Paper 487-2013

SAS® and the New Virtual Storage Systems

Tony Brown and Margaret Crevar, SAS Institute Inc.

ABSTRACT

Storage providers are offering a wave of new and advanced storage subsystems. These offerings promise virtualized, thin-provisioned, tiered, and intelligent storage that is easy to manage and will reduce costs. For many random and mixed workload applications, the promises deliver well and with good performance. Unfortunately, the SAS® I/O workload profiles tend to violate some of the primary design assumptions underlying the configuration of these new systems. This paper addresses the SAS workload-specific issues that need to be considered when configuring these new storage systems for expected performance.

INTRODUCTION

It is no secret that the cost of owning a large storage system has been increasing, although the budget of owning one has been decreasing. In addition to the initial raw hardware cost, there are costs for energy to operate it, data center space to house it, personnel for administration and maintenance, continuous monitoring, and system change and expansion to balance and manage exploding data usage. New drive media and virtualization technologies create opportunities to maximize storage resource utilization and to reduce setup and management costs.

For well-known mixed workloads, this can work well. For ad hoc high-demand I/O workloads that have service-level expectations attached to them, it can be precarious. In this paper, we discuss SAS workload characteristics so that we know what we are trying to provision for. We discuss the goals and means of virtualization, thin provisioning, and storage tiers. And, we discuss their implications for SAS workloads.

SAS WORKLOAD CHARACTERISTICS

The SAS System includes solutions and applications for data management, quality, and analyses. These solutions and applications support SAS reporting and analysis, analytics and advanced statistical modeling, data mining, OLAP, business intelligence, and large data computing. Via the SAS data set structure, the SAS® Scalable Performance Data Server® data structure, and native access engines for most third-party databases on almost every operating system (OS), SAS covers the gamut of data stores. SAS can be used on everything from a single personal PC or enterprise server to a grid, as a thick or thin, virtual or dedicated, operation. Offering vertical solutions in practically every industry such as Financial and Insurance, Retail, Manufacturing, Customer Management, Business Performance, and many more, SAS has a broad business reach. Underneath it all sits the server, storage, and network resources required to house, create, deliver, and protect these powerful systems.

Underlying performance depends on a properly provisioned I/O subsystem. What is important to know about SAS workloads? Why and how are they different from other IT applications and database workload profiles? What should be considered when provisioning the IT systems to support them? We discuss SAS workload characteristics in this section to illustrate the need for proper I/O subsystem architecture, provisioning, and tuning.

We begin with a few facts about how SAS performs I/O.

- SAS performs all I/O through synchronous OS file system calls (for example, OPEN, READ, WRITE, LSEEK).
- SAS does not open files with the O_DEFER, O_SYNC, or O_DIRECT flags.
- SAS performs block I/O in units of BUFSIZE, which is a SAS configuration parameter associated with a SAS data set at creation time.
- SAS supports both random and sequential data access. Sequential access is the dominant access pattern.
- By default, SAS manages reading and writing via an operating system API that uses the operating system's file cache. It does not do direct I/O by default, but it can be configured to.
- For each SAS process, a temporary work space is created as a uniquely named directory located in the file system specified by the –WORK configuration parameter. This storage work space is referred to as SASWORK.
- FSYNCs are not issued for files in SASWORK. Files do not persist if the processes terminate normally.
- A SAS data set is created by appending it to the SAS data set file.
- SASWORK usually exhibits close to a 50/50 distribution of reads to writes. Sequential access is the dominant access mode.

LARGE-BLOCK SEQUENTIAL PREDOMINANCE

The dominant access pattern for most SAS processes is large-block sequential I/O. This is true of data management and exploitation. The type of reporting, analysis, and statistical modeling performed by SAS customers typically involves large data stores accessed sequentially. Given the data sizes of the stores (row lengths and widths), heavy sequential I/O is performed. Operations are favored by underlying I/O subsystems that are optimized for large-block retrieval of large data stores. This is often diametric to many business applications that rely on random access of smaller data units from database structures that are optimized for that.

There are SAS operations that invoke small-block random processing. They are discussed in the next section with considerations for each discussed in the General SAS I/O Considerations section. For general I/O subsystem planning and deployment, the large-block sequential pattern is usually dominant.

RANDOM WORKLOAD COMPONENTS

There are at least three typical scenarios in which a SAS process might exhibit a random I/O characteristic. The first scenario is traversing heavily indexed files for the random retrieval of records against B-tree or hybrid bitmap indexes. The second scenario is OLAP-structure traversals with the SAS® OLAP Server. In the third scenario, there can be data set traversal, retrieval, and update techniques that rely on POINT= type processing, where the OBS pointer is randomly bounced around the data set to get specific records for retrieval or update.

