

Stable High-Capacity One-Hop Distributed Hash Tables

John Risson¹, Aaron Harwood², Tim Moors³

^{1,3}University of New South Wales, ²University of Melbourne
jr@tuffit.com, a.harwood@cs.mu.oz.au, t.moors@unsw.edu.au

Abstract

Most research on Distributed Hash Tables (DHTs) assumes ephemeral, lightly loaded deployments. Each node has a lifetime of a few hours and initiates a lookup once every few seconds or minutes. However, in giant internet data centers, each node has a lifetime of weeks or months and initiates hundreds or thousands of lookups every second. In such an environment, one-hop DHTs are superior to multi-hop DHTs. They use lookup bandwidth more efficiently. We qualify conflicting research to show that a single one-hop DHT can indeed scale to at least a few hundred thousand nodes in stable, high-capacity enterprise networks. Two new designs are presented: One Hop Sites (1HS), a high-capacity DHT tailored for site redundancy; and the One Hop Federation (1HF), a global, hierarchic DHT that resolves an open latency problem. For both, the analysis a) confirms linear scalability to at least a few hundred thousand nodes and b) identifies the most sensitive design parameters.

1. Introduction:

The Two Faces of Distributed Hash Tables

There are two very distinct strands of research on Distributed Hash Tables (DHTs). One consists of multi-hop, peer-to-peer (P2P) DHTs. Five years ago, influential DHTs [1-4] were motivated by peer-to-peer file-sharing applications like Freenet, Gnutella and Napster. A lookup needs several hops to find a key, since each node knows only a handful of the nodes in the DHT. It is not practical for each node to maintain full routing tables if there are many millions of highly transient nodes. This research stream has generated dozens of DHT designs and hundreds of papers. It is unclear whether there have been any industrial deployments, but many interesting applications are in daily use and under development [5].

This paper refers to the other stream as the one-hop DHTs. Strictly, they may use a few hops. This research goes back at least to 1993 [6-9]. Early designs were

tailored for stable, local clusters. A few, recent one-hop designs have made peer-to-peer assumptions – transient nodes and relatively low lookup rates per node [10, 11].

The one-hop DHTs are superior to multi-hop DHTs when nodes have lifetimes measured in weeks or months and when there are hundreds or thousands of lookups per second per node. Given that Amazon and Yahoo deployments have demonstrated the scalability and usefulness of DHTs in stable, high-capacity networks [12], one-hop DHTs deserve research scrutiny.

In Sect. 2, we compare the multi-hop and one-hop DHTs to ask “Why did we forget enterprise DHTs?” In Sect. 3, we revisit the conflicting research on the number of nodes that can be supported by a single, one-hop DHT. This research had made P2P assumptions. Our comparison extends to stable, high-capacity networks to show that one-hop DHTs use lookup bandwidth more efficiently than multi-hop DHTs. Sect. 4 describes One Hop Sites (1HS), a DHT design for large internet data centers needing site redundancy. Sect. 5 describes the One Hop Federation (1HF), a global hierarchic DHT that avoids the risk of unnecessary, high-latency hops at the top of the hierarchy. Tanenbaum and van Steen had flagged the issue as an open research problem [13]. For both designs, our analysis a) confirms linear scalability to at least a few hundred thousand nodes and b) identifies the most sensitive design parameters. Sect. 6 concludes.

2. Why did we forget Enterprise DHTs?

For the last five years, DHT research has been preoccupied with peer-to-peer (P2P) file-sharing networks. Such networks are characterized by short node lifetimes and low key lookup rates per node. Research has partially neglected DHT origins and deployments in stable, high-capacity enterprise networks.

The P2P emphasis is clear in seminal DHTs, though it was sometimes asserted that they are also suitable for stable, high-capacity networks. Chord was explicitly motivated by P2P applications [1]. The main Chord simulations dealt with a continuously churning 1000-node network in which there are 0.05-0.4 join or leave events every second. Tapestry was introduced as a “second-generation” P2P system [2]. However, some of the throughput measurements in Tapestry nodes (up to 7000 messages per second, over 80MB/s) suggested application in high-capacity networks. CAN was primarily motivated by the question, “could one make a scalable peer-to-peer file distribution system?”, though its designers suggested it could also be used for large-scale storage and name resolution systems [3]. Pastry was built as “an object location and routing substrate for wide-area peer-to-peer applications” [4].

P2P assumptions affect DHT designs and simulations. Chord, Tapestry, CAN and Pastry are all multi-hop DHTs. For example, in a network of 2^{15} nodes, the expected hop count is 7.5, 3.75, 14.14 and 3.75 respectively for Chord (degree=30), Tapestry (degree=56), CAN (degree=20) and Pastry (degree=56) [14]. The assumptions are reflected in the most comprehensive performance simulation of DHTs [15]. For “lookup intensive” workloads, each node issues only one query every 9 seconds. Node lifetimes were exponentially distributed with a mean of one hour, to match P2P file-sharing measurement studies.

