VIPR SERVICES STORAGE ENGINE ARCHITECTURE WHITE PAPER

Deploy a modern hyperscale storage platform on commodity infrastructure

ABSTRACT

This document provides a detailed overview of the ViPR Services architecture. ViPR Services is a geo-scale cloud storage platform that delivers cloud-scale storage services, global access and operational efficiency at scale. This abstract appears as the online abstract for EMC.com.

September, 2014

REDEFINE

EMC WHITE PAPER

EMC²

To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, contact your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the EMC Store

Part Number H13518

TABLE OF CONTENTS

EXECUTIVE SUMMARY

According to the 2014 IDC EMC Digital Universe Study, the digital universe is doubling in size every two years. The amount of data we create and copy will reach 44 zettabytes (44 trillion gigabytes) by 2020. Over two thirds of that 44 zettabytes is unstructured data generated by end users, yet 85% of that user-generated content is managed by an enterprise IT department at some point in its lifecycle. This explosive data growth is forcing companies of all sizes to explore more efficient data storage strategies.

It's more than data growth however, that mandates a new approach to information storage. The universe of data is becoming more diverse and valuable—it's not just spreadsheets and word documents. Seemingly everything is connected to a network and creates and consumes varying amounts of data. Your thermostats, your running shoes, your wristband, your glasses, traffic lights – you name it – every device is now a smart device with information that can provide valuable insight into our interconnected world or deliver a competitive edge. Businesses are using new development frameworks and analytics platforms to rapidly build new mobile, cloud and information-based applications that can seamlessly and quickly access and analyze globally distributed data.

Data growth and modern applications mandate that businesses evolve their information storage to address three critical imperatives:

- **Deliver storage services at cloud scale** – Both enterprises and service providers increasingly need to be able to scale seamlessly across data centers and protect critical data globally.
- **Meet new application and user demands** - New applications and users demand instant access to globally distributed content from any device or application anywhere in the world.
- **Drive operational efficiency at scale** - Storage systems must efficiently store petabytes of unstructured data of varying types and optimize storage utilization.

EMC ViPR Software-Defined Storage is a unified storage platform that provides a single control plane for multi-vendor, heterogeneous storage systems. The ViPR Software-Defined Storage platform forms the infrastructure that hosts ViPR Services, which support object, HDFS, and block storage on commodity hardware.

This white paper provides a technical overview of the ViPR Services unstructured storage engine architecture. ViPR Services is a purpose-built geo-scale cloud storage platform that features patent pending technology that meets today's new requirements for cloud-scale storage services, modern applications with global access requirements and operational efficiency at scale.

AUDIENCE

This white paper is intended for solution architects, CIOs, storage administrators and developers and application architects. It provides a detailed technical discussion of the ViPR Services architecture, geo capabilities and multi-site access.

VIPR SERVICES UNSTRUCTURED STORAGE ENGINE

ViPR Services support the storage, manipulation, and analysis of unstructured data on a massive scale on arrays and commodity environments. ViPR features an unstructured storage engine that supports object and HDFS services today and, in the near future, file services. ViPR also includes a block storage engine designed for low-latency block workloads.

- **ViPR Object**

  The ViPR Object Service is a software layer that works across various hardware platforms. It enables you to store, access, and manipulate unstructured data as objects on commodity-based systems, such as the HP SL4540 and on EMC and non-EMC file-based arrays. The Object Service is compatible with EMC Atmos, Amazon S3, and OpenStack Swift APIs.

- **ViPR HDFS**

  The ViPR HDFS Service provides support for Hadoop Distributed File System (HDFS). ViPR HDFS enables organizations to run ViPR Services on commodity-based hardware and build a big data repository at scale. With the HDFS Service, you can use the ViPR storage environment as a big data repository against which you can run Hadoop analytic applications.

