Locating Logical Volumes in Large-Scale Networks

Mallik Mahalingam, Christos Karamanolis, Magnus Karlsson, Zhichen Xu

Hewlett Packard Labs

1501 Page Mill Rd, Palo Alto, CA 94304.

{mmallik, christos, karlsson, zhichen}@hpl.hp.com

Abstract

Storage is increasingly becoming a commodity shared in global scale, either within the infrastructure of large organizations or by outsourcing to Storage Service Providers. Storage resources are managed and shared in the form of *logical volumes*; that is, virtual disks that aggregate resources from multiple, distributed physical devices and storage area networks. Logical volumes are dynamically assigned to servers according to a global *resource utility model.*

This paper focuses on the problem of locating and accessing logical volumes in very large scale. Our goal is to devise mechanisms that are least intrusive to the existing Internet infrastructure. Two methods are proposed, based on *DNS name resolution* and *BGP routing*, respectively. The former is based on the current DNS protocols and infrastructure; the latter requires extensions to the existing BGP protocols. The two approaches are evaluated by means of simulations, based on realistic workloads and actual Internet topology. It is shown that the simpler and less intrusive DNS-based approach performs sufficiently well, for even small caches on the clients.

1 Introduction

Storage Service Providers (SSP) such as ScaleEight [1] and StorageNetworks [2] provide network-based storage solutions for customers that wish to outsource some or all of their data storage and its management. They provide a global storage infrastructure that enables their customers to create, manage and distribute large sets of data across multiple geographic locations.

Clients access such a global storage service in one of two ways. First, directly by means of traditional file system APIs, e.g., through NFS mount-points. These clients are typically hosts that execute application services for the organizations that outsource storage to the SSP. Second, by means of Content Delivery Networks (CDNs) [3, 4], which replicate certain types of the data (originating from the SSP) closer to the edge of the network. We envision that in future storage services, the borderline between SSPs and CDNs will be blurred, as content will be dynamically created and stored at the edge of the network. The emerging technologies for distributed application services [5, 6] and peer-to-peer CDNs [7] point in that direction. Throughout this paper, we use the term *clients* to refer to both these classes of clients.

Typically, the infrastructure of an SSP consists of a pool of storage resources, such as disks, disk arrays and Storage Area Networks (SANs), as well as compute resources (servers) for

providing access to the storage. This infrastructure is physically distributed across multiple geographic locations. SSPs may own their own Data Centers, or their resources may be hosted at Internet Data Centers (IDCs), such as those of Exodus [8] and Qwest [9]. Moreover, we anticipate that, in the future, storage service providers will not necessarily own their own physical resources. Instead, their infrastructure will be provided by on-demand aggregation of resources from multiple disparate data centers, following the principles of a *resource utility* model [10, 11].

Even today, the infrastructure of SSPs and big corporations consists of many, heterogeneous and distributed physical storage resources. In this context, *logical volume managers* are used in order to simplify the management and facilitate the use of diverse resources. *Logical volumes* provide an abstraction for aggregating storage resources spread across multiple disks (that are attached to the same server or the same SAN) to appear as a single virtual storage device [12]. Data is organized within the boundaries of the logical volumes. Data on volumes are accessed through one or more servers that mount that volume. The data may be organized in the form of a file system or a database. To keep the discussion simple, in the rest of the paper, we will refer to data as files.

Clients access a volume by going through the corresponding file server, which coordinates all accesses via a file system API. When a client requests access to a file (performs a lookup), a *file-handle*, which uniquely identifies the file in the system, is returned back to the client. This file-handle contains a *Volume Identifier (VID*) that refers to the logical volume where the file is physically stored [13, 14]. Files accessed by a client may be spread across multiple logical volumes. Therefore, for every file access, the client must resolve the location of a file server that "owns" the logical volume where the file resides.

In a resource utility model, the mapping of logical volumes to physical resources and their assignment to file servers can be dynamic. Therefore, a key problem is how to provide efficient and scalable mechanisms for locating a logical volume and its custodian file server. The system model we assume for our discussion is outlined in section 2. In section 3, we propose a mechanism by which file servers can locate the logical volumes that they are responsible for. Sections 4 and 5 introduce two mechanisms for resolving the identity of a server that provides access to a volume. The main idea behind the proposed solutions is to exploit well-understood mechanisms, with proven scalability in the Internet, and adapt them for locating volumes in very large scale. Our aim is to use existing services (e.g., DNS), with no or minimal changes to the existing infrastructure. The two approaches are evaluated in section 6, using simulation based on both real and synthetic workloads, as well as real Internet topology information. Section 7 discusses related work and section 8 concludes the paper.

