The days of
overprovisioned,
underutilized
storage resources
might soon become
a thing of the past.

Storage Virtualization Gets Smart

KOSTADIS ROUSSOS, NETWORK APPLIANCE

Over the past 20 years we have seen the transformation of storage from a dumb resource with fixed reliability, performance, and capacity to a much smarter resource that can actually play a role in how data is managed. In spite of the increasing capabilities of storage systems, however, traditional storage management models have made it hard to leverage these data management capabilities effectively. The net result has been overprovisioning and underutilization. In short, although the promise was that smart shared storage would simplify data management, the reality has been different.

To address the real challenges of data management and the shortcomings of traditional storage management, we propose a new data management framework based on three observations:

- Data management is best performed by the owner of the data—the data administrator.
- The administrator of the physical storage must be able to control what the data administrator can do.
- Enabling the data administrator to control the data requires tools that abstract out the details of the storage infrastructure, because without such tools we are asking data administrators to manage storage infrastructure.

Storage Virtualization Gets Smart

At NetApp we have built such a framework and have found that it dramatically simplifies management of storage infrastructure by presenting storage management in the context of the data management tasks that the data administrator wants to perform.

In this article we point out the business and data management trends that led us to reject traditional models for data management. We then describe the data management framework that we built and the first products to leverage that framework.

COST AT THE RIGHT SERVICE LEVEL

We have seen a move to more shared infrastructure in an effort to reduce cost. That trend, along with the changing needs of data management, ultimately affects how data and storage management is actually done.

A major IT challenge is how to provision functional services such as e-mail with the right data service attached to that functional service at the right cost. Or to put it differently, e-mail has to have the right number of copies, the CEO's e-mail must not be deleted, and the e-mail service has to tolerate and recover from a variety of failures without breaking the budget. Failure to do so will result in downtime—and that downtime, depending on the day of the month, can affect business.

Dedicating physical resources to each functional service is fundamentally an unsustainable model. The cost in terms of equipment and physical space makes that impossible. Furthermore, existing equipment is underutilized, so the business tries to get more out of it before buying new equipment.

Leveraging shared resources requires some form of virtualization, leading to the adoption of server and storage virtualization. Server virtualization is used to consolidate many underutilized physical servers onto a smaller number of physical servers. Storage virtualization is used to create storage containers that are either bigger or smaller

than the underlying physical disks, thereby improving disk utilization. Like server virtualization, storage virtualization can do more than just improve utilization. Storage virtualization can be the basis for transparent data migration, data replication, thin provisioning, and space- and time-efficient backups (called snapshots).¹ Unlike server virtualization, the full promise of storage virtualization has not yet been realized because of two distinct challenges: the limitations of the underlying technology and the complex administrative handshake required to leverage the features of the virtualization.

As an example of the impact of these challenges, consider the e-mail application that requires backups. The business would like to have backups done once an hour. The reality is that unless those backups are done in a space- and time-efficient manner, the cost is prohibitive. The storage architect, who has already virtualized the storage, observes that the storage system provides a space-efficient backup scheme that meets the application requirement. These backups (snapshots), which take a few minutes to complete, can be performed once an hour. They do not affect the application performance and consume only the amount of space that has actually changed. The storage architect therefore recommends to the business and application administrators that they use these snapshots.

The storage architect now runs into two distinct problems. The first is that the snapshots are not free because the storage system may have systemwide limits on how many snapshots can be taken. The storage architect therefore has to ensure that no single application administrator takes too many snapshots. The second problem is that application administrators typically want application-consistent snapshots, requiring coordination across the application, server, and storage. Coordination in practice requires coordination across multiple administrative groups that understand in detail the complexity of their infrastructure.

