STORAGE SWITZERLAND

Optimizing MS-SQL AlwaysOn Availability Groups With Server SSD

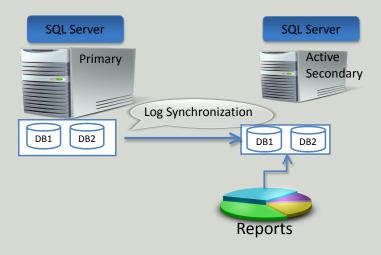**Storage Switzerland, LLC**

George Crump, Senior Analyst

Medium- to large-sized data centers count on applications like Microsoft's SQL Server to support applications that users need to become more productive so that the business can be more competitive. As these applications move into production the businesses become dependent on them and must be able to respond quickly to user requests and of course maintain consistent availability for users.

In the past meeting this demand has meant establishing complex and expensive clusters which required a knowledge of the cluster software but also a potentially complex storage area network (SAN). Traditionally, SAN-attached storage was needed to provide both the primary and secondary server shared access to the same application data in the event of a failure.

As an alternative, third party solutions have been developed that can replicate application data between local storage on two servers allowing for a standby-server type of design without the complexity of a SAN. The problem with these solutions was that the relatively low bandwidth of the inter-server connection created latency which impacted application response time. Until recently IT had to put up with the SAN's cost and complexity since most of these production applications couldn't tolerate this latency.

Microsoft, in an effort to address high cost and complexity issues, and of course broaden the appeal of Microsoft SQL Server in the enterprise, recently introduced a new capability in Microsoft SQL 2012 called "AlwaysOn Availability Groups" (AAGs). AAGs allows Direct Attached Storage (DAS) to provide an affordable alternative to SAN attached storage.

An availability group supports a failover environment for a discrete set of user databases, known as "availability databases", which failover as a group to one or up to four secondary servers. Databases on secondary servers can also be made available for read-only access for reporting and backup tasks, which protects the primary server from being impacted by these activities.

AAGs are part of a growing trend where software application providers are giving IT Managers the means for achieving high availability without the need for a shared storage resource. Examples beyond MS-SQL 2012 include VMware's vStorage Appliance and Exchange 2012. Direct Attached Storage (DAS) is becoming a viable option for keeping costs down while maintaining application uptime.

The AAG tracks modifications to the database as log files and transmits just those changes to secondary servers in either a synchronous or asynchronous fashion. For more extreme protection multiple servers can be set up as secondaries. But in these cases only one secondary server can be designated for synchronous replication, in order to ensure 100% consistency with the primary server.

Synchronous log transmission means that all modifications on a secondary server are acknowledged prior to being fully committed by the primary server. As a result, data consistency is guaranteed, but this comes at the expense of potential performance loss to the SQL Server processing on the primary since it must wait for the acknowledgement.

Using the asynchronous mode, on the other hand, means that the primary server does not wait for acknowledgement. It assumes that the write will be completed so there is no guarantee that data on the network won't be lost during a failure. This mode is typically used for a disaster recovery data center where the geographical separation between the primary and secondary sites would cause unacceptable delays processing on the primary.

(For more details on AAGs see Microsoft's SQL Server 2012 web site)

Optimizing The Environment for Synchronous AAG Mode

There are two key steps in addressing the performance challenges with AAG's synchronous mode. The inter-server network has to be fast and the response time of the DAS in the secondary server needs to improve. Both of these steps should be taken though, as only addressing one still leaves a significant performance bottleneck that may make synchronous mode too slow for production use.

Improving The Network Connection for Synchronous AAG

The typical connection between two servers in the availability group design is a dedicated Ethernet card, often a simple 1GbE connection. Other than the obvious bandwidth issues there is also considerable latency involved with having to process data through the TCP/IP stack. This takes time (latency) and also requires CPU processing resources that the application running on MS-SQL server may need.

An ideal alternative is to use 10GigE or Infiniband to support the communication between the primary and secondary servers. These link protocols provide fast transfer rates and, when used with SMB Direct, can provide very low latencies because log transfers are implemented by hardware using Remote Direct Memory Transfer (RDMA) instead of TCP/IP.

Improving DAS Latency for Synchronous AAG

As is typically the case the first thought in an AAG design is to equip the servers with solid state storage exclusively. While this would address the performance problem, it does raise the cost significantly, especially if the database is of a reasonable size. There is still about a 12X price premium on solid state disk (SSD) vs. hard disk drives (HDD) and practically speaking, only parts of a MS-SQL database will actually benefit from being placed on SSD.

Examples of frequently accessed areas in a database are indexes and Tempdb. A database index allows quickly finding information in a database. Tempdb is similar to a scratchpad. Providing low latency access to these areas can significantly increase database performance. Caching intelligence within the DAS storage can monitor transactions and will locate frequently accessed blocks of data on low latency flash technology. For economy, the entire database will be on HDDs, but flash will contain a copy of portions of the database for fast access.

The Mellanox / LSI Example

An example of this design was recently demonstrated at the PASS Summit. This is an annual conference hosted by the Professional Association of SQL Server, an independent, not-for-profit association dedicated to supporting, educating and promoting the global Microsoft SQL Server community. In this demonstration two Microsoft SQL Servers were set up, one as a primary node the other as a secondary. The connection between the servers was made using a Mellanox RDMA Infiniband I/O) Adapter. The hard drives in both servers were connected to an LSI Nytro MegaRAID card.

Primary and secondary databases are kept concurrent by sending transaction logs quickly over a Mellanox ConnectX-3 Infiniband I/O adapter. The Nytro MegaRAID (covered in this Storage Switzerland Product Analysis) card combines RAID support for HDDs, on-card flash storage

and intelligent caching software. This card stores frequently accessed data on flash, while the remaining storage remains on HDDs to create the most economical solution. The flash technology provides low latency acknowledgements to SQL Server that data has been stored on non-volatile memory.

This combined Infiniband networking and PCIe caching solution gives DBAs and storage administrators the ability to provide database applications both high performance and high availability while enjoying the economy of cost effective DAS storage.

Conclusion

Microsoft SQL Server is widely used by enterprises and large businesses, frequently in performance-critical production applications that simply can't go down. Being able to support this performance and maintain continuous application uptime without the major investment of a SAN is a significant benefit. DAS storage, when used in AAGs, provides that capability, but can still risk a loss in performance. Combining Infiniband with a server-side SSD cache addresses both the performance and high-availability requirements. This solution, like the one demonstrated by LSI and Mellanox, is an excellent example of how to implement a cost effective, highly available database environment without compromising performance.