BEYOND THE 2-TB SCSI LOGICAL UNIT

Suri Brahmaroutu, Software Architect/Strategist

Today's major operating systems, as well as storage virtualization software from vendors such as Veritas and EMC, allow users to create large storage pools comprising multiple storage devices. The size of these pools (also known as volumes, logical units, or virtual disks) is restricted to 2 terabytes (TB) each due to limitations in the Small Computer System Interface (SCSI) protocol. Furthermore, the prevailing disk partitioning scheme under the *x*86 architecture restricts the size of a partition to 2 TB.

Technologies such as Microsoft® Dynamic Disks and EMC® MetaLUNs allow aggregation of multiple logical units into one composite logical unit that can be greater than 2 TB. Thus, it is possible to create and format a single file system or database that is larger than 2 TB. However, there is a performance penalty associated with these composite logical units due to the additional layers of storage virtualization required.

Demand is expected to emerge in the next few years for logical units that are larger than 2 TB natively. A solution is required that enables logical unit and partition sizes to scale beyond 2 TB and into the petabyte disk capacities expected in the future.

New U.S. government archiving and document retention regulations may require affected companies to invest in high-capacity storage systems. Additionally, the increasing deployment of servers equipped with 64-bit processors may fuel demand for larger-capacity storage systems because these servers can address and process larger datasets.

Initially, these larger logical units are likely to appear in high-performance computing (HPCs) environments that deal with data sets larger than 2 TB. Examples are bioinformatics, petrochemical and meteorology applications, and web farms that support music or multimedia download sites. In addition, many storage management applications such as backup applications benefit from the ability to address data as a single logical unit, rather than multiple logical units. Growing single-disk capacities will also drive the need for native support for logical units greater than 2 TB. Soon, 500-GB (or 0.5-TB) ATA drives will be available. If historical capacity increases continue, single drives with more than 2 TB of capacity could be available by the end of the decade. For all practical purposes, these higher single-drive capacities will require that the 2-TB logical unit and partition limit be extended.

This white paper discusses the 2-TB logical unit limitation and the changes to the SCSI protocol and *x*86 disk partitioning scheme that are under way. We identify the associated changes that are required to the operating systems, disk class drivers, and Redundant Array of Independent Disks (RAID) firmware. We organize our discussion around two fundamental operations involved in creating and using SCSI storage systems: creating a logical unit and performing I/O operations to it.

This is not the first time the industry has had to address disk capacity limits. The original design specification for the ATA interface defined 28-bit addressing, which limited the capacity of an ATA hard disk to 137 GB. This limitation forced users to partition larger hard disks into multiple logical units. In 2001, a 48-bit addressing method was specified in the ATA/ATAPI-6 revision, thus extending support for ATA drives with capacities of up to 144 petabytes.

Creating SCSI Logical Units

Storage virtualization software is used to create a SCSI logical unit, which is a large pool of storage that spans multiple physical drives. Like a physical disk drive, the logical unit is typically partitioned and formatted before it can be used. A partition is contiguous storage space on a physical or logical disk that functions as though it were a physically separate disk. Disk partitioning tools create various data structures and write them to the disk when it is partitioned. The format of these structures varies depending on the computer architecture involved. Computers based on the Intel® *x*86 architecture require disks to be partitioned using the Master Boot

Record (MBR) format. In contrast, computers based on the Intel Itanium® architecture require disks to be partitioned using the Globally Unique Identifier (GUID) Partition Table (GPT) format. These operating system-agnostic data structures describe the characteristics, including size, of the partitions and their constituent sectors on the disk.

MBR Limits Disk Partition Size to 2 TB

Within the MBR data structure, a partition table contains information describing how the disk is partitioned. Each entry in the table lists the cylinder/head/sector (CHS) location for each partition, as well as the partition type information. Four-byte fields in the partition table restrict the total number of sectors in the logical unit to $2^{32} - 1$. Thus, under the *x*86 architecture, the maximum size of a partition using standard 512-byte sectors is 2 TB. This partition size limitation must be addressed so that applications can take effective advantage of the large logical units enabled by changes to the SCSI Block Command (SBC) set discussed in the next section.

The industry is addressing the limitation in the MBR partition table by adopting the GPT format used by the Intel Itanium architecture. GPT identifies and defines partitions by their logical block addresses (LBAs), rather than by CHS location. CHS addressing is inconvenient for operating system disk drivers that access disk storage directly because it requires that the drivers be aware of disk drive geometry. With LBAs, disk blocks are numbered in a logical sequence. Disk-drive firmware converts between LBA and CHS equivalents. The GPT format has 8-byte (or 64-bit) LBAs, which can accommodate partition sizes well beyond the current 2-TB limit associated with the MBR format. In fact, a 64-bit LBA addressing scheme can accommodate partitions of up to 16 exabytes.

