OUTDATED

Server-Side Caching

FEBRUARY 2015

WHITE PAPER BY CHRIS M EVANS

L B

StarWind

Trademarks

“StarWind”, “StarWind Software” and the StarWind and the StarWind Software logos are registered trademarks of StarWind Software. “StarWind LSFS” is a trademark of StarWind Software which may be registered in some jurisdictions. All other trademarks are owned by their respective owners.

Changes

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, StarWind Software assumes no liability resulting from errors or omissions in this document, or from the use of the information contained herein. StarWind Software reserves the right to make changes in the product design without reservation and without notification to its users.

Technical Support and Services

If you have questions about installing or using this software, check this and other documents first - you will find answers to most of your questions on the Technical Papers webpage or in StarWind Forum. If you need further assistance, please contact us.

Copyright ©2009-2015 StarWind Software Inc.

In 2016, Gartner named StarWind “Cool Vendor for Compute Platforms”.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

About StarWind

StarWind is a pioneer in virtualization and a company that participated in the development of this technology from its earliest days. Now the company is among the leading vendors of software and hardware hyper-converged solutions. The company’s core product is the years-proven StarWind Virtual SAN, which allows SMB and ROBO to benefit from cost-efficient hyperconverged IT infrastructure. Having earned a reputation of reliability, StarWind created a hardware product line and is actively tapping into hyperconverged and storage appliances market. In 2016, Gartner named StarWind “Cool Vendor for Compute Platforms” following the success and popularity of StarWind HyperConverged Appliance. StarWind partners with world-known companies: Microsoft, VMware, Veeam, Intel, Dell, Mellanox, Citrix, Western Digital, etc.

Contents

Summary

Software-defined storage (SDS) architecture continues to gain mind share in organizations that are virtualizing their storage infrastructure. Within the past 18 months, leading server hypervisor software vendors, including Microsoft, have introduced their own integral SDS offerings that are intended to deliver a "one stop shop" for consumers of their server virtualization wares. However, the SDS technologies from hypervisor vendors have tended to lag behind the wares of third party SDS vendors, such as StarWind Software. The smart vendor seeks to work with, rather than to compete with, the much larger operating system and server virtualization vendor. So, StarWind Software is positioning its Virtual SAN as a technology that augments and complements Microsoft's integral SDS offering. In the process, StarWind resolves some of the technical issues that continue to create problems for Redmond's product.

Introduction and Executive Summary

From the early days of computing there has always been a focus on using expensive resources as efficiently as possible. As the cost of computing has dropped, there is still a need to use resources in the most cost-effective way as the amount of computing resources deployed continues to grow and accelerate year on year.

One case in point is the performance of data storage compared to other computing components such as processors and memory. Hard disks have always been the bottleneck in IT systems but the high cost of DRAM and the need to for a permanent storage medium means there is a limit to the amount of data that can be stored in system memory, which has a direct impact on application performance.

Thankfully the activity profile of data stored on shared external storage arrays shows that only a proportion of data is active at any one time. Caching solutions take advantage of this principle to allow expensive resources such as flash and system DRAM to be use as a temporary store for the most active data in use at any one time.

Although mileage may vary, caching solutions can significantly improve application performance using a faster storage that is as little as 10% of the data stored on the external array, also known as a backing store. This means that issues of performance and throughput can be resolved without the need to move applications to platforms like all-flash storage arrays.

Caching is implemented by intercepting data on the I/O path from the application to the external storage device. This can be to process read requests (data from the array) or write requests (data to the array). The interception process returns data to the I/O request from a location closer to the processor, such as within the server itself. This is achieved by installing software that can sit within the operating system, the hypervisor or the virtual machine, which uses local storage such as flash or DRAM.

Hypervisor-based solutions can be integrated into the hypervisor itself (including the hypervisor O/S kernel) or within a virtual machine. Each solution provides flexibility and disadvantages that have to be weighed up by the end user with reference to their requirements and configuration.

The range of solutions on the market today is significant, with many hardware vendors (of PCIe SSD and SSD products) providing caching solutions to be best utilize their products.

In summary, caching can be a cost-effective way of improving performance where external shared storage systems are in use, particularly when compared to the cost of upgrade or replacement with all-flash solutions.