- **ViPR Block**

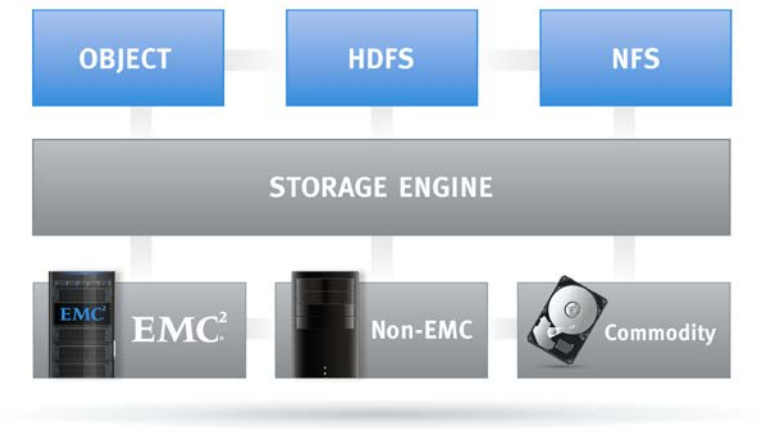
  The ViPR Block Service, powered by ScaleIO, creates a server-based SAN from local application server storage to deliver elastic and scalable block storage. ViPR Block is an alternative to a traditional SAN infrastructure. It combines hard disk drives (HDDS), solid-state drives (SSDs), and Peripheral Component Interconnect Express (PCIe) flash cards to create a virtual pool of block storage with different performance tiers. ViPR Controller can automate provisioning of block storage pools and make the block pools available within the ViPR service catalog. ViPR Block scales to thousands of commodity nodes and automatically

redistributes block storage to optimize performance and capacity usage. ViPR Block on an ECS Appliance is designed to provide general-purpose backend storage for customer-deployed scale-out application infrastructure.

This paper details the architecture of the ViPR Services unstructured storage engine. The unstructured storage engine is the primary component that ensures data availability and protection against data corruption, hardware failures, and data center disasters. The engine exposes multi-head access across object and HDFS and allows access to the same data concurrently through multiple protocols. For example, an application can write object and read through HDFS or vice versa. Today ViPR supports object and HDFS interfaces. In the future it will also support for NFS and CIFS.

Multi-head access is built on a distributed storage engine that provides high availability and scalability, manages transactions and persistent data and protects data against failures, corruption and disasters.

Figure 1. The ViPR Services Unstructured Storage Engine



STORAGE ENGINE DESIGN AND OPERATION

The ViPR storage engine writes object-related data (such as user data, metadata, and object location data) to logical containers of contiguous disk space known as chunks.

Design Principles

Key design principles of the ViPR storage engine include the following:

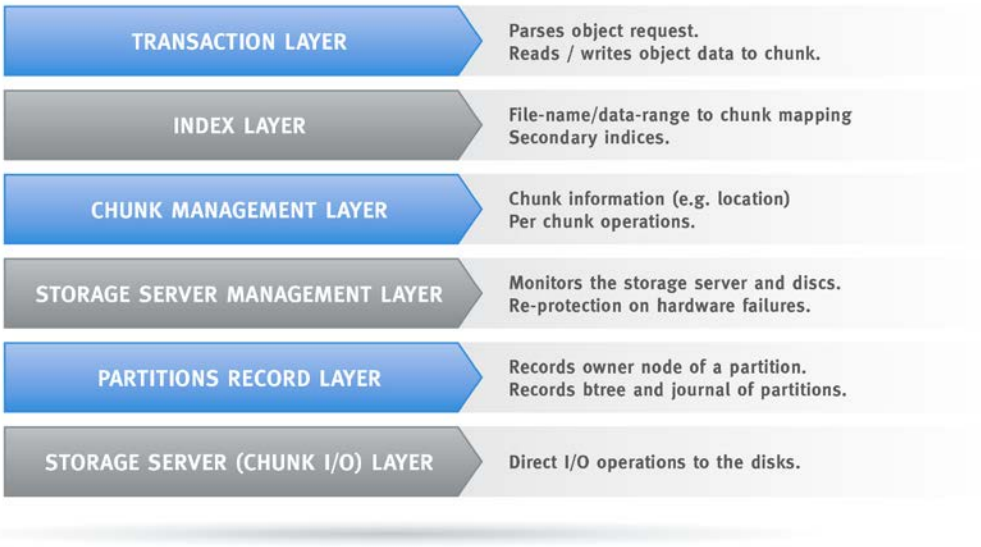
- Stores all types of data and index information in “chunks”
  - “Chunks” are logical containers of contiguous space (128MB)
  - Data is written in an append-only pattern
- Performs data protection operations on chunks
- Does not overwrite or modify data
- Does not require locking for I/O
- Does not require cache invalidation
- Has journaling, snapshot and versioning natively built-in

