International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

DRRD: DYNAMISM RESILIENT RESOURCE DETECTION SCHEME FOR DISTRIBUTED ENVIRONMENT

M. Vidhya Kumari[1], R. Asmetha Jeyarani[2]
[1]PG Scholar, Computer Science and Engineering, Shivani Engineering College, Tiruchirappalli, India
[2]Assistant Professor, Computer Science and Engineering, Shivani Engineering College, Tiruchirappalli, India.
Email : [1]kumarividhya29@gmail.com, [2]asmetha.r@gmail.com

Abstract

In recent years, resource sharing services in grid systems are popular and it stimulate content-delivery overlay network. In existing, Circulated Hash Table is the data structure used for pointing nodes to the keys of a network based on a consistent hashing function. Most of the CHT overlays require O (m) hops per request with O (m) neighbors per node, where m is the network size. A computing resource is constantly portrayed by a resource type and functionality such as Main Memory and processing unit, and resource attributes representing the quantity and specialty. Hereby we propose Grid resource detection with physical proximity. Dynamism resilient resource algorithm, Token-Cluster algorithm, and time sliced resource algorithm are proposed. On adopting these algorithms, less overhead consumption, speedy and dynamism-resilient multi-resource detection is done. Experimental result shows well-organized resource clustering, reduces communication cost, and enhances resource detection success rate flexibly surviving in large scale applications.

Keywords **- Grid systems, Circulated Hash Table, resource detection, overlay network, physical proximity, dynamism resilient, Token-Cluster.**

I. INTRODUCTION

Grid computing leverages a high-degree of resource sharing in a distributed network environment. It enables the sharing, selection, and aggregation of a wide variety of resources including supercomputers, storage systems, data sources, and specialized devices. Overlay networks based on circulated hash table (CHT) have been suggested to manage large-scale Grid resources [9]. A computing resource is always described by a resource type (i.e. functionality) such as CPU and memory, and resource attribute indicating the quantity and special application requirement. To use a CHT overlay for resource detection in a Grid system, all Grid nodes are organized into a CHT overlay. Therefore, CHT overlays map the resource providers and consumers in Grids in a distributed manner with high scalability. However, the adoption of CHT overlays in most current resource management schemes [3], [4], [5] cannot preserve program data locality. Most CHT-based solutions of multi-resource management have limited scalability and fault tolerance. Some solutions apply multiple CHT overlays to manage specific resource groups [3]. This may result in significant increase in management overhead and low dynamism resilience since node joins and departures lead to update of multiple CHT overlays. They store resource descriptors in a few nodes, resulting in imbalanced distribution of resource management workload and loss of many descriptors in dynamism. We desire to have a Grid resource management scheme that is program /data physical proximity and highly scalable and dynamism-resilient.[2]

We establish program/data locality by clustering resources based on their physical proximity and functional matching with user applications. We further develop randomized probing and Token-Cluster algorithms. The novelty of the SCO scheme lies in its low-overhead, fast and dynamism-resilient multi-resource detection.

II. RELATED WORK

CHT overlay networks [10], [8] have been suggested to manage large-scale Grid resources. Mercury [3] is a resource detection protocol for routing multi-resource range-based queries. It can also support explicit load balancing. To support multi-resource queries, Mercury uses multiple CHT overlays. It uses one CHT for each resource, and processes multi-resource queries in parallel in corresponding CHT overlays. However, depending on multiple CHT overlays leads to high overhead for CHT maintenance. SOMO [11] is a scalable, efficient and robust infrastructure for resource management in CHT overlay networks. SOMO performs resource management by relying on a tree structure. It does so by gathering and disseminating system metadata in O (log n) time with a self-organizing and self- healing data overlay. MAAN [4] is a Multi-Attribute Addressable Network that extends Chord to support multi-resource and range queries for grid information services. MAAN addresses range queries by mapping attribute values to the Chord identifier space via uniform locality preserving hashing. It uses an iterative or single attribute dominated query routing algorithm to resolve multi-resource based queries.

