



Consolidating Microsoft SQL Server with HP PolyServe: The User View

Product Insight

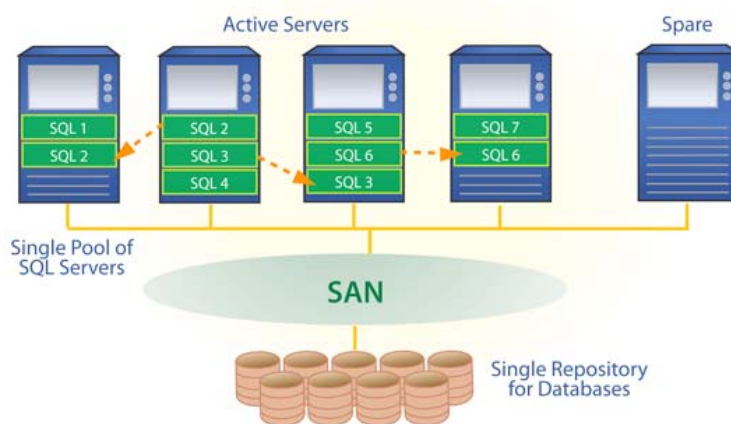
John Webster
Gordon Haff
28 March 2008

Licensed to Hewlett-Packard Company for web posting. Do not reproduce. All opinions and conclusions herein represent the independent perspective of Illuminata and its analysts.

Microsoft SQL Server has enjoyed phenomenal success as a database server. Its relatively low cost, steadily increasing functionality,¹ and ease of deployment have all combined to accelerate its growth. However, that same growth has led to a phenomenon that IT administrators commonly call SQL Server “sprawl”—the rampant, often uncontrolled, proliferation of SQL Server databases, whether under the purview of centralized IT administration, or tucked away under desks or in closets where departmental users have placed them independently of their IT departments.

This sprawl is costly because it can result in inefficient use of hardware, software, and administrative resources. Certainly, there are hardware and maintenance costs that are magnified by this kind of inefficiency, but as we are all becoming aware, poor utilization can also rapidly increase energy consumption. And, not least, server sprawl can also rob SQL Server application users of resource availability and business productivity.

HP’s PolyServe Software for Microsoft SQL Server is a clustered file system configuration that aims to combat SQL Server sprawl, consolidate SQL Server environments, and improve application user resource availability and response time. According to users we recently interviewed, it successfully accomplishes



these goals, and has advantages over better-known server virtualization approaches. In this paper, we discuss HP PolyServe’s inner workings, some of the relevant user experiences gleaned from our discussions, and how HP PolyServe compares to alternative approaches.

¹ See our *Microsoft SQL Server 2008*.

Clustering the File System

Speaking generically, HP PolyServe is a clustered file system (CFS). A CFS enables sharing of volumes, file systems, and even individual files among applications on multiple servers as though they were running on a single system.

A typical single-host file system coordinates access to data among multiple applications running concurrently, and maintains “metadata” to keep track of where directories and files are physically located; their ownership and access rights; when they were created, modified, and accessed; and other housekeeping information. Most file systems store their metadata on the same physical disks that contain the files, using a formally-defined “on-disk file structure.” A cluster file system extends that same functionality across multiple servers, while keeping application programming interfaces (APIs) largely unchanged. The best-performers allow direct I/O access to files on shared disks by each participating server. To accomplish this, a server’s CFS manipulates on-disk file structures the same way, but coordinates its activities with other cluster nodes—typically through the use of some form of “lock manager.”

In the case of HP PolyServe, a Distributed Lock Manager (DLM) allows concurrent read/write access to the underlying data, and can even provide locks at the byte level. HP’s PolyServe team worked directly with Microsoft, licensing its Installable File System (IFS) Kit to develop a Windows-compatible file system.

There is some overhead for a CFS relative to the non-clustered case of independent servers with independent file systems,² because two nodes have to be prevented from working on the same chunk of data simultaneously (using the aforementioned locks). However, each node’s having direct access to a common set of data can be a big performance win relative to approaches that interpose a “heavyweight” intermediate abstraction layer.

² CFSs don’t necessarily replace conventional file systems, but rather sit on top as an additional layer.

