99 Deduplication Problems

Philip Shilane, Ravi Chitloor, and Uday Kiran Jonnala
EMC Corporation

Abstract

Deduplication is a widely studied capacity optimization technique that replaces redundant regions of data with references. Not only is deduplication an ongoing area of academic research, numerous vendors have deduplicated storage products. Historically, most deduplication-related publications focus on a narrow range of topics: maximizing deduplication ratios and read/write performance. While future research will continue to optimize these areas, we believe that there are numerous novel, deduplication-specific problems that have been largely ignored in the academic community. Based on feedback from customers as well as internal architecture discussions, we present new deduplication problems that will hopefully spur the next generation of research.

1 Introduction

Deduplicated storage is an active area of research within both academic and industry because it offers the potential to reduce storage costs by removing redundant data. There are numerous sources of data redundancy including frequent backups, code bases copied by engineers, VMs that are slight modifications of a standard template, etc. Venti [9] was one of the first research systems to detect redundant data within a storage system using hashes of chunks of the content, called fingerprints, which opened up the possibility of dramatically reducing storage capacity requirements and, therefore, costs. To meet performance requirements and reduce resources such as memory and I/O, DDFS [13] and numerous other deduplicated systems leveraged data locality and other techniques to create commercially available products.

Deduplication, as a publication topic, has exploded in the last decade as indicated by the number of search results shown in Figure 1. From a cursory appraisal, most focus on a narrow range of topics such as improving deduplication ratios or system performance, though other topics are certainly covered: specific data types, security implications, hardware changes, etc. We direct readers to a survey article [8] for further details.

Space savings and performance reached competitive levels over a decade ago in commercial products, and deduplication has itself become commoditized. Relying on a deduplicated storage system every day, though, is different than testing a research prototype where limited functionality may be acceptable. Customers want the full set of features they have grown accustomed to on standard, non-deduplicated storage, and their needs continue to evolve as use cases change. While space savings and performance will continue to benefit from improvements, these new problems are the key hurdle for the next generation of deduplication products, especially as primary storage incorporates deduplication.

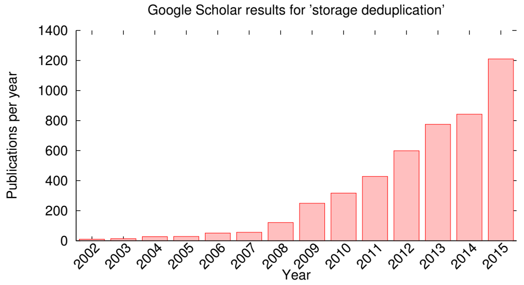

Figure 1: Rapid increase in deduplication publications.

This paper provides a brief summary of new problems specific to deduplication that we feel have not yet received the level of research attention they deserve. To generate these problems, we have spoken with customers in the field who use deduplicated storage, industry experts who analyze multiple products, and engineers who design the future architecture. In this short paper, we identify five classes of new deduplication problems: Capacity (§2), Quality of Service (§3), Security and Reliability (§4), Management (§5), and Chargeback for Service Providers (§6). In several problem areas we highlight existing, initial work. Creating a list of new deduplication problems is an ongoing task, as each advancement triggers another set of problems, and features added to non-deduplicated storage systems are requested on deduplicated products.

2 Capacity

One of the most common questions when using a storage system is, "How much space is left?" For non-deduplicated storage, this is a fairly straightforward question to answer since such systems track allocated and free blocks. Even for a storage system with asynchronous cleaning, such as a log structured storage system, the current number of free blocks can be answered immediately, though the number of free blocks may increase as cleaning progresses. For deduplicated storage, the question is much more complex to answer.

Consider an example system in Figure 2 with client files at the top, unique chunks of data in the storage sys-

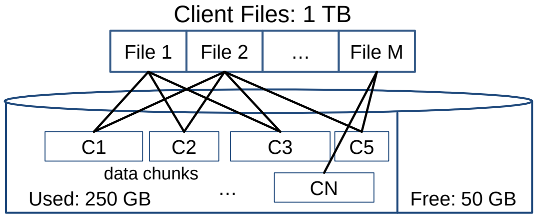

Figure 2: Deduplicated storage showing client files, unique chunks, and free capacity.

tem, and links between files and the chunks that compose the file. Clients wrote 1 TB of file content, and due to deduplication (neglecting compression effects), 250 GB of space is consumed from 300 GB of hard disk drive (HDD) capacity. If an administrator queries free space, the system could respond with 50 GB. The administrator, though, is likely more interested in the question, "How much more can be written?"