Unfortunately, these P2P assumptions lead to DHT algorithms that require more overlay hops than necessary for stable, high-capacity networks. They waste query bandwidth. Given stable nodes and a high query load, two-hop DHTs might consume twice the query bandwidth used by one-hop DHTs. Many multi-hop P2P DHTs use much more.

This waste is significant because query bandwidth is a primary performance measure for giant data-intensive systems. Such systems generally operate at high utilization, close to their query bandwidth limit. AOL’s data centers support over ten billion queries per day [16]. From his experiences with Inktomi, Brewer described this limit as the “DQ principle” [16]:

$$\text{Data per query} \times \text{Queries per second} = \text{constant} \quad (1)$$

While much recent DHT research makes P2P assumptions, some stems from DHT deployments in enterprise data centers (Fig. 1). Huang and Fox reported that DHTs are used for Yahoo user profiles and for Amazon catalog items [12]. The Inktomi search engine accessed a DHT with 10^{12} entries for word-to-document mappings [17]. Inktomi also used a database to manage user data, but it turned out to be a

“mediocre approach, primarily due to cost, complexity and availability” - a highly available DHT would have sufficed [17].

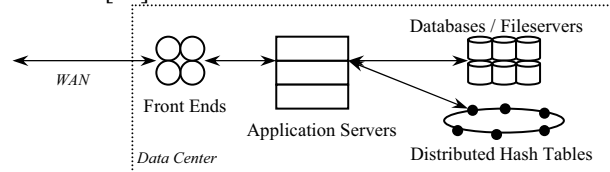


Figure 1. DHTs in an enterprise data center [12]

The earlier DHT research, predating the multi-hop P2P DHTs, seems more suited to the needs of enterprise data centers. The work by Huang and Fox was motivated by DHTs in a Scalable Distributed Data Structure by Gribble et al. [8]. They were in turn influenced by Litwin’s 1993 work on Linear Hashing (LH*) [6]. An LH* client required *at most* two hops for a key search. Litwin’s work has continued in parallel with the P2P DHTs, most recently with LH*_{RS} [9]. Like the P2P DHTs, it does better than hashing schemes which statically partition files over clusters of nodes.

3. Retrospection on One-Hop DHTs

The primary reasons that DHTs have been deployed in enterprise networks are the very high lookup rates and the massive, aggregate main memory indexes. Multi-hop DHTs should be used sparingly in this environment, since they waste bandwidth at very high lookup rates.

A key question, explored in Sect. 3.1, is “How many nodes can be supported by a one-hop DHT?” In Sect. 3.2, we ask “How should topology updates be distributed in one-hop DHTs?” There have been conflicting opinions on both questions [10, 11, 18].

3.1. Optimal Use of DHT Bandwidth

The number of nodes supported by a one-hop DHT can be determined by minimizing the aggregate DHT bandwidth - the lookup bandwidth and the topology maintenance bandwidth. The answer is sensitive to the expected node lifetime and the lookup rate per node, so generalizations can be misleading.

There have been differing conclusions, as shown in Table 1. Rodrigues and Blake concluded that multi-hop DHTs are required only when there are more than tens of millions of nodes [18]. Gupta et al. concluded that their OneHop design was suitable for up to a few million nodes [10]. Tang et al. concluded that one-hop DHTs are only feasible to a few thousand nodes [11]. There is a discrepancy of over three orders of magnitude.

Why the differences? In the OneHop design, Gupta et al. assumed a bound of a few million nodes, because

the upstream bandwidth on Slice Leaders (Sect. 3.2.) exceeds 350kbps when there are more than one million nodes [10]. Rodrigues and Blake [18] assumed two mechanisms consume bandwidth: each node downloads its share of the total file storage capacity when it joins the DHT; each node also downloads full DHT topology information. Rather than work from a bandwidth constraint, Tang et al. [11] optimized the aggregate DHT bandwidth. They assumed that DHT lookups and DHT topology maintenance contribute to the aggregate bandwidth.

Table 1. Comparison of viability of O(1) hop DHTs for various numbers of nodes

| | Rodrigues and Blake [18] | Gupta et al. [10] | Tang et al. [11] |
|---------------|---|---------------------------|----------------------|
| Lookup Rate | - | - | 0.1/node/sec |
| Node Lifetime | ~ 2 days | 2.8 hours | 2.9 hours |
| Bottlenecks | 200kbps per node for file recovery, 50kbps per node for DHT joins | 350 kbps per Slice Leader | - |
| Conclusion | < 10 million nodes | < few million nodes | < few thousand nodes |

So which is the most reasonable assessment of the viability of one-hop DHTs in enterprise networks? All three conclusions were based on P2P file sharing assumption – the node bandwidths or lifetimes were lower than would be found in an enterprise network. Since very high lookup rates are required in enterprise DHTs, we base our initial comparisons on Tang et al. [11], the only work to take lookup rates into account in the one-hop-versus-multi-hop decision.

