Technical Brief: System Performance

Dell XC Web-scale Converged Appliance Powered by Nutanix software

Introduction

Dell XC Series Web-scale Converged Appliances integrate Nutanix web-scale software and Dell's proven storage and x86 server platform to provide enterprise-class features for virtualized environments. As a highly differentiated converged infrastructure solution, the XC Series consolidates compute and storage into a single appliance enabling application and virtualization teams to quickly and simply deploy new workloads. This solution enables data center capacity to be easily expanded — one node at a time — delivering linear and predictable scale-out with pay-as-you-grow flexibility.

The XC Appliance is designed and engineered from the ground-up to provide best-in-class reliability, and to efficiently cope with possible hardware and software failures. It includes a full version of Nutanix software, including Nutanix Distributed File system (NDFS). The distributed software architecture runs a virtual storage controller (Controller VM or CVM) on each node forming a distributed system. All nodes actively work together to aggregate individual direct-attached storage resources into a single global namespace that can be leveraged by all hosts. All storage resources are managed by the NDFS to ensure that data and system integrity is preserved in the event of node, disk or software failure.

The Dell XC Web-scale Converged appliance provides guest virtual machines (VMs) with enterprise storage features and unparalleled scalability, as well as high performance. This unprecedented combination is enabled by the NDFS, which continuously adapts to storage workload patterns to deliver the fastest possible performance to each VM — but without sacrificing capacity, cost, simplicity and feature richness.

Converged appliance

NDFS drives high performance by exposing storage resources on the same host (direct-attached storage) as the guest VMs making requests. This means that the local storage controller (one per XC appliance node) can devote its resources to handling I/O requests made by VMs running on the same physical node. Other controllers running in the cluster are then free to serve I/O requests made by their own local guest VMs. This converged architecture contrasts with traditional storage arrays that have remote storage controllers and resources located across a network (such as SAN or NAS).

The Nutanix architecture has several important performance benefits. First, because storage resources are local, each request does not traverse the network, which drastically decreases latency as it eliminates the physical network from the I/O path. Additionally, the fact that each VM host (the XC appliance) has its own virtual storage controller (Controller VM or CVM), eliminates storage bottlenecks common in shared storage architectures. As new XC appliances are added to the cluster, storage controllers are added at the same rate, providing predictable and scalable linear performance. The scale-out architecture allows for high storage performance, as well as consistent storage performance.

Effective data tiering

NDFS monitors and fingerprints storage access patterns, and treats various data types differently to optimize performance for each guest VM. Frequently accessed (hot) data and random I/O is kept in the fastest storage medium: high-speed memory or flash-based SSD. Less frequently accessed (cold) data and sequential I/O is moved to higher-capacity HDD, all while keeping data fully redundant and protected from failure.

Random data is written to a dedicated area on the local SSD tier called the Oplog. The Oplog is an SSD-based write cache built to handle I/O bursts. It stores data persistently, and quickly responds to guest VMs to deliver both low latency and high performance.

Simultaneous to being written to the local Oplog, data is also sent across the network to the CVM on another node in the system where it is persisted to that node's Oplog. After the data has been stored on two different nodes in the system, the successful write is acknowledged to the guest VM. The result is that any write acknowledged to the guest VM is guaranteed to have been persisted to disk and, therefore, can be subsequently retrieved even if there is a power outage, SSD or node failure just milliseconds after the write had been acknowledged.

Once data is persisted to the Oplog and has been acknowledged to the guest VM, it is coalesced and sequentially drained into the Extent Store. The Extent Store is made up of extents, which are variable-sized contiguous regions of a vDisk (an NDFS file) spanning both the SSD and HDD tiers. The Oplog is continuously drained to maintain space for subsequent incoming writes and to maintain the highest level of performance.

Sequential writes, however, skip the Oplog entirely and go directly to the Extent Store. This is because sequential data is continuous and can be efficiently written to disk in large blocks without performance impact. Additionally, fewer metadata updates are required for each byte of sequential data that is written, thus less time is spent performing metadata updates.

NDFS can also be configured such that sequential data bypasses the SSD tier altogether and is written directly to HDD. High performance is maintained as HDDs already write data sequentially and NDFS can leverage the large number of hard disk drives (spindles). Writing sequential data directly to HDD also reduces the total amount of data stored on the SSD tier, preserving SSD capacity for random data and extending the lifetime of flash-based SSD storage.

Data in the Extent Store will remain on the highest performance 'hot' tier as long as it is being accessed frequently. If data access patterns diminish, however, it is then marked for migration to the high capacity 'cold' tier of storage. The assessment that data has become 'cold' is performed by a system component called Curator, which as its name aptly indicates, curates the file system by performing background tasks to keep the cluster running smoothly.

Each Controller VM has an in-memory read cache called the Extent Cache. Highly accessed data is placed in this local read cache to allow requests to be served directly from memory, thus driving very low latency fetches.

