Automatic Clustering of Grid Nodes

Qiang Xu
Department of Electrical and
Computer Engineering
University of Houston
Houston, Texas 77204
Email: Qiang.Xu@mail.uh.edu

Jaspal Subhlok
Department of Computer Science
University of Houston
Houston, Texas 77204
Email: jaspal@uh.edu

***Abstract*— In a grid-computing environment, resource selection and scheduling depend on the network topology connecting the computation nodes. This paper presents a method to hierarchically group compute nodes distributed across the internet into logical clusters, and determine the relative location of the clusters. At inter-domain level, distance from landmarks (a small group of distributed reference nodes) is the basis for converting the location of nodes inside a complex network structure onto a simple geometric space. The position of compute nodes in this geometric space is the basis for partitioning nodes into clusters. For compute nodes within an administrative domain, minimum RTT is used as the metric to partition nodes into clusters. This approach leads to an efficient, scalable and portable method of clustering grid nodes and building a distance map among clusters. We demonstrate the system for automatic clustering by applying it to computation nodes distributed across five universities in Texas.**

I. Introduction

A Grid can be defined as a large-scale distributed network computing system that is a result of transparent, systematic and effective utilization of geographically and administratively distributed computation resources. A grid-computing system must include a scheduler to select appropriate resources (hosts, network links, etc) for applications. Information service systems such as the Network Weather Service (NWS) [15] are commonly employed by Grid middleware systems like Globus [6] to capture the current state of the grid platform. However, a knowledge of the state of network links (available bandwidth and latency) and the available CPU and memory on compute nodes is not sufficient to solve the resource selection problem. The relative location of the nodes and the network topology interconnecting the nodes are also important factors for the scheduler to decide how to aggregate the available computational resources for a specific application. But the heterogeneous, dynamic, and distributed nature of a grid system makes it very difficult to capture the physical network topology graph. It is often impossible to query the necessary information from routers, switches and other network devices configured and managed by different administrative domains because of technology, security and administrative reasons.

However, for solving the basic resource selection problem, a complete knowledge of the physical network topology is not required. Normally, a coarse and logical description of the network connecting the computing nodes is sufficient. In this paper we focus on grouping "nearby" nodes into clusters and computing the "distance" between these clusters. This information is normally sufficient for a scheduler to determine which groups of nodes can be efficiently employed to solve smaller problems, and which combinations of clusters are likely to be effective for larger problems. Figure 1 illustrates the problem we solve in this work.

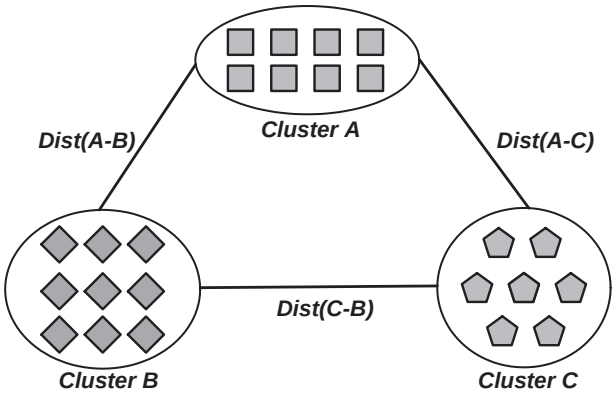

Fig. 1. Mapping network nodes onto a geometric space to construct logical clusters and a distance map between the clusters.

The specific problem we address is as follows: Given a set of compute nodes represented by their IP addresses/hostnames, the goal is to partition them into clusters and place the clusters in Euclidean space to capture their relative locations and inter-cluster distance. The procedure consists of distinct steps for partitioning nodes distributed across the internet into large clusters typically consisting of nodes belonging to a domain, and partitioning of nodes within a domain.

For partitioning of nodes distributed across the internet, we employ the concept of landmarks from GNP (Global Network Positioning) [11] for a scalable and efficient solution. Landmarks are a small group of carefully selected distributed hosts that work as a distributed frame of reference for positioning the compute nodes in a grid computing system. The coordinate set of each node is the latency vector measured from the landmarks to the compute node. Hence, N landmarks will map the compute nodes into an N-dimensional geometric space, where the nodes belonging to the same domain will be close to each other and the nodes belonging to different domains will have longer distances between them. The clustering is

then accomplished based on Euclidean distance and a heuristic distance threshold.