I/O Latency & The Need for Caching

Writing data to permanent storage (such as disk or tape) has always and continues to be the biggest bottleneck in any IT system. In the last 20 years processor DRAM performance has tracked Moore's Law and doubled every 18-24 months. However, over the same period, hard disk drives have scaled their capacity exponentially while seeing a more modest linear growth in performance. This increasing disparity between compute power and the ability to store data in a timely fashion on platforms such as shared storage arrays now represents a serious issue in terms of the ability to continue to scale applications.

One solution put forward by hardware vendors is to implement all-flash arrays or hybrid systems that use flash storage (such as SSDs) as a means of improving the performance of external storage. These solutions deliver high throughput and in some cases low latency but that performance comes at a cost that can be up to 10x the cost of traditional disk systems.

The Pareto Principle and The Working Set

It's hard to justify moving all of the data for a system onto an all-flash array when only part of that data is active at any one time. Typically, most applications follow the Pareto principle, also known as the 80/20 rule, which means that 80% of the I/O load is performed by only 20% of the data. Actual figures vary but the ratios seen match up to the principle in operation. In a shared storage array the data will also typically follow a "long tail" curve, with a small number of LUNs creating the majority of the workload. This means in many cases that a large amount of data sitting on all-flash arrays simply doesn't justify being there.

Hybrid arrays go part of the way to making flash more effective and these solutions can be useful where throughput of data (the volume of traffic being written and read) is the issue. However, if latency is a problem (the time taken to service a single I/O) then external storage doesn't always resolve the problem for two reasons. Firstly external storage has to traverse many layers within the host, including the storage stack, HBA, drivers and storage area network; secondly all of the workload contends over a shared storage fabric and is concentrated into a relatively small number of physical connections into the array, resulting in the risk of I/O queuing between applications.

Server Side Caching Defined

Caching describes the process of storing a copy of data on a fast storage medium (such as DRAM or flash) in order to improve throughput or performance. The aim is to target this more expensive storage at only the subset of I/O requests that need it (the previously described working set). All of the data still resides on a backing store such as a shared storage array making the cache version a temporary copy of the most active data.

Caching Methods

There are two main caching processes that create either a read cache (where the cache services only read I/O requests) and write caching (which services write I/O). Solutions may support one or both methods at the same time. For write I/O caching there are three main methods employed:

- Write-though – in this method the write I/O is written to the cache and immediately written to the backing store before the write is confirmed to the host. Subsequent read requests for the data are then serviced from cache unless the data changes. The write-though method provides high performance for read intensive type workloads and where data has a high likelihood of being re-read soon after write, but it doesn't improve write performance.

- Write-Back – this process writes data initially to the cache and confirms the I/O back to the host immediately. The data is then offloaded to the backing store some time later as an asynchronous process. This method results in high performance for read and write I/O, however the solution is more exposed as data could be lost if a server or cache failure occurs before all data is de-staged to the backing store. Write-back solutions therefore need some additional protection such as clustering to mitigate hardware failure.

- Write-Around – this method writes data directly to the backing store rather than into the cache and so only improves read I/O requests. Write-Around solutions are useful where data being written is unlikely to be immediately referenced after the write occurs. It protects against "cache pollution" where the cache fills with inactive data.

The main benefits of cache solutions have already been articulated in that they improve application performance by using a targeted approach to deploying expensive storage resources. The issue of performance is particularly important in virtualized environments such as VSI and VDI deployments where the effect of virtualization creates the so-called I/O blender effect. This phenomenon means that

I/O requests are almost totally random and unpredictable as servers and applications are distributed randomly across shared datastores. From a technical perspective caching delivers:

- Reduced Latency – I/O responses can be served from local storage eliminating many parts of the I/O stack. Both DRAM and flash storage provide very low latency responses.

- Reduced External I/O – Caching eliminates the need to go to the external backing store for the majority of active I/Os. The reduced load on external storage provides an opportunity to reduce contention on external devices and the storage network.

- Higher Throughput – Low latency translates directly to increased throughput, especially for read-intensive applications.

Financial benefits include:

- Cost Avoidance – purchasing a caching solution that uses a small amount of flash or DRAM storage can be cheaper than upgrading to an all-flash or hybrid storage array.

- Increased Asset Utilisation – using a caching solution can ease the pressure on external backing store, potentially increasing capacity utilisation or avoiding hardware upgrades. The same utilisation benefits apply to any cache hardware purchased, which will be used more effectively on a $/IOPS basis than an external array.