2 System overview

The infrastructure of an SSP resembles any other network in the Internet. We assume it is divided into a number of *Zones*, each with a unique identifier, *Z-ID*. Each Zone consists of one or more Autonomous Systems (AS) and each Autonomous System consists of a number of Autonomous System Regions (ASR). *An* ASR representative maintains a database that

contains information on the logical volumes within its region and their assigned servers. By organizing the system this way, we uniquely identify any logical volume by a Volume identifier (VID), using the convention "*Volume-ID.ASR-ID.AS-ID.Zone-ID*".

File servers typically retrieve their logical volume assignment by interacting with an ASR representative. The volume assignments may be dynamic to accommodate system reconfiguration, fluctuating demand or changing workloads. Automating the resource management in such environments is the focus of several current research projects [10, 11, 15].

When a client requires access to a file, it performs a lookup by sending a lookup request to the file server that hosts the logical volume where the parent directory of the file resides. The file server performs lookup locally on the parent directory and returns the file handle corresponding to the file. Note, that the volume (and server) of the parent directory, where the file handle is constructed, and the volume of the file itself may not be the same, as it is the case in systems such as Archipelago [16] and DiFFS [14]. The file-handle contains a *Volume Identifier (VID*) that refers to the logical volume where the file is physically stored. In order for the clients to access the file, they must resolve the VID and locate the file server that "owns" the corresponding logical volume.

3 Assignment of logical volumes to servers

When a file server comes online, it sends out a request identifying itself, asking for logical volumes that it is responsible for. This functionality is implemented using the DHCP protocol [17]. When an ASR representative within the vicinity of the file server receives the request, it locates the list of logical volumes that the requestor is responsible for and responds back supplying the list to the server. The response contains the configuration information of the logical volumes. For example, in an IP-based SAN, the response may contain Logical Unit Numbers (LUN) and their corresponding target IP addresses, along with other information such as whether a logical volume is stripped, mirrored, etc. The assignment of logical volumes may be pre-configured via storage management tools or may be dynamically assigned by an ASR representative upon receiving the request. Once an assignment is made, the representative for the ASR updates its database to reflect the new state of server-to-volume assignment. These assignments can be dynamically changed to cater for various system conditions such as file server utilization, load balancing, locality, etc. Any reassignment of logical volumes affects only the database of a specific ASR and leaves the rest of the mapping in the system intact.

In very large systems following the resource utility model, we cannot assume that file servers can reach ASR representative via DHCP. Two solutions can be applied in such environments: 1) the file server is pre-configured with a set of logical volumes; 2) the file server is configured with the identity of an ASR representative (not necessarily of its local ASR) which it should contact to retrieve its volume assignments.

4 Logical volume discovery by clients using DNS

In this approach, each *Zone*, *AS and ASR* has one or more designated representatives, which, in practice, are part of the existing DNS infrastructure (authoritative servers) [18]. The root server of the SSP contains information on all zone representatives. Every zone representative maintains a database with all the AS representatives within its zone. In the same way, an AS representative maintains information about all ASR representatives within that AS.