Suppose the business can survive with backups performed every 12 hours and is willing to tolerate some degradation in performance during those backups. Rather than deal with the complex coordination effort, the application administrator may be able to convince the business to buy more storage and use the less space-efficient application-based backup. This type of backup may be less space efficient because it is a full copy of the data followed by a compression step. With application-based backup, there is no coordination effort to limit the number of snapshots, nor is there any need for complex negotiation between the storage, server, and application

groups. The complexity of the administration of the storage infrastructure makes it worthwhile to trade that off against the simplicity of a less efficient mechanism. As a result of this tradeoff, the application is overprovisioned in terms of capacity and underprovisioned in terms of data protection.

Without effective storage virtualization and the ability to leverage that storage virtualization, we have the worst of all possible worlds. We have increasingly capable infrastructure that requires increasingly more sophisticated administrators, making it increasingly difficult to leverage the capabilities of that infrastructure. To compensate for this administrative complexity, we buy more storage. This, of course, creates a different challenge of how we use what we have more efficiently. Which brings us back to wanting to use virtualization...

THE CHANGING NATURE AND NEEDS OF DATA MANAGEMENT

Traditionally, data management was about ensuring that the active copy of the data was available and that some number of replicas could be accessed. The problem was relatively simple and straightforward. The primary storage system was monitored for its health. The backup software tracked the number of copies that existed on tape.

This approach to data management can be described as using the *infrastructure as a proxy* for managing the data. It assumes that as long as you understand what the infrastructure is doing, you can infer how the data is being managed.

Using the infrastructure as a proxy worked fine as long as the number of infrastructure components that represented a single LUN (logical unit number) of data was small. Three technology trends and two regulatory trends broke that assumption.

The first technology trend was the emergence of the disk as a viable technology for storing replicas. Disks, unlike tape, can be used for both read-only and read-write data. In fact, the difference between a read-only copy and a read-write copy is the semantics of the copy, not a property of the medium. As a result, replicas began to be seen as read-write copies that could be used for other purposes.

The second technology trend was that the storage systems themselves became smarter. For example, the ability to create a space-efficient writable copy, such as NetApp FlexClones,² changed the economics of creating multiterabyte copies, enabling use cases that were fundamentally too expensive with full copies. In particular, in test and development environments, the ability to create copies of multiterabyte databases at a fraction of the cost and in

a fraction of the time changes the economics of database development.

Both of these technology trends led to the creation of more copies of data. Unlike tape, disk-based copies can be used for purposes other than just restore; consequently, the copies have their own data management policies. Because the copies depend on the original, however, managing the logical data unit requires management of the primary copy and all of the related copies. Thus, for any single logical unit of data the number of different storage containers increases, breaking the assumption underlying the infrastructure-as-proxy approach to data management—namely, the number of storage containers per LUN is small.

The third technology trend was the emergence of storage virtualization itself. This made it possible to create far more storage containers on the same number of disks. Furthermore, storage virtualization made it far easier to create storage containers.

The first regulatory trend that changed the approach to storage was the need to track data. The ability to know where data is and who has access to it, and the ability to

securely delete data became critically important. Managing at the infrastructure level becomes constrained as the numbers of distinct bits of infrastructure that contain data have to be tracked. This trend, when combined with the proliferation of writable replicas, increases the number of storage containers that have to be managed for any single LUN, thus breaking the assumption underlying the infrastructure-as-proxy approach to data management.

The second regulatory trend was that the data itself had a life span that transcended the lifetime of any storage system. Disks and storage controllers must be replaced

Storage Virtualization Gets Smart

after 10 years. Medical data must be available for 90 years. Using the infrastructure to manage the data becomes problematic as the infrastructure itself changes over time.

As a concrete example of the limitations of using infrastructure as a proxy for data management, consider an Oracle database on a SAN (storage area network). Suppose the Oracle database uses three different LUNs to store different components. Suppose the database needs to be replicated to a remote site to protect against a site disaster. The single Oracle database now uses three more LUNs on the remote system. These secondary LUNs and the three replication relationships must be monitored and tracked.

