



CorData White Paper

Snapshot and Replication

August 5, 2014

The Swimming Pool/Garden Hose problem

Swimming pools hold very large quantities of water. An Olympic-sized swimming pool holds over 660,000 gallons of water. Garden hoses, on the other hand, represent a modest amount of water flow. If your garden hose moves only about ten gallons of water per minute, it would take you about 45 days to fill an Olympic sized swimming pool.

Similarly, if you have a terabyte of data in your storage array and attempt to copy it over a 1Mb network connection, it will take about two and a half hours to copy the data--assuming, of course, that your network connection is not already busy with other traffic.

But what happens when you've got Olympic-sized storage requirements? When you have fifty or a hundred or five hundred Terabytes of data; and your copy time goes to five days, or ten days, or a month-and-a-half?

This is when you might consider using snapshot and replication technology to move data between sites. It's a much more efficient way to manage data in multiple sites than copying your entire dataset.

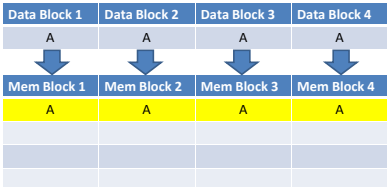
The power of snapshot technology is in the snapshot engine, and with Nexsan's NST storage array, their redirect-on-write technology. The NST uses the Zettabyte File System, or ZFS, originally developed by Sun Microsystems as a high-performance file system for Solaris. It's a 128 bit file system, allowing for very large memory addressing. The technology for the snapshot engine is referred to as "Redirect on Write", an enhanced version of the older "Copy on Write" snapshot technology.



Here's a simplified example of how the NST snapshot engine works.

When new data is written to disk, it's written as a series of data blocks.

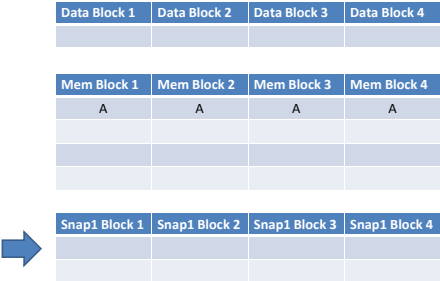
Snapshot Engine Inbound data is stored on disk



23

After the data is written to disk, we start the first snapshot on the data as it sits on disk. Note there is nothing stored in the snapshot area yet, since nothing about the original data has changed. Also note, starting a snapshot takes almost no system resources, so you don't have an impact on performance or applications by creating multiple snapshots, even lots of them.

Snapshot Engine Snapshot 1 starts

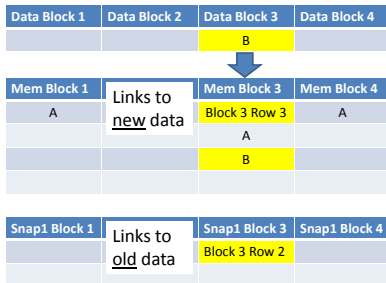


24

Now some new data, an updated data block 'B', arrives. Instead of overwriting the old block 'A', the system stores the new data in a different location on disk, and sets a couple of pointers. The memory pointer now points to the new block 'B', making it a current part of the file. Snapshot 1, however, stores a pointer indicating the old data block 'A'. This is so the snapshot, when it's assembled from its pointers, always shows the file in its original, unaltered state at the time the snapshot started.

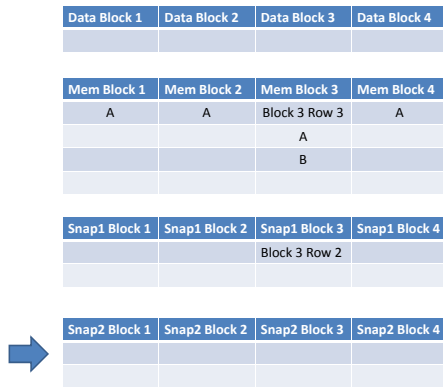
Snapshot Engine

New data arrives, pointers added



Snapshot 2 starts. You can have up to 512 shares, and 2,048 snapshots per share on the NST system.

Snapshot Engine Snapshot 2 starts

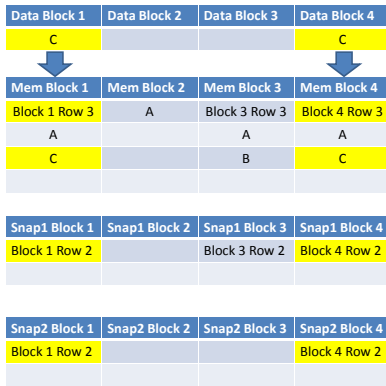




More data arrives, with two updated blocks 'C'. Pointers are updated; so memory, snapshot 1, and snapshot 2 are reflect the update. Performance is very fast on the NST, since metadata such as pointers are maintained in high speed, protected memory.

Snapshot Engine

More data arrives, pointers updated.