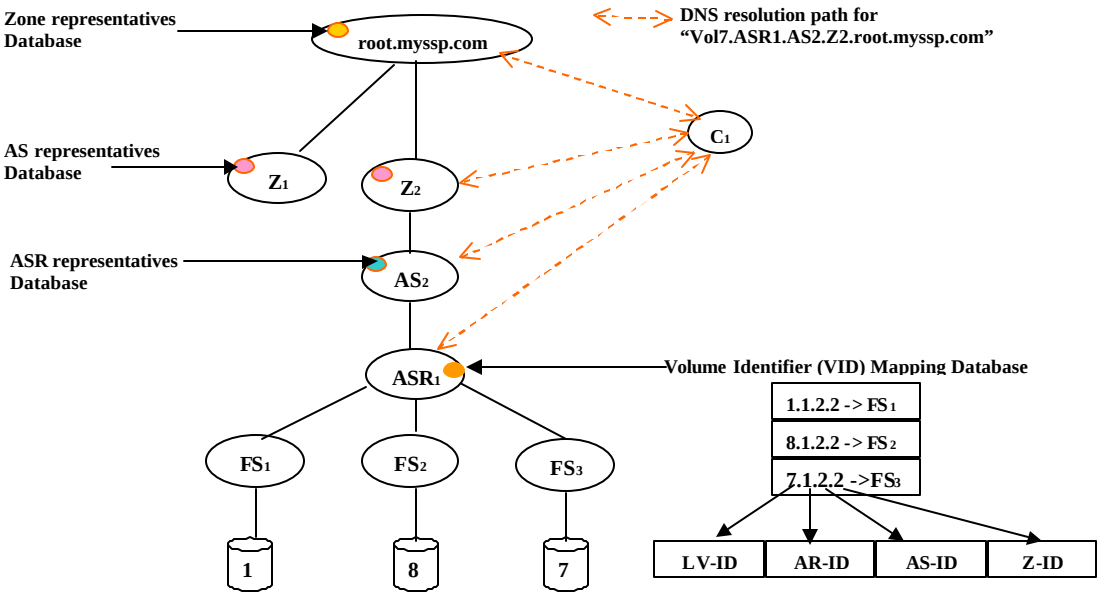


Figure 1: VID resolution using DNS

For a client to access a file, it has to first retrieve a file handle via a lookup process. The client then needs to locate the file server that corresponds to the Volume Identifier (VID) in the file handle. The identity of the server is resolved by exploiting typical DNS name resolution [18]. For example, when a client C_1 receives a file handle that contains VID *7.1.2.2*, it constructs a fully qualified domain name "Vol*7*.ASR*1*.AS*2*.Z*2*.*root.myssp.com*" based on the numerical VID contained in the file handle and the root domain name of the SSP. The root domain name is obtained during the file system mount time The client then resolves this (artificial) domain name through a normal DNS resolution process, as depicted in Figure 1. This process does not require any changes to the existing DNS infrastructure. However, the root server of the SSP needs to be configured to respond to the domain suffix "Z2" by specifying the authoritative representatives for that part of the domain suffix. When a client's requests land at the representative for an ASR, the address of the file server that corresponds to the VID is returned. Results of this query can be cached at the client for improved performance.

Various optimizations are possible in order to speed up the resolution process. One possibility is to have file servers resolve the logical volume mapping, cache the information locally and return the mapping information when a file handle needs to be returned back to the

client. This cached information could significantly reduce the network traffic especially when many clients reference the same logical volume. Cached information can be kept loosely consistent with the actual mapping by performing periodic checks. Also, resolution at the file server can be performed in an asynchronous fashion to hide any extra delays. Invalid references can arise due to volume reassignments or the non-availability of file servers. In this case, clients resort back to the normal resolution process.

Clients can also contact a local DNS server and have that server perform the logical volume to file server mapping. Typically, employing optimizations like this has proven to produce higher cache hit ratio [19] in resolving domain names at the client.

5 Logical volume discovery using suffix-based routing

This section introduces an alternative approach for clients to retrieve the custodian file server of logical volumes, called *Volume Identifier Routing Protocol* (VIRP). Given a VID, VIRP routes the request for VID resolution to the corresponding ASR representative taking the shortest ASR (or AS) path and returns the address of the corresponding file server to the client.

VIRP is based on suffix reachability that is similar to prefix-based routing performed using BGP [20]. There are two variations of the protocol. In the first variation, each ASR representative advertises itself to its neighboring VIRP routers. These advertisements are propagated further to other VIRP routers. For a particular VIRP router, routing advertisement of an ASR representative indicates the shortest path towards that ASR representative.