Furthermore, suppose that on the remote site we create 10 clones of an Oracle database for test and development, thus creating 30 new LUNs. If we were to use the infrastructure to manage the data, we would have to manage 36 LUNs, three replication relationships, and two storage systems for a total of 39 distinct objects, when all that is really being managed is a single Oracle database and its replicas.

ISOLATIONISM AND INFRASTRUCTURE MANAGEMENT
Confronted with the complexity of how to leverage the storage infrastructure, storage architects are presented with two basic approaches: isolationism and infrastructure management.

Isolationism, as typified by ILM (information lifecycle management) or HSM (hierarchical storage management), tries to solve data management by automatic classification of data. The idea behind ILM or HSM is that you can automatically understand the value of the data by inspecting it. Once you understand the value, you can apply policies to the data and thus solve the entire data management problem. The flaw in this model is that although the storage system may contain the data, the data is incomplete because it has value in a business context that cannot be inferred by a simple inspection.

Infrastructure management, as typified by SRM (storage resource management), tries to simplify the problem by cataloging the entire infrastructure. The hypothesis is that if you understand the infrastructure, you can then use the infrastructure as a proxy to manage the data. That approach founders when confronted with the complexity of the SRM space and the fact that the infrastructure as a proxy forces an administrator to manage a single Oracle database by managing 223 distinct components.

UNIFYING STORAGE AND DATA MANAGEMENT

To address the basic challenge of how to effectively exploit storage virtualization without requiring an increasingly large staff to manage that virtualization, we at NetApp took a different approach that we called IDM (integrated data management).³ The goal of this approach is to enable data administrators to manage their data directly, while simultaneously giving the storage architects ultimate control over what anyone could do with the resources they controlled. We call it integrated data management because it integrates data and storage management through automation, and applications and data management through application-specific integration.

The IDM model has three layers. The uppermost layer presents the data service to the data administrator in the context of the application. For example, we provide tools that allow an Oracle database administrator to back up the database directly, and those tools understand how to map the database to the underlying storage containers and storage operations.

Next is a set of new data management abstractions and an automation layer consisting of a role-based access control mechanism, a conformance engine, and a central repository that contains the definitions of these objects.

The lowest layer of the IDM model consists of pervasive storage virtualization, which enables storage infrastructure reconfiguration.

An analogy that we use to explain the IDM model is the ATM. With an ATM, the bank can delegate to the customer the right to withdraw money at any point, but the bank retains ultimate control over how much money can be withdrawn and in some cases the cost of a withdrawal. Using ATMs, banks were able to reduce the amount of cash that was withdrawn, increasing their overall assets. In effect, banks improved their asset utilization while reducing their costs and improving customer satisfaction.

APPLICATION INTEGRATION

Application Integration in an IDM context differs from traditional SRM. In IDM, the goal is to integrate the

application into our data management infrastructure so that the application administrator can control the data management policies of the data. In traditional SRM, application integration is about providing host agents that enable storage administrators to control or at least monitor the storage use of applications. The difference is in the intended audience and its function.

At NetApp we have delivered several products, including SMO (SnapManager for Oracle), which demonstrate the value of this form of application integration. Database administrators use SMO to clone their databases or perform hot backups in a way that leverages the underlying NetApp storage virtualization. SMO, in effect, maps the database and the desired operations on the database—clone or hot backup—into a sequence of operations on the database, server, and storage system, enabling the database administrator to manage the storage virtualization directly without becoming a storage expert.

An ongoing challenge with application integration is coverage. Ultimately, the success of this approach will depend on how easily application vendors or administrators can do the integration.

DATA MANAGEMENT ABSTRACTIONS AND AUTOMATION

An important aspect of IDM is the introduction of a new set of data management objects and the corresponding automation layer that the application and data administrators interact with. The intent of these new objects and automation layer is to abstract out the details of the storage infrastructure and enable higher-level tools or users to interact with their data without having to learn the details of the infrastructure. In addition, this is the layer that provides the control over the storage resource.