The above approach is not always effective for partitioning within a single domain, such as a university, because the distance vectors for nodes inside a domain are often too close to each other for meaningful and accurate clustering. The reason is that the difference in latency from a distant landmark to various nodes within a domain can be smaller than the error tolerance of this approach. Selecting landmarks within the domain does not always resolve the problem because of the relatively low latencies inside a domain, typically under a millisecond. In this case we perform clustering based on direct latency measurements between nodes. This introduces additional measurement complexity, but it is not a problem in practice since the number of nodes within a domain is smaller and pairwise measurements can be made efficiently.

The rest of this paper proceeds as follows. After a discussion of related work in Section II, we present the algorithms and implementation of inter-domain and intra-domain compute node clustering, in Section III and Section IV, respectively. Experimental results are presented in Section V followed by discussion in Section VI. We conclude in Section VII.

II. Related Work

A grid can span a collection of heterogeneous networks, which makes network topology discovery an inherently challenging task. Lowekamp et. al. [10] and Bejearno et.al. [2] both develop novel and practical algorithms to discover the Layer-2 topology by utilizing information from the address forwarding tables. But their work is limited to a LAN or a single administrative domain, where the SNMP MIB information can be accessed. Instead of the physical LAN topology, our method aims to discover the logical topology, and applies to a grid-computing system with nodes distributed across multiple administrative domains.

The term *network tomography* was first introduced by Vardi [14] in 1996 due to the similarity between network inference and medical tomography. The network topology inference problem is to discover the internal logical topology by end-to-end measurements without internal network cooperation. By analyzing packet pairs (two small size packets sent out from the same host with a very small time difference) leaving from the sources and arriving at the destinations, Nowak et al. were able to identify the logical topology with multiple sources and multiple destinations without a knowledge of the physical topology [5], [12]. However, this approach has not been proven to be practical for large-scale network systems, where the overhead of the probes and sensitivity to the behavior of routers and other interconnecting devices are important factors.

Network distance, the metric used in our approach to clustering, is also the basis for several other network prediction and monitoring systems. IDMaps [7] is a measurement infrastructure to support a large-scale distance information query/reply service. It predicts the distance between hosts based on the triangulation inequality from each host's nearest ***Tracer*** node. Global Network Positioning (GNP) [11] represents the complex structure of the Internet by a simple geometric space based on a set of reference hosts called ***landmarks***. Each host has its own geometric coordinates based on distance from the landmarks, and the network distance between the hosts is modeled by Euclidean distance. Theilmann and Rothermel [13] propose dynamic distance maps of the Internet. They create a global view of the Internet from their ***mServers***, which hierarchically decompose the network into regions, and estimate the network distance between hosts. Iso-bar [4] is a scalable overlay distance monitoring system, which clusters hosts based on the similarity of their perceived network distance from designated ***landmarks***. Basically, the objective of using a ***Traccer/Landmark/mServer*** is to allow scalability of large-scale network measurements and analysis as it eliminates the need to measure the distance between every pair of nodes. We have borrowed this idea for clustering and use the term "landmark" in the rest of this paper. Our contribution is to employ these techniques to automatically partition grid nodes into clusters and build a knowledge base that can be used by grid resource managers.

Since compute nodes in a cluster are likely to have similar IP addresses, it is natural to explore that as a basis for clustering. In [8], [9], the similarity between the IP address prefixes is used to divide the hosts/clients into clusters. However, subnets with the same IP prefix can be geographically and logically separated from each other. Clustering in that case requires access to data from BGP (Border Gateway Protocol) tables, which may not be available on a grid. Our method is solely based on coordinated end-to-end latency measurements between sets of nodes, and hence does not make any assumptions about IP prefixes and does not require access to BGP tables.