The answer depends entirely on how much deduplication will be achieved on future writes. A client could fill the system by writing 50 GB of new content, so 50 GB of free space is a conservative answer. Alternatively, a client writing highly redundant data could write many multiples of 50 GB. For a client writing files with consistent redundancy patterns, the number of new files that can be stored can be approximated by dividing the remaining space by the incremental capacity used by each file [2]. Unfortunately, one of our customers mistakenly thought that all workloads would have similar deduplication ratios and promptly filled their system with non-deduplicable data. Space estimation must strike a balance between preventing a system from becoming full and overwhelming a customer with unnecessary alert messages. The best approach is to provide multiple ways to track space usage: report 50GB of free space as well as estimates of how much more can be written.

As a storage system becomes full, it is also natural for an administrator to ask, "What should I delete to free space?" The answer may be driven by regulatory compliance, internal policies, or the importance of various files. The properties of deduplication lead to a follow-up question, "How much space will I free by deleting a file?" Deleting a file that is logically (*i.e.* from the writer's perspective) 10 GB may not result in freeing 10 GB of capacity. Consider File 1 in Figure 2 that refers to three chunks also referenced from File 2. Deleting File 1 will not free any chunks, though it will free a small amount of meta data associated with the representation of File 1 in the system. Customers do not want backup failures, so when a system is getting full, they immediately ask how much space files take. One of our customers attempted to free space by deleting backups, which not only failed to free much space but also created a compliance failure.

To better support capacity planning, a deduplicated storage system should provide detailed space usage reports. One simple (but incorrect) strategy is for a system to track, as a file is written, how many bytes are deduplicated versus stored. While that value is true at that moment in time, storage is dynamic. Suppose that File 1 is written followed by File 2. At that moment, chunk $C5$ is unique to File 2, so that is the incremental storage needed for File 2, which could be recorded. If File 1 is deleted, now File 2 is the only file referencing chunks $C1$, $C2$, and $C3$, so the amount of space returned when deleting File 2 changes as other files are written and deleted. It is also valuable to identify data that does not deduplicate, so that it can be transferred to potentially less costly, non-deduplicated storage.

When an administrator has multiple deduplicated systems to balance, she might like to know both how much space will be freed by removing data from a source node and much space will be used on a target node [3]. While the discussion has focused on system capacity, a related topic is how to maintain deduplication benefits as data moves between tiers such as flash, HDDs of various speeds and capacities, and cloud storage.

Future Research Opportunities: An offline process could periodically calculate the unique content for each file, or such statistics could be calculated during garbage collection. An inline process could provide hints about capacity that are at least accurate within an approximation threshold. A new interface may be needed for administrators to query the system to determine how much space can be freed by deleting one or more files.

3 Quality of Service

While deduplication can decrease the cost of storage, a storage system must also meet the quality of service (QoS) requirements for a client. We use the term QoS in a broad sense to encompass a variety of topics covering latency and throughput goals requested by an administrator for a range of priority configurations. While there has been extensive research on QoS for non-deduplicated storage systems [11], there has been little for deduplicated storage [12].

Deduplication adds additional levels of indirection to map from a file representation to data chunk locations, and deduplication tends to turn sequentially written content into references to chunks scattered across the HDDs. Figure 3 shows a typical deduplication architecture with access protocols at the top, a RAM (and possibly flash) cache for important data and meta data, and a HDD system. Structures on HDD include a file representation consisting of an array of fingerprints for a file's data chunks, chunks stored in multi-megabyte containers, and an index that maps from fingerprint to container.

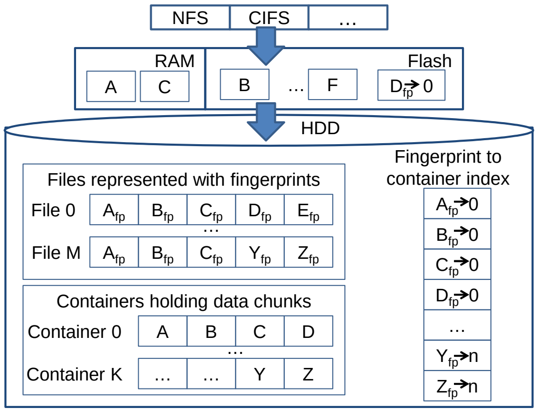

Figure 3: A deduplication architecture with levels of indirection and shared content that complicate QoS goals.

