The Value of a Scale-out Storage Architecture

Introduction

Scaling storage sounds simple: Just add more disk drives! But there's a lot more to it than that, and the enterprise storage industry is running into a wall with conventional approaches. That's why next-generation datacenter and cloud storage systems adopt a different approach from the scale-up storage systems of the past. They divide the storage workload across a network of semi-autonomous nodes, just like cloud systems from companies like Google and Facebook.

Challenges

Legacy storage systems were never meant to scale. Rather than engineering systems and protocols for flexibility, the storage industry focused on reliability and data services. But this leaves them with an architecture that simply cannot meet the needs of modern IT organizations.

Scaling issues start with the disk drives. The traditional way to build a storage system is to use RAID to permanently "marry" a set of hard disk drives together. Then these are placed in a disk shelf, which connects to a disk controller. All of this is locked into place when the system is set up. Before any data is written to a disk, its final configuration must be determined.

In the 1990's, the only way to expand an existing storage system was to "scale-up" by adding more shelves and hard disk drives. But the controller remained the same, along with whatever CPU, memory, and I/O resources it contained. Although it is possible to add a shelf or two of disks to most so-called modular storage systems, the controller eventually becomes a bottleneck of the whole system.

Scaling beyond the controller is more difficult than it sounds due to the legacy storage protocols most operating systems demand. SCSI-based protocols like Fibre Channel and iSCSI encode the target controller in every packet, making it difficult to shift the workload to another controller. File-based protocols like NFS and SMB are somewhat more flexible but the underlying system architecture for most systems is still SCSI-based.

The conventional solution was to create a cluster of storage controllers and disk shelves. These systems are tightlycoupled, sharing information about every I/O operation with every controller. Eventually they too run into bottlenecks when this traffic gets too heavy. Plus, it's difficult for them to relocate data when hard disk drives are deployed in a fixed RAID arrangement.

Solution

The problem of scaling has been solved in Internet-scale companies by replacing the tightly-coupled cluster with a system of loosely-coupled nodes. Each node handles a part of the work, and everything is re-balanced when nodesare added or removed. This is how Google can build an index of the whole Internet, and it's also how the OneBlox works.

The main issue with so-called hyperscale approaches is compatibility. Conventional datacenter applications are not built to deal with a scale-out storage solution that uses objects instead of files or blocks. Vendors have built gateways that translate legacy storage I/O into object or API storage access, but these too have the same scalability limitations as a storage array.

OneBlox Scale-out Architecture

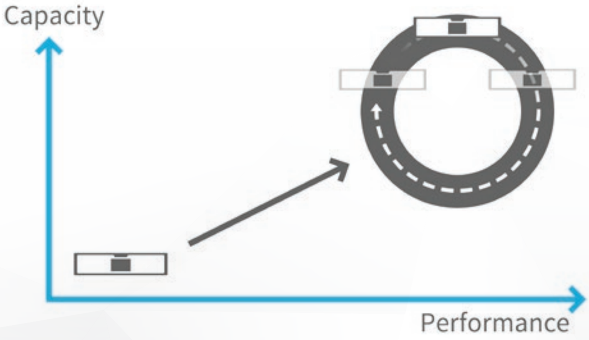
At OneBlox, we have reimagined the way storage should be built. Our system has a true scale-out architecture yet maintains compatibility with conventional operating systems and applications. It's a true shared-nothing architecture, with each node only maintaining a map of what it contains and its place in the system. Here's how it works.

As storage I/O is received by any OneBlox appliance, everything is broken up into 32 KB objects and then cryptographically hashed using the industry-standard SHA1 algorithm. This creates a unique ID for each object. Then the objects are distributed across the entire Ring. These objects can also be passed to another OneBlox Ring in a mesh, allowing data to be sent to another location, local or remote, for safekeeping.

Disk capacity and OneBlox appliances are organized into a Ring, and each element takes on an equal amount of incoming data. Our data distribution algorithm makes sure the data is balanced across the resources, protecting against 2 drive failures or 2 OneBlox failures and that extra copies are kept in different locations in the Ring. This not only ensures data availability but also accelerates performance since every part of our system is involved.

As disks and OneBlox appliances are added, our Ring architecture quickly and easily expands the whole system. Data is automatically re-balanced fairly when any component is added or removed. And in the event of a failure, the entire Ring pitches in to recover the data and get things running normally again.

There is no centralization of resources in a OneBlox Ring. Each node stores its own data and our algorithm means it always knows where to look for data on other nodes. The Ring can start with just one OneBlox appliance and grow on demand with no reconfiguration and still provide optimal performance scalability.



Conclusion

With OneBlox, IT organizations have a sophisticated, yet simple, scale-out storage architecture. OneBlox appliances work together to distribute capacity and performance and customers can add hard drives or additional OneBlox as needed. Unlike traditional storage systems, OneBlox gives the flexibility modern data centers require.