Understanding Data Locality in VMware Virtual SAN

July 2014 Edition

TECHNICAL MARKETING DOCUMENTATION

vmware®

Table of Contents

Introduction

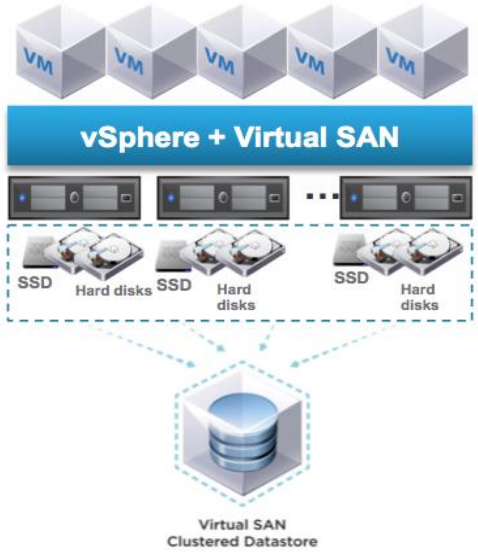
VMware Virtual SAN is a new hypervisor-converged, software-defined storage platform that is fully integrated with VMware vSphere. Virtual SAN aggregates locally attached disks of hosts that are members of a vSphere cluster, to create a distributed shared storage solution.

Virtual SAN is a hybrid disk system that leverages both flash-based devices to provide optimal performance, and magnetic disks, to provide capacity and persistent data storage. This delivers enterprise performance and a resilient storage platform.

The distributed datastore of Virtual SAN is an object storage system that leverages the vSphere Storage Policy Based Management (SPBM) framework to deliver application-centric storage services and capabilities that are centrally managed through vSphere virtual machine storage policies.

This document discusses how Virtual SAN design choices exploit data locality to deliver superior aggregate performance while retaining simplicity and efficiency.

Figure 1: VMware Virtual SAN



Virtual SAN Design Goals

One of the primary design goals for Virtual SAN is to deliver a new level of storage management simplicity and ease-of-use to the VMware administrator. Virtual SAN achieves this by creating an abstracted pool of storage resources that can be used easily and efficiently across the entire vSphere cluster.

While capabilities exist to "tune" performance for individual workloads, Virtual SAN strives to eliminate most of the manual effort associated with traditional per-workload storage management. Many of the design choices found in Virtual SAN are a direct result of observing thousands of production vSphere clusters.

Data Locality

In computer science, "data locality", also known, as "locality of reference" is the behavior of computer programs according to which a workload accesses a set of data entities or storage locations within some period of time with a predictable access pattern.

There are two main types of data locality:

- Temporal locality - The probability that if some data (or a storage location) is accessed at one point in time, then it will be accessed again soon afterwards.

- Spatial locality - The probability of accessing some data (or a storage location) soon after some nearby data (or a storage location) on the same medium has been accessed. Sequential locality is a special case of spatial locality, where data (or storage locations) are accessed linearly and according to their physical locations.

Data locality is particularly relevant when designing storage caches. For example, flash devices offer impressive performance improvements – at a cost – so efficient use of these resources becomes important.

Virtual SAN Design Considerations

Like any storage system, VMware Virtual SAN makes use of data locality. Virtual SAN uses a combination of algorithms that take advantage of both temporal and spatial locality of reference to populate the flash-based read caches across a cluster and provide high performance from available flash resources.