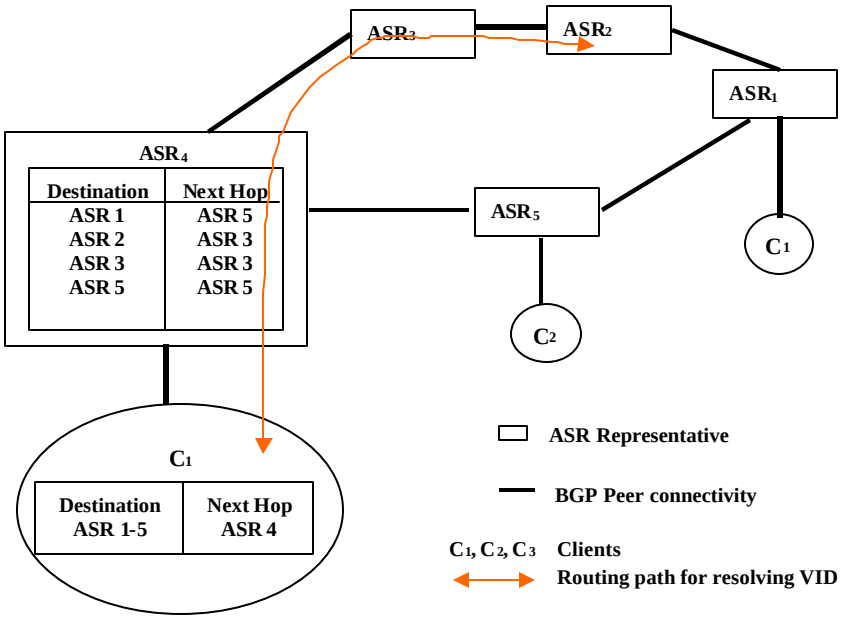


Figure 2: Example showing VIRP advertisements and routing VID resolution

For example, Figure 2 shows the routing table at VIRP router ASR_4. The routing table contains the next hop address for other ASR representatives following the shortest path. As shown earlier in section 2, VID contains a Volume ID, an ASR ID, an AS ID, and a Zone ID. Clients resolve VID by routing the request to the ASR representative corresponding to the ASR part of the VID. The routed request takes the shortest path leading to the target region. For example, a client C_1 that wishes to resolve a VID that belongs to ASR_2 will first route to ASR_4 and then take ASR_3 as the following hop and route to ASR_2. In VIRP, the clients receive routing advertisements but do not perform any advertisements.

Alternatively, to reduce the size of VIRP routing tables, the advertisement can be performed at the AS level. We introduce a representative for each AS to receive requests from clients and direct them to ASR representatives. The AS representatives advertise themselves as it was done in the previous case. Once a client request is routed to an AS representative, the latter can forward the request to an ASR representative by performing a local lookup using the ASR-ID. The respective ASR representative responds to the client with the address of the file server using the volume part of the VID. This greatly reduces the number of entries kept in the routing tables but it requires defining additional protocols for interaction between AS and ASR representatives. To give the readers an idea of the savings on routing table size, assume that an ASR corresponds to a network prefix on the Internet. There are 150K unique prefixes whereas the number of AS on the Internet is on the order of 10K.

There are several ways to deploy this type of infrastructure. One way is to reuse the existing BGP routing infrastructure by adding new protocols. A more practical way is to construct an *overlay network* to build this infrastructure [21]. Such an overlay network can be constructed at application level for easy deployment.

6 Evaluation

The performance of the proposed DNS-based and BGP-based approaches is evaluated by means of simulations. The simulation model is based on an Autonomous System (AS) view of the actual Internet topology as of October 2001, and a real-world, globally distributed workload. We chose this to be a web workload for two reasons. First, we believe that Content Delivery Networks will be one of the main applications of a globally distributed file system, and secondly, it is one of the few workloads that today have millions of globally distributed clients. The metric used to compare the two approaches is *client perceived latency* in resolving a VID.

6.1 Simulation Methodology

Our simulation model uses three sets of inputs in order to calculate the client perceived latency for the approaches: An Internet topology, a set of volumes and their locations, and finally the location of the clients and a list of chronologically ordered accesses to these volumes. The input parameters are all summarized in Table 1.

The Internet topology was generated using BGP routing table information obtained from a leading ISP, *Telestra.net* [22], during October 2001. From these routing tables an undirected

graph is constructed, in which nodes represent Autonomous Systems and edges represent their peering relationship. The generated graph contains approximately 13.000 nodes and 150.000 edges and we assume a uniform edge cost. The distance between two nodes in the topology is measured in number of AS-level network hops on the shortest path between those nodes. The placement of the DNS servers in this Internet topology is decided in the following way. We generated a list of nodes sorted in descending order of their fan-out (number of nodes that are just one hop away from one specific node). The node that has the highest fan-out is selected to be the representative for "*root*" and removed from the list. Next, the set of zone representatives are picked from the top of the list and then are removed from the list. The AS and ASR representatives are chosen in the same way.

Table 1: The main parameters of the experimental platform and their corresponding values. The shaded parameters are the ones that we vary in the experiments.