III. Compute Nodes Clustering

The intuition behind our approach to clustering is that compute nodes within a logical cluster will show similar network behavior, such as latency and bandwidth, when measured from distant nodes. Conversely, nodes that are not close to each other in network sense will show very different behavior. Latency is the metric used in our hierarchical clustering. While the inter-domain latency could range from several milliseconds to several hundred milliseconds, the intra-domain latency is usually within 1 millisecond. Hence, our approach to inter-domain and intra-domain clustering is different.

A. Inter-domain clustering

The inter-domain clustering procedure can be divided into two phases: i) Using the latency as the metric, we map the nodes in the given computation grid onto an N-dimensional geometric coordinate space. Each compute node is represented as an N-dimensional vector formed by measured latency from a set of N landmarks. ii) Based on the Euclidean distance between node pairs and a heuristic distance threshold, we build an undirected graph with edges representing distance. Clusters

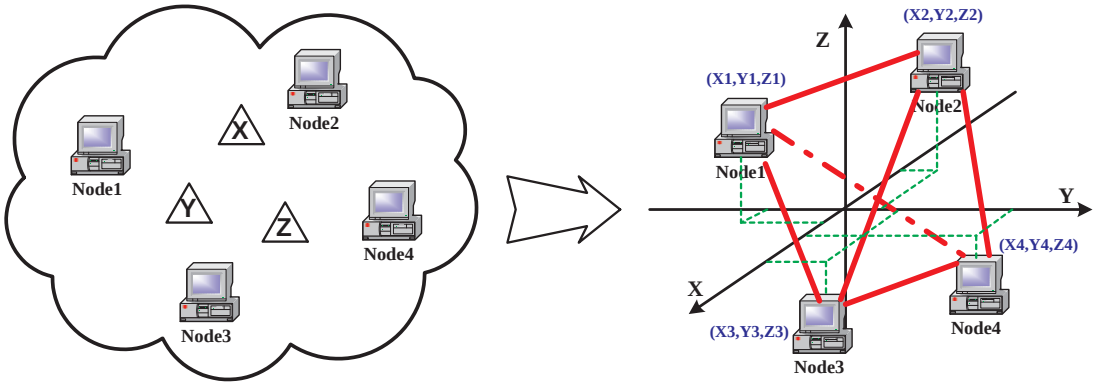

Fig. 2. Network Structure to Geometric Space

are created by repeatedly solving the maximal clique problem over this graph. We explain these steps in more detail.

1) Grid Nodes to Geometric Space: Based on the model proposed in the Global Network Position [11], the Grid nodes are mapped to an N-dimensional geometric space as illustrated in Figure 2. The N-dimensional space is based on a small set of N distributed nodes known as Landmarks. Each compute node is characterized by its geometric coordinates, which are derived from the Round Trip Times (RTTs) between the node and each landmark.

If there are N landmarks, L_1 to L_N, and M compute nodes H_1 to H_M, a total of $N \times M$ measurements are needed to capture the distance between every landmark and every compute node. The advantage of using landmarks is that the number of measurements is reduced to $N \times M$, from a possible $M \times M$ in a naive approach. Note that M increases with the size of the grid platform while N is a preselected constant denoting the number of landmarks. The difference is critical for large networks and makes this approach scalable. Another advantage of this landmark based approach over pairwise measurements is due to the asymmetry of the Internet; the path and latency from node A to node B is not always the same as those from node B to node A. With the landmark approach, since all distances are measured from landmarks, the asymmetry is not a problem. Finally, since all measurements are made from a few landmarks only, the approach is logistically much easier to manage.

2) Network Distance and Maximum Clique/Cluster Discovery: After mapping the grid nodes to a geometric space, clustering decision is based on the Euclidean distance.

$$\text{Euclidean\_Distance}(\vec{A} - \vec{B}) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

First, a complete undirected graph G(V, E) is built, where V is the set of nodes, and E represents the distance between the corresponding pairs of nodes. If the Euclidean distance between two compute nodes is greater than a threshold, the edge between that pair of nodes is removed. The Euclidean Distance Threshold is $\sqrt{n} \times T$, where n is the total number of landmarks, and T is determined heuristically based on maximum expected latency between two hosts inside a typical administrative domain. The new graph is again represented as G(V, E), where V is the set of compute nodes, and E is the set of edges whose weight is less than the threshold. The algorithm for building clusters from this graph is presented in Table I. This algorithm essentially finds the largest clique of nodes in the graph, builds a cluster from them, and then removes those nodes from the graph. The procedure is repeated until all nodes are assigned to clusters.