The storage engine writes data in an append-only pattern so that existing data is never overwritten or modified. This strategy improves performance because locking and cache validation are not required for I/O operations. All nodes can process write requests for the same object simultaneously while writing to different sets of disks.

The storage engine tracks an object’s location through an index that records the object’s name, chunk ID, and offset. The object location index contains three location pointers before erasure coding takes place and multiple location pointers after erasure coding. (Erasure coding is discussed more later in this document.) The storage engine performs all storage operations (such as erasure

coding and object recovery) on chunk containers. The following diagram illustrates the various services and layers in the storage engine and their unique functions.

Figure 2. ViPR Services Storage Engine Service Layers



Each layer is distributed across all nodes in the system and is highly available and scalable. This unique storage engine architecture has the following unique capabilities:

- All nodes can process write requests for the same chunk simultaneously and write to different sets of disks.
- Throughput takes advantage of all spindles and NICs in a cluster.
- Payload from multiple small objects are aggregated in memory and written in a single disk-write.
- Storage for both small and large data is handled efficiently with the same protection overhead.

This unique architecture brings new levels of performance and efficiency for the storage of massive volumes of unstructured data.

VIPR WRITE, READ, AND UPDATE PATHS

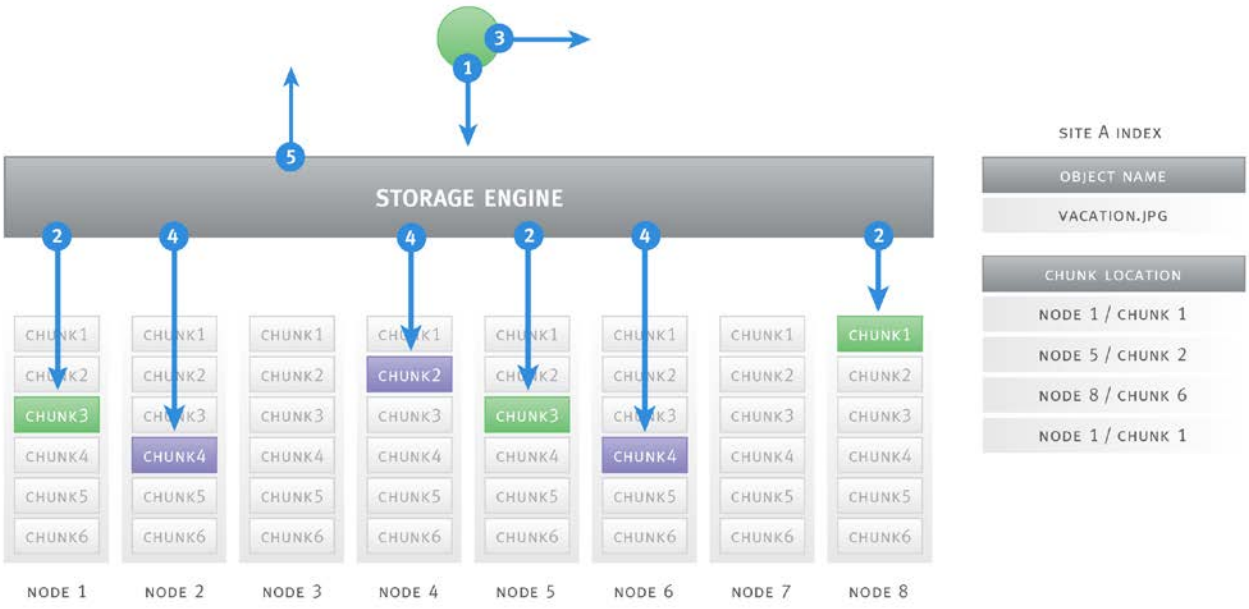
ViPR Write Path

This section illustrates the ViPR commodity data flow when a user submits a create request. In this example, illustrated in Exhibit 3, a user submits a write request through one of the supported interfaces and the storage engine stores the object's data and metadata on nodes and disks within a single site. The detailed steps follow.

