ADAM LEVIN

SAN vs. NAS for Oracle

A TALE OF TWO PROTOCOLS

Adam Levin has been a UNIX sysadmin for over 10 years. He's currently employed at Audible.com, a company that sells audiobooks and other spoken word content over the Internet.

levins@westnet.com

A COUPLE OF YEARS AGO, OUR COMPANY decided to centralize our storage. It was a pretty big plunge, but we knew it was the right way to go, based on the type of data we had and the way we served it to our customers. Our company sells spoken-word audio (radio programs, newspaper reads, audiobooks, speeches, special events, etc.). These audio files are available in a variety of formats, and we settled on a "many small servers" architecture to deliver the content. Therefore, we decided to set up a large number of relatively small Web servers, all reading the same data off of a central disk storage unit. We needed high capacity and high availability, along with good performance and ease of data integrity (backups and restores). Since we were going to be spending good money on a solution, we decided to try to leverage that solution for our Oracle database full of customer records as well. The economies of scale, as well as the reliability and performance, of a large central disk unit made complete sense. Eventually, though, we realized that the complexity and expense of a storage area network didn't make sense in our application, and the simple network-attached solution would end up being better for us.

We eventually settled on a Network Appliance unit, for several reasons. One of the keys for us was that we wanted to keep our Oracle environment on "real" disks, as opposed to network-attached disks. The Network Appliance allowed us to support both network-attached storage (NAS) and, through a Brocade switching fabric, a storage area network (SAN). Therefore we decided to go with SAN for our Oracle database, so that we could control the file system, and NAS for our audio content, letting the NetApp deal with file system details. We expected that for Oracle, the SAN would perform better. We liked the idea of keeping Oracle on native file systems, since we were used to them and knew how to work with them, back them up, etc. Besides, we didn't need to share the file system among several or many machines.

There were a couple of catches, though, that caused us problems from the beginning, and planning for the future eventually led us in another direction.

The Oracle machine was an aging Sun Enterprise E4500 with SBus I/O boards—no PCI to be found, and no room to add I/O boards. NetApp would have loved to sell us their connector kits, but they didn't support SBus, so we decided to go ahead and buy Sun SBus host bus adapters (HBAs—the cards speak SCSI and connect the host computer to the SAN disk unit, as opposed to a NIC, or network interface card, that speaks Ethernet and connects the host computer to the network or NAS unit). These are 1Gb HBAs, even though the switches and the NetApp support 2Gb. So, right away we took a performance hit. In addition, PCI bus transfer rates are better than SBus transfer rates, but we didn't want to spend too much money on the upgrade.

To keep costs down further, we opted to stay with Sun's file systems rather than switching over to Veritas. We used Sun's MPXIO (an I/O multipathing driver similar to Veritas DMP, or direct multipathing) with some success, but there were some problems there, too. We wanted redundancy and fault tolerance as well as the added performance of multiple paths to the disks. The NetApp has a clustered filer head, meaning it's got two filer heads that generally work independently, but they are interconnected so that if one fails, the other will take over, on both the network and the SAN.

We bought two Brocade switches that came with the NetApp, so that both filer heads and all hosts could be connected to both switches, to further add redundancy and failover capability to the system.

Ultimately, with both filer heads and all the Sun machines all connected to both switches, we should have had a lot of redundancy, but the configuration (both hardware and software) was quite complex. The complexity made it not only difficult to understand, but also difficult to troubleshoot and to accurately test various failure scenarios. This proved important later on, after we went live with the whole thing.

We had hashed out all of the hardware configuration and install details with NetApp's people, but of course they could only help us so much, since we were technically using an unsupported configuration. They warned us that we might have some difficulties.

Going Live

The first problem was that MPXIO was an all-or-nothing proposition. Technically, we should have only been using half the paths to our NetApp as primary paths, and the other half (the paths that went to the second NetApp filer head) as standby. This is how it normally works when you use Veritas's DMP with NetApp's supported adapters and drivers in your Sun machine. Since we weren't using those, but, rather, all Sun gear, we couldn't set it up the recommended way. Instead, our Oracle server could see all of the paths through both filer heads as primaries, and would use them all. This meant that half of our traffic was going over the inter-cluster link on the NetApp—not an ideal situation. The NetApp generated warning messages about this, and it's likely that our performance was degraded because of it (the inter-cluster link is a 1Gb fiber connection).

The second problem was that nobody in our department had much knowledge of SAN architectures. We're a small group and were so busy putting out fires that we hardly had time to deal with it. This was more significant because of the number and complexity of the connections. We had tested NetApp filer head failover, HBA failure on the Sun, and cable failures. We even tested switch failure by unplugging one of the Brocade switches. It turns out, though, that if the other switch fails (the one that we naturally failed to test), then the whole SAN fabric gets very, very confused. This happened once, several months after going into production, and caused a downtime for our customers.

Next Pass

As the company and our customer base grew, it became time to upgrade the Oracle server. This gave us an opportunity to reevaluate our system configuration, including the disk system. Our data warehousing project was really taking off, and getting data from the production Oracle machine to the warehouse was taking longer and longer. Additionally, while the snapshot capability of the NetApp is impressive, the SAN snapshots work slightly differently, and use up more disk space because of it (there are benefits, though, in that you can take a snapshot, create a LUN out of it, and mount it as writable, effectively "copying" your data in seconds). We also began investigating Oracle 9i RAC (RAC stands for Real Application Cluster, an Oracle way of saying that you have multiple machines serving the same data to your applications, for redundancy). RAC would have required Veritas in several ways (Cluster Server and File System, primarily, but that would have followed neatly into Database Edition). We still wanted to avoid the added expense of Veritas, not to mention the complexity involved.

