Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

A Decentralized Approach To Topology Aware Clustering of Peer-to-Peer Systems

Chou-kai Wang, Shyh-In Hwang, and Ertai Hwang
Department of Computer Science and Engineering
Yuan Ze University
s917416@mail.yzu.edu.tw
s927418@mail.yzu.edu.tw
shyhin@cs.yzu.edu.tw

Abstract-*Various approaches of clustering based enhancement of peer-to-peer systems have resulted in significant performance boost. One of the promising techniques is termed as distributed binning. Combined with one hop lookup services, the proposed hierarchical clustering aims to provide finer clustering towards reducing routing stretch while maintaining high scalability and acceptable overhead. The clustering schemes can further be exploited in constructing a topology aware overlay structure. Such overlay satisfies not only general applications taking advantage of its decentralized and fault-tolerant nature, but also those which favors locality over distant nodes.*

1. Introduction

Many research works have been devoted to the improvement of structured peer-to-peer overlays. Different approaches assume different measurement metrics and diverging degree of trade-offs. Although the ultimate goal may be to design a low-overhead architecture with high-performance and great scalability, such idealism remains out of reach for now.

Instead of trying to construct the ultimate peer-to-peer overlay, one can improve upon current technology incrementally. One way is to utilize topology information to assist the routing algorithm in certain architectures. Some overlay networks can also be constructed with topology in mind. In other architectures topology-related information can be used to locate nearby nodes for routing decision or neighbor selection.

Incorporating topology information into structured overlays can break the uniformity of node distribution, resulting in fault tolerance and load balancing issues. However, not all kinds of peer-to-peer application strictly require globally distributed properties. These properties, as important as they are, can be relaxed to a point that achieves greater performance without breaking the system.

Thus, assuming that topology awareness is helpful, several novel ways to discover topology-related information can be exploited to further enhance the performance of current peer-to-peer networks with no loss of high scalability. This research aims to provide a way for nodes on the peer-to-peer overlay to discover topologically nearby nodes and properly form clusters in a distributed manner. Such clustering structure combined with certain peer-to-peer architecture can create a new kind of overlay for applications that wishes to exploit locality property or prefer nearby nodes to distant ones.

In the following section we briefly explore previous researches on exploiting proximity properties and clustering of network nodes. In section 3 an improved clustering architecture is proposed based on distributed binning briefly covered in the previous section. For the purpose of justifying our scheme, evaluation and simulation results are outlined in section 4. Finally in section 5, we conclude this research and present possibilities for future works.

2. Locality Aware techniques

Utilizing locality property is known to be an effective way of enhancing the performance of structured peer-to-peer systems due to their nature of distributing nodes uniformly without regard for physical network structure. Some of the researches related to this topic are outlined below.

2.1. Topology Aware Overlay Construction

In a version of CAN, each node measures its distance to a set of landmark nodes to position itself on the Internet. Mapping of such relative positions to coordinates in the *d*-dimensional coordinate space constitute a topology aware overlay. This resembles in many ways the GLS [30], which also binds node distribution to geological location.

Grouping nearby nodes and forming a hierarchical system is another popular approach adopted by [11], [10], [16], [18] and [14]. In the architecture proposed by [16], peer-to-peer overlays is constructed in a hierarchical and tree-like form. Most effort in this research has been put on the routing issues among groups. The concept of hiding the dynamics of transient nodes behind each group indeed reduces routing latency and maintenance overhead. Furthermore, individual groups in the hierarchy do not need to use the same architecture and topology aware construction is also an option rather than a must.

A cluster based overlay construction in [11] also takes advantage of topology information. A central server is set up to perform network-aware clustering and node registration. Clustering can be done as easily as grouping nodes according to their IP addresses. For further performance enhancement, in each group the clustering server assigns some *delegate nodes* to reduce load on the central server and providing local cache functionality. This hybrid approach although may not look pretty from the eyes of the faculty, might be an acceptable comprise.