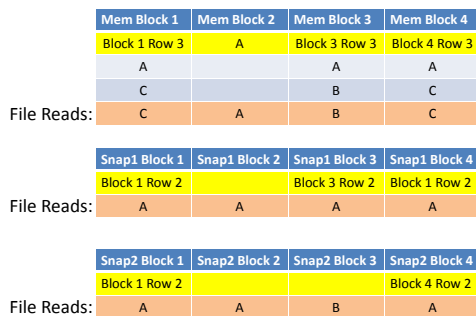


27

Now you see the current contents of the file versions on disk and in Snapshot 1 and Snapshot 2, all assembled from the unchanged data as originally stored, and pointers. Anytime a file is read, weather the current contents or a snapshot, the file is reassembled very quickly from pointers so your Snapshots always reflect how the file looked at the time a snapshot started.

Snapshot Engine

Data is assembled from pointers



28



Backing up your data from snapshots

Backup systems are able to use snapshots to copy the snapped data to tape or to another site without affecting the servers using that live data. You create snapshots much more frequently than backups, since they don't affect server applications to do so. You can automatically perform a snapshot every fifteen minutes if you wish.

Storage systems such as the Nexsan NST Unified Storage Array automates snapshots, backups, and replicate data to other sites, so it becomes very easy to completely protect your valuable data from problems. If your data is corrupted, you can easily convert the affected data back to the most recent snapshot or go back to a time before the data was impacted and pull out or copy single files.

Replication works this way too. You either replicate synchronously, where both sites always match up with the same data, or asynchronously, where you push data to the other site and it gets there as quickly as it can. Snapshots are used for asynchronous replication, and they're de-duplicated prior to transmission to further save on bandwidth. With synchronous data mirroring, you have absolute mirrored data, but data can only change as fast as the remote location can receive and acknowledge it. Systems like the NST use very fast system interconnects, SAS or fibre channel, to create synchronized data on two separate NST systems up to 25 kilometers apart utilizing long-wave fibre channel cable runs.

For a metro cluster or wide area network, asynchronous replication works best. The NST supports both types of synchronization, so you may use synchronous mirroring for critical data that must be mirrored in real-time or asynchronous mirroring to mirror data off-site. The NST even supports many-to-one synchronization, so you can use one site as a replication site supporting multiple NST systems.

With most backup systems, data reacquisition is a big problem. When your backup data is on a tape cartridge, you must read it off tape and back onto an available storage array first, in order to redeploy the data. And this can take days or weeks when you're dealing with many terabytes of data.

With remote asynchronous replication, all you need is an adequate network connection and your applications are back up and running in minutes.

Why perform snapshots and replication in the storage array rather than with my backup software?

Many backup software companies offer replication capabilities. They use local backup servers to run software which snapshots and replicates data between sites, and they support disparate storage hardware attached via storage networks. Sometimes this is a good solution if you've inherited a bunch of different hardware, all with different brand names.



But if you have the chance, it's much better to perform snapshot and replication in the storage array. This is because you eliminate many performance bottlenecks between your application servers, the backup servers, and the stored data. And you eliminate large expenses, because all those servers and software licenses and support contracts to run the backup software add up to big money. And you eliminate complexity, because you reduce the many different hardware and software systems which must be integrated, since they all come from different companies, representing different management tools. That can be very expensive.

The hidden big expense with backup systems – intersystem dependencies

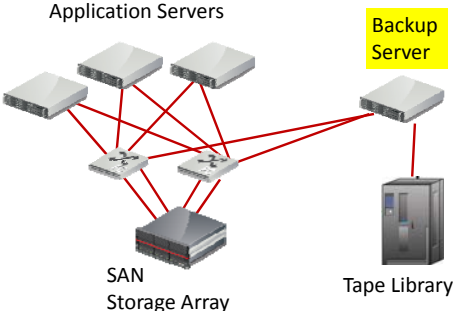
The biggest expense with backup software is not the software licenses, but interdependencies which can cause complex problems when you upgrade or change any of the components of your data protection architecture.

For instance, your backup software vendor approves only certain server hardware, network cards, storage arrays, operating systems, and network topologies. And their list changes for every version of the software they release. If you bring in new hardware, you have to make sure it complies. And if the software vendor brings out a new release, you may have to upgrade operating systems application software, server hardware, and even your network cards to insure it all works together properly. That can be very, very expensive.

If you consider all the requirements and costs for backup, scheduling, data movement, compression, de-duplication, and application agent software licenses, you can see the cost advantages of a smart storage system like the Nexsan NST which performs snapshot and replication right in the storage array. With nothing more than a couple of NST systems far enough apart, you have a complete data protection and disaster recovery solution, and it's all managed from a single pane of glass. And with price-correct storage attached, there's no need to deal with de-duplication, a storage solution with slow write performance, premium prices, added complexity, and the fact that some data simply doesn't compress or de-duplicate very well.