SWORD [5] is a resource detection service for wide-area distributed systems. It locates a set of machines matching user-specified constraints on both static and dynamic node characteristics. It has a technique for efficient handling of multi-resource range queries that describe application resource requirements. By leveraging Chord topology and routing mechanisms, the DAT trees are implicitly constructed from native Chord routing paths without membership maintenance. SEMM [6] provides a preliminary study of exploiting a Sequential cycloid overlay for resource management.

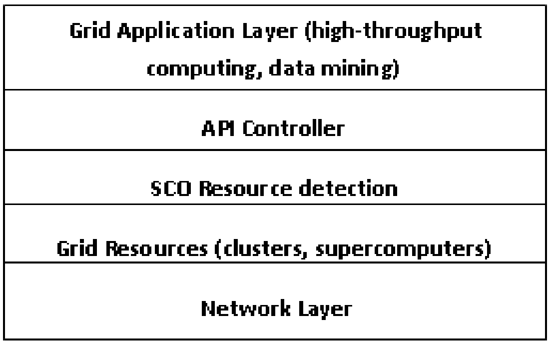

Figure1: Grid Protocol Architecture with Sequential Cycloid Overlay.

SCO distinguishes itself by addressing detailed issues in resource detection and proposing strategies for locality preserving, dynamism-resilience and efficient resource detection. Figure 1 shows the Grid Protocol Architecture with Sequential Cycloid Overlay. These features contribute to the high scalability and efficiency characteristics of SCO in Grid resource management. Comprehensive simulation results confirm the high performance of SCO.

III. SEQUENTIAL CYCLOID OVERLAY NETWORK

We present below the architecture and processing layers of SCO. This is a CHT-based hierarchy or Physical Proximity Grid resource clustering and detection. SCO is built by extending from the cycloid overlay proposed in [8]. Cycloid is a lookup efficient overlay network generalized form the cube-connected cycles [9]. A d-dimensional cycloid is built with at most n=d 2d nodes. Like CCC, a cycloid has a constant node degree equals to its dimensiond.

An object is assigned to a node whose ID is closest to its ID. The overlay network provides two main functions: Insert(ID, object) stores an object to a node responsible for the ID, Lookup(ID) retrieves the object through CHT-based searching. Each node maintains a routing table recording its neighbors in the overlay network for object lookups. Most properties of cycloid can be found in [8]. A landmark clustering is adopted to generate proximity information [12].

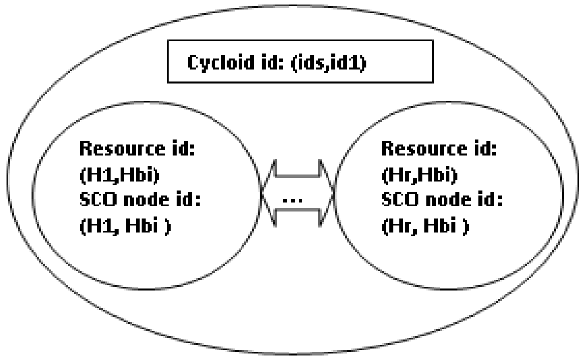

Figure 2: A Cycloid overview in SCO architecture

We assume m landmark nodes that are randomly scattered in the Grid. Each node measures its physical distances to the landmark nodes. A vector of distances < d1, d2... dm > is used to perform clustering. Two physically close nodes have similar vectors. We use space-filling Hilbert curve [12], [1] to map each m-dimensional landmark vectors to a real number. The number is called the Hilbert number of a node denoted by Hb. Its purpose is to preserve the physical proximity among the nodes selected for the same cluster in SCO. The SCO architecture builds a topology-aware cycloid architecture on a Grid. Specifically, each node generates its ID (H, Hb), where H is the consistent hash value of its IP address. Therefore, physically close nodes with the same Hilbert number are in the same cluster, and those with similar Hilbert number are in nearby clusters. To build each node's routing table, SCO uses proximity-neighbor selection technique [6]. As a result, a topology-aware cycloid is constructed, in which the logical proximity abstraction derived from overlay matches the physical proximity information in reality.

