ISSN:0975-9646

Sivanthiya T et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2325-2328

Creation of Replica Path Using Maxdisjoint in Tree Based Routing

Sivanthiya T,Ramyapreethi S,Geethalakshmi P,Jagadeesh Mailu Dheevambiga P

ME-Students, Department of CSE, Muthayammal Engineering College.

Abstract-The peer-to-peer network (p2p) consist of participating nodes and these nodes are interconnected. Based on the linkage between the nodes p2p is classified as structured or unstructured p2p networks. Using DHTs in structured networks provide lookup service similar to hash table. It can be efficiently retrieve the values associated with the given key, but sometimes it can deny the access to keys. So Tree Based Routing DHTs are used in the unstructured networks to avoid this kind of attack. In unstructured network there are trusted and selfish node, if the particular node selfishes to send the data to the other node then the server creates the replica path using Tree Based Routing DHTs and it defines MAXDISJOINT. The maximum disjoint routes are created using MAXDISJOINT, here the sender can view the nodes from d to d-1,where d is desired number of disjoint routes. Hence sending and receiving of data is done at the low response time without producing too high a load. Finally nodes can send the information with less disturbances and routing security is considerably improved.

Index Terms-Peer-to-Peer Network,Distributed hash table,MAXDISJOINT,Disjoint routes.

1 Introduction

Peer-to-peer (P2P) networking is a distributed application architecture that partitions tasks or to a particular array slot. This system often implement an abstract ***overlay network. This network*** consists of all the participating peers as network nodes. There are links between any two nodes that know each other: i.e. if a participating peer knows the location of another peer in the P2P network, then there is a directed edge from the source to the destination in the overlay network. Based on how the nodes in the overlay network are linked to each other, the P2P networks are classified as structured or unstructured. In ***structured P2P networks***, peers are organized by specific criteria and algorithms. Structured P2P networks employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired file/resource, even if the resource is extremely rare. But these are sensitive to extremely transient join/leave/failure behavior of peers. ***Unstructured P2P networks*** do not impose any structure on the overlay networks. Peers in these networks connect in an ad-hoc fashion based on a loose set of rules. Ideally, unstructured P2P systems would have absolutely no centralized elements/nodes. Hence they are highly resilient to peer failures. Usage of Distributed hash tables (DHTs) provide a lookup service similar to a hash table (key, value), any participating node can efficiently retrieve the value associated with a given key. Sometime nodes can deny access to keys or misroute lookups. Hence Tree based DHT is used were disjoint routes in DHTs of this type. Here the number and placement of replicas necessary to produce d disjoint routes from any source node to the replica set is prescribed. With this scheme, tolerate d _ 1 malicious peers, whether they are attacking the storage and retrieval of data items or the routing infrastructure. Goal of this project is to replacing the DHT (Distributing Hash Table) by MAXDISJOINT, and creation of replica path by using this MAXDISJOINT is done and send the preferred data using replica path when the selected node being selfish to send the data to the destination. Existing System DHTs are inherently less resilient because of the usage of Equivalence Based Routing Scheme. Distributed hash tables share storage and routing responsibility among all nodes in a peer-to-peer network. These Networks have bounded path length unlike unstructured networks. Hence nodes can deny access to keys or misroute lookups. When deployed over the Internet, DHTs may be impacted by the failure or compromise of peers in the overlay and performance guarantees no longer hold. In fact, it may not be possible to fetch a desired object at all. Hence Existing System is not efficient to detect attack such as storage and retrieval attacks because many p2p networks allow nodes to join without prejudice, leaving the network vulnerable to attack. This paper realizes the benefits of replication and route diversity in concert through replica placement. Consider a class of DHTs that route messages using a scheme is called tree-based routing. There exists a replica placement called as MAXDISJOINT that creates disjoint routes in DHTs of this type. This creates the impact of MAXDISJOINT on routing robustness compared to other placements when nodes are compromised at random or in a contiguous run. So in the proposed system the sending and receiving of data is done at a low response time without producing too high a load. Nodes can send the information without any disturbance and routing security is considerably improved.