Before we proceed further, here are a few definitions to fill out our knowledge on the caching process.

Cache Effectiveness – this is a term used to describe how effective a cache is in accelerating read and write I/O and is typically expressed as a "hit ratio" or the percentage of I/O accelerated over I/O processed.

Caching Algorithms – cache algorithms determine which data needs to reside in the cache and which can be removed, especially as the cache becomes full. Examples include "Least Recently Used", which invalidates the least recently accessed data. Other examples include "Least Frequently Used". Some algorithms will also attempt to predict what data should be in cache and pre-fetching data before it is requested. This can help to eliminate the initial read penalty for new data.

Caching Issues

Of course the use of caching solutions has negatives as well as positives. Here are some of the major things to consider when using caching solutions and how they can be addressed.

Data Integrity – write back caching (as an example) keeps updates in the cache and asynchronously writes them to disk at some later time in order to smooth out and so improve write I/O responses. However if the cache (or server) were to fail before the updates were committed, then the backing store would be inconsistent. This problem can be catastrophic and cause the loss of data across an entire file system. The answer to this problem is to cluster servers and replicate write I/O outside of the single server, providing a secondary copy for recovery purposes.

Data Concurrency – there are circumstances when data stored in the cache can become out of date. This problem can occur in a clustered environment when a data block is updated by one host, thereby invalidating the copies in cache on other cluster members. A similar issue occurs with write-back solutions, which need to ensure cached write I/O is replicated to all cluster members as write I/O is processed and accessed from a shared backing store.

Cache Warm Up – caches need to be populated with data to become effective. This loading process is known as cache warm-up. If the cache is transient (i.e. lost at server reboot) then the warm-up process will need to occur at each boot up, making the cache less effective until this process completes.

Implementation Methods

Cache software can be implemented in many different ways.

Operating System

In non-virtualized environments, cache software must be installed on each server operating system individually and matched with the cache resources. The cache software intercepts I/O as it is read and written from external storage, returning the results from the cache rather than the external device. Many solutions appear as a device driver to the host, effectively emulating physical storage in the process.

Virtual Server Implementations

In virtual server environments, caching can be implemented in a number of ways:

- Within the Hypervisor – here the caching software is integrated into the I/O path of the hypervisor, for example in VMware vSphere environments as a VIB (vSphere Installation Bundle) in ESXi or device driver. The device driver makes decisions on which data to cache in a transparent fashion to the hypervisor itself.

- Within a VM – in these solutions, the caching software sits within a virtual machine on the hypervisor. The VM presents storage to the hypervisor through standard storage protocols such as iSCSI and NFS. The hypervisor sees the storage as standard externalised storage resources, while the storage VM is responsible for providing cached data blocks and offloading updates to the backing store (if that feature is supported).

There is much debate as to whether the implementation of caching should be performed within the kernel of a hypervisor or within a virtual machine. Kernel based solutions have the following issues and benefits:

- Installation and removal of the cache software may require a system outage.

- Upgrade of the software may require a system outage and/or be restricted to the version of the hypervisor in use.

- Kernel-based solutions are inherently tied to the hypervisor they are written for and may not have been ported across all virtualisation solutions.

- The resources consumed by the cache software (such as processor and memory) may not be transparently reported to the user by the hypervisor platform.

VM-based solutions have the following issues and benefits:

- Installation and removal is usually a simple process that doesn't require a hypervisor reboot.

- Upgrades are not tied to the version of the hypervisor in use and can be introduced easily (for example refreshing caching algorithms).
- Solutions aren't tied to a specific hypervisor (although the vendor may not support multiple platforms).

- Resource consumption of the cache VM is easy to monitor and report.
- Solutions may result in higher latency than kernel-based due to the additional translation of I/O into standard storage protocols (depending on how the data path has been implemented).
- Solutions may be more difficult to manage in terms of performance and resource consumption as they contend with actual VM workloads (and so may need dedicated resources).

There are clearly benefits and disadvantages in using either location for caching software and making a recommendation on the best solution isn't a standalone process. Choosing the right product needs to be taken into consideration with other deployment factors.

