VSM Whitepaper

Table of Contents

Introduction

Versity Storage Manager (VSM) is currently managing over one exabyte of data globally in some of the world's most demanding storage environments. Versity initially released VSM in 2013 after porting the Sun Microsystems' open-source SAM-QFS project to Linux. VSM is suitable for environments with up to 48 tape drives, 750 million files, and aggregate throughput of one petabyte per day.

VSM interfaces with existing enterprise business applications and backup solutions through a standard POSIX file system interface, making it easy to expose low-cost archival storage resources to end users. Rich and flexible archive policies optimize the flow of data both in and out of cloud storage services and archival storage devices such as tape libraries, disk arrays, and on-premises object storage systems.

VSM's ability to broker data across a diverse mix of archival storage systems enables customers to optimize performance, reliability, and cost factors for different data types.

VSM is engineered for maximum utilization of storage and networking hardware. Our customers use VSM to read and write Petabytes of archival data per day. With advanced technology, friendly business practices, and responsive customer support, Versity is an archiving solution you can love!

VSM Key Benefits

- **OHSM Compatible**
  VSM and OHSM share the same original code base -Sun's SAM-QFS. This allows VSM to "restore" OHSM metadata dump files and read OHMS archival data. Converting from OHSM to VSM is fully supported, and easy to accomplish with zero data migration and minimal training.

- **Lowest TCO**
  VSM allows large storage sites to operate their own infrastructure with cloud scale efficiency. Versity sites are able to obtain TCO levels up to 10x lower than Amazon Glacier while delivering unlimited amounts of data at no incremental cost over high-speed local area networks.

- **Hybrid Cloud**
  VSM provides enterprise class archiving performance natively to both traditional on premises tape systems and private or public cloud systems. VSM does not rely on object to tape gateways which limit performance, functionality, and utilization. VSM's POSIX front end

interface allows customers to deploy object storage systems without changing their applications.

- **Open-Source Format**
  VSM stores all customer data in the open-source GNU tar file format so that archive content is always available with or without vendor software. This practice eliminates vendor lock-in and provides an extra layer data availability assurance.

- **Transparent Interface**
  VSM presents itself as a POSIX filesystem or can be used directly for object interactions. There is no need to change legacy or new codes to use the system.

- **Automated Orchestration**
  VSM employs rich policies to automatically move data to the desired media, and automatically manages cache space so there's no need to worry about it.

Background

VSM was commercially launched in 2013 and has been established as a leading large scale archive management tool supporting sustained throughput of up to 12GB/s (1 petabyte per day) under real world conditions. VSM has been deployed globally in nearly every significant archive industry vertical and across sites ranging in size from Fortune 10 to small research sites. The technology behind VSM was based on Sun Microsystem's open-source SAM-QFS product. Versity ported SAM-QFS to Linux, assembled a team of leading archive storage experts, released dozens of incremental improvements including adding support for object storage and has been rapidly evolving and advancing the product for over five years.

How VSM Works

VSM includes a SAN file system that runs on nodes with access to shared block devices. A single VSM node primarily consists of two components, the **Intelligent Cache™**, and the **Archive Engine™**.

The **Archive Engine™** efficiently packages and groups archive data. Files are grouped intelligently by user, group, path name pattern, size, creation/modification time, access time, or any combination of these so that like data and data destined for a common target are packed together. Files are containerized into the GNU TAR format, both for its open data format and for read/write efficiency. This format has distinct benefits for small files, which are grouped and then sent to archive, thereby shaping traffic into workloads that are optimal for obtaining maximum throughput of the target archive storage devices and media. Upon retrieval, the system recalls files individually without retrieving the entire tar file. Large files are divided into chunks then grouped into data sets that are efficient for streaming tape writes. This capability allows VSM to stripe large files across an arbitrary number of drives simultaneously. Unlike other products that stripe across tape drives, VSM allows files to be read back with a different number of drives than were used during the write so that administrators can control drive resources based upon workloads.

Archive resources can include any combination of tape, private cloud, public cloud, and disk. The archive resources are specified by configurable policies. For instance, a customer could specify a configuration that saved copy 1 to a tape library, copy 2 to a different tape library, copy 3 to AWS, and copy 4 to a private cloud system at a remote site. Storage resources may be reserved or prioritized, making things like placing certain types of data on separate media within the same archive possible. For example, it is possible to ensure that multiple copies within the same tape library are always written to different pieces of physical media.

All data including objects are presented as standard POSIX files, so there is no need to modify or re-write existing enterprise applications in order to take advantage of private cloud or public cloud storage resources. Versity's file packaging process minimizes the overhead associated with object storage servers and enables parallel uploading for extreme file to object performance. For example, a current VSM reference customer utilizing private cloud hardware sustains full saturation of 2 10G interfaces. For applications that utilize object directly it is possible to get/put objects directly to and from VSM using the S3 protocol.

The **Intelligent Cache™** stores files and indexes them. Archive metadata is stored directly in the filesystem. Metadata stored in this way is a major advantage because there is no external

database that can get out of sync or lag behind the file system. Metadata remains online for search, browsing, and use by applications. Metadata operations such as an 'ls -l' do not cause tape mounts. Both metadata and data are check summed to ensure end to end data correctness.

Cache space on shared block devices is managed between high and low thresholds that are specified by the site administrator. VSM automatically manages space to balance new incoming files, archiving activities, and data retrieval. Archived files are optionally released (removed) from the cache only when the high threshold is reached so that files remain accessible on the fastest storage whenever possible. Storage administrators may specify that files with certain attributes such as path, owner, size, type, etc. should always remain in the cache for fast access.

When data is ready to be used by an application or user, it enters a process called staging. Staging activity is fully automated and transparent to applications, although there is visibility into the stage queue for the storage administrator if needed. Staging order is set by policy and may be configured to favor the copy that is most readily available. This is usually the copy on the fastest media, but this depends on site specifics like connection speeds and charges for access. Like archive workloads, stage workloads are sorted and optimized to enable efficient media handling and maximum throughput. Stage resources may be managed to ensure optimal system availability. For example, the number of tape mounts or drives utilized by a specific user or by a specific set of files may be limited.

To help identify candidate data for archiving and to help with ingesting files from various enterprise file systems, Versity provides the **Archive Fabric Module™** (AFM). AFM is a tool that will analyze external filesystems and help identify good candidates for archiving, as well as copy or move bulk data into the VSM system.

VSM runs on commodity hardware. The following components are required:

- Shared storage device for **Intelligent Cache™** (any low latency SAN device)
- 1 or more SSD LUN(s) for metadata
- 1 or more LUN(s) for cache data
- Metadata size = 1GB per 1 million files, 4GB per directory
- Fiber Channel or IB network for SAN
- Commodity server nodes for VSM
- Tape, disk, private cloud or public cloud infrastructure

VSM is delivered to customers as an rpm. The customer executes the binary and may use the installation guide to step through the installation and configuration process. Installation typically takes anywhere from one to four hours, provided that the hardware and operating systems have been installed and prepared for VSM. On site professional services are available for deployment, configuration, and administration if desired.

Versity executives and software engineers are available via a dedicated Slack channel, email, or phone to help with installation, configuration, support and other questions.

Roadmap

VSM is compatible with Verity's next generation mass storage platform, ScoutAM. ScoutAM is a free upgrade for existing VSM subscribers. Please see the ScoutAM Whitepaper for detailed information on the features and capabilities of Versity's newest product.

Conclusion

VSM enables organizations to use commodity hardware to manage large data collections resulting in the lowest total cost of ownership while maximizing performance and eliminating challenges associated with archiving and managing large volumes of data.