IV. PHYSICAL PROXIMITY GRID RESOURCE DETECTION

Using a flat CHT, resource descriptors are gathered in different repository nodes based on resource type. This has posed a challenge to achieving program/data locality, by which a node can locate physically close resources for its multi resource query by only probing its nearby nodes in the CHT overlay. The idea is to map functional resources in the same physical cluster to logically close nodes to satisfy specific application demands. The logical distance between a node and a cluster on the SCO reflects the physical distance between the node and the resource.

A. Physical Proximity Resource Clustering

In general, resources required by a Grid application is specified by a set of resources such as CPU, memory, band-width, I/O subsystem, etc.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

An effective resource detection algorithm locates resources across a wide area based on a list of predefined attributes. We specify each resource in node i by a resource descriptor Dr, consisting of the following 4-tuple.

Theorem: If nodes j and k are directory nodes of resource requested by node i, and IDi ≤ IDj < IDk or IDi ≥ IDj > IDk, then directory node j's resources are physically closer to node i than directory node k's resources. The load balancing algorithm in [7] can be further adopted to achieve more balanced descriptor distribution between the directory nodes. Since this is not the focus of this paper, we do not present the details of the load balancing.

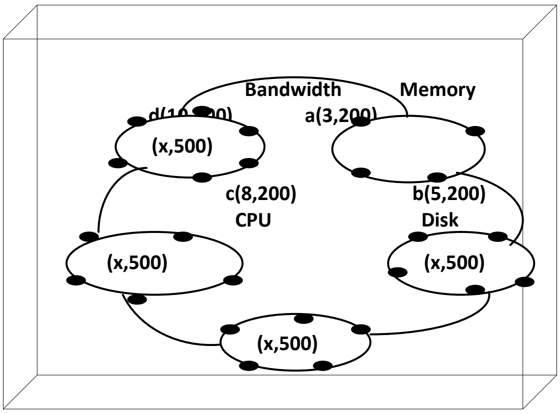

Figure 3: Cycloid node routing link states.

B. Physical Proximity Resource Detection

When node i queries for multiple resources, it sends a request Lookup(Hr , Hbi) for each resource r. Each request will be forwarded to its directory node in node i's cluster. If the directory node has no requested descriptor, it probes nodes in nearby clusters. Theorem 4.1 indicates that the resources of directory nodes in closer clusters are physically closer to the requester. Hence, a node should probe its logically close neighbors in order to locate physically close resources. We present the successor and predecessor clusters of node j's cluster as sucCluster(j) and preCluster(j), respectively. Firstly, a node probes the directory nodes in these clusters simultaneously. Then, the directory nodes in sucCluster(sucCluster(j)) and preCluster(preCluster(j)) are probed. This process is repeated until the desired resource descriptors are found. However, such sequential probing is not robust enough to handle dynamism where nodes join and leave the system continually. We incorporate the algorithm in [7] and develop proximity-aware randomized probing algorithm (PRP) to resolve the problem. In the PRP algorithm, a node first applies sequential probing. If no response is received during a predefined time period, the node randomly chooses two nodes in an increasing range of proximity and repeats the probing process.

Since resource descriptors are allocated to different nodes in a cluster based on resource functionality, the probed nodes should be the directory nodes of the requested resource. Based on the resource clustering algorithm, we know that the probed nodes should have the closest cyclic ID to the probing node. Therefore, the probing node can reach them by targeting an ID composed of its cyclic ID and a randomized cubical ID chosen in an increasing proximity.

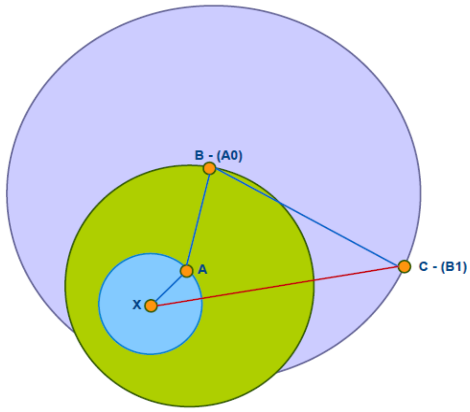

Figure 4: Physical proximity of resources from C – X is greater when pass through C – B – A – X.