We began diagramming what the cluster would look like and the kinds of connections we would require. We'd have to upgrade our Brocade switches, too, because we'd have more than the small eight-port switches could handle. Plus, we still didn't know why losing a switch caused our fabric to fail.

We quickly began to realize that NFS could save us a lot of time and trouble, and since NetApp and Oracle 9i RAC already work together and support each other, we figured it would work out well. It turned out that we didn't go RAC after all, but just stayed with 9i, because of the nature of some of our applications: they open connections to the database and then stay connected. If the database node fails, all connections to that node fail and must be restarted, causing problems with our site. Since they have to be restarted anyway, it's simpler and cleaner to have one Oracle machine, and a warm-spare standing by in case of a hardware failure. With our Oracle data on the NAS volumes, switching between machines is trivial. We're still working on getting those applications to fail gracefully, at which point we can start looking at Oracle clusters again.

We got Oracle involved, asking them for all of the relevant material concerning running Oracle 9i and 9i RAC on NetApp Filers. We also got some documentation from NetApp regarding Oracle 9i, RAC, tuning, and things related to that. We spoke to Sun about multipathing and failover, and decided to go with their Sun Trunking software, in addition to using IP Multipathing (IPMP, a standard network configuration option included in Solaris where we can set up some network ports, or trunks with the trunking software, as standby for the primaries). Since this is native to the Sun Solaris machine and designed to be primary/failover, it works very smoothly.

Once the new Sun hardware was in, it was time to start working on the actual setup and configuration. We set up the Etherchannel paths on the Cisco switches and the IPMP trunks on the Suns, and began testing. We had been given a list of NFS options recommended by Oracle and NetApp, with the caveat that we should test to see which options actually gave us the best performance. We tested I/O using dd to copy data back and forth, and we used the SE Toolkit's nx.se program to monitor network traffic, as well as timing the tests with the time command. We found that the extra NFS options (forcedirectio and noac, mainly) weren't worth it, and the defaults performed as well or better. We did decide to stick with TCP rather than UDP, for integrity of the data in the event of network problems. UDP is faster than TCP because it's designed for speed rather than accuracy, and we use UDP for some of our other NFS data that's read-only. However, TCP gives us flow control and error correction, which is important for our data integrity.

We had, and continue to have, a problem with the Etherchannel and Sun trunks. It turns out that trunking works best primarily when many smaller machines are talking to one large, trunked machine. The default algorithm for balancing the traffic load across the ports in the trunk is to use MAC address hashing. Basically, each machine that connects to the trunked host machine is assigned a hash value that corresponds to one of the network ports in the trunk. So, if you have 100 clients, they'll be balanced pretty close to 25 per port in a four-port trunk.

In our case, though, the Oracle machine is just one machine, and even though we have two ports trunked, the traffic appears to come from only one of them. We changed the default MAC hashing on the Sun to round robin, which forces packets to share all ports. This can cause out-of-order problems, which we don't believe we've seen, at least not yet.

The NetApp also provides the ability to go round robin, but it's all or nothing; round robin would cause other hosts to suffer through out-of-order problems as well, and we don't think we want to do that just yet, so while the Sun talks to the NetApp balancing the traffic over both interfaces in the trunk, the four-port trunk on the NetApp side is only using one interface of the four, and sending traffic back to only one of the two Sun interfaces.

We're still experimenting somewhat with these performance issues. We're seeing burst speeds of 60MB/sec from the Oracle machine to the NetApp, which is very respectable for a 1Gb connection, but not so good considering the trunk. Hopefully, we can work out a plan to get the Oracle machine and the NetApp to talk to each other over all ports instead of just one. At the very least, though, the trunking gives us fault tolerance, because if one port fails, there's no network hiccup while the other port(s) in the trunk continue to carry traffic, which is important for our site and our customers. The most important element now is that even without the full trunk running, we're getting better I/O than we saw with the old Oracle machine over the SAN.

Another problem we're currently seeing is a high number of mutex locks per CPU on the Oracle server. We never saw this before. We've started working with Sun engineers on the problem, and they believe it's related to a network configuration or hardware problem, perhaps a bad network port. It remains a mystery for the moment, but since the new machine is so fast, it hasn't caused us significant problems, so we've got time to figure out what's causing the mutex problem and to solve it.

Overall, we're very happy with the NAS solution. The snapshots are a breeze to manage, and they'll be even more effective when we license the SnapRestore product, which will enable us to restore one file or an entire volume of data in seconds. We've freed up significant disk space after switching away from LUN snapshots, and we've been able to significantly lessen the amount of time required to transfer data from our production database to the data warehouse, which is also connected via NAS to our NetApp.

REFERENCES

Sun Cluster: http://www.sun.com/software/cluster/faq.xml

Oracle 9i RAC: http://www.netapp.com/tech_library/3165.html

Oracle 9i Grid Computing: http://otn.oracle.com/products/oracle9i/grid_computing
/Oracle9iGridCookbook.html