

STORAGE SWITZERLAND

SCALE OUT NAS STORAGE – GRID VS. CLUSTER

WHICH STORAGE INFRASTRUCTURE IS BEST FOR THE MODERN DATA CENTER?



George Crump, Senior Analyst

The modern day data center's storage infrastructure is under assault. Virtual servers and desktops have demanding and widely varying I/O patterns, while back office and customer facing applications need high performance and reliability. Even file sharing has become a critical function and the key component to successful collaboration. This has led to a need for new storage systems that can scale to meet the demands of these environments.

Clustered, scale out storage seems to be the go-to solution from storage suppliers, with many of the major vendors going out of their way to acquire such technology if they don't have it already. While scale out storage via a clustered implementation has been the most common, scale out storage via a grid design is now gaining in popularity and is an option worth considering.

What is a Scale Out Cluster?

While it's impossible to describe every permutation of clustered, scale out storage on the market, there are some commonalities to the design. A clustered, scale out system is built a node at a time. Each node is typically a

low-end, server-class system that has storage capacity, I/O and CPU processing capabilities. As you add each node to the environment the aggregated capacity, bandwidth and processing power is supposed to scale at a linear rate. Each of these nodes is connected by a private backplane network for internode communication.

Most of these scale out clusters no longer have a single, or a few nodes that are responsible for ingesting data. Instead, the first available node will receive any inbound data which is typically replicated or encoded. "Replicate" means that the controller duplicates and transfers the entire object (typically a file) to another location in the cluster. Replication will almost always imply some sort of mirror of data instead of the more space-efficient parity based protection, which is what encoding does.

Encoding is one type of data dispersion. Think of encoding as the process of segmenting data so that individual blocks can be written on every, or most of the nodes in the environment. The goal is that when that file is read back all the nodes can send their segments and the connecting client or application will see the aggregated performance of the cluster.

In most clustered, scale out storage systems these encoding or replication processes means that write I/O is always bottlenecked by the receiving node, since the entire file or object must be received by a single node at a time. For every write to a node two or more writes have to occur on other nodes. This can consume 50% of the IOPS when handling replicas. The encoding or replica process also means that there is a significant amount of I/O that occurs on the backplane network as data is encoded or replicated and then copied to other nodes as fast as possible.

Since multiple nodes can receive file data at the same time, in a write-heavy environment this means the backplane network has to support very high throughput in order to move all of the data around the cluster. While a high performance backplane network can be built it certainly adds to the expense of the storage system. Using a lesser network means relegating the system to something other than primary storage where performance is critical.

Finally, the encoding process also handles the data protection algorithm as well. Many clustered storage systems have the ability to set protection levels by volume or even by file type. The encoding process has to match up the type of file or destination volume and then generate the appropriate parity to meet that protection level. Once again, this is more work for the encoder and more traffic for the backplane network.

In addition to all of this behind-the-scenes processing, each node is going to be responsible for the 'front and center' data services like volume management, snapshots, replication and cloning. All of these services place a burden on each node in the scale out cluster as well.

Clusters using replication instead of encoding may allow for simpler node use but they have an added capacity requirement. More capacity means more floor space, more power and more cooling. For example, 100TB of raw data can require 200 or 300TB in a replication-based cluster design. This of course must all be managed, housed,

powered and cooled.

To resolve this issue, individual nodes on the system cannot be low-end servers as described above. Instead these systems must have multiple processors, multiple cores and multiple network connections to keep up with the amount of work they have to perform both behind the scenes and with traditional data services. Once again this adds cost to the storage system. It also adds a layer of complexity as putting more components into the node adds to the likelihood of a node failure which can then have a cascading effect on the rest of the infrastructure.

When it comes to high availability (HA), scale out storage clusters provide fail over HA only. They can ensure the restart of services on another node, but there may be interruptions to users. Fault Tolerance (HA without interruption) requires further customization of the node which once again, add costs and complexity.

What is Grid Storage?

Grid storage systems like those offered by [Gridstore](#) take a different approach to addressing the scalable storage challenge. Similar to clustered scale out storage, a storage grid is built a node at a time from low-end servers. The goal of a scalable grid storage system though is to use these economical servers and not require the use of an expensive, high speed, backend network in order to provide performance that's acceptable for primary data use.

To accomplish this goal grid storage systems change where the encoding process is performed. Instead of doing this encoding on the storage nodes themselves the encoding is actually done on the client that's storing the data.

A virtual controller is loaded onto each connecting client that understands how the grid is designed. It also provides intelligence like locking and meta data look-ups, which improves performance of the infrastructure as a whole and lessens the load on each node. A major function of this local encoding is that it optimizes data for storage on the nodes before that data hits the network. As it starts each write operation, data is segmented and parity bits (for redundancy) are generated. The client writes each segment directly to each node instead of having the node segment the data.

Ramifications of Grid Storage

First, Grid storage allows for the use of simpler, lower power and less expensive nodes. The lower the complexity of the node the lower the number of things that can go wrong or the things that need to be constantly fine tuned. Simplification and cost savings also comes from not having to install a specialized backplane network, simply plug the grid into the existing network infrastructure.

There is also a better potential for performance since the workload is being distributed between the clients and the nodes. Companies that provide grid based storage like Gridstore find that they can handle 3X as much throughput, can rebuild failed nodes 3X faster and can be 3X as efficient with capacity (when compared to replicated clusters).

Grid storage can also provide fault tolerance without the cost or complexity. To date, fault tolerant systems were beyond the reach of the majority of companies due to the high cost of providing redundant fail over components within every part of the system. The lower bar become HA

by clustering systems together so they appear and act as a single system while providing a fail over mechanism when a server fails, and the services restart on another.

A grid takes a different approach and delivers fault tolerance while eliminating the cost and complexity of the cluster model. Instead of a client communicating with a single node which does all the work to then distribute data across the cluster - a grid distributes the processing normally done on the storage node to virtual controllers that operate in each client. The virtual controller calculates the parity stripes and writes the fractional data of the file in parallel streams directly to each storage node. If any of the storage nodes fails (or even multiple nodes) during the I/O, the client is unaware of it and unaffected. The I/O will complete with no disruption or data loss.

By eliminating the need for a cluster and allowing the virtual controller to optimize data before it hits the network, Gridstore has found an innovative way to offer fault tolerance with the cost or added complexity.

Summary

Scale out storage continues to grow in popularity but many customers, assuming that cluster based, scale out storage is their only option, simply tolerate its shortcomings. Grid storage solutions like those available from Gridstore are an excellent alternative. By eliminating the cost, complexity and overhead of a cluster, these solutions provide scalable performance and capacity at a more cost effective price point while raising the bar from just high availability systems to completely fault tolerant systems.

About Storage Switzerland

Storage Switzerland is an analyst firm focused on the virtualization and storage marketplaces.

For more information please visit our web site: www.storage-switzerland.com

Copyright © 2011 Storage Switzerland, Inc. - All rights reserved