Early CFSs mostly saw duty as failover mechanisms for Unix servers. If one server failed, another could quickly take over because it had direct access to all of the failed server’s storage. CFSs also gained considerable popularity in high performance computing (HPC) where a large number of nodes often access a large quantity of common data. (Furthermore, HP data tends to be “read-mostly”—that is, read more often than it’s written—which makes coordinating access easier.)

Although it, too, made inroads into smaller-scale HPC clusters, PolyServe actually started out in 1999 with a focus on commercial applications. Its initial product, UnderStudy, was a classic HA-clustering product for Linux, providing application and hardware monitoring and failover for two servers. Over time, PolyServe went on to establish itself in a diverse set of applications including SQL Server, BizTalk, SAP, Windows Media Server, Citrix, SharePoint, Dynamics, Tibco, and others. In February of 2007, HP acquired PolyServe.

HP PolyServe Basics

In the universe that includes all possible virtualized computing solutions, HP PolyServe lives between the operating system and the storage subsystem. It is neither server virtualization (in the sense of virtual machines) nor storage virtualization (in the sense of abstracted storage devices). HP PolyServe essentially creates a pool of server and storage hardware resources that are dedicated to running a collection of SQL Server databases.³ All SQL Server files are stored in a single repository, here depicted as sharable SAN storage.

This architecture can simplify management of SQL Server by pooling database instances and allowing administrators to move them. Storage is likewise all part of a common pool.

³ Or other data types or application instances. The balance of this note focuses on HP PolyServe in an SQL Server environment.

The HP PolyServe Server Environment

It is common to find a 1:1 server-to-database relationship for SQL Server deployments. This causes the sprawl described earlier. When departmental users create a new database, they commonly add another dedicated server as well. The situation is often not much better even in a datacenter.

Under HP PolyServe, any database instance can run on any server—new or legacy—in the HP PolyServe cluster. New database instances are added to the server cluster, rather than a new physical server. Software updates are also performed at the cluster level across all servers within the cluster, rather than server by server.

A single instance of a database can be moved without copying from one server to another with minimal disruption to the application user either under normal circumstances (often called dynamic rehosting) or when a server failure within the cluster occurs. Therefore, there is no reason to duplicate hardware resources for each database instance requiring high availability. Servers within an HP PolyServe cluster can be configured as active-active standbys in the event of a failure. Alternatively, a standby server can back up one or more active servers.

Failover happens in the length of time required to stop and start an SQL Server instance—as opposed to the more typical sequence that also involves a SCSI release, SCSI reserve, file system unmount/mount, and `chkdsk` steps. HP estimates 30 seconds as a typical length of time, although, as with all clustering software, it depends upon the number of transactions that need to be replayed during startup of the instance after a failure.

An IT administrator can also shift resources to or away from database instances depending on existing workloads to cover peak load periods like month-end processing.

HP PolyServe supports clustering of unlike servers, even ones running different versions of the same OS. Therefore, an IT administrator can build

clusters of up to 16 servers from both existing and new server assets, without particular regard to hardware differences. SQL Server versions can also be mixed within a cluster, although Microsoft licensing sometimes constrains what can be mixed on a single physical server.

The HP PolyServe Storage Environment

The HP PolyServe cluster file system shares a common pool of SAN storage. As a consequence, the storage environment is managed in much the same way as the server environment. There is no need to create special configurations for high availability server pairs, as all storage is shared across a high-availability cluster. Capacity is added to the storage pool without disruption. LAN-free backup processes can be applied to all databases within the storage pool, rather than individually with each SQL Server instance. Using a shared file system also avoids mismatching drive letters or shares between servers (resulting in a “failed failover”).