	Parameter	**Value**
Topology	Distribution	Part of real Internet
Volumes	Number	20,000 or 80,000
	DNS nodes	4/10/5/100 (Z/AS/ASR/Volumes) or 4/40/5/100
Objects	Number	90,000 or 1 million
	Distribution	Sequential or Random
Clients	Number	5,400 Client clusters
	Distribution	According to real AS location
	VID access pattern	WorldCup98 or Random

The object references were obtained from web logs of the World Cup Soccer 1998 event [23]. The logs contain references to nearly 90K unique files. These files are mapped on 20K and 80K volumes, respectively for the two scenarios. While clearly the World Cup site would not in reality be located on this many volumes, a client would not access solely one site. Instead a client would be accessing many different volumes of various sites. Our client workload can thus be seen to represent a widely scattered surfing pattern that is close to a worst-case scenario for the DNS approach. The placement of objects to volumes is done in two ways: *sequential* and *random*. For each of these algorithms, N files (where N = unique files / no of nodes) need to be placed on each volume. For the sequential algorithm, the first N unique files encountered in the web log are placed on node 1.1.1.1. The following N unique files are then placed on node 2.1.1.1, and so on. As more frequently accessed files tend to show up earlier in the web log, this algorithm will place popular files closer to each other. The random algorithm, on the other hand, places the first N files encountered in the web log on a random node, the next N files on another random node, and so on.

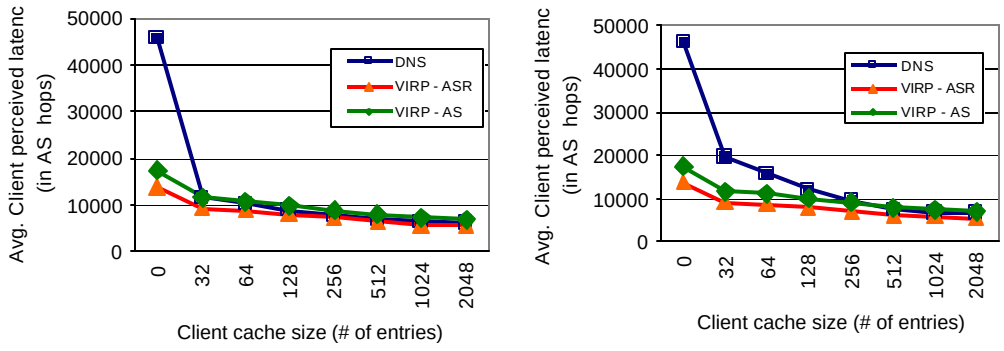
The clients' locations and access patterns were also obtained from the 98 World Cup logs. These contain accesses made by roughly 2.6 million clients over the course of 90 days (includes accesses made 30 days prior and 30 days after the event). To be able to assign these clients to the AS node they actually resides on in reality, we developed a program that converts IP address of a client to the corresponding AS ID. This clustering generated about 5.4K unique client clusters that are located in the same number of unique ASs.

We use two different client access traces to evaluate the proposed schemes: *WorldCup98* and *random*. The former is taken straight from the client accesses of the World Cup log; the latter is a uniformly random VID accesses. In the World Cup log, all clients in one AS access, on the average, 1K unique objects, while in the random one, the simulation is terminated after 2K unique objects are referenced by each AS.

To measure the client perceived latency, 20% of the ASs were randomly chosen and used in the simulations. They represent 500K clients generating close to 20% of the total client accesses. For each AS, a list of objects that the clients in that AS accessed is generated. In our model, every server (DNS server or VIRP router) that is queried adds to the client perceived latency. We express the client perceived latency in terms of the number of AS hops involved. This has been shown to be a fair measure of latency [24]. Network contention is not taken under consideration. For the simulation, we have used simple LRU caching at the clients to store the resolved VIDs. The impact of the size of this cache and all other shaded parameters in Table 1 are examined in the next section.