Examples include:

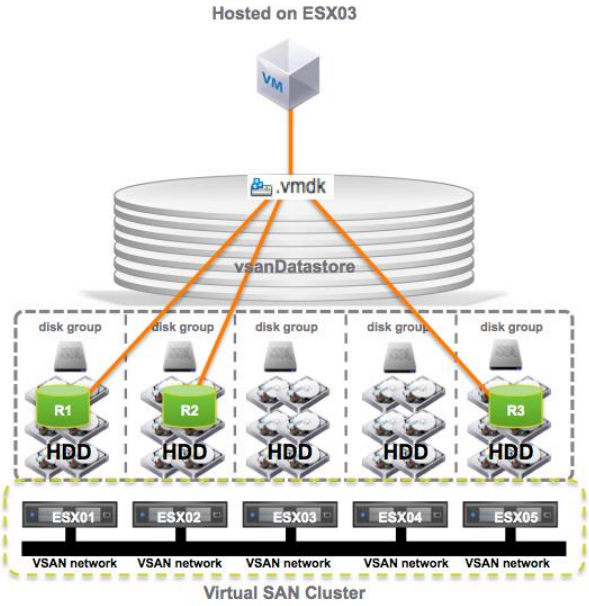
- Every time application data is read by a virtual machine, Virtual SAN saves a copy of the data in the Read Cache portion of the flash device associated with the disk group where the copy of the data resides. Temporal locality implies that there is high probability that said data will be accessed again before long.

  In addition, Virtual SAN predictively caches disk blocks in the vicinity of the accessed data (in 1MB chunk at a time) to take advantage of spatial locality as well.

- Virtual SAN uses an adaptive replacement algorithm to evict data from the Read Cache when it is deemed unlikely that the data will be accessed again soon, and uses the space for new data more likely to be accessed repeatedly.

- Virtual SAN makes replicas of storage objects across multiple servers for protection purposes. Reads are distributed across the replicas of an object for better load balancing. However, a certain range of logical addresses of an object is always read from the same replica. This approach has two important benefits:

Figure 2: Objects Replicated Across Multiple Servers



1. Increases the chances that the data accessed is already in the Read Cache.

2. A data block is never cached in more than one flash device.

To be clear, a fundamental design decision for Virtual SAN is to not implement a persistent client-side local read cache. The decision was based on the following observations regarding local read caching on flash:

- Local read caching results in very poor balancing of flash utilization (both capacity and performance) across the cluster.

- Local read caching requires transferring hundreds of gigabytes of data and cache re-warming when virtual machines are vMotioned between hosts to keep compute resources balanced.

- Local read caching offers negligible practical benefits in terms of performance metrics, such as latency.

Let's see in more detail the rationale behind this architectural approach of Virtual SAN.

Data Locality and latency

One theoretical argument in favor of caching VMDK data on the same host as the virtual machine is that of improved read access latencies by avoiding the network latency overhead. If true, this argument would only apply to reads, and not protected

writes.

As with all clustered storage systems, protected data must be safely written on more than one server, meaning that all writes must be network writes. Thus, any local caching approach will only be theoretically effective for reads, and not for protected writes.

The majority of customers with production Virtual SAN deployments (and for that matter any hyper-converged storage product) are using 10Gigabit Ethernet (10GbE). 10GbE networks have observed latencies in the range of 5 - 50 microseconds. (Ref: Qlogic's Introduction to Ethernet Latency - http://www.qlogic.com/Resources/Documents/TechnologyBriefs/Adapters/Tech_Brief_Introduction_to_Ethernet_Latency.pdf)

Those latencies are achieved with sustained workloads and they include packet processing in the network stack (TCP/IP). Those numbers are consistent with what we observe with the ESX network stack. In the case of Virtual SAN this happens on the VMkernel network adapter.

One may argue that given the low access latencies of flash devices, network latencies may become apparent when data is accessed remotely. However, a deeper look shows this not to be the case.

A typical enterprise grade flash device (SSD or PCIe card) has advertised optimal latencies in the range of 20 - 100 microseconds (usec). However, this is applicable only for operations issued to the device one at a time. Most flash storage devices have limited internal parallelism; they can typically process only a few operations in parallel.

In real-world use cases and certainly in the case of virtualized environments, storage systems use flash devices to maximize the I/O Operations per Second (IOPS) provided to the workloads. Because of the limited internal parallelism of the devices, the storage system needs to maintain a queue of I/O operations on the device to achieve a high number of IOPS (by keeping the pipeline full). Depending on the device, the queue depths required to maximize IOPS may range from 32 to 256 operations.