Reading a portion of a file involves accessing the file representation (not necessarily laid out as sequentially as in this example), the fingerprint index, and then reading the chunk. In this example, an administrator may expect that for File M, chunks *C* and *Y* are sequential, though they are actually stored in different containers. Because of complicated container layout and caching effects, it is difficult to predict how many disk I/Os are needed for a client request. Though there has been work on improving restore performance [6], latency and throughput requirements were not supported. One new customer experienced higher than promised performance until more users and data were added, when the performance decreased to the expected levels, and they complained. This highlights the importance of predictable performance even if it means throttling potential peaks.

Next consider the QoS technique of caching when there are multiple clients with different priority levels. Consider two clients, `High` reading File 0 and `Low` reading File M in Figure 3. A simple QoS approach is to assign cache space in RAM to the two clients relative to their priorities. `High` could receive twice as much cache space as `Low`. Due to redundant content (chunks *A* through *C*), though, content brought into the cache to serve `High`'s read requests (and counting against `High`'s time/work quota) can also serve `Low`'s reads. This can lead to an nonintuitive performance result where `Low`'s latency is better than `High`'s. There is a pathological case where `Low` reads the same data blocks immediately after `High` was charged for the work of bringing chunks into the cache. Then `High` may use up its quota, while `Low` has available quota to perform other work. The priority level can even change over time for a single file, since a file written as a low priority backup may be read at a later time as a high priority restore.

Deduplication before or during network transfer affects network utilization since logical bytes transferred may differ from physical bytes transferred. One service provider wanted differentiated backup and replication performance for each tenant since backup and replication windows were tenant-configured.

Besides client initiated I/O, deduplicated storage systems often have resource-intensive background tasks that need QoS control. Garbage collection is one such process that involves determining which chunks are referenced from live files versus which chunks can be freed. Other asynchronous tasks include off-site replication and integrity verification. More than one customer has been in a situation where deleted data could not be garbage collected fast enough for new writes to be stored or replication did not complete quickly enough. There may be opportunities to reorder the processing of data to serve both background tasks and client I/Os.

Future Research Opportunities: A QoS system must be aware of the potential latency of various operations and have sufficient resources to meet the overall goals. Accounting for shared content across clients is likely a policy decision with multiple reasonable options. Work/time for retrieving a shared chunk could be charged to the first client to access the chunk or the cost could be shared over all clients that access the chunk, though maintaining accurate accounting is complex. Making resource-intensive tasks more efficient will directly reduce over-provisioning. Finally, to test QoS, customer traces and synthetic load generators are needed with realistic deduplication patterns.

4 Security and Reliability

Deduplication raises new concerns when balancing security with space savings. To maintain security during deduplication, the idea of using a hash of the chunk as the encryption key has been explored [1]. This ensures that repeated chunks will encrypt to the same byte string and deduplicated while preserving privacy. Because of the computational costs of encryption, it is an interesting future path to perform deduplication before encryption, though the possibility of information leakage must be addressed. By timing data transfers, it may also be possible to infer what already exists on a deduplicated server [5]. End-to-end security must consider whether data has to be decrypted and re-encrypted multiple times when transferring from client to server to local storage and then to a remote system, while preserving deduplication benefits. A service provider wanted to achieve the space savings of deduplication across tenants, while their tenants wanted to control and periodically change their own keys.

An area that has received less attention is the complex relationship between deduplication and data reliability [10]. By reliability, we mean the likelihood that needed data will be available when requested, as compared to security which involves preventing unauthorized access, knowledge, and tampering. One of the central tenants of data reliability is to create multiple copies to protect against data loss, while deduplication removes redundancies. In particular, backup and archival systems–designed to protect data–were among the earliest adopters of deduplication because of the inherent redundancy. And yet, when people first learn of deduplication, they often question the risk of data loss.

To increase data reliability, deduplicated systems typically offer several techniques. First, RAID can protect a disk array against failures. Second, numerous versions or snapshots can be retained since they only require the incremental space for modified content. Third, data is often replicated off-site, and since only the unique data needs to be replicated, replication can complete faster than transferring a full data set. Intuitively, the combination of RAID, versioning, and replicating counterbalances a risk of data loss due to deduplication, but analyzing reliability quantitatively is an open question.