Tang et al. [11] showed that the aggregate DHT bandwidth in degree-diameter optimal DHTs [19] is minimized when

$$d \ln^2 d = \frac{fl \ln n}{3} \quad (2)$$

where d is the node degree, the f is the lookup rate, l is the expected node lifetime and n is the number of nodes in the DHT.

Based on Eq. 2 we make an initial investigation of optimal one-hop DHTs for the lookup rates and node lifetimes that might be expected in enterprise data centers. Fig.2 shows that the optimal number of nodes in a one-hop DHT is sensitive to lookup rate and node lifetime. The ‘ephemeral’ (P2P) and ‘stable’ (enterprise) lines show operating points at which one-hop DHTs minimize the aggregate bandwidth. That is, at these points, multi-hop DHTs will always use more bandwidth. Consider the point labeled “26” on the “ephemeral” line in Fig. 2. For a 10,000 node DHT, if the lookup rate is over 26 lookups/node/sec *and* the

expected lifetime of each node is 2.9 hours or more, then a one-hop DHT always consumes less bandwidth than a multi-hop DHT. Similarly, if the lookup rate is less than 26 lookups/node/sec *and* the expected node lifetime is less than 2.9 hours, then a one-hop DHT always uses more bandwidth than a multi-hop DHT.

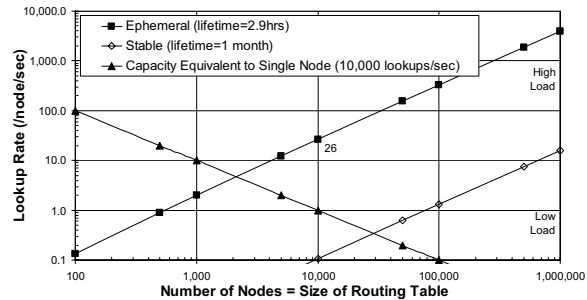


Figure 2. O(1) hop DHTs to minimize bandwidth for various lookup rates

The ‘equivalent capacity’ line of Fig. 2 shows an aggregate lookup rate of 10,000 lookups/node/sec. If the lookup requirements are near or below this line, then a DHT should not be used for its high lookup bandwidth – a single commodity node could do just as well. The range of lookup rates per node in Fig. 2 is commensurate with lookup rates in DHT prototypes: Tapestry nodes could process up to about 7000 messages per second [2]; in the prototype by Gribble et al. [8], the average node processes 480 reads and 106 writes per second.

A similar discussion applies to the point labeled “77” on the “low load” line in Fig. 3. For a 10,000 node DHT, if the node lifetime is over 77 hours *and* the load on each node is 1 lookup/sec or more, then a one-hop DHT always consumes less bandwidth than a multi-hop DHT. Similarly, if the node lifetime is less than 77 hours *and* the load on each node is less than 1 lookup/sec, then a one-hop DHT always uses more bandwidth than a multi-hop DHT.

The main lesson from Fig. 2 and Fig. 3 is that, when stable nodes support a high lookup load, one-hop DHTs are viable at least to a few hundred thousand nodes. The limit of a few thousand nodes in Tang et al. [11] only applies to ephemeral P2P nodes with low lookup rates per node. Eq. 1 assumed the aggregate bandwidth consists of DHT lookup and topology maintenance traffic. It did not consider the bandwidth to recover key-value pairs after failures. Sect. 4 takes recovery bandwidth into account, but it does not change the basic conclusion – stable, high-capacity, one-hop DHTs are viable at least to a few hundred thousand nodes.

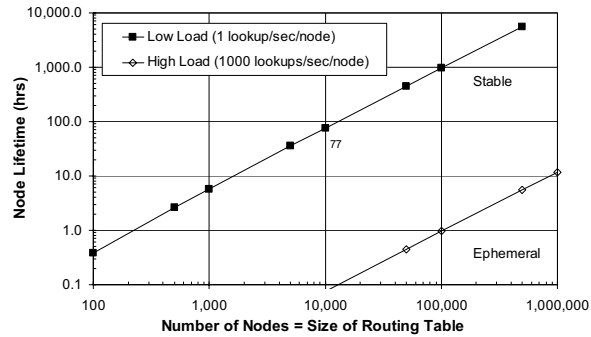


Figure 3. $O(1)$ hop DHTs that minimize bandwidth for various node lifetimes

3.2. Towards reliable topology updates in $O(1)$ hop DHTs

Given that one-hop DHTs meet the requirement for enormous lookup rates, we now explore one-hop topology maintenance schemes. At high lookup rates, slow propagation of DHT membership changes will increase the impact of lookup timeouts. We observe that one-hop topology maintenance is merely an application of multicasting, so the designs in Sections 4 and 5 can draw on 15 years of research on reliable, scalable multicasting [20].