Other Solutions

There are two other ways in which to implement caching. Qlogic, an HBA vendor for Storage Area Networks (SANs) offers a product called FabricCache that caches data in the HBA itself. The HBA is installed with a second PCIe card that acts as the storage device.   The FabricCache product operates transparently to the operating system and so provides more flexible management and maintenance.

The second option is to consider caching within the application itself, including the database tier. Solutions already exist from the major database providers to increase the amount of data in memory in order to improve performance. Application-based solutions do require more management overhead as each system requires dedicated resources and tuning individually.

Vendor Roundup

Vendor solutions are categorised by the way in which the cache solution is installed.

Operating System

- StarWind Software Virtual SAN - StarWind's solution is part of their Virtual SAN offering. Caching is implemented using a combination of DRAM and flash storage to accelerate performance and protect data in the event of failure of any node in the Virtual SAN cluster.

- HGST ServerCache – O/S-based caching solution that uses solid-state disks or system DRAM (read caching only) to improve the performance of external storage. Supports Windows, Windows Hyper-V and Red Hat Enterprise Linux 6.x (and equivalent distributions).

- HGST ClusterCache – technology developed from the acquisition of Virident in 2013. ClusterCache offers standard caching features with the ability to integrate with other Virident solutions to provide data resiliency capabilities.

Hypervisor – VM based

- Atlantis Computing ILIO and USX – memory-based caching using VMs for virtual machine and virtual desktop solutions.

- Infinio Systems, Accelerator – VM-based solution that caches NFS datastores with a VM on each server in an ESXi cluster. Accelerator appears as an NFS datastore to the ESXi host, providing read-only caching functionality. Data is deduplicated across the VMs of the clusters to provide increased virtual caching capacity.

Hypervisor – Kernel Based

- PernixData Inc, FVP – clustered caching solution delivered as an ESXi VIB and installed as a device driver within the ESXi kernel. FVP supports a range of flash storage hardware and system DRAM to create both write-back and write-through cache solutions.

- VMware Inc, vSphere Flash Read Cache – hypervisor integrated caching solution build into VMware vSphere ESXi. This allows pooled cache resources to be assigned to individual virtual machine hard disks (VMDKs). Cache resources are assigned in fixed rather than variable quantities.

- VMware Inc, Virtual SAN – caching implemented using SSDs as part of Virtual SAN distributed storage. vSphere 6 implements all-flash Virtual SAN that uses SLC flash for write I/O caching and MLC for permanent data storage.

- Proximal Data, AutoCache – hypervisor-based solution for vSphere and Hyper-V. Samsung Electronics acquired proximal data in November 2014.

Cross-Platform

- SanDisk FlashSoft – FlashSoft caching solutions support virtual environments (VMware vSphere) and Windows Server and Linux operating systems.

Hardware-based Solutions

- Qlogic Inc, FabricCache – SSD accelerated caching of I/Os at the Fibre Channel HBA layer using a dedicated PCIe SSD flash card in the server.

Contacts

Chris M Evans

Chris M Evans has worked in the technology industry since 1987, starting as a systems programmer on the IBM mainframe platform, while retaining an interest in storage. After working abroad, he co-founded an Internet-based music distribution company during the .com era, returning to consultancy in the new millennium. In 2009 he co-founded Langton Blue Ltd (www.langtonblue.com), a boutique consultancy firm focused on delivering business benefit through efficient technology deployments. Chris writes a popular blog at http://blog.architecting.it, attends many conferences and invitation-only events and can be found providing regular industry contributions through Twitter (@chrismevans) and other social media outlets.

Langton Blue Ltd

Web-site:	www.langtonblue.com	Twitter:	@langtonblue
E-mail:	enquiries@langtonblue.com	Phone:	+44 (0) 845 275 7085

StarWind Software

US Headquarters	EMEA and APAC
☎ +1-617-4497717 📠 +1-617-507-5845	☎ +44 20 3769 1857 (UK) ☎ +49 302 1788 849 (Germany) ☎ +33 097 7197 857 (France) ☎ +7 495 975 94 39 (Russian Federation and CIS) ☏ 1-866-790-2646

Support Portal:	https://www.starwind.com/support	Sales:	sales@starwind.com
Support Forum:	https://www.starwinds.com/forums	General Information:	info@starwind.com

StarWind

www.starwind.com **StarWind Software, Inc.** 35 Village Rd., Suite 100, Middleton, MA 01949 USA