For example, one might analyze whether deduplicated data on a RAID storage array is any more or less safe than retaining multiple copies of data on a non-RAID array. The same comparison could be made to a cluster that retains k copies as this is a common approach. To phrase the problem more broadly, "How safe is data within a *storage environment*?" We use the term storage environment, as multiple storage systems may be part of the analysis. Since deduplication reduces data sizes, it becomes feasible to replicate entire data sets to remote location(s), so the reliability of those systems is part of the reliability solution. Also, in some cases a non-deduplicated primary storage system may be included in analysis since it may have snapshots and/or be integrated with deduplicated, secondary storage. While there is ongoing research on the failure rate of disks and RAID systems [7], future work is needed on end-to-end reliability including deduplication effects.

Future Research Opportunities: One option is to complete an empirical measurement of data loss rates for various flavors of deduplicated storage, if storage vendors will release such information. A second option is to model the risk of each component of the system using published failure characteristics and calculate the reliability for the storage environment. Then, techniques to increase reliability can be explored.

5 Management

While research often focuses on increasing storage capacity and performance, a critical feature for administrators is manageability, and deduplicated storage adds new wrinkles for storage administrators to consider. While numerous sub-problems fall within the management topic, we focus on sizing, migration, and reporting using a hypothetical storage administrator as an example.

One of the first question our storage administrator considers when buying a storage system is, "Does the capacity fit my organization's needs?" As discussed in §2, deduplication complicates capacity calculations, and an administrator needs a mechanism to estimate deduplication benefits [4]. Some storage vendors provide tools to analyze the amount of deduplication on an administrator's current data sets, but such tools have limitations. Analysis tools may be slow to run, can require an unreasonable amount of system resources (IOPS, bandwidth, network, etc.) while analyzing data, and may not be allowed to analyze confidential data due to data leakage concerns. An administrator may then wish to partition the space among multiple internal applications, and the sizing problem returns at a smaller granularity. Partitioning storage is itself a new content-sharing problem involving technical and policy questions.

Once the storage administrator has received and partitioned deduplicated storage, she next wishes to migrate data off of a retired system. If the retired system is non-deduplicated, it can take days to transfer hundreds of terabytes of data. On the other hand, if the retired system is deduplicated, there is the potential to transfer post-deduplication content. While migration is often supported when the retired and new systems are the same product, migration between different products is often impossible. Unfortunately, the chunking and hashing algorithms may be incompatible between different products, and such details are often proprietary, with little incentive for rival companies to standardize migration. We have unfortunately disappointed some customers who wished to migrate post-deduplicated bytes from another deduplicated system to our own, which was not supported due to implementation differences.

Finally, the storage administrator has the system installed and loaded with data. Her next concern is measuring the ongoing health of the system through reporting, which is used to manage the system and demonstrate that storage dollars are well spent. While reporting includes deduplication issues previously discussed, additional levels of detail are needed. Reports should specify capacity and performance for storage volumes, which is complicated by the shared nature of deduplication. Deduplicated storage has more configurable "knobs" to optimize than standard storage, so reports should provide enough detail and context that problems, if they arise, can be understood and resolved.

Potential solutions: Tools for sizing deduplicated storage systems must become more efficient in terms of resource requirements and running time. For migration, there is the potential for standardization across vendors or tools to transform one deduplication format to another. To improve reporting tools, further input from administrators is needed as well as internal mechanisms to measure the behavior of deduplicated storage at the granularity administrators need. Finally, management for deduplicated storage will continue to be influenced by functionality supported by non-deduplicated storage systems.

6 Chargeback for Service Providers

Service providers create some of the most difficult challenges for deduplicated systems. By consolidating multiple tenants onto one environment, they push the boundaries of most systems in terms of QoS, security, and management at scale. A new, and fundamentally challenging problem, is chargeback. Ultimately, a service provider can succeed only if it can charge its tenants effectively and appropriately. First, there is the question of what services can be billed to the tenant. While most organizations charge solely by capacity, others would like to charge for a larger range of resources (CPU, IOPS, and storage and network bandwidth), as well as by service level (response time, security, reliability, etc.).