To illustrate topology maintenance in one-hop DHTs, consider just two designs - OneHop by Gupta et al. [10] and 1h-Calot by Tang et al. [11]. In OneHop [10], particular DHT nodes are designated “slice leaders”. Notifications of node arrivals and departures are sent directly from all nodes to a slice leader. To minimize bandwidth, the notifications are buffered. They are then forwarded across to other slice leaders and down through overlay multicasting trees, so that all nodes eventually receive the notification. While they do decrease the volume of topology maintenance traffic when nodes have short lifetimes, slice leaders have disadvantages. They become a primary system bottleneck (Table 1). They delay notifications by tens of seconds. Their failover mechanisms have not been specified, so it is impossible to reason about recovery times. Simulations have shown, though, that the slice leader failures prolong recovery times from one to five minutes, when 45% of a 2000-node OneHop DHT fails [10]. Consequently in 1h-Calot, every node is a root of an implicit overlay multicasting tree [11]. The “branches” in each tree are selected in much the same way that “fingers” are chosen in the $O(\log n)$ -hop DHTs.

How well do these schemes compare against the reliable multicasting research literature? The choice of tree-based multicasting is sound – it has been shown to be the most scalable of the reliable multicasting

topologies [21]. The reliability mechanisms of 1h-Calot are likely to be ineffective in data center deployments. Messages are only acknowledged locally on the multicast branches – there is a significant probability that topology updates are lost before reaching all nodes. As a result, 1h-Calot nodes re-announce their existence after 0.7 lifetimes, but this may be weeks or months on stable nodes.

More generally, such sender-based reliability is vulnerable to acknowledgement implosion at the sender and acknowledgement delay up the multicasting tree. Recently, Slingshot [22] was shown to recover multicast packets two orders of magnitude faster than the well-known Scalable Reliable Multicast, that is, within a few milliseconds. Both are receiver-based schemes that avoid the problems of sender-based reliability. Slingshot relies on IP multicasting for best-effort delivery, and forward error correction amongst receivers for recovery.

In the designs of Sections 4 and 5, we assume that DHT topology maintenance is accomplished by receiver-based recovery (e.g., [22]) with :-

- a) Best-effort IP multicasting in data centers [20].
- b) Best-effort multicasting over single-hop DHTs, whenever IP multicasting is not available in the wide-area network (WAN). The 1h-Calot design, without acknowledgements and re-announcements, meets this requirement [11].

4. Site Rings

Whereas independent node failures might dominate a P2P network, correlated failures deserve special attention in enterprise networks [23]. An entire data center might lose power or WAN connectivity. We present the OneHopSites (IHS) design, by which DHTs can cooperate across protected sites, and recover from network partitions between those sites. We argue that sites are best served by independent DHT rings (Sect. 4.1). By quantifying normal and recovery traffic for the IHS design, we confirm that a single-site, one-hop DHT ring can indeed scale to a few hundred thousand nodes (Sect. 4.2). However, if a significant volume of the DHT load is write traffic, the inter-site bandwidth can limit each DHT site ring to about ten thousand nodes.

4.1. Assumptions

Three key assumptions underpin the IHS design. Firstly, because of correlated failure modes, high-capacity sites deserve independent DHT rings. In Sect. 3.2., we saw how query failures persist for five minutes after correlated failures in a 2000-node OneHop ring [10]. For other DHTs, it has been said

that correlated failures have little impact. For example, in Tapestry simulations of less than 1000 nodes, it was shown that correlated membership events fail a proportion of the lookups for less than one minute [2]. However, we argue that the simple design choice to confine rings to sites reduces failure impact at very little cost. It reduces uncertainty about inter-site recovery performance at a few hundred thousand nodes – a scale untested by DHT simulations to date. It gives simple, strong guarantees on the independence of replicas. In the event of a network failure between DHT sites, there are *no* DHT lookup failures.

Secondly, we assume simple replication between site pairs. IHS relies on the “smart client” approach [24] to shield end users from DHT failures, and to redirect DHT calls to another site in the event of a major outage (Fig. 4). Key-value pairs are replicated on each site.

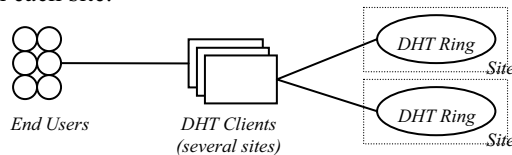


Figure 4. Smart DHT Clients

To support replication between sites, the analysis in Sect. 4.2. assumes the DHT message flows of Fig. 5. DHT reads and writes can be directed to any node on either site. Weak consistency is assumed, such that DHT writes continue during major network failures between sites. Each node trades heartbeats only with its immediate successor and predecessor. If a failure is detected and agreed locally by successor and predecessor, the whole ring is notified. Each node maintains a Local Node Cache – the full list of DHT members on the local site. The immediate neighbors guarantee lookup correctness, whereas the timeliness of updates to the Local Node Cache determines lookup performance. Each node also keeps a Remote Node Cache – the small number of nodes in the remote DHT ring that are responsible for the same key space. Nodes generally write to the remote replica in a single overlay hop. Each node knows Remote Node Gateways. These are members of the remote DHT that bootstrap the Remote Node Caches. A node contacts the Remote Node Gateway only once in its life – subsequent changes to the Remote Node Cache are discovered during normal DHT messaging.