1. An application submits a request to store an object named vacation.jpg.
2. The storage engine receives the request. It writes three copies of the object to chunk containers on different nodes in parallel. In this example, the storage engine writes the object to chunk containers on nodes 1, 5, and 8.
3. The storage engine writes the location of the chunks to the object location index.
4. The storage engine writes the object location index to three chunk containers on three different nodes. In this example, the storage engine writes the object location index to chunk containers on nodes 2, 4, and 6. The storage engine chooses the index locations independently from the object replica locations.

5. When all of the chunks are written successfully, ViPR acknowledges the write to the requesting application.
6. After ViPR acknowledges the write and the chunks are full, the storage engine erasure-codes the chunk containers.

Figure 3. ViPR Write Path

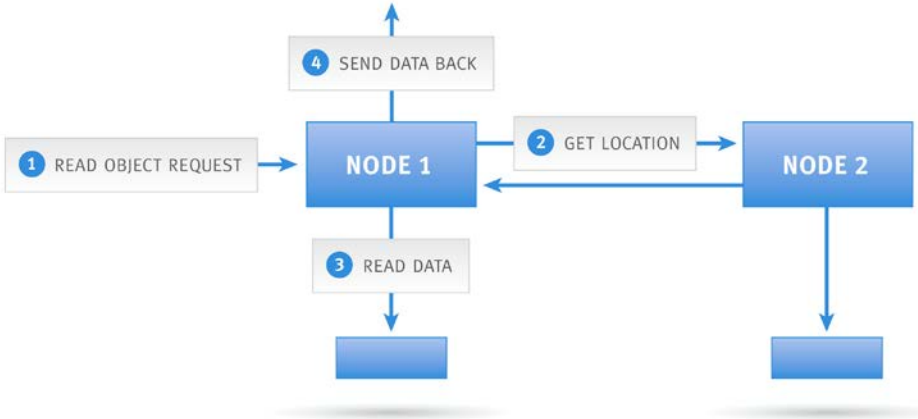


ViPR Read Path

This section and Figure 4 illustrate the ViPR commodity data flow when a user submits a read request. At a high level, the storage engine uses the object location index to find the chunk containers storing the object. It then retrieves the erasure-coded fragments from multiple storage nodes in parallel and automatically reconstructs and returns the object to the user. The detailed steps follow.

1. The system receives a read-object request for vacation.jpg.
2. The ViPR storage engine gets the location of vacation.jpg from the object location index.
3. ViPR reads the data and sends it to the user.

Figure 4. ViPR Read Path



Update Path

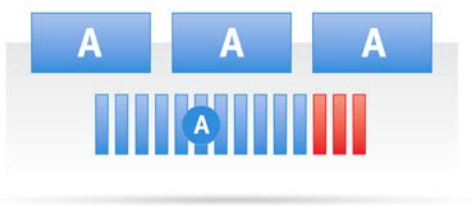
ViPR updates existing objects using byte-range updates, in which only the changed data is rewritten. The entire object does not need to be rewritten. The storage engine then updates the object location index to point to the new location. Because the old location is no longer referenced by an index, the original object is available for garbage collection.

ERASURE CODING

Erasure coding provides storage efficiency without compromising data protection or access. The ViPR storage engine implements the Reed Solomon 12/4 erasure-coding scheme in which a chunk is broken into 12 data fragments and 4 coding fragments. The resulting 16 fragments are dispersed across nodes at the local site. The storage engine can reconstruct a chunk from a minimum of 12 fragments.

ViPR requires a minimum of four nodes to be running the object service at a single site. The system can tolerate failures depending on the number of nodes. When a chunk is erasure-coded, the original chunk data is present as a single copy that consists of 16 data fragments dispersed throughout the cluster. When a chunk has been erasure-coded, ViPR can read that chunk directly without any decoding or reconstruction. ViPR doesn't need to read all the fragments to read a small object, but can directly read the object itself in the fragment which contains it. ViPR only uses the code fragments for chunk reconstruction when a hardware failure occurs.