Second, even when looking to charge by capacity, they struggle. While the service provider could simply bill for logical bytes transferred and stored, the provider may wish to pass along the savings due to deduplication to the tenant. Otherwise, a tenant could save money by purchasing their own deduplicated storage system. Furthermore, users who know they have data that deduplicates at a higher ratio resent being overcharged. The question becomes "How can you effectively charge a deduplicated price?" One service provider initially signed up tenants to three year contracts with an assumption of 8X[1] space savings but found itself losing money when space savings were lower in practice. Other service providers try to bucketize deduplication ratios for their tenants: tenants with 2-4X deduplication get charged one rate, tenants in the 4-8X range get another rate, and so on. This may simplify the calculation of exact space usage, but during an audit, a service provider must prove a tenant's data falls in the assigned bucket, and some tenants believe that random sampling and estimates are insufficient. Still other service providers have taken the alternative of calculating each user as existing in its own deduplicated space. Unfortunately, this requires service providers to truly separate all their tenants and lose any benefit of cross-tenant space efficiency.

Third, there is the issue of timeliness. Since tenants are directly paying for storage usage, they may expect more immediate feedback than employees within a company sharing internal storage. As an example, capacity reporting across employees in a company could be determined asynchronously, perhaps weekly, due to complexities of determining usage per employee. Tenants may want to track their usage as storage operations take place, so faster approaches are needed.

Future Research Opportunities: For service providers to succeed with deduplicated offerings, they need accurate measurements for post-deduplication resource usage (capacity, I/O, network, etc.). This may involve per-tenant rolling estimates. Techniques are also needed to assign/reassign tenants to deduplicated storage to maximize deduplication and meet QoS requirements [3].

7 Discussion

Deduplicated storage is a maturing field with numerous publications and commercially available products. While the research community has largely focused on a small set of problems (space savings, performance, etc.), and those topics will continue to yield innovation, there are numerous, novel problems we need to address so that deduplicated storage can become more fully functional. From conversations with customers and engineers, we have identified five broad problem areas: capacity, QoS, security and reliability, management, and chargeback for service providers. These problem areas are not intended to be exhaustive, but will hopefully motivate the next generation of research and deduplication products.

8 Acknowledgments

We thank Stephen Manley for his valuable feedback.

References

[1] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. P. Wattenhofer. FARSITE: federated, available, and reliable storage for an incompletely trusted environment. In *OSDI*, 2002.

[2] M. Chamness. Capacity forecasting in a backup storage environment. In *LISA*, 2011.

[3] F. Douglis, D. Bhardwaj, H. Qian, and P. Shilane. Content-aware load balancing for distributed backup. In *LISA*, 2011.

[4] D. Harnik, E. Khaitzin, and D. Sotnikov. Estimating unseen deduplication–from theory to practice. In *FAST*, 2016.

[5] D. Harnik, B. Pinkas, and A. Shulman-Peleg. Side channels in cloud services: Deduplication in cloud storage. *Security & Privacy, IEEE*, 8(6):40–47, 2010.

[6] M. Lillibridge, K. Eshghi, and D. Bhagwat. Improving restore speed for backup systems that use inline chunk-based deduplication. In *FAST*, 2013.

[7] A. Ma, R. Traylor, F. Douglis, M. Chamness, G. Lu, D. Sawyer, S. Chandra, and W. Hsu. RAIDShield: characterizing, monitoring, and proactively protecting against disk failures. *ACM TOS*, 11(4):17, 2015.

[8] J. Paulo and J. Pereira. A survey and classification of storage deduplication systems. *ACM Computing Surveys*, 47(1):11, 2014.

[9] S. Quinlan and S. Dorward. Venti: A new approach to archival data storage. In *FAST*, 2002.

[10] E. W. Rozier, W. H. Sanders, P. Zhou, N. Mandagere, S. M. Uttamchandani, and M. L. Yakushev. Modeling the fault tolerance consequences of deduplication. In *IEEE Symposium on Reliable Distributed Systems*, 2011.

[11] D. Shue, M. J. Freedman, and A. Shaikh. Performance isolation and fairness for multi-tenant cloud storage. In *OSDI*, 2012.

[12] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti. iDedup: latency-aware, inline data deduplication for primary storage. In *FAST*, 2012.

[13] B. Zhu, K. Li, and H. Patterson. Avoiding the disk bottleneck in the Data Domain deduplication file system. In *FAST*, 2008.

[1] logical bytes divided by post-deduplication bytes