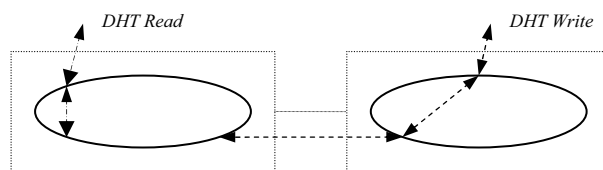


Figure 5. IHS reads and writes

Thirdly, we assume that nodes are loaded and stable. The analysis conservatively assumes read and write loads that have been achieved by DHT prototypes. The total offered load consists of 1000 reads/sec/node and 400 writes/sec/node. This is three to four orders of magnitude larger than loads in most simulations of the P2P DHTs [15]. The analysis shows that it is feasible to support over 100 million lookups per second on a single IHS ring. Nodes are stable, but to represent scalability conservatively, we assume lifetimes of only one week.

4.2. Analysis

Under these assumptions, Fig. 6 gives a breakdown of the DHT bandwidth per node.

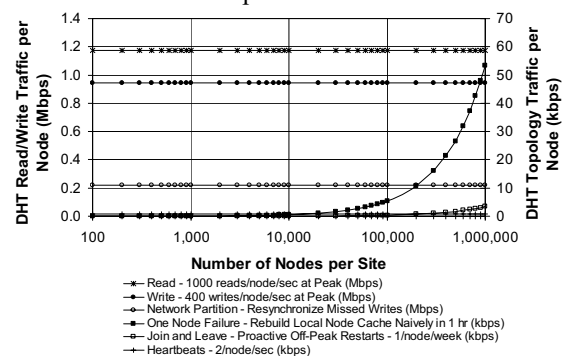


Figure 6. Total DHT traffic per node on O(1) hop sites

Fig. 6 demonstrates the key scalability advantage of the one-hop DHTs: bandwidth per node is independent of the number of nodes when the total offered load (averaged per node) is a constant. This is not true for the $O(\log n)$ DHTs. Nor is it true for “lightweight group multicasting” solutions, in which a lookup is multicast to many nodes – one responds, the remainder discard the message [25]. The bandwidth per node is clearly dominated by read/write traffic. Under the assumptions of Sect. 4.1., there is no need to design for a subtler balance between read/write traffic and topology maintenance traffic, as in [26]. The topology maintenance traffic amounts to only a few kbps, even for one million nodes. The linear scalability of one-hop DHTs make it easier to predict a) the performance of large deployments from small test models, and b) the capacity available during planned maintenance activities [16].

Fig. 6 also illustrates recovery bandwidth. The inter-site traffic to resynchronize replicas after a network partition (loss of connectivity between sites) was 0.22Mbps/node. It was assumed that the resynchronization time is equal to the duration of the network partition. In practice, the choice between resynchronization time and resynchronization

bandwidth will be determined by the application's specific weak consistency requirements. The analysis conservatively assumed that all missed writes need to be delivered to the other site, after the failure is repaired. Fig. 6 shows the bandwidth to rebuild the Local Node Cache by copying it naively from another local DHT node in one hour. Here again there is a balance between recovery time and recovery bandwidth. At 100,000 nodes (5.3kbps), there is room to increase the rebuild speed by one or two orders of magnitude. The "join and leave" curve of Fig. 6 shows the traffic generated as every node is proactively restarted each week. The topology notification is IP multicast to every collocated node when a node leaves, and then again when it rejoins. Proactive rolling restarts are one way to remedy transient failure and performance degradation [12].

Where Fig. 6 addressed bandwidth per node, Fig. 7 shows the breakdown of traffic entering each site: the total read and write client traffic (recall Fig.4); the total read and write traffic between sites; and the inter-site traffic to resynchronize writes after network partitions. At 10,000 nodes, each of these components of site bandwidth is between 2 and 4 Gbps. The exception is the very small inter-site read traffic – neighbors of failed nodes can redirect read traffic to replicas on the other site.