TABLE I

ALGORITHM TO CLUSTER GRID COMPUTE NODES

INPUT: An undirected graph G(V, E)

OUTPUT: Sets of nodes belonging to different clusters C_i (i=1 to K).

Initialize i=1.

1. Use the enumerative algorithm [3] to find a maximal clique of Graph G.

2. Assign the clique nodes to Cluster C_i .

3. Remove all nodes in Cluster C_i and all edges connected to those nodes from the graph G.

4. If G is NULL, Terminate. Else increment i = i+1 and Goto Step 1.

B. Intra-domain clustering

The inter-domain clustering algorithm discussed above will typically group all nodes in a domain in a single cluster. Inside an administrative domain, there can be multiple compute clusters belonging to different departments and research groups. In our experience, the landmark based approach to inter-domain clustering is not very effective within a domain. We employ directly measured RTT between pairs of nodes as the basis to discover the logical topology and partition the network into local clusters.

First, the latency between each pair of nodes inside the administrative domain is measured. This will require $M \times M$ latency measurements, where M is the number of nodes in

the domain. However, the latency is small (within 1ms) and the number of nodes is a fraction of the nodes in a full computational grid. Hence we believe this is practical.

We again build an undirected graph G(V,E), where V is the set of nodes within the domain and E represents edge distances. For edge (a, b), the distance is the average of RTT(a→b) and RTT(b→a). The basic idea of intra-domain clustering is as follows. For nodes within the same cluster, the distance of the edges between them should be within $\beta\%$ of the least expensive edge to each of them and within $\beta\%$ of the least expensive edge within the cluster. The algorithm is outlined in Table II. We heuristically choose β to be 20 in our experiments.

TABLE II
ALGORITHM TO CLUSTER COMPUTE NODES INSIDE A DOMAIN

INPUT: A undirected graph G(V, E)

OUTPUT: Sets of nodes belonging to different clusters C_i (i=1 to K).

C_i.min_edge always represents the weight of the least cost edge in C_i (or 0 if there are no edges within the cluster)

1. Each node in G initialized as a 1 node cluster.

2. For each edge E(a,b) in G in non-decreasing order

If (a and b are in different clusters Cluster(a) and Cluster(b)) then

If E(a,b) $<$ [(1+$\beta\%$)* Cluster(a). min_edge] AND
E(a,b) $<$ [(1+$\beta\%$)* Cluster(b). min_edge]

Merge Cluster(a) and Cluster(b)

IV. Implementation

A clustering framework has been developed based on the algorithms discussed. The system was applied to 36 nodes within Texas. Nodes at University of Texas, Austin, Carnegie Mellon University, Pittburgh and Rice University, Houston were used as landmarks. The intra-domain clustering was applied to 4 real clusters inside University of Houston. We discuss some of the interesting issues in the implementation — latency measurement and latency threshold.

A. Latency Measurement

Ping, developed in December of 1983 [1], is a popular tool to measure the RTT. It exploits the IP protocol by sending out an Echo-Request ICMP packet to the target host, which forces the target to return an Echo-Response ICMP message. Unfortunately, due to the malicious use of ICMP services, increasing number of routers and firewalls will not forward or respond to ICMP packets. Also, operating systems may limit the rate of ICMP response. All these mechanism reduce the efficiency and effectiveness of Ping.

TCP Ping stands for ping-like tools that exploit TCP's 3-way handshake in connection establishment. Unless the incoming TCP packet is blocked at target host, a response of SYN+ACK or RST is received as a response to a SYN request. The time between SYN and SYN+ACK/RST at the sender can be recorded to determine the RTT.

In our experiments, we use HTPing [16], which stands for *Hurricane TCP Ping*. It directs probe packets to a single target with fine inter-probe intervals (on the order of milliseconds).

