technical knowledge base

versión para imprimir

Technical knowledge base document

Europa

Submitted Date: 22/01/06
Title: Notice: Performance and Configuration Tips for Enterprise Virtual Arrays (EVA) with Active-Active Operation
Document ID: c00594722
Last Modified Date: 22/02/06

You may provide feedback on this document

SUPPORT COMMUNICATION - CUSTOMER NOTICE

Document ID: c00594722
Version: 1

Notice: Performance and Configuration Tips for Enterprise Virtual Arrays (EVA) with Active-Active Operation

NOTICE: The information in this document, including products and software versions, is current as of the Release Date. This document is subject to change without notice.

Release Date: 2006-01-22
Last Updated: 2006-01-22

DESCRIPTION

1 Summary

Active–Active (AA) functionality on an EVA disk array enables any virtual disk (Vdisk) of the EVA to be accessed through either of its two controllers. EVAs running XCS firmware (EVA4000/6000/8000) implements AA. When running VCS 4.x firmware, EVA3000/5000 also provides this functionality. Active–Passive (AP) functionality on the EVA3000/5000 running VCS 3.x firmware and earlier enables Vdisk access only to the controller owning the Vdisk.

With AA a read request can use different data paths depending on which controller receives the request. In one case, the data path is the same as that used with VCS versions 3.x and earlier. In the second case, the data path is *longer*, being the same as that used by an EVA to mirror its write cache. Consequently, the performance of EVA/s running VCS 4.x, under read-dominated workloads may be less than that of the same EVA/s running earlier versions of VCS firmware. EVA write performance can also be impacted by reads and vice versa. This occurs because write-cache mirroring may be a performance limiter on EVA3000s and EVA5000s under intense write workloads, especially those dominated by large–block transfers.

These possibilities should be considered in terms of the following facts:

1. Most operating systems enable paths from hosts to Vdisks to be configured so that proxied reads can be avoided, either by direct manipulation of paths or indirectly through load-balancing disciplines. A controller that receives requests to a Vdisk that it does not own is called a proxy controller with respect to that Vdisk. The definition of proxied read is a read that goes through the *proxy controller*.

2. Any impact on performance that can be caused by AA depends significantly on workload demand, the largest impact occurring on arrays running at higher than 50% array (not CPU) utilization.

3. The impact on performance of VCS–based EVA/s is similar to that seen on XCS–based EVA/s, yet no performance issues associated with AA operation have yet arisen on the XCS-based arrays.

The purpose of this document is to discuss the points just presented. Section 2 provides a brief description of how AA is implemented on EVA arrays. Section 3 describes how pathing and load balancing is done on a number of operating systems, and is intended to provide the reader with a starting point for further investigation, rather than supplying a full solution.

DETAILS

2 Active-Active Operation

AA enables a Vdisk to be accessed through either of the EVA controllers. However, each EVA Vdisk has a special association with one of its controllers. This association is referred to as ownership in this document. The firmware responsible for performing all operations involving virtualization, including cache and RAID, executes on the processor of the controller that owns the Vdisk to which an I/O request is targeted. If the controller that receives the request does not own the Vdisk, the controller will negotiate with the other controller over an array mirror port to satisfy the request. Such an I/O request is called a proxied request in this document. A controller that receives requests to a Vdisk that it does not own is called a *proxy controller* with respect to that Vdisk.

Proxied I/O requests, whether read or write, have associated with them a small overhead caused by the negotiation mentioned. In addition, proxied **read** requests involve extra data transfers. The controller that owns the target Vdisk forwards data from its cache memory to the cache memory of the controller that received the read request. This transfer is done over an EVA's mirror port. Such transfers inject traffic on the mirror port and other internal controller buses that would otherwise not be present. Therefore, the performance of an EVA can depend on the percentage of read requests that are serviced as proxied requests.

This additional traffic does not occur for proxied write requests because write cache is mirrored on all EVA/s that have AA functionality. Therefore, the performance of an EVA under proxied writes is similar to that of the same EVA under non-proxied writes. Write-cache mirroring can be disabled, per Vdisk, on EVA3000s and EVA5000s running VCS revisions of 3.x and earlier. This cannot be done on VCS revisions of 4.0 or newer, nor on any revisions of XCS firmware. If write-cache mirroring has been disabled to boost array performance, it may be advisable not to upgrade the firmware to VCS v4.001 (or higher).

As mentioned earlier, the processor of the controller owning the targeted Vdisk executes all operations involving virtualization, including cache and RAID. These operations represent the bulk of the work that a processor must do in order to service a request. If all the Vdisks of an EVA are owned by a single controller, that controller's processor would do the bulk of all the work necessary to process the requests of a workload, resulting in a workload imbalance across the controllers. Such an imbalance should be avoided by distributing the ownership of Vdisks across an EVA's controllers, determined by the I/O demands to which those Vdisks are subjected.