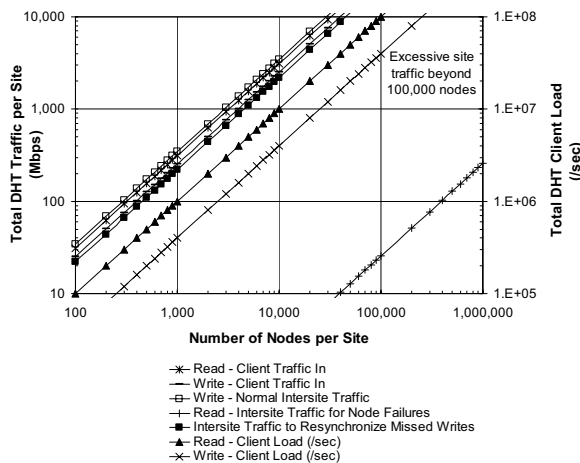


Figure 7. Total DHT traffic per site on O(1) hop sites

We somewhat arbitrarily cap the design at about 10,000 nodes – inter-site transmission over tens of gigabits per second is expensive. The bulk of the inter-site load is due to write traffic. If the offered load is only read traffic, the system could grow to over 100,000 nodes per site.

5. One Hop Federation

Whereas Sect. 4 explored one-hop DHTs for sites, Sect. 5 presents a hierarchic design for a global Federation of One-Hop DHTs (1HF). DHTs have been considered for several global applications needing reliability and massive aggregate throughput: name resolution [27]; location services for internet telephony [28]; a naming and inter-domain routing architecture for heterogeneous networks [29]; and a public, general-purpose DHT to support services like instant messaging, multicasting, and video streaming [5]. We take no position here on the suitability of DHTs for these applications. Instead, we resolve the open "long hop" problem [13] that is common in the top layer of hierarchies like 1HF (Sect. 5.1). A global ring lookup to resolve a key to its target region might go via the opposite side of the globe, even when the target object is in a nearby region. We confirm 1HF's linear scalability to a few hundred thousand nodes (Sect. 5.2).

5.1. Assumptions

There have been several attempts to deal with the problems of scale and locality at the top of global, hierarchic lookup networks [13, 30-32]. A single root is a well-known bottleneck for extreme lookup and update rates. A common way to scale *lookup* capacity is to multicast lookups to root clusters [31]. The target objects are replicated at many clusters. Such designs have yet to be proven for a high load of lookups *and* updates, such as might be found in a global location service [32]. When the target objects are updated regularly, or when they move regularly, existing solutions fall short. If one multicasts lookups to find the most up-to-date object, then much query bandwidth is wasted. If objects move, then how can we avoid unnecessary, high-latency, inter-continental hops? For this reason, "deciding which (root) sub-nodes should handle which entities in very large-scale location services is still an open question" [13].

To address this open issue, 1HF makes two assumptions. Firstly, it assumes rendezvous over hierarchic DHTs. One-hop DHT rings are arranged hierarchically, where regional rings have subtended organizational rings and organizational rings have subtended site rings. Regions and organizations deserve their own rings for the same reason that sites did in Sect. 4.1 - independent rings survive network partitions.

Several authors have proposed hierarchic DHTs without rendezvous. Some explicitly require 'ring identifiers' to augment key-based routing – if an object moves, a new ring identifier needs to be found [33,

34]. Others embed location information in content identifiers [35], which are changed when an object moves [33-35]. We recently proposed a rendezvous mechanism so that objects are known by persistent, location-independent identifiers (Fig. 8) [28]. 1HF uses a similar approach.

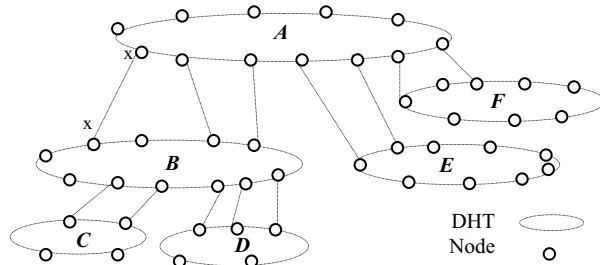


Figure 8. Global Hierarchy of DHT Rings [28]

Secondly, 1HF assumes a federation of regional hierarchies of one-hop DHTs. It does away with the global ring in previous hierarchic DHTs [28, 33, 34]. While 1HF is similar to [28], that design contained the “long hop” flaw identified by Tanenbaum and van Steen [13]. The indirection to resolve a location-independent key to a region might require a hop to the opposite side of the globe, even if the target object is nearby.

Fig. 9 shows the functions of the 1HF regional rings. The organization and site rings are omitted because a) they operate per the previous design [28] and b) the regional 1HF rings are likely to be the bottleneck. Regional rings are different to other 1HF rings in that they store only key-to-ring indirection records. Key-to-value records are kept by subtended rings, so that operations that write to the key-to-value records are transparent to the regional rings. The key-to-ring indirection record is updated when a) a key is inserted or removed and b) when the target key-value record moves from one region to another. It is expected that most key mobility is *within* a region, in which case no updates to the regional rings are required. The direct hop from the indirection record in one ring to the responsible node in another ring is accomplished by the Remote Node Cache (Sect. 4.1). Indirection records are expected to vastly outnumber any other record in the regional ring.