While high queue depths result in better IOPS and throughput, they also increase the observed latency per operation; each operation spends more time in the device queue waiting to be processed. For most flash devices, maximum IOPS are achieved with latencies that are close to and some times exceed 1 millisecond (msec).

As a real-world example, consider the recent Intel DC S3700 enterprise SSD, a device with very dependable performance characteristics. The device is rated to deliver up

to 75,000 4K reads with a queue size of 32 operations. At the same time, its quality of service with that queue depth is rated to 4K reads in 1 msec, 99.9% of the time (Ref: Solid-State Drive DC S3700 Series - http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/ssd-dc-s3700-spec.pdf -- see page 7).

Another typical SSD used in enterprise solutions is the Intel DC S3500. The device is rated to deliver up to 75,000 4K reads with a queue size of 32 operations. In this case, the latency range is even higher. The device is rated to deliver 4K reads in 2 msec, 99.9% of the time, using a queue size of 32.
(Ref: Solid-State Drive DC S370 Series - http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/ssd-dc-s3500-spec.pdf).

Given the above data, it becomes clear that the latency introduced by the network is negligible compared to the flash device latencies. The network adds 10-100 usec to a storage device latency that is 1 - 2 msec range. This has been confirmed in lab experiments: it ends up being extremely difficult to measure the additional latency solely introduced by the network.

For the vast majority of workloads, Virtual SAN's approach delivers very good read performance from read cache, faster than many traditional storage arrays.

Virtual SAN delivers this high level of read performance while efficiently using SSD resources (only one copy of cached data) as well as preserving an extremely simple experience for vSphere administrators.

Data Placement and Resource Utilization

Given that network access of cached reads doesn't impose a noticeable performance penalty, what additional advantages result from the Virtual SAN design, which distributes the cache across the cluster?

There are several:

- Improved Load Balancing and Resource Utilization

  By being able to distribute data on any host and any device in the cluster, Virtual SAN can achieve superior resource utilization. All resources: capacity, IOPs and throughput are available to all the VMs in the cluster, regardless of which server they reside on.

  Better overall resource utilization in the cluster results in better aggregate performance by eliminating 'hot spots' on individual hosts or devices. It also

results in superior economics –maximize the bang-for-buck achieved from expensive flash devices.

- Less Data Migration

  With read caches local to the client, one has to pay the penalty of bulk data movement and/or cache re-warming every time a virtual machine is vMotioned across hosts in the cluster. These bulk data moves may take longer than expected, and/or consume an inordinate amount of network resources. As a result, there is a considerable performance impact on a VM, for a while after vMotion is completed.

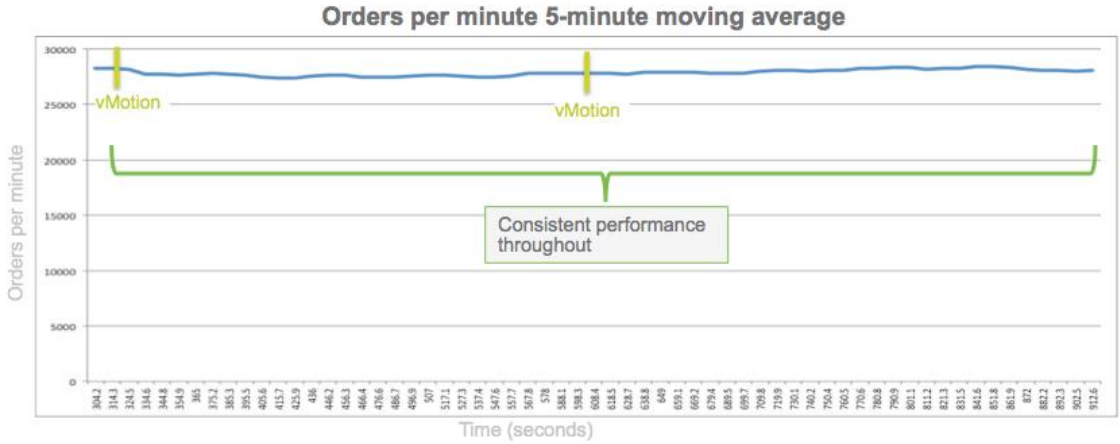
  Load balancing the compute resources (CPU and memory) in a cluster should not mean that one needs to also do expensive data moves. vSphere DRS (Distributed Resource Scheduler), for example, does not need to calculate the cost of moving cache contents when optimizing VM placement in the cluster.