The User Experience

We interviewed a number of users of HP PolyServe Software for Microsoft SQL Server to get a sense of real-world deployments. In general, these users are managing large systems environments, and are dealing with hundreds of instances of the SQL Server database running on many different server and storage hardware platforms. Their stories contained some common themes:

Control of SQL Server Sprawl

Each user we interviewed was managing in excess of 200 SQL Server instances with HP PolyServe. The largest had more than 1,200 databases, with 400 already using HP PolyServe and with 800 left to migrate over. All reported that growing frustration with managing their SQL Server environments, poor resource utilization (usually no more than 15 percent server utilization), and inconsistent application availability were factors that forced them to look for a more strategic alternative. One compared the growth of SQL

Server instances to the explosive growth of Tribbles aboard the Starship Enterprise.⁴

All started by first migrating existing databases from either single- or dual-server installations to the cluster; they then added databases as required. Technically speaking, the migrations were a relatively straightforward process. However, resistance by user groups to the consolidation effort was common.

Getting Departmental Users On Board

For example, one user reported that his IT server administration group needed to break an established pattern whereby departmental user groups implemented their own SQL Server databases without support and guidance from centralized IT. Of course, that didn't keep them from looking to centralized IT for support later on! After months of selling the consolidated SQL Server concept to each departmental user group and the CFO, this IT administrator was able to deliver better application availability to each of his user groups with a consolidated platform model. He also now runs SQL Server on a services delivery basis whereby user groups enter into a Service Level Agreement (SLA) and are charged for resource usage.

Another user advised that having some idea of the financial impact of an HP PolyServe implementation to the hosting cost model would be advisable when moving from a one-to-one server/application hosting model to a clustered, one-to-many server/application hosting model. The real (known) cost of the existing model can then be compared to a new projected cost model under HP PolyServe.

It may also be advisable to familiarize database administrators (DBAs) with SAN technology if this will be their first experience with managing databases on networked storage. Yes, networked storage has been around for years, but some DBAs have yet to experience a close encounter.

Microsoft Cooperation

We are often asked whether consolidation isn't a Bad Thing for this software vendor or that systems company. After all, poor utilization would seem to lead to more license revenue and hardware sales. In practice, we find that users who are able to make efficient use of their computing infrastructure are often willing to spend more than those are having trouble getting value out of what they have. In this case, larger centralized databases from IBM and Oracle are one possible alternative to SQL Server sprawl. Thus, the users we interviewed didn't report any Microsoft push-back at all. Perhaps, partly as a result of Microsoft's support, none discussed looking at switching to a different database.

Alternatives

All of the interviewed users did, however, report investigating other ways to more effectively manage burgeoning SQL Server environments before encountering and finally going with HP PolyServe. One alternative was to use larger servers configured as dual-node Microsoft clusters to consolidate single instances and to improve SQL Server availability. The other approach considered was to migrate and consolidate instances of SQL Server to VMware. Each was found to be inadequate. Here we dig a bit deeper into the reasons why these alternatives came up short.

SQL Server Dual-node Clusters

For Microsoft SQL Server clustering, the common objections were lack of scalability and the requirement that server pairs consist of identical servers.

The general consensus was that larger, dual-node SQL Server clusters only get half-way toward ultimately solving the problem of SQL Server sprawl. While they may be less expensive from a software standpoint, they are easily more expensive from a hardware resource perspective. Large nodes are duplicated and have to be identical. That usually means having to source new hardware rather than using existing resources. Additionally, dual-node clusters were seen to be limited in the number of

⁴ tinyurl.com/2s95zy

instances they could support while maintaining acceptable performance levels. Adding/moving processing capacity within this environment was also seen as disruptive, requiring a multi-step SCSI reserve/release and file system unmount/mount/chkdsk process every time a disk resource needs to move between servers.

It probably didn't help that clustering has never been a primary focus of Microsoft. So, while Microsoft Cluster Server has improved in features and robustness over the years, it's never really become a front-and-center database capability—especially beyond basic failover function—in the same manner that IBM and Oracle have developed clustering in their DBMS products.

VMware ESX Server

Several of the users we spoke to investigated VMware as an SQL Server consolidation platform; two got so far as to perform some testing. Here, we detected uneasiness with regard to performance as more database instances were added. The uneasiness also crept over to Microsoft's perceived support, or more correctly, lack thereof, for SQL Server on VMware.

We have noted in our research that VMware users generally begin VMware deployments by first migrating those applications that are seen to be the easiest to move, then progressing to the more difficult and often more critical applications later on as IT administrators become more familiar and confident. At this point in server virtualization's evolution—especially on x86—running databases and other heavy-duty, I/O-intensive, and performance-critical business applications within VMs remains a leading-edge, if not bleeding-edge, use case for many shops. It's not unheard of, but it's still fairly uncommon in production environments, especially at high scale points, and especially in comparison to the virtualization of less-demanding applications.