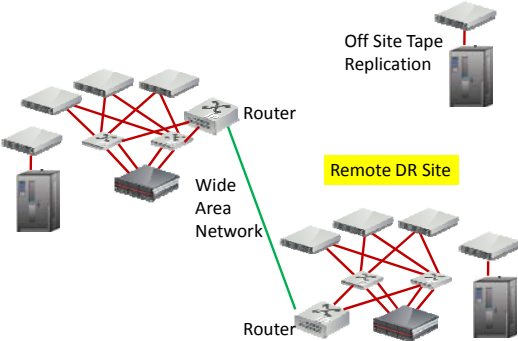
Let's explore how a Nexsan NST storage cluster completely protects your data. Here's a typical backup solution with a local tape library to store the backups.

Typical Backup Solution



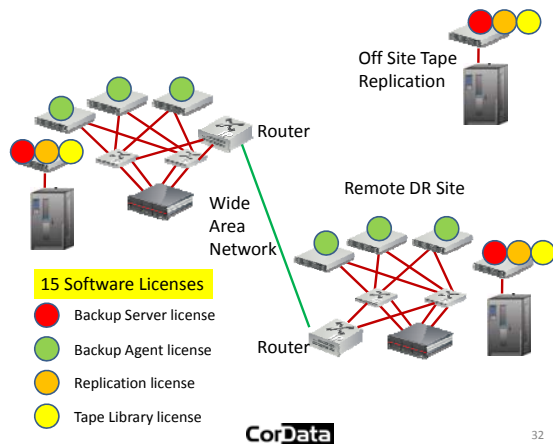
Here's a typical second site for disaster recovery. Note the second off-site tape library and backup server; this is to recover data from tapes at the remote site. Now, you've got a second set of servers to access the data, and a second storage array to use to reacquire the data. And off-site tape vaulting or replication, just to be sure your data is safe.

Typical Disaster Recovery Solution



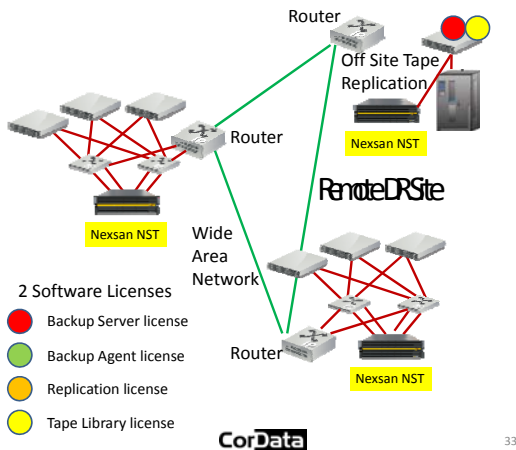
Here are all the different software licenses it takes to perform this sort of a backup and disaster recovery solution. And the data reacquisition time to recover the data from tape to disk at the remote site could take days of load-time if you're recovering the data from tape back onto disk before it can be used.

Typical Disaster Recovery Solution



Now, let's use the Nexsan NST system to perform the same functions. Note we will use scripts or the standard Microsoft Volume Snapshot Service, VSS, to perform all the application agent functions. This is an industry standard approach, rather than a proprietary backup solution, to normalizing application data before a snapshot or backup is performed. Notice how the local and remote site backup servers disappear, as well as all the associated software license costs. Now, you only have to copy data from the NST to tape at the tape replication site. But most importantly, the data recovery time goes from days to minutes. With the NST, you've always got a usable image of your data on disk—ready to go.

Modern Disaster Recovery Solution





Optimizing your architecture to reduce costs, complexity, and effort

As your data assets grow, you'll want to re-architect your data protection strategies to better match with your data recovery goals. But it's difficult to do in many cases. You have existing systems in place which represent years of development time, and groups of employees who have years of experience with the many software tools you are employing.

However, we usually see immediate customer returns from upgrading from a backup software-based vendor solution to a storage array-based solution. And in almost all cases, you can continue to use the backup system and scheduler you already have; it's the expensive application agent and remote replication licenses you can eliminate—and stop paying for.

When you see the financial advantages to a storage system-based backup and disaster recovery solution, and then add to that the obvious advantages of much faster data reacquisition and recovery times, and then add to that a substantial reduction in complexity and inter-system dependencies; the choice is clear. It's time to stop the flow of wasted spending.

About CorData

CorData is a storage systems integrator based in the Washington, DC area. We work with US intelligence, military, and civilian agencies, commercial clients, and research and development organizations world-wide.

For more than thirteen years, CorData has provided data storage insights to help our clients achieve their key business objectives.

Headquarters
620 Herndon Parkway
Suite 340
Herndon, VA 20171
703-986-3808
info@CorDataSys.com
www.CorDataSys.com

Copyright Notice
Copyright 2014 CorData. All rights reserved.