For this reason, existing hyper-converged storage products that use client-local caches recommend to their customers that vSphere DRS is disabled in order to reduce vMotion migrations. That eliminates one of the big benefits of virtualization.

Virtual SAN is not architected with data locality dependencies, and thus delivers more consistent performance. The results from an example test illustrate this. The workload that was run is a typical transactional application, and the orders per second (on a five-minute moving average) were measure throughout the duration of the test.

In this test, a virtual machine started running on a host that did not contain any disk replicas (i.e. all disk replicas were running on other hosts). After five minutes the VM was migrated to another host with no disk replicas (marked as the first vMotion event in Figure 3).. Five minutes after that it was migrated to a host with one local replica. The application performance remained consistent during and after the migration.

Figure 3: Virtual SAN Workload Performance without Data Locality after Migration Operations



Effective Data Locality

The above arguments are not meant to dismiss the usefulness of read caches local to the client. Such caches can be effective under the following conditions:

- The latency of accessing a local cache is orders of magnitudes faster (lower latency) than going over the network. With today's technologies, that is possible when the cache is in RAM (latencies measured in nanoseconds).
- When the medium where the local cache resides is not constrained by the narrow parallelism of storage devices. Again, RAM is the obvious choice here.
- When a small local cache can go a long way. A good example is when you can share a lot of read-mostly data among many similar virtual machines on a host. Having a small cache is also warranted for a constrained and relatively expensive resource such as RAM.

An obvious use case that meets the above conditions is Virtual Desktops. These deployments usually involve:

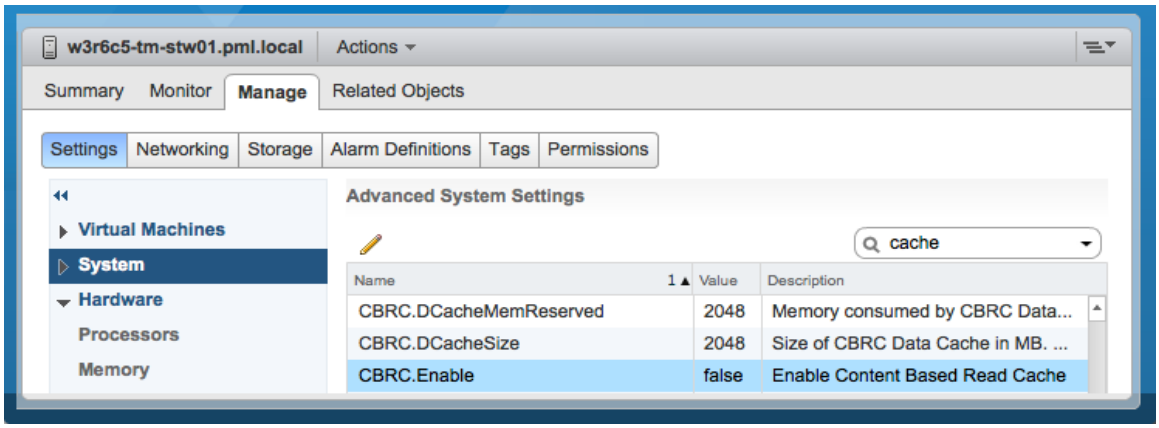
- High consolidation ratios (many virtual machines per host)
- Large ratio of common data between similar OS images, which can be effectively de-duplicated in the local cache.

These types of use cases can benefit by using some form of in-memory caching

solution that uses de-duplication, such as VMware's Content-Based Read Cache (CBRC).