6.2 Performance Results

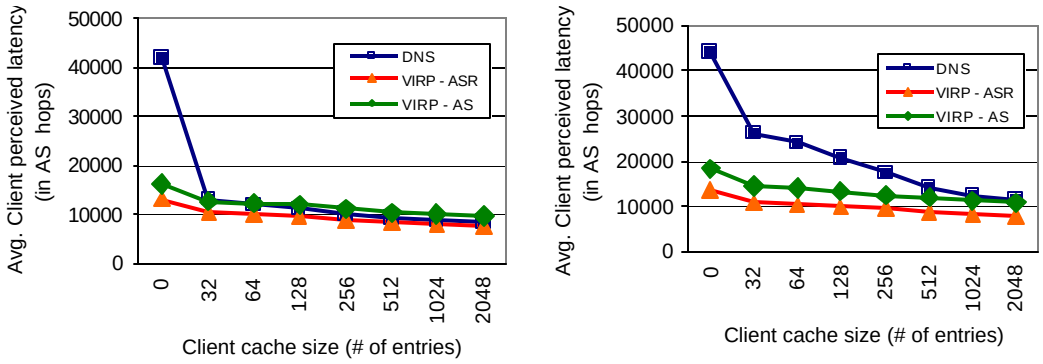
The initial intuition was that the DNS approach should have a higher client perceived latency than the VIRP approaches, when the VID lookup cache size is small and/or when the locality of VID lookups is poor. In this section, we will investigate how much locality the DNS approach needs in order to be comparable to the VIRP approaches, and provide a rough estimate on how many VID lookups need to be cached at each client for this to be achieved.



(a) Sequential object distribution ***(b) Random object distribution***

Figure 3: Results for the DNS, VIRP-ASR and VIRP-AS approaches. Number of volumes: 20,000. Number of objects: 90,000. Client access pattern: WorldCup98.

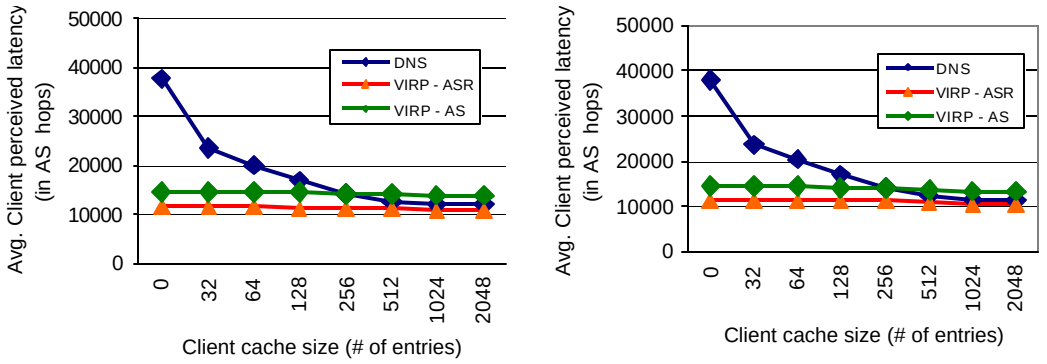
Figure 3a shows the results for the DNS, VIRP-ASR and VIRP-AS approaches using sequential object distribution. In the figure, the x-axis represents the various client cache sizes and the y-axis represents the average client perceived latency due to the VID lookup process. VIRP-ASR has the lowest client perceived latency as it requires only one lookup message and it traverses the shortest path between the client and the server. For VIRP-AS, there is a potential for one more message, thus the slightly worse performance. The most interesting point in this graph is that the DNS approach performs well even for small client cache sizes. For the sequential object distribution of Figure 3a it starts to perform well at 32 entries, but for the random case in Figure 3b, this point is only increased to 256 entries. For a straightforward implementation of the client cache, this translates to a modest 1KB and 8KB of memory space, respectively.



(a) Sequential object distribution ***(b) Random object distribution***

Figure 4: Results for DNS, VIRP-ASR and VIRP-AS approaches. Number of volumes: 80,000. Number of objects: 90,000. Client access pattern: WorldCup98.

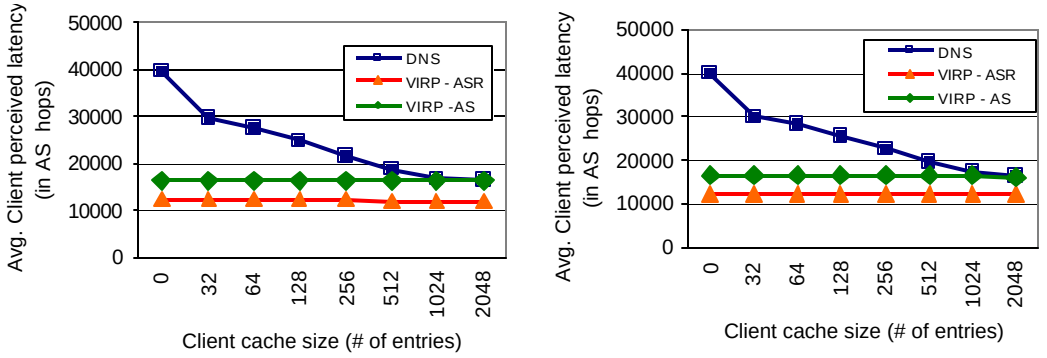
Figure 4 shows the effects of what happens if the number of volumes is increased four times to 80,000 volumes. As the locality will be poorer than before, we would expect the DNS approach to perform even worse. But for the sequential object distribution it hardly matters for clients with a cache, as the DNS approach performs as well as before. However, for the random object distribution the cache size required for DNS to become comparable to VIRP-AS is larger. It is now around 2K entries, translating into 64KB of memory space.