Another way to extend virtual disk capacity beyond 2 TB is to increase the standard sector size from 512 bytes. However, disk sector size is primarily controlled by hard-drive vendors and there is currently little momentum in the industry to implement a larger sector size.

To implement the GPT, operating system vendors must provide GPT disk partitioning support and tools to migrate a disk from MBR to GPT. Linux tools are currently available. Microsoft has not yet announced its plans. The additional changes required to support 64-bit LBAs are discussed in the following section.

Performing I/O Operations

SCSI is one of the primary interfaces used to connect storage devices to a computer. The interface definition is maintained by the T10 Technical Committee comprising representatives from major computer, hard-drive, and component vendors. The SCSI interface has evolved into several successive SCSI standard versions since 1986. The current SBC definition standardized in 1997 uses 4-byte-long LBA addressing, which limits the addressable size of a logical unit to 2 TB. A revised standard called SBC-2, due to be a published in early 2005, extends the LBA addressing to 8 bytes to accommodate logical unit sizes larger than 2 TB.

SCSI I/O

Most I/O requests begin when an application opens a file handle and calls an I/O routine. The routine is ordinarily supplied by a language library (such as C++) or an environment subsystem such as the Win32® application program interface (API). The subsystem (or language library, if linked with the program) calls a native I/O system service call. The I/O Manager accepts the service call and associated file handle, creates an I/O request packet (IRP), and delivers it to the file system. The file system exercises great control over the I/O operation at that point and eventually forwards the request to the volume manager. The volume manager creates a new IRP and sends it to the disk driver. Up to this point, none of the processes deal at the logical unit level. Beginning with the disk driver stack, however, the I/O processes must deal at the logical unit level and hence are potentially subjected to changes to support larger than 2-TB logical units.

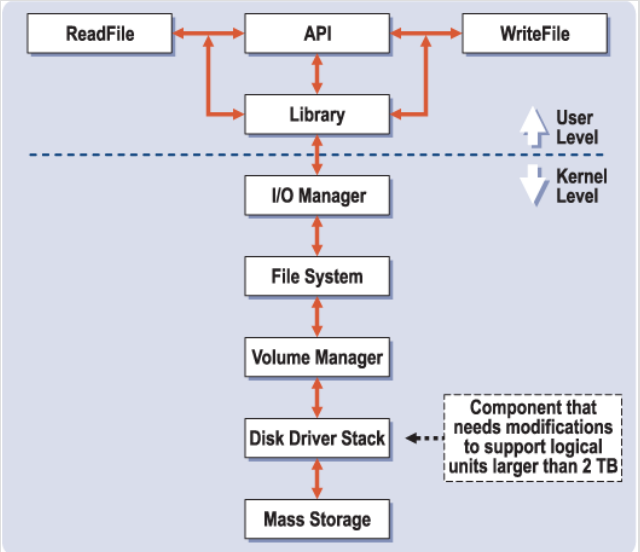

Figure 1. SCSI I/O Flow

Implementing the revised specification will require modifying several software components that handle I/O requests. Figure 1 in the sidebar, "SCSI I/O," depicts the high-level components—from application to physical storage array—involved when reading or writing to a SCSI logical unit. The details of the logical unit partitioning scheme and changes associated with the revised SBC set are hidden (or abstracted) from user applications and most of the operating system. For this reason, it is likely that the changes will directly impact only the disk driver stack and the RAID controller firmware.[1]

Disk Driver Stack

The current SBC standard defines a 10-byte Command Descriptor Block (CDB), a data structure used to communicate SCSI commands. The CDB includes a 4-byte LBA field. The SBC-2 draft standard defines a new 16-byte CDB definition that increases the LBA field to 8 bytes. The disk driver stack must be upgraded to support the SCSI commands that use the new 16-byte CDBs.

Figure 2 is a high-level view of the Microsoft disk driver model, which is composed of disk class, port, and miniport drivers. The Linux driver model is similar, with block I/O, middle-, and lower-layer drivers.