C. Dynamism-Resilient Resource Management

In addition to exploiting the physical proximity of the network nodes to minimize operation cost, an effective resource management scheme should also work for Grids in a dynamic environment. When node i joins in the system, it reports its resources via Insert((Hr , Hbi),Dr), and receives the descriptors in its responsible ID region from its neighbors. When a node departs from the system, it transfers descriptors to its neighbors. For example, if node (2, 200) joins the system in Figure 3, then the descriptors in the range (0, 200) and (2, 200) are transferred from node (3, 200) to node (2, 200). If node (3, 200) leaves, it transfers its descriptors to node (10, 200) or (5, 200) based on the ID closeness. If node (3, 200) is the only node in its cluster, it transfers its descriptors to its closest node in its closest cluster. The consistent hashing for key assignment protocol requires simple re assignment of resource descriptors. SCO resorts to periodical resource reporting to avoid useful descriptors from being lost in the clustering and detection process. If a directory node has failed, its resource descriptors are lost. Within T, the lost resource descriptors will be reported to a new directory node. To prevent the descriptor space from being flooded with outdated descriptors, the directory nodes execute garbage collection periodically. Consequently, instead of relying on specific nodes for resource descriptors, SCO always stores a resource descriptor in a directory node, and the Lookup(Hr , Hbi) requests can always be forwarded to the node.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

D. Token-Cluster Algorithm

We introduce Token-Cluster forwarding algorithm to further enhance the efficiency of the SCO scheme. Like most multi-resource management approaches, SCO uses m lookups for a query of m resources. Thus, rather than using m lookups, a node can combine the m lookups into one lookup message to be sequentially routed within a cluster. Moreover, since a node with available m resources needs m Insert() messages for resource clustering, which are routed in the same manner as the Lookup() messages, the two kinds of messages can be integrated. Furthermore, since the messages of all nodes in a cluster are routed in the same manner and the nodes need to report their available resources periodically, the messages for resource clustering and detection of all the nodes can be combined.

At this time, the token has no Dr of available resources and the Dr left are for resource requests. The primary node uses PRP algorithm to forward the token to another cluster. At the cluster, this process is repeated until the token is empty, i.e. all requested resources are discovered. Therefore, in the algorithm, only one message is generated periodically. Combining a number of messages into a single message for forwarding within a cluster and between clusters significantly reduces cost.

V. PERFORMANCE EVALUATION

The dimension of the cycloid simulated is 11. Thus, the CHT overlay network can accommodate 4096 nodes. We compare the performance of SCO with MAAN [4], Mercury [3], and SWORD [5]. To be comparable, we used Chord for attribute hub in Mercury and SWORD. The experimental results show advantages in using SCO over the competing overlays for the same purpose. We generated 1000 requests. The number of resources in a request ranges from 1 to 5 with step size of 1. We used a transit-stub topology generated by GT-ITM [10] with approximately 5,000 nodes.

A. Cost in Grid Resource Detection

In this experiment, we randomly generated 5000 resource requests, and recorded the distance between the resource provider and requester of each request. Figure 5 shows the CDF of the percentage of allocated resources against the physical hop distance. We can see that SCO is able to locate 97% of total resource requested within 11 hops, while others locate only about 15% within 10 hops. The more resources are discovered in shorter distances, the higher efficiency of Grid applications. The results confirm the unique physical proximity feature of SCO to enable users to locate physically close resources.

B. Performance in a Dynamic Grid Environment

MAAN and SWORD incur lower success rates than SCO and Mercury. Because of dynamism, some requests may be lost. More requests fail to arrive at their destinations successfully when the node arrival/departure rate increases, leading to decrease of success rate. SCO and Mercury distribute resource descriptors among all nodes in the system, while MAAN and SWORD mainly depend on 11 nodes, so any departure in the 11 nodes will result in the loss of a high volume of resource descriptors, resulting in sharp drop off of success rate. In a dynamic environment, the number of nodes probed in the probing phase will be more than that of static environment due to node joins and departures. These results verify the superior performance of Mercury and SCO, compared with MAAN and SWORD in handling network dynamism.