(a) Sequential object distribution *(b) Random object distribution*

Figure 5: Results for DNS, VIRP-ASR and VIRP-AS approaches. Number of volumes: 20.000. Number of objects: 1 million. Client access pattern: Random.

The last set of experiments was designed to stress the approaches even further to see how they hold up for a random client access pattern with a larger number of objects. Few workloads will have access patterns that are truly random, however, this will provide us with a worst-case scenario for the approaches. Figure 5 and 6 show the results for the random client-access pattern when the number of objects is 1 million. It can be seen that the VIRP approaches perform better than the DNS approach for small sizes of caches, but their performance remains more or less unaffected by the client cache size.



(a) Sequential object distribution *(b) Random object distribution*

Figure 6: Results for DNS, VIRP-ASR and VIRP-AS approaches. Number of volumes: 80.000. Number of objects: 1 million. Client access pattern: Random

This is due to the random accesses to volumes. There is little reuse of VIDs as the lookups are completely random, thus there is also little use of the client cache for storing individual VID lookups. However, for the DNS approach there will still be access locality for the entries that store the zone, AS and ASR lookups as there are far lower number of these in the system than volumes. This explains why DNS benefits from a larger cache but not the VIRP

approaches for this experiment. Thus, even for modest cache sizes, the performance of the DNS approach is comparable to that of VIRP.

6.3 Summary of simulation results

Our simulation shows that VIRP with ASR level aggregation outperforms all other approaches we compared against. The drawback with the VIRP approaches is that they require protocol modifications to the existing routing infrastructure. The DNS approach, on the other hand, can be deployed on existing infrastructure. Its performance is comparable to VIRP for reasonable client cache sizes even when the locality is poor. For reasonable cache sizes, the type of the object distribution has lesser effect on the client perceived latency. In general, we believe that the deployment of the DNS approach is preferable as its performance is comparable to the VIRP approaches, while using existing infrastructure.

7 Related Work

Existing distributed storage systems, such as AFS [13, 14], are designed for deployment in campus environments. These systems maintain a *volume location database* (VLDB) to track the servers in the system where volumes reside. For example, AFS maintains a VLDB for every "cell" of the system. The VLDB is typically replicated on two or more *Volume Location Servers*, for availability reasons. An AFS client within a cell is manually configured with a list of *Volume Location Servers* that it can contact to resolve the volume location. This is not a feasible choice for large-scale geographically dispersed networks such as the Internet. Also, AFS does not provide any mechanisms by which *file servers* can locate the logical volumes they are assigned to; this information is hard-wired in the servers' configuration.

Volume managers such as that of Veritas [25],[26] and storage virtualization systems [27] aggregate multiple, disparate physical storage resources using the volume abstraction. These solutions are applicable to small-scale systems, a single SAN and a single data center. Neither they provide service for hosts in the network to discover their assignments nor they allow clients to resolve the owners of logical volumes.

Techniques used by SSPs such as Scale8 [1] are not published. Karamanolis *et al*. [14] describe mechanisms by which a file server keeps limited information about the peers that the logical volumes under its custody have references to. Their proposal is primarily an optimization of our DNS approach, where caching is used at the file server.

8 Conclusion

Storage is increasingly becoming a commodity resource shared in global scale. The emerging business model of outsourcing storage (or its management) to third-party service providers amplifies this trend. In this context, storage resources are virtualized and shared by means of logical volumes. This paper addresses the problem of locating and accessing logical volumes in global infrastructures, as those of Storage Service Providers or large corporations.