A topology-centric approach is taken by [14] that drastically shifts the paradigm to the other end. Unlike other peer-to-peer overlays which construct with no consideration of physical topology and then incorporating topology aware ideas into their own structure, the authors took an extreme apparoach and thought only about topology. Inspired by [12], BGP tables are retrieved from Internet core routers to specifically identify network structures. With such information at hand, topology specific clustering can be done in a precise manner. Unfortunately, not all part of network structure is managed via BGP. NAT boxes and internal networks hiding thousands of nodes behind them also poses problems. Nonetheless, as the authors claim, this extreme approach can at the very least serve as a lower bound of all other topology aware clustering techniques.

2.2. Clustering of Nodes

As can be seen in section 2.1, clustering of nearby nodes in the peer-to-peer network seems to be a popular design approach. Clustering similar data or nodes in a traditional sense has been explored in great detail in the data-mining area. Scenarios such as given n^2 distance matrix or exact feature-vector of n nodes, finding the optimal clustering are fully explored problems. Although such optimal clustering even with global information, is a NP-Hard problem, approximates are still possible. However, these algorithms are centralized in nature and not suitable for our research. Utilization of [6] and [7] in a naive way often result in a centralized clustering scheme.

Ratnasamy's work [18] proposed another rather simple technique to properly cluster all nodes in the system. Each node measures RTT values to certain well-known landmark nodes, simply by using ICMP ping, and uses these M values as an ordering to join one of the M! groups. This is called ***distributed binning***. Although this method already over-clusters the systems in that with only 10 landmarks we can have as much as 3628800 nodes, the authors proposed further fine grained clustering using the set of RTT values as a ***level vector*** in M-dimensional space. Only nodes with the same ordering of M RTT values and same ***level vector*** can be in the same group.

This scheme may seem fine at first glance, but the high number of clusters one can have with as low as 10 landmarks makes us skeptical about the resulting accuracy. Because transient variance on RTT values can yield very different results. In order to improve RTT measurement accuracy, apparently pings have to be placed carefully and more frequently as one might desire. But current Internet has not been so friendly as it used to be, differentiating from malicious DDoS attack to RTT measurement is a daunting task for any administrator and difficult for IDS instruments to handle.

3. Architecture

In this section, we present our hierarchical architecture for topology aware clustering in the peer-to-peer systems. In section 3.1, the idea of hierarchical clustering is presented. Such structure raises issues discussed in the same section. In section 3.2, a global view of our architecture and definitions of algorithms are presented. Finally in section 3.3, we explore an application of our clustering scheme: a topology inherent peer-to-peer architecture.

3.1. Hierarchical Clustering

Clustering on the basis of distributed binning is suspected with accuracy problems as explained previously. Another obvious approach is through recursive refinement. The basic idea is to perform more than one pass of binning to a set of nodes. First a major scale clustering using distributed binning take place in the global scale, possibly forming continental scale groups. Then each individual group perform further clustering, possibly forming groups in the national scale. If condition calls for another clustering in a smaller scale, further clustering will be performed recursively.

To performing such recursive clustering in a dynamic clustering scheme, we must explore the number of necessary levels of this hierarchy and choices of landmarks in each level to achieve optimal clustering.

3.1.1. Landmark Selection

Good landmark selection is critical for distributed binning to succeed. In a fixed hierarchy we have the luxury of analyzing the topology of underlying networks and make smart selection of landmarks. So, at the first level of clustering, traditional selection of some well-known and fixed landmarks is a viable solution. Only at further levels of clustering do we need dynamic selection.

The challenge we face is to select, at these further levels, good landmarks having the following properties: stable, well-connected and distributed. An unstable landmark can seriously interfere with the binning process. Well-connected means on normal operations a landmark can respond well to other nodes' ping requests. Nodes behind NAT networks or firewalls blocking ICMP traffic does not qualify as well-connected nodes.

The distributed property are easier to satisfy. In a DHT based system, node identifiers are chosen in uniformly distributed manner without regard for physical topology. Choosing landmarks based on some fixed identifier can actually present good landmark candidates. For example, slice leaders discussed in [17] can be good landmarks satisfying this property. Of course, more efforts can be made to make sure that we have well behaving landmarks. Minimum physical hop count between each landmark and degree can all be taken into consideration in the selection process.