The common concern expressed for consolidating SQL Server instances to a VMware server was an inability to predict the performance of I/O-

intensive database applications hosted on a VMware server. As a result, our HP PolyServe users felt that it would be difficult for them to stand behind SLAs for SQL Server.

The background of this concern lies in the way that I/O from guest virtual machines (in this case, virtual machines containing an SQL Server instance) gets handled. Normally, when an application wants to read from disk, it issues a system call such as a `read()`; the file system then maps to a location on the disk volume; a device driver requests the block (assuming SAN, rather than NAS, storage) from the physical storage device through a storage controller.⁵ There may be some additional mappings related to Logical Volume Managers or software RAID, but those are the basics.

With virtualized guests, on the other hand, quite a few additional steps have to take place. This is because the guest application device driver doesn't typically communicate with "real," i.e. physical, storage controllers. Rather, the driver typically turns I/O requests into accesses to I/O ports that communicate with virtual adapters using privileged x86 IN and OUT instructions, which are then trapped and appropriately processed by the virtual machine monitor (VMM, a.k.a. the hypervisor). Without going into further gory detail, suffice it to day that these levels of indirection and translation add overhead.

Characterizing VM I/O overhead is no easy task, and it's hard to draw generalizations.⁶ However, it's fair to generalize that I/O-intensive workloads (as are common with databases) stress server virtualization more than tasks that are more compute-bound. Based on a variety of published papers and studies, overheads in the 25 to 30 percent range for I/O-intensive environments seem to represent a "bad case," though not a pathological or worst-case result. Pass-through I/O (the ability to pass I/O directly from a VM to a physical

⁵ A clustered file system doesn't change these basics other than to check for conflicting requests from other cluster nodes.

⁶ tinyurl.com/39exkh

adapter) and enhancements at the processor level (Intel VT-d and AMD IOMMU) are among the options on the horizon to reduce I/O overhead in virtualized environments, but they remain in the future.

Implementation Considerations

HP PolyServe for Microsoft SQL Server includes a number of standard HP PolyServe elements, plus an integration layer that ties the core elements of HP PolyServe together with SQL Server. These elements include:

- Matrix Server
- Cluster volume manager, also referred to as the HP PolyServe Volume Manager
- Cluster file system (CFS), also referred to as the HP PolyServe File System (PSFS)
- Application Control Engine
- Matrix Manager

Here we outline some basic HP PolyServe implementation and environmental considerations.

Matrix Server

Matrix Server is the core shared data clustering software module for HP PolyServe. Up to 16 physical servers can make up an HP PolyServe cluster. Pre-requisites include:

- Intel or AMD-based servers (or server blades)
- Microsoft Windows Server 2003 or Windows Server 2003 R2
- Storage Area Network (SAN)-attached storage, either Fibre Channel (FC) or iSCSI

The physical cluster can be composed of a mix of servers and storage from multiple vendors, as long as they can share standard SAN and Ethernet connections, and run standard Windows operating environments.

Volume Manager

The Volume Manager manages storage from the perspective of the server cluster. It presents a single

pool of storage to the cluster, no matter how many of what types of arrays the storage pool is composed of. Using the Volume Manager console, an administrator can create volumes from free disk space on one or multiple arrays. Volumes can also be striped across multiple arrays. LUNs can be configured with the same or different capacities, and may exceed 2 TB each. An HP PolyServe cluster can support up to 256 16 TB file systems per cluster, and has 4 KB and 8 KB block size support. The Volume Manager can be used to expand the size of the file system without disruption.

Cluster File System

HP PolyServe's Cluster File System (CFS) is symmetric (i.e. no access mediation by a single file server or master server), fully journaled, and NTFS-compatible. With a distributed lock manager (DLM) it supports multiple, simultaneous, direct access by up to 16 SAN-attached servers within the HP PolyServe cluster. In typical deployments, NTFS is used for system boot, while HP PolyServe CFS is used for the application data.