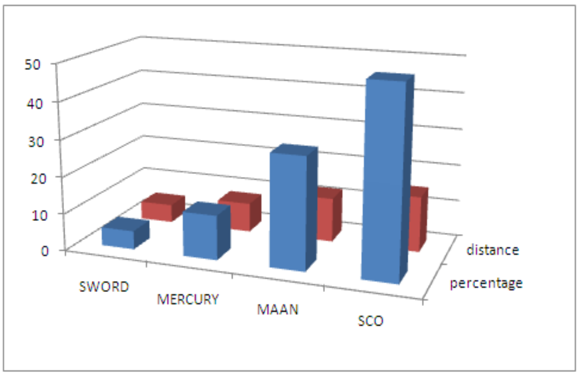

Figure 5 CDF of the percentage of allocated resources against the physical hop distance.

VI. CONCLUSION

Rapid development of Grids demands a scalable and efficient resource management scheme to sustain distributed performance in a dynamic wide-area environment. The major contributions of this work are summarized below: (a) This paper presents a SCO by extending the cycloid CHT overlay. The SCO pools physically close resource together in logically-close clusters. (b) We have developed physical proximity algorithms to enable users to dynamically discover physically close resources with required functionalities in their neighborhood. Most previous schemes fail to preserve the locality and require users to discover resources in the system-wide scope. (c) The SCO scheme uses a single large CHT overlay with low overhead. It achieves balanced workload distribution and resilience to resource failure. Most previous schemes use multiple CHT-based overlays causing high overhead or one CHT overlay causing workload imbalance.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Both of those are more suitable for static Grid configurations with limited applications. (d) The Token-Cluster algorithm enhances system efficiency. Simulation results reported demonstrate the superiority of using SCO in Grid reconfiguration for large-scale and dynamic applications.

The proposed framework is still under intensive system and middleware development. Further research aspects: (1) Prototyping of the proposed SCO (2) Developing benchmark programs to test the efficiency and validate the claimed advantages. (3) Apply virtual machine techniques to extend the SCO model to secure Grid applications. (4) Integrate Grid and P2P technologies with machine virtualization techniques for global-scale Internet applications.

REFERENCES

[1] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmaier. Space filling curves and their use in geometric data structure, Theoretical Computer Science, 181(1):3-15, 1997.

[2] Haiying Shen and Kai Hwang, "Locality-Preserving Clustering and Discovery of Resources in Wide-Area Distributed Computational Grids",University of Southern California, Los Angeles, 2010.

[3] A. R. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting scalable multi-attribute range queries. In Proc. of ACM SIGCOMM, pages 353-366, 2004.

[4] M. Cai, M. Frank, and et al. MAAN: A multi-attribute addressable network for grid information services. Journal of Grid Computing, 2004.

[5] D. Oppenheimer, J. Albrecht, and et al. Scalable wide-area resource discovery. Technical Report TR CSD04-1334, EECS Department, Univ. of California, Berkeley, 2004.

[6] H. Shen. SEMM: Scalable and Efficient Multi-ResourceManagement in Grids. In Proc. of GCA, 2008.

[7] H. Shen and C. Xu. Locality-aware and churn-resilient load balancing algorithms in structured peer-to-peer networks. TPDS, 2007.

[8] H. Shen, C. Xu, and G. Chen. Cycloid: A Scalable Constant-Degree P2P Overlay Network. Performance Evaluation, 63(3):195-216, 2006.

[9] J. Smith and R. Nair. Virtual Machines,Virtual Machines:Versatile Platforms for Systems and Processes. Morgan Kaufmann Publisher, 2005.

[10] E. Zegura, K. Calvert, and et al. How to model an Internet-work. In Proc. of INFOCOM, 1996.

[11] Z. Zhang, S.-M. Shi, and J. Zhu. Somo: Self-organized metadata overlay for resource management in P2P CHT. In Proc. of IPTPS, 2003.

[12] Z. Xu, M. Mahalingam, and M. Karlsson. Turning heterogeneity into an advantage in overlay routing. In Proc. of INFOCOM, 2003.