Furthermore, in the one-hop lookup services defined in [17], a form of central control without single points of failure is performed well by these slice/unit leaders. With this centralized control, the election and notification of landmarks to all other nodes in the system can be greatly simplified. If we exercise this one-hop lookup inside a group consisting of nearby nodes, not only the maintenance overhead introduced by one-hop lookup is reduced to local scale, but also do we greatly reduce routing stretch. The latter effect comes from the fact that the source node already knows the IP address of destination node in the same group. If our scheme is applied on applications in which intra-group communication occurs more frequently, the overall performance can be greatly enhanced.

3.2. Algorithms and Definition

In this section, we describe our system architecture as a whole and define algorithms and terms.

First, recursive binning is defined as follows.

R-binning: abbreviation of recursive binning, a scheme to recursively refine clustering using multiple levels of distributed binning.

Recursive Binning Algorithm

1. Let *C* be a set of all nodes.
2. If *C* satisfies ***ending condition,*** then terminate.
3. Partition *C* according to ***distributed binning***. We have the resulting partition: $P = \{ g_1, g_2, g_3, ..., g_m \}$
4. For all g_i in *P*, let $C = g_i$ and perform this algorithm starting from step 2.

Define Ending Condition:

Let *C* be a set of nodes, if the size of *C* **does not** exceeds ***m***, then *C* satisfies ending condition.

This algorithm of ***recursive binning*** serves the purpose of progressively refining the clustering until a some condition are met and each node belongs to a unique group. ***Ending condition*** is defined to limit the size of each group. The reason for this limitation is to prevent overly refined clustering and to limit the size of a single bin to not overload one-hop lookup. However, it may not be the only rule to determine the termination of recursive binning. Other mechanisms to trigger and terminate further clustering can be developed as future works.

In the leaf groups resulted from clustering, a version of one-hop lookup in [17] is adopted to form a structured overlay among all nodes in the same group. This one-hop lookup maintains node membership only at leader nodes. Other regular nodes only receive information regarding available slice leaders and landmark nodes. This modification further reduces the overhead of one-hop lookup, although maintenance of membership of all nodes can also reach regular nodes as an option.

3.3. Topology Inherent P2P

Some peer-to-peer overlays ([3], [16]) already take into consideration many possible variations integrated with topology. In [3] the partition of ***d***-dimensional identifier space can be mapped to a topological partition of nodes. In [16] the hierarchy of groups can be topologically or geographically structured.

Here we present our scheme of utilizing topological awareness in Chord, an unstructured peer-to-peer overlay network. The main idea is to incorporate the r-binning in the generation of node identifiers such that nearby nodes on identifier space are topologically nearby. However, the original routing mechanism in Chord is not modified.

To facilitate the discussion, we assume a node identifier to be a 160-bit number divided into the prefixing bin identifier and the remaining suffixes. The remaining suffixes can be a randomly generated number or the IP address of the node. To determine the bin identifier, the landmark identifiers and measured RTT values in the binning process are hashed to a 128-bit number. This in effect produces unique bin identifier for each bin. MD5 is sufficient as a hash function in this method. Figure 16 provides an illustration.

A problem with the above identifier generation is that it only guarantees that nodes within the same bin are placed in nearby locations on the identifier space. Nearby bins are not actually nearby groups of nodes on the identifier space. To get around this problem, the identifier can be seen as a m-digit number in which higher digits denote higher levels of binning. Assuming 4 landmarks in each level of binning, we assign a 2-bit identifier to represent each landmark. Thus the sorted landmarks according to measured RTT values becomes a 8-bit integer in which every 2 bits represent a landmark. N levels of binning will produce N 8-bit integers forming the bin identifier. Stuffing the bin identifier to the higher 8*N bits of the node identifier and randomly assigning the rest of the lower bits produces a node identifier. A typical case can be that 128 bits of bin identifier allows for 16 levels of recursion. In general 8*N bits of bin identifier is generated by N levels of recursion, and the remaining bits are randomly assigned.