If these activities represent a large component of the total workload, you should consider segregating them to separate physical file systems that are optimized for random access. In these instances, the larger SATA drives that provide high numbers of IOPs are recommended for disk provisioning.

GENERAL SAS I/O CONSIDERATIONS

There are a number of general SAS considerations for I/O architecture. They have been published in numerous papers noted in the “References” section. A short synopsis follows:

- It is typically better to use a larger number of fast and small disks versus a smaller number of large disks to aggregate I/O throughput for most of SAS (large-block, sequential I/O patterns). Smaller, faster spinning disks leave less unused space and provide the throughput aggregation necessary to drive high megabytes per second I/O.
- Refer to the notes above concerning Random Workload Components
- See Figures 1 and 2 below for a comparison of drive media types, including their general IOPs and throughput ratings. Note these are the drive manufacturer maxium performance and actual performance characteristcs under load will be significantly different. For example, a 15K rpm SAS drive will nominally yield 20 - 25 Megabytes/second (MB/s) throughput under heavy sustained workload, which is far below the drive’s maximum rating of 198 MB/s in Figure 2.
- If possible, do not share dedicated physical disks with applications that are not from SAS. Unfortunately, with widely striped or everything-striped systems, this usually isn’t practical or possible.
- Use read ahead and release behind when possible.
- Use dynamic multi-pathing and host bus adapter balancing to spread I/O workloads evenly.
- Match per-file system throughput rates to a minimum of 75 to 100 MB per second per core.
- Try to use a 64 KB or higher multiple-stripe unit over as many physical disks as practical in RAID 5.
- For systems that use predominantly large files (for example, greater than several gigabytes in size), it is helpful to enforce a 64 KB or higher SAS BUFSIZE option (matching the storage transfer size for sequential optimization). For systems that use predominantly small files, this isn’t necessary. The SAS BUFSIZE option can use its default value (which is determined by an internal algorithm that varies BUFSIZE based on data set rows that will fill a 16 K internal buffer page). The default value will vary for each data set and is usually optimal for a large number of small files used by the SAS engine.
- For SAS workloads with very large non-shared files, consider using direct I/O (DIO). See http://support.sas.com/resources/papers/proceedings09/327-2009.pdf.
- DIO helps prevent the server’s operating system file cache from becoming the I/O bottleneck when a large number of SAS processes sequentially scan multi-gigabyte tables.

Drive Media General Comparisons

Drive Type	RPM/SIZE	MB/sec Sustained XFER Rate
SATA	7200/ 2 TB	134
SATA	7200/ 4 TB	181
SAS	15K/ 300 - 600 GB	198
SSD SLC	100 - 400 GB	536
SSD MLC	200 - 400 GB	495

FASTER

Figure 1. Chart of General Drive Media Comparisons

The chart above shows general drive performance by media type and size. Note the significant increased MB/sec throughput with faster spnning disk drives, and the dramatic jump provided by solid state drives (SSD). Note the drive ratings vs. actual results above in General SAS I/O Considerations. In addition experience has shown SSD read rates far outperform SSD write rates, which are getting better with newer model flash drives.

EMC® – Media - Average Response Times

Drive Type	Spin Rate	Average Response Time
HDD	7200	~12 ms
HDD	10K	~9 ms
HDD	15K	~ 6 ms
FLASH	NA	~1 ms

FASTER

**EMC Symmetrix VMAX – VP, FAST VP
EMC Central Region – Applications Solutions Practice

Figure 2. EMC Chart of Average Media Response Times

The chart above provided by EMC shows the rapidly reducing drive response time as the drive spin rate increases, and the significant decrease when FLASH is used. Very low response times drive better random I/O performance.

SASWORK-SPECIFIC CONSIDERATIONS

SASWORK is typically the hardest worked file system in a SAS installation. It is a common, temporary working space shared by users, all of which might be producing large data demand in an unpredictable way with an unpredictable schedule, typically with a 50/50 read and write ratio.

By default, a utility directory (UTILLOC) is embedded in the SASWORK file system. In UTILLOC, utility work files associated with threaded operations are housed. The UTILLOC directory, alone, can produce significant throughput demand. The data manipulation characteristics that are performed typically involve heavy sequential I/O. Because data multiplication can also occur (for example, a SAS SORT makes three copies of the data set that is sorted during its operation), the SASWORK file system must support heavy I/O with high throughput.