Figure 5. ViPR Services Erasure Coding



Garbage Collection

In append-only systems, update and delete operations result in files with blocks of unused data. This is done at the level of chunks and unused chunks reclaimed by a background task. This maintains the efficiency of the system.

STORAGE EFFICIENCY FOR LARGE AND SMALL FILES

ViPR is adept at handling both a high volume of small files as well as very large. ViPR efficiently manages small files, for example, files that are 1KB to 512KB. Using a technique called box-carting, ViPR can execute a large number of user transactions concurrently with very little latency, which enables ViPR to support workloads with high transaction rates. When an application writes many small files with high I/O, ViPR aggregates multiple requests in memory and writes them as one operation. ViPR will acknowledge all the writes at the same time and only when the data is written to disk. This behavior improves performance by reducing round trips to the underlying storage.

ViPR is also very efficient when handling very large files. All ViPR nodes can process write requests for the same object simultaneously and each node can write to a set of three disks. ViPR never stores more than 30 MB of a single object in a chunk and supports multi-threading. Throughput takes advantage of all spindles and NICs in a cluster, which enables applications to use the full bandwidth of every node in the system to achieve maximum throughput.

GEO PROTECTION

EMC ViPR geo-protection provides full protection against a site failure should a disaster or other calamity force an entire site offline. ViPR geo-protection works across all types of ViPR-supported hardware. A geo-protection layer in ViPR protects data across geo-distributed sites and ensures that applications seamlessly function in the event of a site failure.

ViPR geo-protection manages the storage overhead that occurs when replicating multiple copies of data across sites. In addition to providing local protection, ViPR efficiently replicates data across multiple sites with minimal overhead. ViPR Geo-protection layer is a

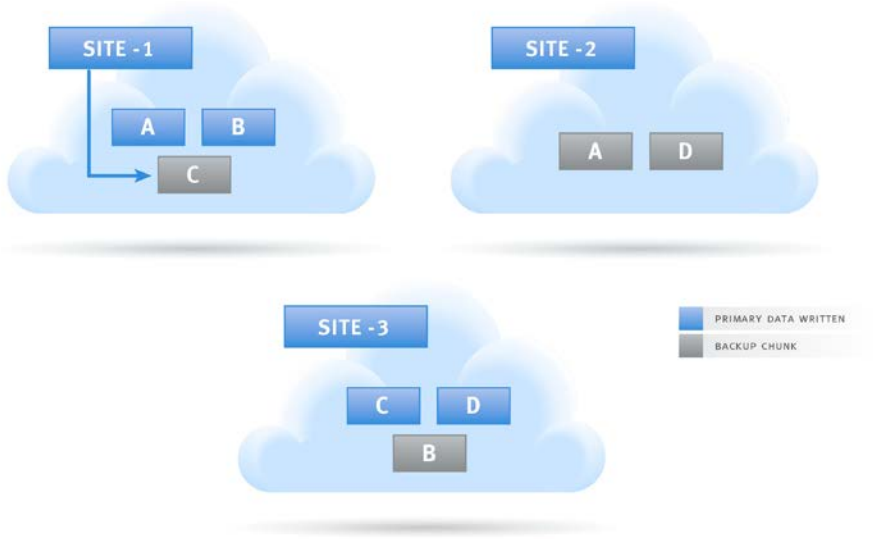
higher-level construct on top of local protection that is provided in ViPR Commodity. This means customers can deploy a mix of file based array platforms in one site and commodity based platforms in another site and have geo-replication between both sites.

ViPR Geo-protection has two main components. First, data protection that ensures data is protected in case of a site failure and, second, a storage overhead management scheme that reduces the number of copies when used in a multi-site environment.

Data Protection

In order to protect data across sites, ViPR ensures that for every chunk written, there is a backup chunk in another site. This means that ViPR always keep a single backup copy in any other site to ensure data is available in case of a disaster. Figure 6 illustrates the backup of chunks. The blue indicates the primary data written and the gray indicates the backup chunk.

