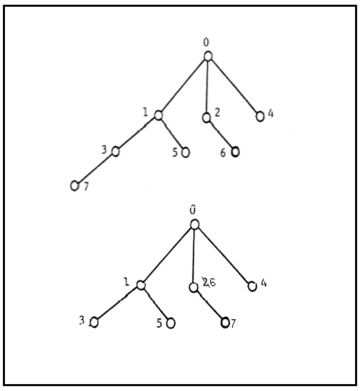

Fig. 3. Tree structure

Best Case

All nodes in the network are in alive state and request has been satisfied at the first hop itself.

Average Case

Some of nodes are in alive and some are in non-alive state and the request is satisfied at the next hop.

Worst Case

Most of the nodes are in non-alive state and the request is satisfied at the last zone.

TABLE I: SIMULATED DATA

	No.of Non-alives	No. of alives	No.of resources	No of hops by Existing Algorithm	No of hops Proposed Algorithm
Best	0	8	12	12	9
Average	2	6	8	14	10
Worst	4	4	5	6	4

The Existing algorithm[2] is designed to reach all nodes in the network. Some of the alive nodes are not reached while traversing the above algorithm. In addition dead or non-alive nodes are reached which results in time consumption and it reduces the efficiency. In the following diagram 010,001&101 are assumed as non alive nodes. If we follow the traversal using H-Grid algorithm 010(non alive) is reached .

Finally an application is developed for better resource discovery. The algorithms considered in this paper fall into the category of random walk and flooding based search algorithms. In hypercube interconnection networks, search for next neighbor node works as follows. An initial node accepts the request from the user. Each request includes one or more requirement for resources (memory, CPU, Printer and files). Then the initial node initiates a search process to discover the desired number of results that satisfy the required resources. The program is written to identify the resources connected to systems and then the task is transferred to the node based on availability. Application includes node creation, availability of the node and tasks transfer. Node initializes a search for a certain resource.

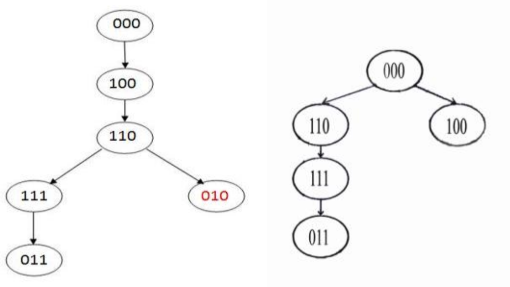

Fig. 4. Existing and proposed diagrams

VIII. CONCLUSION

A study was done on the existing interconnection structures and resource discovery algorithms and their performance. The proposed algorithm is successful in achieving the objectives like less number of hops, reaching all the alive nodes.Numerical results show that dynamic search can obtain short search time and provide a good tradeoff between the search performance and cost. It decreases the cost of resource discovery time. Hence an algorithm is implemented to reduce the time required almost equal to half to that required for normal search. Under different contexts, dynamic search always performs well. When combined with knowledge-based search algorithms, its search performances could be further improved.

REFERENCES

[1] A. Sharma and S. Bawa "Comparative Analysis of Resource Discovery Approaches in Grid Computing Computer Science & Engineering," *Journal of Computers*, 2008, vol. 3, no. 5, pp. 60-64.

[2] A. Gallardo, L. D. de Cerio, and K. Sanjeevan "HGRID: A Self-configuring Grid Resource Discovery," *Journal of Computing and Information Technology* - CIT, 2008, no. 4, pp. 333–338.

[3] R. Buyya and Murshed M., "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," *Concurrency and Computation: Practice and Experience (CCPE)*, 2003, vol. 14, Issue 13-15.

[4] H. Zhiying and W. Z. Liu "A Hyper-cube based P2P Information Service for Data Grid," in *Proceedings of the Fifth International Conference on Grid and Cooperative Computing* (GCC'06), 2006, China.

[5] A. Iamnitchi and I. Foster, "On fully decentralized resource discovery in grid environments," *Second International Workshop on Grid Computing*, 2003, pp. 51-62.

[6] A. Iamnitchi and I. Foster "A Peer-to-Peer Approach to Resource Discovery in Grid Environment," *The University of Chicago*, 2002, USA.

[7] S. Basu, S. Banerjee, P. Sharma, and S. J. Lee, *Node Wiz: Peer-to-peer Resource Discovery for Grids*, 2006, HP Labs, USA.

[8] Y. Ma, B. Gong, and L. Zou, "Resource Discovery Algorithm Based on Small World Cluster in Hierarchical Grid Computing Environment," *Seventh International Conference on Grid and Cooperative Computing*, 2008, pp. 110-116.

Mrs. M. Shanthi is doing her Ph.D. in Mother Teresa Women's University, Kodaikanal and working as an Assistant Professor , MCA Dept. in Velammal Engineering College, Chennai. She is having an academic experience of about **20** years and has presented papers in National and International conferences and published research papers and articles in Magazines and Journals.

Dr. A. Anthony Irudhayaraj is presently the Dean of Information Technology. He is an active member of IEEE and advisor of IEEE Computer Society Student Branch Chapter. He is the Research Supervisor in a number of universities presently guiding four research scholars. He is also the reviewer and member of the editorial board for a number of journals.