Some recommendations for the SASWORK file system follow:

- Do not share SASWORK physical disks with other SAS file systems' spindles when possible.
- For logging file systems, create file systems for separate log files.
- If too many SAS processes share the same file system for SASWORK, file system metadata updates and write loads can become performance issues.
- Pay careful attention to the size, relative work rate, and number of files opened in a SASWORK file system. Some customers are attempting to build multi-terabyte SASWORK file systems, which puts undue pressure on the metadata, logging, and file system management. Multiple smaller SASWORK file systems are preferred if they can provide adequate capacity for the process size.
- Typically, SASWORK is striped with RAID 5 and higher levels of parity are not needed for scratch workspace.
- The amount of disk space needed in SASWORK is a function of the application logic (input data size) and coding efficiency.
- Carefully monitor the file systems used from SASWORK for full-disk issues, which terminate SAS processes.
- Create multiple SASWORK file systems based on traffic, utilization, workload, and spindle contention.
- Separate theUTILLOC directory from SASWORK when possible to alleviate file system pressure.

Now that SAS I/O characteristics have been discussed, we can turn our attention to new storage arrays. How are storage arrays different from previous lines of storage? How should changes be managed to best support SAS I/O loads?

STORAGE ARRAY GROWTH

Before the newest storage technology was available, most storage systems consisted of file systems placed across LUNs, which were striped across most or all of the RAID ranks in an array to attain the size, capacity, and spindle-throughput required by large application loads like SAS. The disk resources beneath these file systems were thickly provisioned, which meant that they were physically pre-allocated, fixed in size, and fully dedicated. Even now, the benefits of such an arrangement are clear—sufficient space with safety and growth overhead can be provisioned, and sufficient spindle-throughput dedicated to provide predictable I/O performance. However, the downside to thick provisioning is that if a file system needs a high-water mark of 600 GB for normal space usage (to accommodate the busiest processing periods) for example, usually 800 to 100 GB would be thickly allocated to it to provide space plus safety and growth room on a 24-hour-a-day, 7-day-a-week basis. Safety and growth room is paramount because growing a physically allocated file system is difficult. It often requires completely rebuilding the physical system with serious outage and risk. And, as storage arrays grow larger, the costs of disk resources and ongoing management become very expensive. This type of storage provisioning is referred to as "thick provisioning." (See the example in Figure 3.) This often leaves a lot of under-utilized storage.

Because of SAS large-block I/O orientation (which requires high megabytes-per-second throughput), many physical drives were involved to aggregate the throughput per spindle. Depending on the actual size of the data (capacity), if large disks were used (for example, 300 GB or larger), more disks were required to garnish spindle throughput than were actually needed for capacity. This created even more disk space that was not used. If the busiest periods of file system usage were only a small fraction of the total hours in a week, there could be considerable disk space that is largely unused (for example, necessary to be there, just not filled up) on a weekly basis.

This unused disk space catches the eye of storage administrators who are pressured to cut expenses, maximize resources, and forestall future disk purchases required for growth by other applications on the array. As data and processing rapidly grow, storage arrays grow larger in size. With thick provisioning, the raw number and size of disk devices increase substantially. Architecting and managing them becomes very time-consuming and increasingly difficult. Management tasks related to extending LUNs, balancing usage, and providing safety (for example, backup and recovery) become more difficult for already overloaded storage administrators and often require unwelcome outages. As a result, companies sought ways to reduce or eliminate the costs associated with these older storage technologies and provisioning methods.

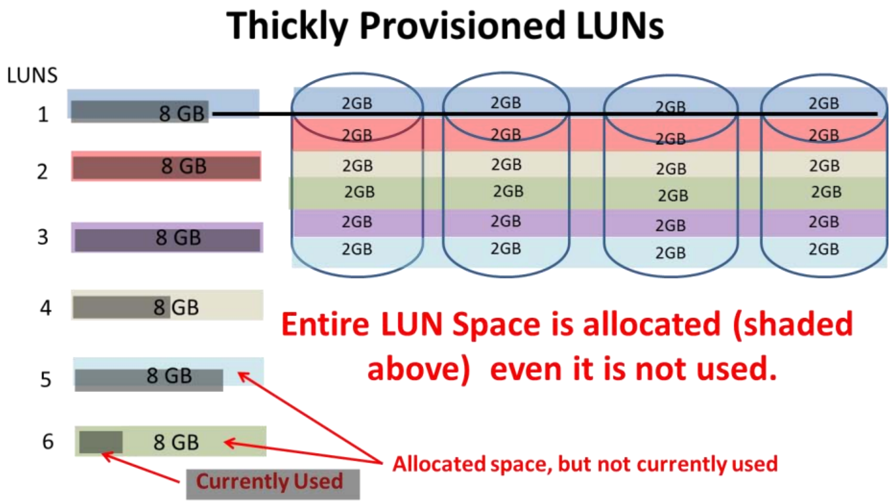

Figure 3. Thick Storage Provisioning

The darkly shaded space represents storage that has been written to. The lightly shaded portions of the LUNs represent space that has not been written to and is empty. With safety overhead and the number of spindles required for throughput, the lightly shaded space often represents more than is physically used by the application.