These schemes may seem straight-forward when only one level of clustering is performed. With our r-binning scheme, a node can experience multiple levels of clustering and in each round of binning produces a new identifier. Key migration in this scenario can pose serious problems to the stability and scalability of the system. However, networks grow slowly enough for further clustering to take place if we choose our ***ending condition*** carefully. In this perspective, key migration once in a while can be seen as an acceptable overhead. Furthermore, we can define the first m levels of r-binning as a must to cope with the fast-growing network in the early days of deployment.

As for one hop lookup, only slight modifications are needed. In [17] slice and unit leaders are chosen from some fixed positions on the identifier space. In our schemes these leaders are chosen from fixed positions on each portion of identifier space to which the bin belongs. Because in the same bin (same bin identifier) the node identifiers are randomly generated, these leaders have the same distributed properties as in [17].

Because of the simplicity of the above schemes, they can be adapted to other peer-to-peer overlays with minimal modification. Especially those architectures in which a node first route requests to those with similar identifiers, topology-inherent properties can greatly reduces latency stretch and provide more efficient routing.

4. Simulation

The variety of peer-to-peer architectures have already called for measurement, modeling and analysis methodologies. Some researches develop metrics for dynamic measurement ([24], [27]), some explore models for simulating real world peer-to-peer overlays ([26], [27]), and others analyze characteristics inherent in such decentralized networks ([25]).

Because what we explore in this work is mainly the topology structure of underlying IP networks, modeling of such structure is of first priority. In the works of [21], a transit-stub model is proposed that more accurately models Internet than traditional random graph or power law models. Many peer-to-peer works also give high credit to this model. Therefore we adopt the transit-stub model as a foundation for evaluation of our clustering scheme. To measure path latencies each edge on the graph need to be assigned a latency. The method in [21] is adopted here which assigns 20ms of latency for inter-transit latency, 2ms for inter-stub latency, and 5ms for stub-transit latency.

Another topology generator called inet [32] is also very useful in many researches works. Inet is based on the power laws to approach the Internet topology. What differentiates inet from other power-law random graphs is that it also deploys other techniques to achieve enhanced similarity to the Internet regarding many metrics. Unfortunately, like other power-law based topologies, clustering coefficient in inet is quite low meaning that the performance of clustering is not significant. For this reason we only use inet topologies in some of the simulations for comparison purpose only. As to the assignment of latency in the inet topology, such power-law random graphs can only rely on random assignment. We assign a random latency between 10ms and 100ms for each edge.

In our simulations of r-binning, number of landmarks for each level is set to 4 and the miximum recursion level is 20. In the case of d-binning, number of landmarks is chosen to be the best-performing number in that particular topology. Landmarks in both schemes are randomly chosen for simplicity with acceptable performance.

In figure 1 and 2, gain ratios for both scheme are compared. As is obvious, r-binning performs well as anticipated and is insensitive to the number of landmarks. It is also observed that level vector has the potential of improving d-binning, but such fine grained clustering based only one level of binning is not entirely reliable.

In figure 3 r-binning is performed with varying limits on the maximum level of recursive refinement. It is reasonable that as more levels are permitted gain raio also increases as well. This result is shown here to demonstrate that with small number of levels of refinement the performance of clustering is promising enough to justify such scheme. It is not intended to tune performance out of this parameter.

Finally, simulation results of latency stretch are presented in figure 4. Only transit-stub topologies are considered because the low clustering coefficient of inet topology does not provide enough enhancement to justify applying r-binning to Chord architecture. In this simulation, peer-to-peer nodes are randomly distributed to every AS in transit-stub topology. Chord construction without regard for topology are marked as random on the figure to provide base line comparisons.