2 Related Work

To place our work in context, we discuss related work on replica placement, peer-to-peer routing security, and general peer-to-peer security issues.

2.1 Replica Placement

Replica placement has long been studied in the realm of distributed computing. Many studies have compared the performance of different placement schemes in terms of quality of service, availability, and time to recovery in different types of serverless systems [6], [9]. The first DHT-based replication schemes were only concerned with availability and thus local replication, i.e., replicas placed close to the master copy in the ID space. As detailed herein, such placements have very little routing robustness. A very important paper, which proposed the first deterministic nonlocal replica placement scheme for DHT-based systems, was that of Ghodsi et al. [12]. To realize benefits of this approach for routing security, it is therefore necessary to instantiate particular schemes for different DHTs or classes of DHTs and evaluate their routing characteristics. This is exactly the problem considered in this paper. The instantiation of symmetric replication that was presented in [12] was equally spaced replication.

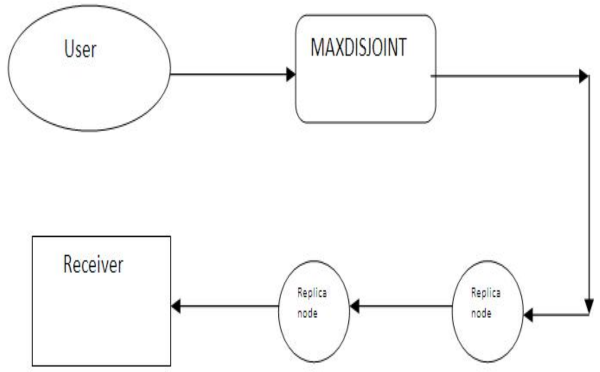

Fig.1.Data flow replica placement

The paper contained a thorough evaluation, which showed that the technique reduces the message overhead in node joins and leaves, provides better load balance, and improves fault tolerance. However, routing robustness was not considered. It is worth noting that the MAXDISJOINT placement is equivalent to equally spaced replication in Chord. In other DHT implementations, however, MAXDISJOINT provides added flexibility in terms of tuning routing robustness that equally spaced replication does not provide.

3 Distributed Hash Tables With Tree-Based Routing

The geometry impacts neighbor and route selection, which can have an impact on flexibility, resilience, and proximity performance as studied in. Although we use the term "tree-based routing," we are not referring to the geometry, but the routing algorithm.

3.1 Tree-Based Routing DHTs

Consider a DHT with an ordered id space I with size N=|I| and a branching factor B such that logB N is integral. The branching factor is used by each node to construct its routing table. We define a DHT constructed in this manner to be a tree-based routing DHT. If the paths from any source node to all possible destinations are aggregated, the resulting topology is a tree, which is how tree-based routing gets its name.

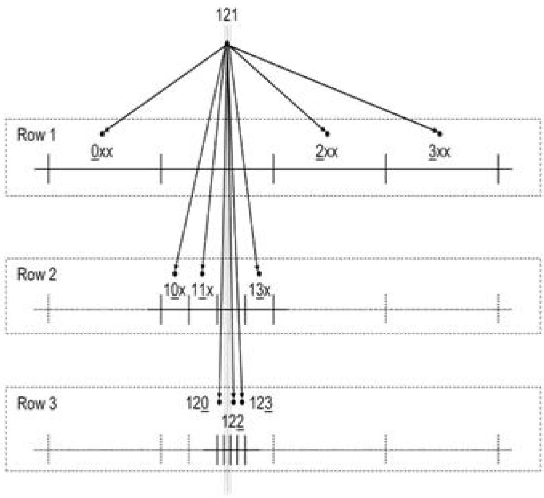

Fig. 2. Pastry routing table structure for the hypothetical node 121(N= 64; B = 4).The highlighted region marks the segment to which node 121 belongs.