Content-Based Read Cache (CBRC) is a vSphere hypervisor feature that is currently utilized exclusively with Horizon View and is called View Storage Accelerator. It is a very fast in-memory (volatile RAM) dedup'ed Read Cache residing on every host where the data is accessed. This feature allows for a read cache layer to be constructed in memory that is optimized for recognizing, handling, and de-duplicating virtual desktop client images.

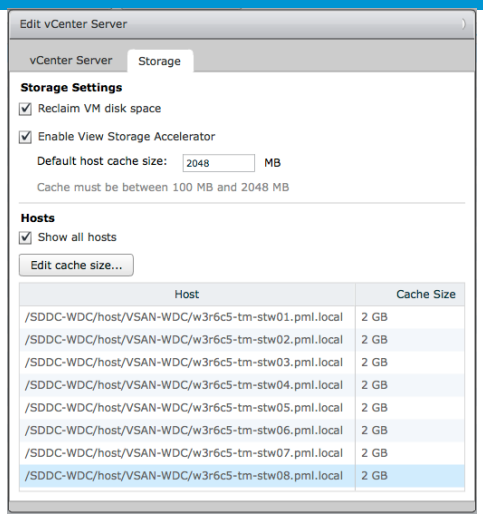
Figure 4: Content-Based Read Cache Feature in vSphere Hypervisor



View Storage Accelerator is configured and enabled in the vSphere hypervisor and managed by the Horizon View Composer. View Storage Accelerator delivers a significant reduction in read IOPS, as high as 90% for certain operating system images and workloads.

The amount of reduction in IOPS enables large scaling capabilities for the virtual desktops in the case of I/O storm workloads. That's typically found in large virtual desktop deployment scenarios, when many VMs go through a boot sequence at the same time.

Figure 5: View Storage Accelerator Host Settings



Virtual SAN can be combined with View Storage Accelerator and thus use local read caching based on RAM, which is the more effective application of data locality.

Conclusion

In conclusion, the benefits of better balancing and improved overall resource utilization in a cluster using hypervisor-converged storage with flexible data placement consistently outweigh any practical performance benefits of local caching on flash devices.

Local read caches may add additional expense due to bulk data transfers during VM migration (vMotion), they do not apply to protected writes, and they can introduce additional complexity if not fully integrated with other management tools in use.

Local caching is best used only where the cache medium is much faster than network and storage device latencies, and where a small cache can go a long way. The View Storage Accelerator feature, based on Content-Based Read Cache, is designed to work exactly within these constraints and is an effective use of local caching for virtual desktop deployments.

The VMware engineering team considered all potential options when designing Virtual SAN. Their choices reflect the need to deliver superior performance, great economics as well as an extremely simple and consistent management experience for vSphere administrators.

Acknowledgments

I would like to thank Christos Karamanolis, Principal Architect of VMware R&D, and lead engineer of Virtual SAN. This paper would not have been possible without him, his contributions. Christos deep knowledge and understanding of the product was leveraged throughout this paper.

I would also like to thank Chuck Hollis, Chief Strategist of the Storage and Application Services Business Unit for his contributions and Charu Chaubal, group manager of the Storage and Availability Technical Marketing team for his contributions to this paper.

About the Author

Rawlinson Rivera is a senior architect in the Cloud Infrastructure Technical Marketing group at VMware focused on Software-Defined Storage technologies such as Virtual SAN, Virtual Volumes, as well as the integration of VMware storage products with the OpenStack framework.

As a previous architect in the VMware Cloud Infrastructure and Management Professional Services organization, he specialized in vSphere and cloud enterprise architectures for VMware' Fortune 100 and 500 customers.

Rawlinson is among the few VMware Certified Design Experts (VCDX#86) in the world and is the author of multiple books based on VMware and other technologies.

Follow Rawlinson's blogs:

- http://blogs.vmware.com/vsphere/storage
- http://www.punchingclouds.com

Follow Rawlinson on Twitter:

- @PunchingClouds

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com