B. Latency Threshold

The Euclidean Distance Threshold for deciding if a pair of nodes belong to the same cluster is $\sqrt{n}\times T$, where n is the total number of landmarks, and T is determined heuristically based on maximum expected latency difference between two hosts inside a typical administrative domain. In our implementation, we choose T to be 1ms.

We argue that if the latency difference between a pair of nodes and each landmark is less than 1 ms, then the nodes are likely to be in the same physical domain. The latency between any two nodes is composed of queuing delay, transmission delay and propagation delay. Hence, the difference in latency includes the difference in these three components of total delay. Since the packet size for latency measurement is small, the transmission delay is negligible. Queuing delay difference is expected to be small because the routes between landmarks and nodes within the same administrative domain will be the same except for a few hops close to the nodes under consideration. So, the propagation delay difference will dominate the total latency difference. The physical range of packet travel within a millisecond is roughly in the range of 2000 meters, which is fairly long inside a physical domain. Hence, 1 ms is a reasonable estimate for an upper bound on latency difference between a landmark and the nodes in the same subnet.

V. Experiments and Results

The clustering framework developed in this paper was applied to groups of nodes for validation. We discuss the inter-domain and intra-domain experiments separately.

A. Inter-domain experiment

For this experiment we selected 36 nodes at locations across Texas. These nodes are distributed in different administrative domains — Rice University in Houston, Texas A&M at College Station, Texas A&M at Galveston, and University of Texas at Dallas. The 3 landmarks are nodes located at UT, Austin, CMU, Pittsburgh, and Rice, Houston. The resultant mapping of the nodes to geometric space is illustrated in Figure 3, Figure 4 and Figure 5. In Figure 3, nodes at UT, CMU and Rice are used as landmarks, in Figure 4 nodes at UT and CMU are used as landmarks, and in in Figure 5, nodes at UT and Rice are used as landmarks. In all these figures, the symbols used to represent nodes for mapping and partitioning from each university are as follows: + UT Dallas, ○ TAMU Galveston, * TAMU College Station, and ◇ Rice.

It is clear from Figure 3 that nodes are cleanly partitioned into their natural clusters when the three landmarks are employed for mapping to a 3-D geometric space. Figure 4 and

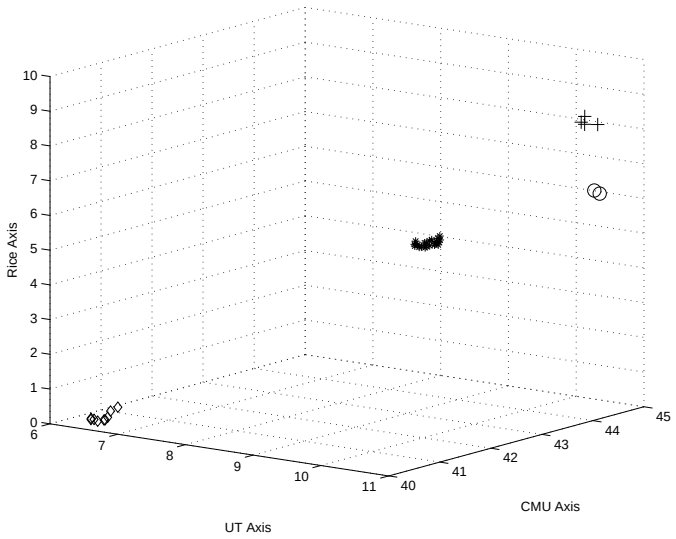

Fig. 3. Nodes from different domains distributed in 3-D space with UT, CMU and Rice as axes. (+ UT Dallas, ○ TAMU Galveston, * TAMU College Station, and ◇ Rice)