First, tree-based routing DHTs are deterministic; that is, given a message destined for a node d, each node has one and only one routing table entry through which the message can be forwarded to d.

3.2 Creating Disjoint Routes with Tree-Based Routing

Determinism and routing convergence provide a natural avenue for creating disjoint routes. Routing convergence guarantees that once a path enters a segment of the id space, it will never proceed to a node that is outside of that segment. If two paths can be created originating in different segments, then the paths are guaranteed to be disjoint. Furthermore, the determinism property ensures that any two routing table entries will route to different segments. Therefore, we can create disjoint routes simply by routing through different routing table entries.

4 Definition Of Maxdisjoint Placement

MAXDISJOINT assigns each replica an identifier, which is used to determine its placement. The placement algorithm takes as input N, the identifier space size of the DHT; B, the branching factor; and d, the desired number of disjoint routes. As described in the replica placement algorithm, in each round replicas are placed starting at the master key and working in the direction of increasing identifiers. The algorithm is presented as such for its simplicity. However, the steps within each round can be performed in any order. Each step is functionally equivalent to the others in its round. Therefore, a real implementation may reorder the steps in each round to distribute the replicas more uniformly across the identifier space. This will help to provide load balance and tolerate runs of contiguous failed nodes.

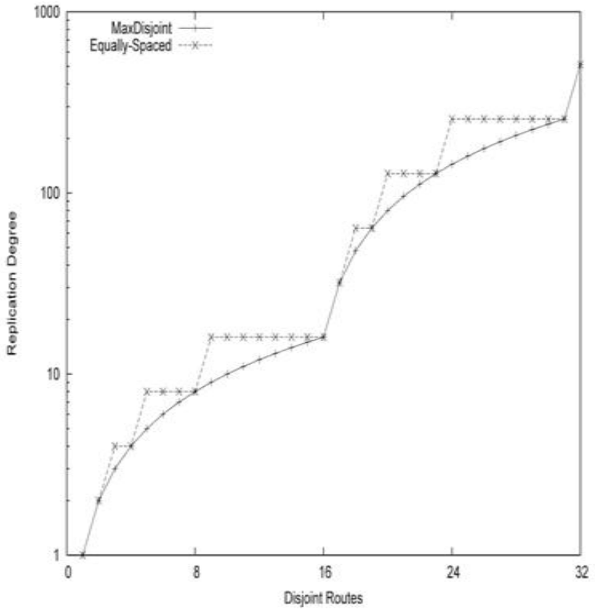

Fig. 3.Replication degree for increasing numbers of disjoint routes (B = 16).

Doubling the replication degree avoids the cost of shifting existing replicas, but may come with the added burden of storing an excessive number of replicas.