Full data integrity is supported without partitioning the SAN-attached storage array(s). It also supports automatic recovery of multiple server failures without disruption to the surviving servers within the cluster.

Application Control Engine

The application control engine detects and responds to operational abnormalities in cluster hardware, interconnections, and software using a "heartbeat" mechanism across the Ethernet LAN connections that provide the interconnection between servers in the cluster. It also controls where applications run within the cluster. The LAN connections should be redundant, as HP PolyServe can automatically reconfigure around failed LAN connections. If a server fails, the application control engine will automatically restart the application on another server in the cluster, starting with a server designated by an IT administrator to be first on a ranked list of failover candidates. If a software failure occurs (while the hardware remains stable),

the application control engine will automatically restart the application in the same place.

One user advised that care should be taken in the planning stages to consider how a cluster recovery should proceed if required under a disaster recovery scenario where a cluster outage is experienced. How many servers will be required to restart the cluster? Which databases are the most critical and need to be back online first vs. other, perhaps less important, ones?

The application control engine can also be used by an IT administrator to start and stop applications running anywhere within the cluster, as well as to move a database instance from one to another server if for example the workload becomes too great for a server where an application is currently running.

Finally, the application control engine supports the concept of virtual hosts. A virtual host consists of an IP address and one or more applications that can be moved as a group among servers in the cluster. This enables remote clients to connect to an application running within the cluster no matter where it happens to be running.

Matrix Manager

Matrix Manager is a single “pane of glass” management portal for an entire HP PolyServe cluster. It shows the status of all cluster hardware and software components. It is also used for configuring, updating, moving, and managing data protection processes like backup. From the console, an IT administrator can:

- Add or remove server and storage resources non-disruptively
- Create new virtual hosts or move a virtual host to another server
- Create the prioritized list of server failover candidates noted above
- Review the cluster event log
- Create tailored views of the cluster associated with specific applications, for example

- Set-up event monitors and alerts

Many IT administrators rely on Command Line Interfaces (CLIs) to effect management processes that are specific to their environments. Thankfully, Matrix Manager also includes a CLI and supports the use of command line tools.

HP PolyServe Software for Microsoft SQL Server

This is HP’s integration layer between the HP PolyServe components described above and each SQL Server instance. It includes:

- A Health Monitor that periodically queries the status of each instance of SQL Server running within the cluster to assure its continual functioning
- The creation of Virtual SQL Servers, a concept that is similar to the Virtual Host concept presented earlier that allows SQL Server instances to be moved within the cluster to compensate for a physical server failure or for load balancing purposes
- A SQL Server Registry Replicator that stores configuration data for each SQL Server
- An SQL Server Installation and Hotfix Updater Agent—a “push” installer of SQL Server software patches, updates, and hotfixes. Using this agent, updates can be applied consistently across all servers within the cluster as a “one-time” process.

HP’s implementation enables separation of responsibilities and access between and among server administrators, storage administrators, DBAs, and other IT staff members. Integration with Active Directory (AD) also enables role management solely through AD, as well as directly through HP PolyServe.

Conclusion

Cluster File Systems have arguably never really lived up to their promise in commercial computing. Even the most sophisticated such in the Unix world ended up being used far more frequently for simple pair-wise failover than for more sophisticated “single system image” (SSI) deployments that

tackled management complexity as well as simple application uptime.

However, today, HP PolyServe has found a home in large-scale Microsoft SQL Server deployments as a consolidation platform. Managing a few SQL Server databases is easy for IT shops to wrangle. But when these instances become many, and when they start growing at the rate of 40, 50, or 100 per year, IT has a problem—one that cries out for effective consolidation. Our user interviews made that point loud and clear.

HP PolyServe not only increases the availability of each individual database instance to application users, but it greatly simplifies management. It's not the only approach to enable consolidation, but the users we spoke with had considered the options and decided that HP PolyServe has some distinct advantages, whether in flexibility, management ease, the ability to handle high I/O workloads, cost reduction, or a combination of the above.



Through subscription research, advisory services, speaking engagements, strategic planning, product selection assistance, and custom research, Illuminata helps enterprises and service providers establish successful information technology.