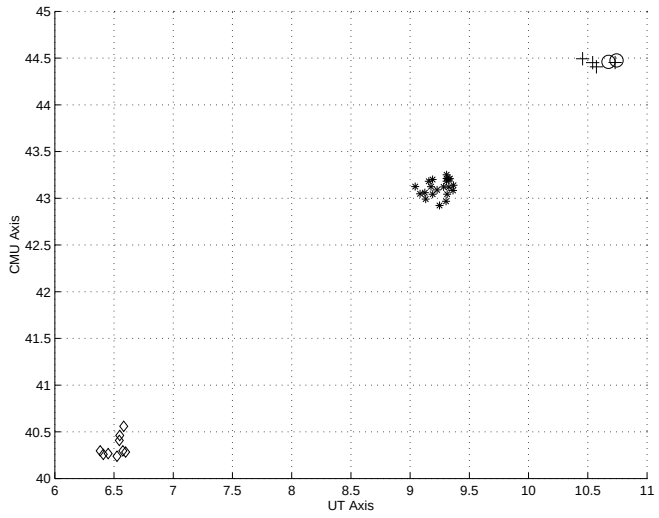

Fig. 4. Nodes from different domains distributed in 2-D space with UT and CMU as axes. (+ UT Dallas, ○ TAMU Galveston, * TAMU College Station, and ◇ Rice)

Figure 5 show mappings to a 2-D geometric space employing only 2 landmarks. One thing to note is that, in Figure 4, the nodes at UT Dallas and TAMU Galveston are too close to each other to be divided into different clusters. The implications is that, in this situation, landmarks at CMU and UT Austin are not sufficient to map the network topology. Employing a Rice node as a landmark instead of CMU leads to the mapping in Figure 5, while employing a Rice node as an additional landmark leads to the mapping in Figure 3. In both these cases, the nodes are cleanly divided into 4 different clusters. The point is that the selection and the number of landmarks are important parameters for good partitioning in this method.

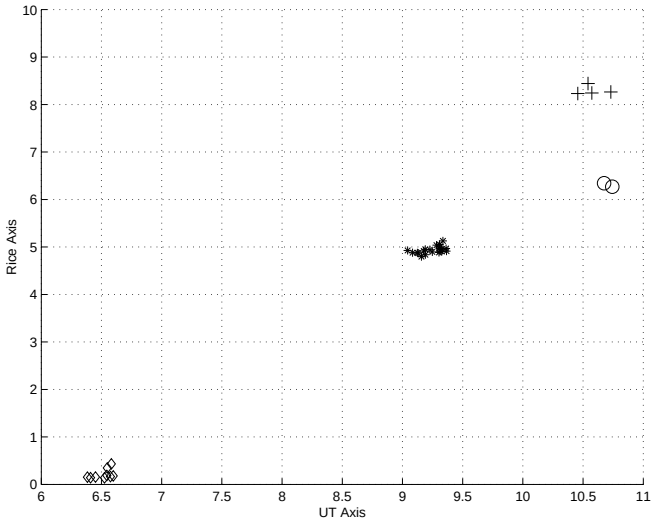

Fig. 5. Nodes from different domains distributed in 2-D space with UT and CMU as axes. (+ UT Dallas, ○ TAMU Galveston, * TAMU College Station, and ◇ Rice.)

B. Intra-domain experiment

In this experiment we use compute nodes of 4 clusters at University of Houston, that we refer to as *PGH201*, *Stokes*, *Opteron* and *Itanium*. PGH201 and Stokes are clusters belonging to 2 different research groups. Opteron and Itanium are clusters owned by the Texas Learning and Computation Center. NPACI Rocks cluster distribution is installed on all 4 clusters.

The RTTs between nodes within the same cluster and between nodes in different clusters are listed in Table III. We have represented the results also in 3D in Figure 6 to be analogous with inter-domain results, even though the methodology is different. Our clustering framework recognizes PGH201 and Stokes as separate clusters, but puts Opteron and Itanium (2 physically different clusters) in the same logical cluster. This can be observed from Figure 6. After talking with the system administrator, we discovered that Opteron and Itanium are actually within the same subnet. This means that our clustering is logically correct.