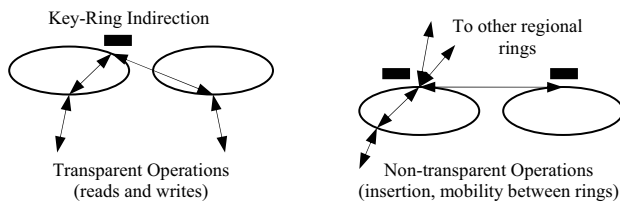


Figure 9. Functions of the 1HF regional rings

5.2. Analysis

Fig. 10 shows the aggregate capacity of the 1HF regional rings, in terms of the maximum number of DHT operations and the maximum number of “public” main-memory objects. We envisage that the majority of objects will be private, contained within organizational or site rings. The DHT operations load is limited by the bandwidth of the regional nodes, conservatively set to 2Mbps. Ballani and Francis estimated that there are 5000 service provider points of presence (PoPs) globally, based on 200 tier-1 and tier-2 ISPs and an average of 25 PoPs per service provider [36]. Assuming four gateways per POP, there may be over 20,000 nodes participating in 10 to 100 regional rings.

Like 1HS, the 1HF capacity scales linearly in both the DHT throughput and the total number of stored objects. The maximum DHT throughput is sensitive to the proportion of non-transparent operations, the number of regions and the query bandwidth of regional nodes.

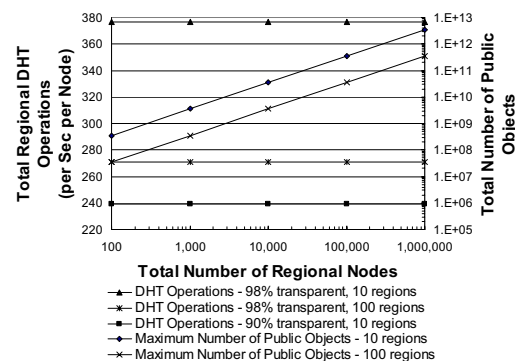


Figure 10. Capacity of 1HF regional rings

6. Conclusions

Our objective in this paper is to increase research momentum on one-hop DHTs. These are superior to multi-hop DHTs in stable, high-capacity enterprise networks because they use query bandwidth more efficiently.

Two designs have been presented. One Hop Sites uses separate DHT site rings for resilience to network partitions. One Hop Federation is a design for a global hierarchy of one-hop DHTs. It does away with the global ring, used in many hierarchic DHTs, to solve a known latency problem. Analysis has confirmed that both designs scale linearly to over one hundred thousand nodes.