All AA-enabled EVA/s include "Intrinsic Vdisk (or Logical Unit) Transition" functionality. If at least 2/3 of the total read requests (over a one hour period) to a Vdisk are proxied reads, ownership of the Vdisk is transitioned to that controller.

Performance could be impacted if proxied reads begin to saturate part of the data bandwidth in the proxy controller. This can be indicated by monitoring the "Total Mirror MB/s" using EVAPerf during heavy I/O loads. Monitoring this metric on the EVA 3000 and/or EVA 5000 before migrating to v4.x may determine if upgrading to AA is prudent. If the sustained rate summed across all Virtual Disk Groups per controller is greater than 50 MB/s (EVA 3000 and/or EVA 5000 only) for either controller then the additional overhead due to proxied read access could add latency to read and write I/O response times. The impact can be mitigated, in most cases, by managing the host path selection to the Vdisk/s so that proxied access is avoided using host-based multi-pathing capabilities. NOTE: The EVA 4000, EVA 6000, and EVA 8000 have a higher bandwidth because the hardware managing caching and mirroring is different.

3 Path Control and Load Balancing

The effects on performance of proxied I/Os can be eliminated or controlled using host software on the operating systems (OS) on which EVA/s are supported. This section provides a high-level summary of such software available on several OSs.

3.1 Windows – MPIO DSM

There are two pathing options available.

3.1.1 Using SQST

On Windows, MPIO DSM is the multi-pathing software used for EVA/s running VCS 4.x or XCS firmware. MPIO uses all paths to a Vdisk, but it allows load balancing across all such paths. The default MPIO DSM load-balancing policy is the Shortest Queue Service Time (SQST) discipline. The path used to submit an I/O request to a Vdisk is that which results in the shortest service time, i.e., the shortest time for the array to service requests. (Note: the service time as used here is the "response time". The difference in terminology arises because latencies are measured from the viewpoint of the host, where queues that result in "wait times" are host based rather than in the array.)

As mentioned in Section 2, a read request has a longer data path when it is sent to a proxy controller than when it is sent to a non-proxied controller. This means that the average service time (in the sense used above) of all pure-read workloads will be lower along the non-proxy path than it will be along the proxy path (see graphs in Section 4 that compare 50% proxy cases to 0% proxy cases). Therefore, the average service time for a workload with an arbitrary read/write ratio will also be lower along the non-proxy path than it will be along the proxy path. The SQST load-balancing discipline has a tendency to select the non-proxy path, on average. SQST is the recommended policy for all EVA/s that have Active-Active operation.

3.1.2 Preferred path setting

Configuring the primary paths to a Vdisk. When this is done, all I/O requests are sent to the Vdisk along that path. By properly configuring primary paths to the controller that owns the Vdisk, you can avoid non proxied I/Os operations.

3.2 HP-UX

There are several ways with which pathing can be addressed under HP-UX. The software includes Logical Volume Manager's (LVM) *pv-links* feature, *Secure Path*, and *Veritas Dynamic Multi Pathing (DMP)*.

3.2.1 LVM's Pv-Links

PV-links allows one I/O path, called a primary path, and several failover paths, called secondary paths, to be assigned to a Vdisk. Proxied I/Os can be eliminated by choosing the primary path to a Vdisk to pass through the controller that owns that Vdisk.

If a secondary path through a proxy controller for a Vdisk becomes active as a result of a failure of the primary controller, any performance impact will be transient, corrected by the "Intrinsic Vdisk Transition" functionality of the EVA.

3.2.2 Secure Path

With Secure Path a preferred path to a Vdisk can be set. When this is done, all I/O requests are sent to the Vdisk along that path. All other paths are considered to be alternate paths that are not active in sending I/O requests to the Vdisk. Proxied I/Os can therefore be avoided by assigning such a preferred path to the Vdisk through the controller that owns the Vdisk.

If a preferred path to a Vdisk is not set, then Secure Path will use all available paths to that Vdisk. In addition, any load balancing will take place across all of these paths.

3.2.3 Veritas DMP

This is a third party add-on solution that provides good flexibility on path assignment and load balancing. Further information on Veritas products should be obtained from Veritas.

3.3 OpenVMS

OpenVMS sends I/O requests to a given Vdisk through a single path (per host), i.e., requests are not striped across multiple array host ports. If a path fails, OpenVMS will find another working path and use it exclusively. The OpenVMS user can explicitly define the path to any Vdisk, but they cannot define failover paths. Failover paths are selected automatically by the OS.