The data management abstractions are the dataset, policy, and resource pool. The *dataset* represents a collection of data and all of its replicas, or alternatively, a collection of files or structured data and their replicas. It refers to the data contained within a set of storage containers, such as LUNs, and not the storage containers themselves. A dataset is the basic handle to the data used by all data administrators or applications that integrate with the framework. A *policy* describes how the storage containers that contain a dataset should be configured. This configuration specifies both how the dataset should be provisioned and how it should be protected. A *resource pool* is a collection of physical storage resources managed by storage administrators.

Automation is provided through a conformance engine, which performs the crucial task of ensuring that

the dataset policy and the actual configuration of the storage that contains the dataset match. The conformance engine continually monitors the storage infrastructure, comparing the dataset policy with the current configuration of the storage. If there is a mismatch, then the conformance engine can reconfigure the storage to satisfy the policy. For example, if a dataset is not being replicated, and the policy requires that it be replicated, the conformance engine will create storage objects on the secondary system and configure replication between the primary and secondary systems.

A flexible RBAC (role-based access control) system controls which operation can be done on which object by which administrator.⁴

Consider the e-mail example described earlier. The storage architect first configures two resource pools: one for primary storage and another for remote storage. The storage architect then creates a policy called *e-mail server* that describes how the dataset is to be replicated and provisioned. To provision a new e-mail server, the data administrator simply selects the e-mail server policy, configures some per-dataset-specific parameters such as name and size, and then selects the appropriate resource pools for the primary and remote storage. The conformance engine then creates the primary and secondary storage objects and configures the replication relationship. Later, if the storage architect changes the e-mail server policy—for example, increasing the frequency of replication—the conformance engine will reapply the policy to the storage, changing the replication parameters.

PROTECTION MANAGER

NetApp's first system to incorporate datasets is Protection Manager.⁵ Using Protection Manager, storage architects can define policies that describe how datasets should be protected. These policies can then be delegated to backup administrators who can in turn use them to protect specific datasets. The conformance engine in Protection Manager ensures that the datasets are correctly protected and can notify the storage architect or backup administrator of any specific issues such as out-of-space or failed backups.

CONCLUSIONS

The changing needs of business and data management are driving the IT industry to find better ways to manage data. Storage virtualization simplifies data management because it allows dynamic reconfiguration of the shared storage resource. Without the right management infrastructure between the storage virtualization and the owners of the data, however, the leverage will be lost.

Storage Virtualization Gets Smart

Traditional storage management models are inadequate because they either focus exclusively on infrastructure or do not enable the owner of the data to manage the data.

Integrated data management is an approach we have taken at NetApp to leverage the virtualization in our core platform to simplify data management. This approach has yielded some promising early results. There is more work to be done, however. Q

REFERENCES

1. Hitz, D., Lau, J., Malcolm, M. 1995. File system design

for an NFS file server appliance. Network Appliance Technical Report, TR-3002 (March).

2. Introduction to Data ONTAP 7G. Network Appliance Technical Report, TR-3356, October 2005.

3. Joshi, V.S., Chalaka, R. 2006. Integrated data management: an innovative approach to improved IT productivity and flexibility. Network Appliance Technical Report, TR-3467 (April).

4. Ferraiolo, D.F., Kuhn, D.R. 1992. Role-based access control. 15th NIST-NCSC National Computer Security Conference: 554-563.

5. Desai, J. 2007. NetApp Protection Manager protecting your data: policy-driven data management. Network Appliance Technical Report, TR-3524-0207 (February).

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

KOSTADIS ROUSSOS (<http://www.krung.net/blog>) is a technical director at Network Appliance, where he spends most of his time working on products that leverage storage virtualization and application integration to simplify data management. He is currently focused on architecting tools.

© 2007 ACM 1542-7730/07/0900 \$5.00