Figure 6. ViPR Services Chunk Backup

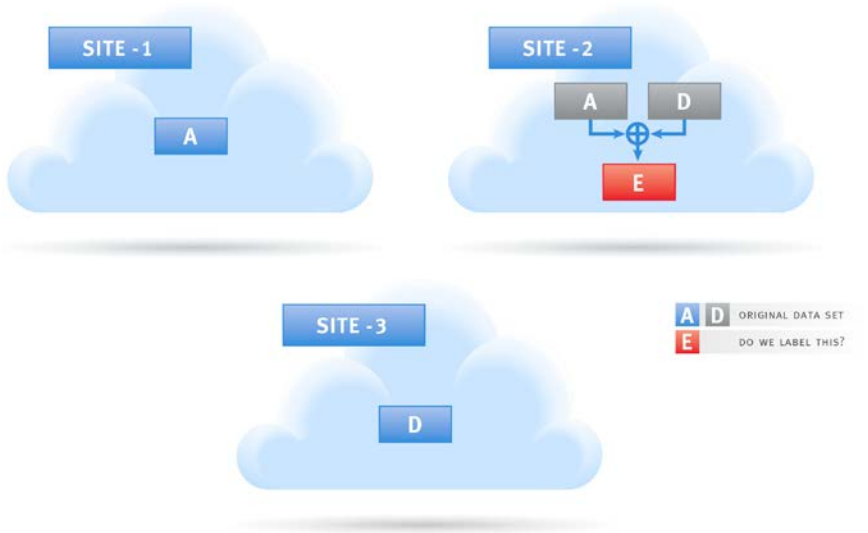


Storage Overhead Management

Once data has been protected and the Object virtual pool contains more than two virtual data centers (VDC), ViPR starts background tasks to efficiently manage storage overhead. In order to reduce the number of copies in multi-site deployment, ViPR implements a contraction model. The data contraction operation is an XOR between two different data sets that produces a new data set. This new data set is then kept and the original data is removed.

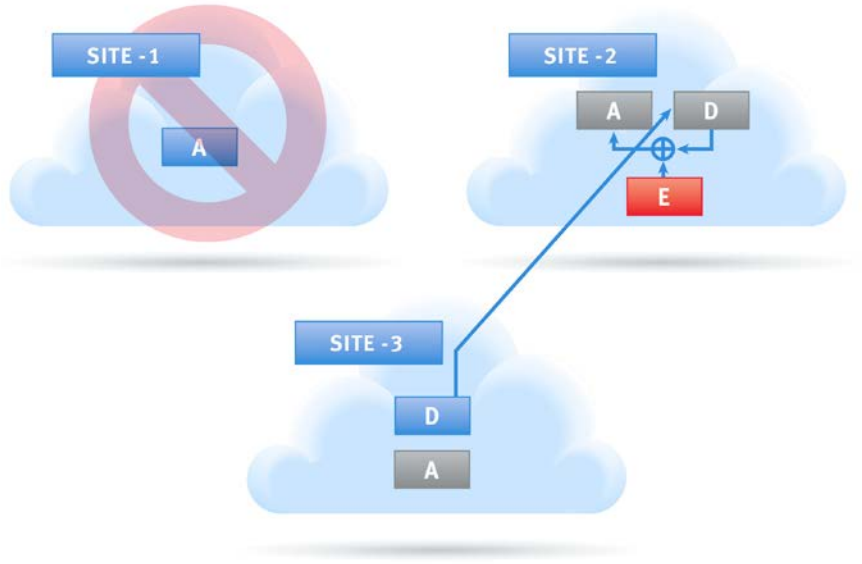
In Figure 7, A and D is the original data set, and when contracted produce the value E. The value E is then kept on site 2 and A and D backups are removed.

Figure 7. ViPR Data Contraction Operation



In case of a site failure the XOR operation that was previously done on data set A and D that produced E, can then start a data expansion process that retrieves the data affected by the failing site. In the diagram below the failure of site-1, causes ViPR to start a reconstruction process that sends back data set D from site-3 to site-2 and using that expansion process ViPR is able to retrieve back A.