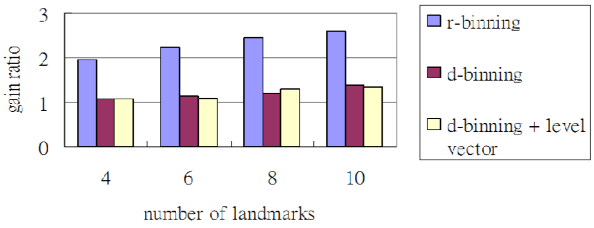

Figure 1: gain ratio comparison with inet 10K nodes

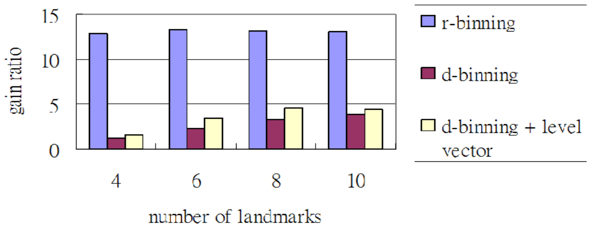

Figure 2: gain ratio comparison with TS 10K nodes

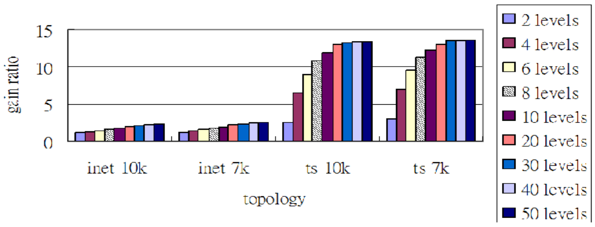

Figure 3: r-binning with limits on recursion level

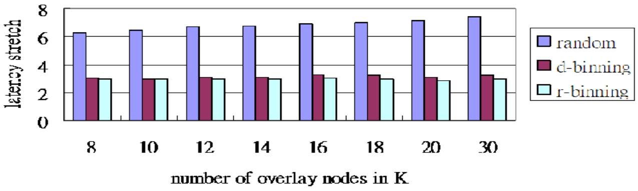

Figure 4: latency stretch for Chord, topology TS3K

5. Conclusion

The research presented here aims to utilize topology information in a distributed way as an enhancement of peer-to-peer overlay construction. These kinds of topology-sensitive overlays often provide performance boost against the original design with little sacrifice on overhead and fault-tolerance. In Ratnasamy's research, much performance gain is achieved using distributed-binning. Recursive-binning (r-binning) is presented here to further refine network clustering by recursively applying distributed-binning. One-hop lookup proposed by Gupta [17] aids the selection of landmarks during the r-binning process and further reduces latency stretch within same bin down to one with acceptable maintenance overhead.

Although this research focus on the Chord overlay, it is not limited to this single architecture. Any peer-to-peer overlay that routes message to nearby nodes on the identifier space can benefit from this enhancement. Applying r-binning on the overlay construction of other structured or unstructured peer-to-peer architecture makes an interesting research topic in the near future. Applications layered on top of the peer-to-peer network such as storage and group communication can also take advantage of topology properties and the improvements can be measured to further justify our scheme. Last but not least, improvements on identifier generation based on topology can be made to further satisfy the distributed property than our current scheme.

6. References

[1] Ion Stoica, Robert Morris, David Liben-Nowell, David Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, in Proceedings of ACM SIGCOMM, Aug 2001.

[2] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph, “Tapestry: An Infrastructure for fault-resilient wide-area location and routing”, Technical Report UCB/CSD-001-1104, University of California, April 2001.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. “A Scalable Content-addressable Network”, in Proc. Of ACM SIGCOMM, Aug 2001.

[4] The Gnutella protocol specification, 2000. http://dss.clip2.com/GnutellaProtocol104.pdf .

[5] A. Rowstron, and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”, in International Conference on Distributed Systems Platforms(Middleware), Nov. 2001.

[6] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. Gryniewicz, Y. Jin, “An Architecture for a Global Internet Host Distance Estimation Service”, in IEEE INFOCOM , 1999.

[7] T. S. Eugene Ng, Hui Zhang, “Towards Global Network Positioning”, in Proceedings of ACM SIGCOMM Internet Measurement Workshop.

[8] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose, “Modeling TCP Throughput: A Simple Model and its Empirical Validation”, in Proceedings of the ACM SIGCOMM ’98 conference on Applications, technologies, architectures and protocols for computer communication.

[9] Fred Von Lohmann, “Peer-to-Peer File Sharing and Copyright Law: A Primer for Developers”, in Second International Workshop on Peer-to-peer Systems(IPTPS ‘03).

[10] L. Ramaswamy, B. Gedik, L. Liu, “Connectivity Based Node Clustering in Decentralized Peer-to-Peer Networks“, in Third International Conference on Peer-to-peer computing(P2P ’03).