5 EXPERIMENTS

It is not efficient to detect attack such as storage and retrieval attacks because many p2p networks allow nodes to join without prejudice, leaving the network vulnerable to attack. Miscellaneous attacks, which target other aspects of the system, such as admission control or the underlying network routing service.The first class of attack is commonly addressed with replication. Objects are replicated at several peers in the network to increase the likelihood that there will be a correct replica available. The benefits of replication on load balancing and overall performance have also been studied. The first work that considers how the placement of replicas affects object reachability through the routing infrastructure. The benefits of replication and route diversity in concert through replica placement. In this paper, a class of DHTs that route messages using a scheme is called as tree-based routing. The number and placement of replicas necessary to produce d disjoint routes from any source node to the replica set. With this scheme, it is able to tolerate d _ 1 malicious peers, whether they are attacking the storage and retrieval of data items or the routing infrastructure.Therfore the input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. The system Implementation is to uniquely identify each replica, A key identifier pair (k, v) is used, where k is the key identifier and v is virtual key identifier. For each replica, v gives the location of the master key. By definition, the master key k is denoted by the pair (k,k). Here ordered pair is required because two data items may have replicas that reside on the same peer. When a key k is inserted into the DHT with replication degree d, the key identifier pairs is computed for each replica: (k, k) (k1, k) (k2, k) . . . (kd_1, k). Once the key identifier pair for each replica is computed, the DHT key insertion mechanism is used to insert the replicas. That is, lookup for each key identifier is performed. Next, the DHT lookup primitive must be modified to accommodate the new replication scheme. When a peer is queried for a key, the query node must compute the locations of all replicas. The key identifier is used to route to the replicas. Once a replica's home node is found, the key identifier pairs are compared to return the appropriate replica. It is worth noting that no additional routing table entries are required to route to the replicas. In addition, if the query node dispatches the lookups for the entire replica set simultaneously, there may be an improvement in performance because the query node can return the first correct response received (which may have returned along a route shorter than the route to the master key). However, if the added load of the extra lookups puts strain on the system, the performance may improve only slightly or even degrade. This hypothesis is evaluated through experimentation. Finally, when peers join or leave the DHT, the DHT join and leave mechanisms can be used by simply ignoring the virtual peer identifiers in each key identifier pair. Note that DHT replica placement schemes that place replicas in the neighborhood of the master key require modification to the node join and leave mechanisms. To maintain the replication degree, replicas will need to be shifted for every join or leave. Ghodsi et al. [12] discuss the effect of churn on symmetric (equally spaced, MAXDISJOINT) replication and show that only Oð1Þ messages are needed to maintain the replication degree for every join or leave messages for a successor-list (neighbor set) placement, where r is the replication degree.

6 CONCLUSION

In this paper, a class of DHTs, which employ a tree-based routing scheme, that for every DHT of this class there exists a replica placement that can create a provable number of disjoint routes.To defined MAXDISJOINT, a replica placement that creates disjoint routes in a full distributed hash table that employs a tree based routing scheme. Through simulation, this placement creates disjoint routes effectively in DHTs that are sparsely populated. In addition, MAXDISJOINT creates disjoint routes without modification of the underlying routing scheme; therefore, its implementation is independent of the underlying DHT chosen, provided that the underlying DHT employs tree-based routing. Furthermore, to demonstrated that disjoint routes have a positive impact on routing robustness and the probability of routing success when nodes are compromised at random or in runs. Specifically, to showed that a vast majority of queries can be resolved successfully even with a quarter of nodes compromised.

REFERENCES

[1] J. Aikat, J. Kaur, F.D. Smith, and K. Jeffay, "Variability in TCP Round-Trip Times," Proc. ACM SIGCOMM Internet Measurement Conf.(IMC '03), pp. 279-284, 2003.

[2] M.S. Artigas, P.G. Lopez, and A.F.G. Skarmeta, "A Novel Methodology for Constructing Secure Multipath Overlays," IEEE Internet Computing, vol. 9, no. 6, pp. 50-57, Nov./Dec. 2005.

[3] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure Routing for Structured Peer-to-Peer Overlay Networks," Proc. Symp. Operating Systems Design and Implementation (OSDI '02), pp. 299-314, 2002.

[4] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," Proc. Symp. Operating Systems Design and Implementation (OSDI '99), pp. 173-186, 1999.

[5] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-Bandwidth Multicast in Cooperative Environments," Proc. ACM Symp. Operating Systems Principles (SOSP '03), pp.298-313, 2003.

[6] Y. Chen, R.H. Katz, and J. Kubiatowicz, "Dynamic Replica Placement for Scalable Content Delivery," Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS '02), pp. 306-318, 2002.

[7] F. Dabek, M. Kaashoek, D. Karger, R.Morris, and I. Stoica, "Wide Area Cooperative Storage with CFS," Proc. ACM Symp. Operating Systems Principles (SOSP '01), pp. 202-215, 2001.

[8] J.R. Douceur, "The Sybil Attack," Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS '02), pp. 251-260, 2002.