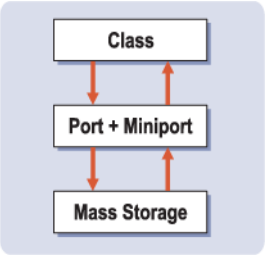

Figure 2. Microsoft Disk Driver Model

- **Disk Class Driver** — Microsoft and Linux vendors provide class (or block I/O) drivers that manage specific types (or classes) of devices such as disk drives. The main role of the disk class driver is to construct a SCSI CDB when an I/O request packet (IRP) is received. The class driver packages the CDB into a SCSI Request Block (SRB) and passes it to the SCSI port driver. Therefore, the class driver must be modified because it builds the CDB.
- **SCSI Port and Miniport Drivers** — Port drivers are typically provided by operating system vendors. Miniport drivers are provided by hardware vendors. The SCSI port driver programs the disk controller and sets certain path- and target-specific values in the SRB. The SRB is then forwarded to the miniport driver. The SCSI port driver does not alter the contents of the CDB. For this reason, it is unlikely that port drivers must be updated. It is also unlikely that changes to miniport drivers will be required.

Microsoft and Linux Operating System Support

It can be concluded from the previous discussions that operating system vendor support is key to the industry transition. These vendors must provide GPT disk partitioning support and tools to migrate a disk from MBR to GPT, as well as updated drivers. Linux disk partitioning and migration tools are already available and Microsoft is expected to provide such tools in the future. Similarly, the Linux version 2.6.*x* block layer, which is equivalent to the Microsoft class driver, supports 64-bit LBAs. (This functionality has not yet been tested thoroughly by the Linux community.) Microsoft has not yet announced its plans concerning modified disk class drivers.

RAID Firmware

RAID controller firmware must also be updated to reflect the changes to the SCSI standard. The RAID firmware processes each CDB received from the upper-layer driver. The firmware may break a CDB into several CDBs before passing them to the respective physical disk drives. RAID vendors are updating firmware to appropriately handle CDBs with 8-byte LBAs. Next-generation products that support larger than 2-TB logical units are expected from various RAID vendors in 2005.

File Systems

Large volumes of data sets typically reside in file systems, rather than databases. For this reason, file-system capacity is important in any discussion of very large (that is, petabyte) storage systems. The Microsoft Windows NT® file system (NTFS) architecture is designed to accommodate up to 2^{64} bytes, or 16 exabytes, of file

1. The volume management service should not require changes to support a 64-bit LBA. However, some volume manager vendors may have chosen to hard code LBA limits into their applications. These applications must be rewritten to enable logical unit sizes larger than 2 TB.

storage. Linux extended file systems (EXT) 2 and EXT 3 can accommodate 8 and 16 TB, respectively, per file system in 32-bit processor environments. In addition, several UNIX®-based file systems, including XFS, ReiserFS, and IBM's JFS have higher limits. Thus, in theory, it is possible to implement a very large file system with a single namespace over a given storage capacity. No additional changes are required in these file systems to support logical units of greater than 2 TB.

Support in Legacy BIOS for Bootable LUNs Larger Than 2 TB

Currently, the BIOS and operating system boot loader code assumes an MBR-style partition table. This code must also be modified to accommodate the GPT. In addition, the drivers that service BIOS INT 13 calls must be modified. INT 13 calls are used by the operating system and applications for I/O access to hard disks before disk controller drivers are loaded. To support bootable LUNs that are larger than 2 TB, the drivers that service INT 13 calls must be equipped to handle 64-bit LBA addressing. Because there is little momentum in the industry for bootable partitions that are larger than 2 TB, Dell has no plans to incorporate GPT support into its BIOS.

CONCLUSION

With upcoming changes to the *x*86 partitioning scheme, SCSI disk drivers, and SCSI RAID firmware, logical unit configurations that are greater than 2 TB will become feasible. In fact, single-petabyte configurations will be feasible, although logical units must be a manageable size so that complete datasets can be backed up and restored efficiently. These practical constraints are likely to dictate a maximum logical unit size that is less than a petabyte.

The changes discussed in this white paper require a coordinated effort among operating system and server and storage hardware vendors. In addition, software vendors developing large databases and file systems, as well as system vendors such as Dell must ensure that new driver stacks function properly by participating in industry interoperability "plugfests" and investing in extensive validation efforts to ensure backward compatibility. Complete end-to-end solutions that support 2-TB and larger logical units are expected to appear beginning mid-2005.

For More Information

- *Virtual Storage Redefined - Technologies and Applications for Storage Virtualization*, Paul Massiglia and Frank Bunn, VERITAS Software Corporation, April 2003.
- SCSI Block Commands - 2 (SBC-2) Working Draft: **www.t10.org/ftp/t10/drafts/sbc2/sbc2r16.pdf.**