Current versions of OpenVMS do not distinguish between proxy paths and non-proxy paths. At boot time, OpenVMS tries to distribute Vdisks across all the available paths to an EVA. However, the probability is ½ that the path chosen to a particular Vdisk will be through the proxy controller of that Vdisk. If in fact a path to the Vdisk is chosen through the proxy controller, a user can alter the path from the host by using OpenVMS CLI commands. On a single node cluster, an alternative is to allow the EVA’s “Intrinsic Vdisk Transition” function to change the path to the Vdisk’s owning controller.

In a clustered environment (more than 1 node using an EVA), the situation is more complex. OpenVMS makes no attempts to coordinate path selection across the nodes of a cluster. Consequently, one node could use a proxy path to a particular Vdisk while another node uses a non-proxy path to that same Vdisk. In this case, the EVA’s “Intrinsic Vdisk Transition” function is not guaranteed to work. OpenVMS CLI commands must be used if it is necessary or desirable to avoid proxied I/Os.

3.4 NetWare

Novell Native MPIO can be used to assign non-proxy paths to Vdisks on an EVA.

3.5 Linux

Interactions between Linux and EVA/s should be done using the Qlogic MPIO driver. This driver is not included with Linux but is supplied by HP.

With this driver a user can ensure that I/Os are sent to a Vdisk only through the non-proxy controller of that Vdisk. Furthermore, load balancing across HBA/s and all the host ports of the non-proxy controller is done. If failover to the other controller becomes necessary, the ownership of the Vdisk will be transferred to that controller.

3.6 Solaris

There are two ways to manage I/O paths under Solaris. The first is using MPxIO, which is bundled with Solaris. The second is using Veritas Volume Manager (VxVM) and Dynamic Multi Pathing (DMP), which is a third party product and has additional cost associated with it. Further information on Veritas products should be obtained from Veritas. The two solutions work similarly for the purposes of this discussion.

The software can be used with and without load balancing. When no load balancing is used, a set of paths to a Vdisk can be assigned through which I/O requests are sent to that Vdisk. Proxied I/Os can be avoided by assigning each path to a Vdisk to the controller that owns it.

> **NOTE:** A separate set of “passive” paths can also be assigned as failover paths to the other controller, if desired.

Load balancing applies to all available paths to Vdisks. Therefore, depending on the load-balancing discipline used, proxied I/Os may be introduced.

3.7 AIX

MPIO on AIX can be used to manage pathing. It can be used in a manner similar to Solaris MPxIO (the no load-balancing case) to assign non-proxy paths to Vdisks.

Antemeta software, which is a third party product and represents additional cost, works much like Solaris MPxIO and provides load balancing possibilities.

3.8 Tru64 Unix

A host running Tru64 Unix will use all available paths to a Vdisk of an EVA. I/O requests to a particular Vdisk are distributed across all these paths using a round-robin discipline modified with final selection dependent on path queue depths. This means that a candidate path is chosen from the set of all possible paths via a round-robin algorithm. However, if the number of outstanding requests on that path is larger than the number of outstanding requests for other paths, that path is skipped. Queue depth monitoring results in path-oriented load balancing. If path selection were performed by a strict round-robin algorithm, half of all I/Os to a Vdisk would be proxied I/Os.

Tru64 Unix represents an OS for which performance penalties associated with AA could be realized.

Additional Information

RECEIVE PROACTIVE UPDATES: Receive support alerts (such as Customer Advisories), as well as updates on drivers, software, firmware, and customer replaceable components, proactively via e-mail through HP Subscriber's Choice. Sign up for Subscriber's Choice at the following URL: http://www.hp.com/go/myadvisory

---

Hardware Platforms Affected: HP StorageWorks 3000 Enterprise Virtual Array, HP StorageWorks 5000 Enterprise Virtual Array, HP StorageWorks 6000 Enterprise Virtual Array, HP StorageWorks 4000 Enterprise Virtual Array, HP StorageWorks 8000 Enterprise Virtual Array
Components Affected: Not Applicable
Operating Systems Affected: Not Applicable
Software Affected: Not Applicable
Third Party Products Affected: Not Applicable
Support Communication Cross Reference ID: IA00589322

You may provide feedback here

To help us improve our content, please provide feedback and any additional comments below. If you have a problem or a question that needs immediate attention, please submit a call or contact your HP Response Center instead.

relevance to my question

- Provided the information I needed
- Helped me and/or pointed me to the information
- Did not pertain to my question/problem

quality

- Is good, please provide more like it
- Is hard to understand and/or needs more detail
- Has incorrect technical information
- Took the recommended action, but it did not solve my problem

access to resources

- Cannot access the Web site listed
- Cannot find the resource listed

Comments:
As much as possible, please identify the sections of the document to which your comments relate. When you have finished, press the *Submit* button to mail your evaluation and comments.

clear » submit »