The paper briefly describes ways to assign computational resources (servers) to volumes and how this mapping is performed in various system models. We then focus on mechanisms for

clients to locate and access logical volumes, in a very large, dynamic infrastructure. That is, locate the servers that provide access to specific volumes. In environments of the scale and volatility required in a "resource economy", a centralized volume location database does not provide a satisfactory solution. First, it does not scale sufficiently (e.g., for tens of thousands of volumes); second, we cannot expect a centralized "knowledge" of the entire system's configuration.

The motivation for the work presented in this paper was to investigate solutions that are based on well-understood and provably scalable mechanisms. In that spirit, two approaches are proposed to address the problem. The first is based on existing DNS infrastructure and protocols to resolve hierarchical volume identifiers. The second proposes extensions to existing BGP routing protocols to efficiently locate host servers of volumes.

Our initial assertion was that the BGP-based approach would perform better than the DNS approach. However, experimental results based on simulations indicate that even for modest volume-id caching on the clients, the benefits of BGP are negligible. Moreover, the DNS approach is based completely on existing protocols and it is not intrusive to the existing infrastructure. So, its deployment would be straightforward. On the other hand, the BGP approach requires extensions to existing protocols and routing table management, making it much harder to be deployed in a real environment. The latter is not justified by the marginal performance benefits this approach offers.

9 References

[1] ScaleEight, http://www.scale8.com/.

[2] StorageNetworks, http://www.storagenetworks.com/.

[3] Akamai, http://www.akamai.com.

[4] DigitalIsland, http://www.digitalisland.com.

[5] Ejasent, http://www.ejasent.com.

[6] Zembu, http://www.zembu.com.

[7] J. Kangasharju, J. W. Roberts, and K. W. Ross, "Object Replication Strategies in Content Distribution Networks," presented at 6th Web Caching and Content Distribution Workshop, Boston, MA, USA, 2001.

[8] Exodus, http://www.exodus.com.

[9] Qwest, http://www.qwest.com.

[10] K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and B. Rochwerger., "Oceano - SLA Based Management of a Computing Utility," presented at Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, 2001.

[11] J. Wilkes, J. Janakiraman, P. Goldsack, L. Russell, S. Singhal, and A. Thomas, "Eos - The Dawn Of The Resource Economy," presented at HotOS-VIII Workshop, Schloss Elmau, Germany, 2001.

[12] D. Teigland and H. Mauelshagen, "Volume Managers in Linux," presented at FREENIX Track: 2001 USENIX Annual Technical Conference, Boston, Massachusetts, USA, 2001.

[13] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West, "Scale and Performance in a Distributed File System," *ACM Transactions on Computer Systems*, vol. 6, pp. 51-81, 1988.

[14] C. Karamanolis, L. Liu, M. Mahalingam, D. Muntz, and Z. Zhang, "An Architecture for Scalable and Manageable File Services," Hewlett-Packard Labs, Palo Alto, Technical Report HPL-2001-173, July 2001.

[15] IBM, "Autonomic computing," : http://www.research.ibm.com/autonomic.

[16] M. Ji, E. W. Felten, R. Wang, and J. P. Singh, "Archipelago: An Island-Based File System for Highly Available and Scalable Internet Services," presented at 4th USENIX Windows Systems Symposium, 2000.

[17] DHCP, http://www.dhcp.org.

[18] DNS, http://www.dns.net/dnsrd/.

[19] E. Sit, "Study of caching in the Internet Domain Name System," Massachusetts Institute of Technology, May 2000., 2000.

[20] Y. Rekhter, T. Li, and M. 1995, "A Border Gateway Protocol 4 (BGP-4) - RFC 1771," in *Request for Comments: 1771*, 1995.

[21] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. Kubiatowicz., "Bayeux: An Architecture for Scalable and Fault-tolerant WideArea Data Dissemination," presented at In Proc. of the Eleventh International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2001), 2001.

[22] Telstra, "Raw BGP Data," http://kahuna.telstra.net/bgp2.

[23] Worldcup98, "Worldcup98 soccer event - Web logs," : http://ita.ee.lbl.gov/html/contrib/WorldCup.html.

[24] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-Informed Internet Replica Placement," presented at 6th Web Caching and Content Distribution Workshop, Boston, MA, USA, 2001.

[25] VERITAS, "Veritas Volume Manager," http://www.veritas.com.

[26] M. Hasenstein, "The Logical Volume Manager (LVM)," http://www.sistina.com/lvm_whitepaper.pdf.

[27] StorageApps, "SANLink," http://www.hp.com/products1/storage/san/sanlink/index.html.