TABLE III
RTTS INSIDE AND BETWEEN 4 CLUSTERS AT UNIVERSITY OF HOUSTON (IN MILLISECONDS)

Clusters	PGH 201	Opteron	Itanium	Stokes
PGH 201	0.09	0.32	0.32	0.30
Opteron	0.25	0.09	0.09	0.50
Itanium	0.30	0.10	0.10	0.35
Stokes	0.40	0.50	0.60	0.10

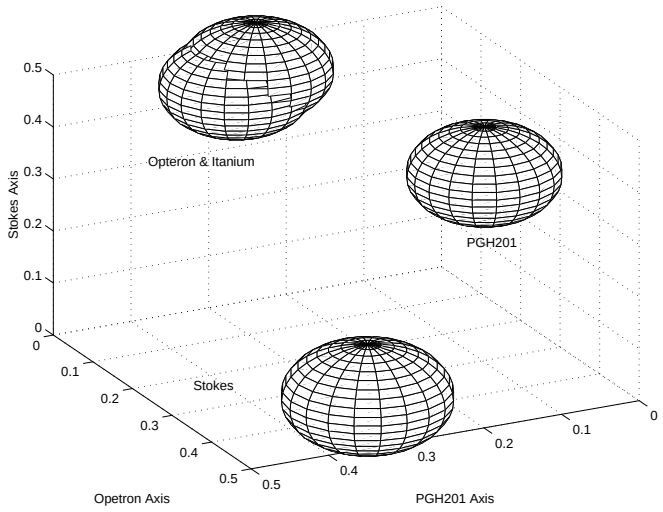

Fig. 6. Clusters inside UH domain distributed in 3-D space

VI. Discussion

A. Landmark selection

One interesting question in this research is how many landmarks are sufficient to correctly cluster compute nodes? In experiments conducted with the Iso-bar overlay distance monitoring system [4], 6, 15, 60 and 106 landmarks were selected. The results show that there was very little difference in distance prediction accuracy. Although the goals of this research are different from the Iso-bar, we use a similar Euclidean network distance as the basis for clustering. Hence, it is reasonable to assume that a small number of well selected landmarks can be sufficient.

B. Maximal clique and distance threshold

In inter-domain clustering, since the nodes belonging to the same logical cluster must be close to each other in Euclidean space, it can be argued that simply using a distance threshold to decide whether a pair of nodes is in the same cluster is sufficient. The issue is whether it is necessary to compute maximal cliques that we employ for clustering. Indeed clustering can be done effectively in most cases by simply aggregating nodes based on a distance threshold. However, the clique based procedure is necessary to generate unique disjoint clusters in all cases.

Consider the set of nodes in in Figure 7. The two cliques shown represent the intuitively "correct" clustering. However, since one pair of nodes across these clusters are close to each other in the "border" region between the clusters, a simpler partitioning could have decomposed it into 3 clusters - with one cluster consisting of the 2 border nodes. Further, the actual cluster formation would depend on which nodes were examined first. However, the maximal clique algorithm will always partition the nodes into unique clusters independent of ordering - into the 2 clusters shown in Figure 7 for this example. In practice, we have not come across such scenarios, but checking for cliques makes the procedure more consistent and robust.

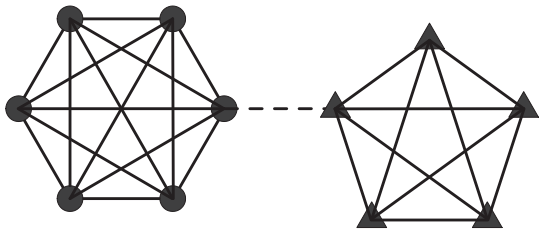

Fig. 7. Border nodes between 2 administrative domains

C. Clustering based on domain suffixes

The simplest inter-domain clustering algorithm is to put nodes with different domain name suffixes in different clusters, e.g., putting all nodes with suffix *rice.edu* in a different cluster from all nodes with suffix *uh.edu*. However this approach is not always effective. Sometimes two different domains may share the same backbone network infrastructure and their nodes should belong to the same logical cluster. Conversely, nodes belonging to the same domain name suffix can be part of different backbone networks. Our approach is not affected by these factors as it determines the logical location of nodes based on direct measurements from landmarks.