Figure 8. ViPR Data Expansion Operation



The following table illustrates the storage overhead introduced by ViPR based on the number of sites.

Figure 9. ViPR Storage Engine Storage Overhead

Number of Data Centers	Overhead
1	1.33x
2	2.67x
3	2.00x
4	1.77x
5	1.67x
6	1.60x
7	1.55x
8	1.52x

Traditional approaches to protection make extra copies of the data and the overall protection overhead increases with the number of additional sites. With ViPR, storage efficiency scales inversely: the more data centers, the more efficient the protection. The ViPR geo-protection model is unique in the industry and delivers three critical benefits:

- **Data reliability and availability** - ViPR can tolerate one full-site disaster, and two node failures in each site when using commodity based platforms with a minimum of eight nodes.

- **Reduced WAN traffic and bandwidth** - Data written to any site always has the complete data available locally, which avoids WAN traffic when recovering from node or disk failures.
- **Optimized storage efficiency and access** - Efficient algorithm that enables ~1.8 copies across 4 sites without WAN read/write penalties.

MULTI-SITE ACCESS

ViPR provides strong, consistent views of data regardless of where the data resides. With geo-protection enabled, applications can access data immediately through any ViPR site, regardless of where the data was last written. For example, an application that writes object-1 to site A can access that data immediately from site B. Additionally, any update to object-1 in site B can be read instantly from site A. ViPR ensures that the data returned for object-1 is always the latest version, whether you access it from site A or B.

ViPR achieves this strong consistency by representing each bucket, object, directory, and file as an entity and applying the appropriate technique on each entity based on its traffic pattern. This technique minimizes WAN roundtrips, which reduces average latency.

Achieving Strong Consistency with Asynchronous replication

In ViPR all replication for backup data is done asynchronously. This means that after the write has committed, ViPR returns back immediately to the client. Then a background asynchronous replication task ensures data is protected across sites. In a multi-site access model ViPR ensures that clients are always getting the latest version of the data regardless of the location where that data resides.

ViPR Global Index

In order to provide a more detailed explanation of how strong consistency works in ViPR, we need to understand ViPR global index. The global index is key to ViPR architecture. The index is stored and replicated across multiple sites. All index information is stored in chunks similar to object data. The index is always replicated to all the sites. This information is relatively small and enables other sites to execute lookup steps locally most of the time.

Whenever data is stored in a site, it becomes primary in that site, and this site becomes the index owner for that data. Consequently, if there is an update to that data, this site needs to be contacted in order to ensure strong consistency. However, this may result in WAN penalties for such updates. ViPR improves that by providing lease and heartbeat techniques which improve performance and avoid WAN hops.

In order to ensure strong consistency ViPR uses background mechanisms like lease and heartbeat across sites to maintain the guarantees that an object is not modified. The optimization is suspended when the object is continuously updated, otherwise the mechanism remains in effect.

Achieving strong consistency in using asynchronous replication is a key fundamental capability in ViPR that relieves application developers from the worry of having to deal with the inconsistencies that is introduced by eventual consistency replication that may affect the application behavior.

CONCLUSION

Hyperscale means being able to massively scale a compute or storage environment in response to new demands. Hyperscale is now a necessity to support mobile, cloud, Big Data and social applications and the massive amounts of data they create and consume. Hyperscale is achieved by using standardized, off-the-shelf components that, individually, don't provide performance and reliability. However, at scale, the pool of components, together with intelligent software, provide the necessary reliability and performance. Most people associate hyperscale with large public cloud infrastructures such as Google, Facebook and Amazon Web Services. EMC has changed that and made hyperscale capabilities and economics available in any data center.

Software-defined storage makes it possible to bring cloud-scale storage services to commodity platforms. ViPR is the intelligent software in the hyperscale equation. The ViPR Services software is architected using cloud principles which make it unique in the industry as a platform that delivers cloud-scale storage services, support for modern applications with global access requirements and offers operational efficiency at scale.

REFERENCES

ViPR Hardware Checklist:

http://www.emc.com/techpubs/vipr/commodity_install_checklist-1.htm