[11] B. Krishnamurthy, J. Wang, Y. Xie, “Early Measurements of a Cluster-based Architecture for P2P Systems”, in ACM SIGCOMM Internet Measurement Workshop, Nov. 2001

[12] B. Krishnamurthy, J. Wang, “On Network Aware Clustering of Web Clients”, in ACM SIGCOMM, 2000.

[13] M. J. Freedman, D. Mazieres“, Sloppy Hashing and Self-organizing Clusters,” in Second International Workshop on Peer-to-peer Systems(IPTPS ‘03).

[14] K. W. Ross, E. W. Biersack, P. Felber, L. Garces-Erice, G. Urvoy-Keller, “Topology Centric Lookup Service“, in Proceedings of COST264 Fifth International Workshop on Networked Group Communications (NGC)

[15] M. Castro, P. Druschel, Y. C. Hu, A. Rowstron, “Exploiting Network Proximity in Peer-to-peer Networks”, in Proceedings of the International Workshop on Future Directions in Distributed Computing(FuDiCo 2002).

[16] L. Garces-Erice, E. W. Biersack, P. A. Felber, K. W. Ross, and G. Urvoy-Keller, “Hierarchical Peer-to-peer Systems”, in Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par).

[17] A. Gupta, B. Liskov, R. Rodrigues, “One Hop Lookups for Peer-to-peer Overlays”, in the 9th Workshop on Hot Topics in Operating Systems(Hot OSIX).

[18] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, “Topologically Aware Overlay Construction and Server Selection,“, in Proceedings of IEEE INFOCOM'02.

[19] M. Castro, P. Druschel, Y. C. Liu, A. Rowstron, “Topology Aware Routing in Structured Peer-to-peer Overlay Networks”, Tech. Rep. MSR-TR-2002-82, Microsoft Research.

[20] J. Ritter, “Why Gnutella Can’t Scale. No, Really.” http://www.darkbridge.com/~jpr5/doc/gnutella.html, Feb. 2001.

[21] E. Zegura, K. Calvert, and S. Bhattacharjee, “How to Model an Internetwork”, in Proceedings of IEEE INFOCOM ’96.

[22] KaZaA, http://www.kazaa.com .

[23] eDonkey, http://www.edonkey.com .

[24] S. Sariou, P. Krishna Gummadi, S. D. Gribble, “A Measurement Study of Peer-to-Peer File Sharing Systems, “ in Proceedings of Multimedia Computing and Networking 2002 (MMCN '02).

[25] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroju, Steven D. Gribble, Henry M. Levy, John Zohorjan, “Measurement, Modeling, and Analysis of a Peer-to-Peer File Sharing Workload, ” in Proceedings of Multimedia Computing and Networking 2002 (MMCN '02).

[26] M. Jovanovic, F.S. Annexstein, K.A. Berman, “Modeling Peer-to-Peer Network Topologies through Small-World Models and Power Laws, “ in the Proceedings of IX Telecommunications Forum(TELFOR 2001).

[27] Mario T. Schlosser, Tyson E. Condie, Sepandar D. Kamvar, “Simulating Peer-to-Peer File Sharing Network, “ in the 1st Workshop on Semantics in Peer-to-Peer and Grid Computing.

[28] ns, http://www.isi.edu/nsnam/ns/ .

[29] R. Kurmanowytsch, M. Jazayeri, E. Kirda, “Towards a Hierarchical, Semantic Peer-to-Peer Topology, ” in Second International Conference on Peer-to-Peer Computing.

[30] J. Li, J. Jannotti, D. De Couto, D. Karger, R. Morris, “A Scalable Location Service for Geographic Ad Hoc Routing, ” In Proceedings of the 6th ACM International Conference on Mobile Computing and Networking.

[31] J. Wang, “Gnutella Bandwidth Usage, “ http://resnet.utexas.edu/trouble/p2p-gnutella.html .

[32] C. Jin, Q. Chen, and S. Jamin, "Inet topology generator," Technical Report CSE-TR-433-00, EECS Department, University of Michigan, 2000.