D. Scalability

While the inter-domain clustering presented in this paper is highly scalable, clustering within a domain does require latency measurements between each pair of nodes and hence scalability is of concern. However, we believe that this may not be a serious problem in practice for the following reasons:

- The number of nodes within a domain in a grid is typically within a range of 100s of nodes, an order of magnitude less than a multi-domain grid.
- Latency between nodes is small, typically under a millisecond. Hence, the overhead of latency measurement by a node with a large number of other nodes also takes a very short time. Further, we have not observed any significant reduction in accuracy when measurements are done by multiple nodes in parallel, possibly because of the small size of probe packets.
- Clustering is expected to be a relatively infrequent event.

However, a packet storm when starting measurement process on a large network may be a cause for concern. There are ways in which the measurement overhead can be reduced, e.g., by hierarchically selecting representatives for groups of nodes, and limiting some measurements to these representatives. These issues are not addressed in this work.

VII. Conclusion

A procedure to aggregate computation nodes into logical clusters, and measure the distances between these clusters, is of value to grid schedulers for selecting appropriate resources in a large-scale grid-computing system. In this paper

we present a method to hierarchically group compute nodes distributed across the internet into logical clusters, and build a distance map among these clusters. Organization of grid nodes into hierarchical clusters also helps resource monitoring and management on a grid.

In comparison with existing clustering techniques, our method is efficient, robust, scalable, and portable. One of the key features of our approach is the use of landmarks to improve the efficiency of measuring distances between grid nodes. While the concept of landmarks is not new, our contribution is applying this idea to help solve grid resource selection and resource management problem. Experiments across a set of clusters distributed across Texas validate the effectiveness of this approach to automatic clustering of grid nodes.

ACKNOWLEDGMENTS

We would like to thank our colleagues at University of Texas, Austin, Carnegie Mellon University, and Rice University, for providing access to computation nodes for experiments. We would also like to thank all members of our research group, in particular, Tsung-I Huang, Gautam Puranam and Jatindera Walia, for their contributions to this work. Finally, the anonymous referees made several suggestions that helped improve this paper.

This material is based upon work supported by the National Science Foundation under Grant No. ACI-0234328 and Grant No. CNS-0410797. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Support was also provided by University of Houston's Texas Learning and Computation Center.

REFERENCES

[1] The story of the ping program. `http://ftp.arl.mil/~mike/ping.html`.

[2] Y. Bejerano, Y. Breitbart, M. Garofalakis, and R. Rastogi. Physical topology discovery for large multi-subnet networks. In *Proc. IEEE INFOCOM*, pages 342–352, 2003.

[3] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letter*, pages 375–382, 9 (1990).

[4] Y. Chen, C. Overton, and R. H. Katz. Internet iso-bar: A scalable overlay distance monitoring system. *Journal of Computer Resource Management, Computer Measurement Group, Spring Edition*, 2002.

[5] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang. Maximum likelihood network topology identification from edge-based unicast measurements. In *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 11–20, New York, NY, USA, 2002. ACM Press.

[6] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.

[7] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. Idmaps: A global internet host distance estimation service. October 2001.

[8] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *Proc. IEEE INFOCOM*, August/September 2000.

[9] B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. *ACM SIGCOMM*, August/September 2000.

[10] B. Lowekamp, D. O'Hallaron, and T. Gross. Topology discovery for large ethernet networks. *SIGCOMM Comput. Commun. Rev.*, 31(4):237–248, 2001.

[11] T. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *Proc. IEEE INFOCOM*, New York, NY, June 2002.

[12] M. Rabbat, R. Nowak, and M. Coates. Multiple source, multiple destination network tomography. In *Proc. IEEE INFOCOM*, Hong Kong, March 2004.

[13] W. Theilmann and K. Rothermel. Dynamic distance maps of the internet. Technical Report TR-1999-08, 1999.

[14] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *J. of the American Statistical Association*, pages 365–377, 1996.

[15] R. Wolski, N. T. Spring, and J. Hayes. The network weather service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 15(5–6):757–768, 1999.

[16] A. Zeitoun. Htping. `http://www.eecs.umich.edu/~azeitoun/tools.html`.