Curator leverages a series of Map Reduce algorithms to efficiently scan the metadata in a distributed manner by analyzing different portions of metadata on each node in the cluster. It manages many file system operations, including data tiering, disk rebalancing, defragmentation, repairing data redundancy after either a disk or node failure, and much more. Curator runs once per hour and whenever a critical event occurs. Examples of critical events include disk or node failure, the SSD tier filling up beyond a given threshold, and a disk (HDD or SSD) filling up beyond a given threshold relative to other disks.

Conversely, data can also be promoted from the HDD tier to the high-speed SSD tier when data access frequency increases. Data is promoted to the SSD tier if it is accessed three times within 20 minutes, and if there is between 10 seconds and 10 minutes separating all three read requests. This allows for data to be migrated to the SSD tier without unnecessarily promoting data that is simply accessed for very brief periods of time.

All disks on a particular tier are leveraged by the local Controller VM for I/O. This distribution of writes fully utilizes the bandwidth available for all disks, for both reads and writes. It also provides 'RAID-0 like' level performance, but without the downside risk of an entire node becoming unavailable if a single disk failure occurs. Similarly all Controller VMs in the cluster participate in Oplog and Extent Store replication.

Data locality

A common characteristic of a virtual machine cluster is the fact that VMs will migrate from host to host within a cluster, throughout the day and over time in order to optimize CPU and memory resources. Because NDFS serves data locally to guest VMs using direct-attached storage, it is necessary for the VM's data to follow when it moves between hosts.

In a traditional shared storage environment data is accessed over the network so a VM's data stays in the same place (on the central array) even if the VM migrates throughout the cluster. Because of the distributed and scalable nature of the Nutanix architecture, however, NDFS keeps data as close to the VM as possible for the fastest performance, and to minimize both crosstalk and network utilization.

After a VM has completed the vMotion, the Controller VM on the destination host will take ownership of the migrated VM's files (vDisks on NDFS), and will then begin to serve all I/O requests for these vDisks. Accordingly, writes will also be written to the local Controller VM on local storage to ensure that write performance remains as fast as before the VM migration event.

Read requests for newly written data will be served locally, and previously written data will be forwarded to the source host's Controller VM. In the background, Curator will dynamically move the VM's remote data to the local XC appliance so that all read I/O is performed locally, and does not traverse the network.

High-performance snapshots and clones

Traditional VMware snapshots can degrade performance and are not typically recommended for use in production environments. Performance degradation occurs because the hypervisor has little to no knowledge about the back-end storage medium. Consequentially, it has to use existing storage constructs — VMDKs, which are essentially files on disk. When a VMware snapshot is taken, the VMDK is marked read-only and is essentially locked in place. Subsequent writes will go to another file called a "delta" file. When subsequent snapshots are taken, the delta file is treated like the original VMDK and is marked read-only. An additional delta file is created for each incoming write.

The performance challenge is that writes are made to delta files in the form of a change log, which records every change made to a file since the snapshot was taken. This means that reads need to access not only the most recent delta file to which the block of data was written, but also every change made to the original data. More snapshots results in more changed data, which in turn imposes a substantial performance penalty.

Nutanix snapshots are designed for production-level data protection, allowing for the benefits of 'VMware like' snapshotting, but without the associated performance penalties. Nutanix tracks data using a distributed metadata database, which was designed with efficient snapshots as a key requirement. Nutanix utilizes a redirect-on-write algorithm that dramatically improves system efficiency when performing snapshots.

First, rather than writing data to a change log, writes are allocated a new block. No lookup of the existing data is required when writing. Secondly, the system intelligently handles the way snapshot trees are tracked in metadata in order to optimize performance and capacity, while simultaneously minimizing system overhead. When any of the blocks inside of the group are written to by the child, the child vDisk gets a copy of the parent metadata for a group of extents (Extent Group). This essentially eliminates any overhead as the snapshot chain grows because each clone has its own copy.

Additionally, the metadata system allows for multiple vDisks to be requested simultaneously in a single request. This greatly minimizes metadata lookup overhead for blocks that have not yet been inherited (since they have not been written to yet). Once the snapshot tree expands beyond five branches the tree is 'cut,' which involves copying the remaining parent metadata down to child vDisks. This is important for data like OS files, which tend to only be read but not written to. In other words, their metadata would not otherwise be inherited.

Closely related to snapshots is the concept of a clone, which is essentially a writable snapshot. NDFS uses the same underlying mechanism for performing cloning as it does for snapshots, so it benefits from the same metadata optimizations. The main difference is that when a clone is taken, both the clone and the original become children of the snapshot, meaning the parent snapshot will now have two writable children.

Learn More at Dell.com/XCconverged.

TB_XC_Appliance_System_Performance_040915