7. References

- [1] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for Internet applications, *IEEE/ACM Trans. on Networking* 11 (1) (2003) 17-32.
- [2] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, Tapestry: A Resilient Global-Scale overlay for Service Deployment, *IEEE Journal on Selected Areas in Communications* 22 (1) (2004) 41-53.
- [3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A scalable content-addressable network, *Proc. of the conf. on Applications, technologies, architectures and protocols for computer communications*, August 27-31 2001, pp. 161-172.
- [4] A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, *IFIP/ACM Middleware 2001*, Nov 2001.
- [5] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, OpenDHT: a public DHT service and its uses, *Proc. of the conf. on Applications, technologies, architectures and protocols for computer communications*, Aug 22-26 2005, pp. 73-84.
- [6] W. Litwin, M.-A. Niemat, and D. Schneider, LH* - Linear Hashing for Distributed Files, *Proc. ACM Int'l Conf. on Management of Data SIGMOD*, May 1993.
- [7] R. Devine, Design and Implementation of DDH: A Distributed Dynamic Hashing Algorithm, *Proc. 4th Int'l Conf. on Foundations of Data Organizations and Algorithms 1993*.
- [8] S. Gribble, E. Brewer, J. M. Hellerstein, and D. Culler, Scalable, Distributed Data Structures for Internet Service Construction, *Proc. 4th Symp. on Operating Systems Design and Implementation OSDI 2000*, October 2000.
- [9] W. Litwin, R. Moussa, and T. Schwarz, LH*RS - a highly-available scalable distributed data structure, *ACM Trans. on Database Systems* 30 (3) (2005) 769-811.
- [10] A. Gupta, B. Liskov, and R. Rodrigues, Efficient Routing for Peer-to-Peer Overlays, *First Symp. on Networked Systems Design and Implementation NSDI*, March 2004.
- [11] C. Tang, M. Buco, R. Chang, S. Dwarkadas, L. Luan, E. So, and C. Ward, Low traffic overlay networks with large routing tables, *Proc. of ACM Sigmetrics Int'l Conf. on Measurement and Modeling of Comp. Sys.*, Jun 6-10 2005, pp. 14-25.
- [12] A. C. Huang and A. Fox, Cheap recovery: a key to self-managing state, *ACM Trans. on Storage* 1 (1) (2004) 38-70.
- [13] A. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2002.
- [14] G. S. Manku, M. Bawa, and P. Raghavan, Symphony: Distributed Hashing in a Small World, *Proc. 4th USENIX Symp. on Internet Technologies and Systems*, March 26-28 2003.
- [15] J. Li, J. Stribling, R. Morris, F. Kaashoek, and T. Gil, A performance vs. cost framework for evaluating DHT design tradeoffs under churn, *Proc. IEEE Infocom*, Mar 13-17 2005.
- [16] E. Brewer, Lessons from Giant-Scale Services, *IEEE Internet Computing* 5 (4) (2001) 46-55.
- [17] E. Brewer, "Combining systems and databases: a search engine retrospective," in *Readings in Database Systems*, 4th ed, 2004.
- [18] R. Rodrigues and C. Blake, When Multi-Hop Peer-to-Peer Lookup Matters, *The 3rd Int'l Workshop on Peer-to-Peer Systems*, San Diego, CA, USA, February 26-27 (2004)
- [19] D. Loguinov, J. Casas, and X.-M. Wang, Graph-theoretic analysis of structured systems: routing distances and fault resilience, *IEEE-ACM Trans. on Networking* 13 (5) (2005) 1107-1120.
- [20] S. Deering and D. Cheriton, Multicast routing in datagram internetworks and extended LANs, *ACM Trans. on Computer Systems* 8 (2) (1990) 85-110.
- [21] B. Levine and J. J. Garcia-Luna-Aceves, A comparison of reliable multicast protocols, *Multimedia Systems* 6 (1998) 334-348.
- [22] M. Balakrishnan, S. Pleisch, and K. Birman, Slingshot: time-critical multicast for clustered applications, *IEEE Network Computing and Applications* 2005.
- [23] P. Yalagandula, S. Nath, H. Yu, P. Gibbons, and S. Srinivasan, Beyond availability: towards a deeper understanding of machine failure characteristics in large distributed system, *Proc. First Workshop on Real, Large Distributed Systems*, December 2004.
- [24] C. Yoshikawa, B. Chun, P. Eastham, A. Vahdat, T. Anderson, and D. E. Culler, Using smart clients to build scalable services, *Proc. of the Usenix Annual Technical Conference*, Jan 1997.
- [25] K. Ostrowski, K. Birman, and A. Phanishayee, The power of indirection: achieving multicast scalability by mapping groups to regional underlays, (2005)
- [26] J. Li, J. Stribling, R. Morris, and F. Kaashoek, Bandwidth-efficient management of DHT routing tables, *Proc. 2nd Symposium on Networked Systems Design and Implementation*, May 2-4 2005.
- [27] V. Ramasubramanian and E. Sirer, The Design and Implementation of a Next Generation Name Service for the Internet, *ACM SIGCOMM 2004*, Portland, Oregon, USA, August 30-September 3 2004.
- [28] J. Risson, S. Qazi, T. Moors, and A. Harwood, A Dependable Global Location Service using Rendezvous on Hierarchic Distributed Hash Tables, *Proc. of the IEEE 5th International Conference on Networking*, Apr 23-28 2006.
- [29] J. Pujol, S. Schmid, L. Eggert, M. Brunner, and J. Quittek, Scalability analysis of the TurfNet naming and routing architecture, *Proc. of the 1st ACM workshop on Dynamic Interconnection of Networks*, Sep 2005, pp. 28-32.
- [30] G. Ballintijn, M. Van Steen, and A. Tanenbaum, Exploiting Location Awareness for Scalable Location-Independent Object IDs, *Proc. Fifth ASCII Ann. Conf.* 1999, pp. 321-328.
- [31] T. Deegan, J. Crowcroft, and A. Warfield, The main name system: an exercise in centralized computing, *ACM SIGCOMM Computer Communication Review* 35 (3) (2005) 5-14.
- [32] M. van Steen, F. Hauck, P. Homburg, and A. Tanenbaum, Locating Objects in Wide-Area Systems, *IEEE Communications Magazine* (1998)
- [33] A. Mislove and P. Druschel, Providing administrative control and autonomy in structured peer-to-peer overlays, *The 3rd Int'l Workshop on Peer-to-Peer Systems*, June 9-12 2004.
- [34] L. Garces-Erice, E. W. Biersack, K. Ross, P. Felber, and G. Urvoy-Keller, Hierarchical P2P Systems, *Proc. ACM/IFIP Int'l Conf. on Para. and Dist. Comp.*, Aug 2003.
- [35] N. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, SkipNet: A Scalable Overlay Network with Practical Locality Properties, *Proc. Fourth USENIX Symp. on Internet Technologies and Systems USITS'03*, March 2003.
- [36] H. Ballani and P. Francis, Towards a global IP anycast service, *Proc. of ACM SIGCOMM 2005*, Aug 22-26 2005.