MODERN ARRAY TECHNOLOGIES

The newest line of storage arrays use virtualization technologies that enable thin provisioning and storage tiering across drive technologies. Storage vendors have crafted offerings that provide automation of many administration and management tasks, faster setup and maintenance with little or no outage, and less physical expense. Using faster drive technologies, such as the fast-spinning fiber channel or SAS drives or new solid-state storage, file system performance can be boosted, sometimes very significantly.

VIRTUALIZATION, THIN PROVISIONING, AND STORAGE TIERING

One of the main selling points of virtualization, thin provisioning, and storage tiering is cost reduction. This is accomplished in several ways:

- Storage virtualization. Modern arrays use virtual management architecture, software, and automation utilities to lower setup and management costs.
- Thin provisioning. By eliminating unused space (for example, use fewer drives to get the job done with no unused or under-used space) and having a higher utilization of drives, fewer drives are used. Drives are fully populated and sometimes over-subscribed in shared pools.
- Storage tiers. Automated storage tiering across drive types initially assigns data to the appropriately performing storage type (at a better cost). Then, you can interactively promote or demote data based on performance characteristics to only use expensive storage media when it is actually needed.

In this paper, each of these technologies is discussed. We explore what they are, how they work, and what effect they can have on SAS I/O workloads.

STORAGE VIRTUALIZATION

Storage virtualization management creates an abstraction between the user and actual physical storage by presenting a logical data space to the user, and then mapping that logical data space to actual physical locations. This is typically accomplished by mapping chunks of data residing on back-end resources (for example, physical LUNs on the storage subsystem) to front-end LUNs or logical volumes presented to a computer host. By mapping the logical data space to smaller chunks of data via a metadata mapper or algorithm that controls location mapping on the fly, the tight coupling between the logical access and physical management is removed.

The intention of this uncoupling is to permit the underlying physical array systems to move the data around and act on the small data chunks in a way that does not noticeably impact the logical usage of the data. (For example, you don’t really notice your data is being moved to faster storage as you are using it.) The entire file system is not being touched at once, but it is being operated on in small chucks in a way that interleaves with your current use of the data. The effectiveness of what can be done with these small chunks is largely dependent on their sizes. The EMC VMAX, for example, uses small 768 KB chunks, which support very flexible operations. This enables promised new benefits such as:

- Not requiring the physical data to reside in the exact same physical space (for example, same disks) all the time. Every time that you access it, it might be in a different physical space because the array management specified its location based on array management needs.
- Allowing data to be backed up, cloned, or snapped without noticeable impact to users.
- Allowing data from overloaded disks to be migrated to less busy disks while the data is still being used.
- Allowing data to be migrated to higher- or lower-performing disk devices (for example, from SATA to fiber channel to solid-state disks) based on on-the-fly usage monitoring.
- Allowing the creation of pools of data based on performance needs and drive types.
- Enabling thin provisioning of data.
- Using software-based management to:
    - Simplify the creation and management of logical and physical data resources.
    - Automatically handle many ongoing maintenance tasks.
    - Greatly simplify the expansion and reduction of file systems and volumes.

It is possible to have multiple layers of virtualization or mapping so that the output of one layer of virtualization can be used as the input for a higher layer of virtualization. A good example of this is a virtualized volume manager being fed by virtualized storage from a SAN.

THIN PROVISIONING

One of the primary things virtualization enables is thin provisioning (or the logical appearance of having more resources than are physically provided). Remember that in thick provisioning, storage resources have to be physically pre-allocated and available at all times with safety space, whether it is used at a high percentage or not. The benefits of virtualization remove some of the management obstacles that tie arrays to thick provisioning—the ability to easily move resources on the fly, the ability to easily grow data pools without interruption, and the automated management of logical and physical data.

Thin provisioning allows disk resources to be driven closer to 100% usage with less system administrator involvement. Because thin-provisioned storage can be instantly grown within available pools, less physical storage can be initially pre-allocated to a logical view. Your file system might be logically defined as having 500 GB of storage, but, at any given time, it might only comprise (on the physical disk) a little more than it actually is storing in capacity (for example, 200 GB). The difference between the 500 GB defined and the 200 GB actually used is not physically allocated. It is available in the pool for other file systems to use. This eliminates unused space by not having an overdedication of what was actually populated. Less storage can ostensibly be purchased because no unused space exists. Electricity and management costs are lower as are disk acquisition costs. Figure 4 shows a very simplistic illustration of thinly provisioned storage.

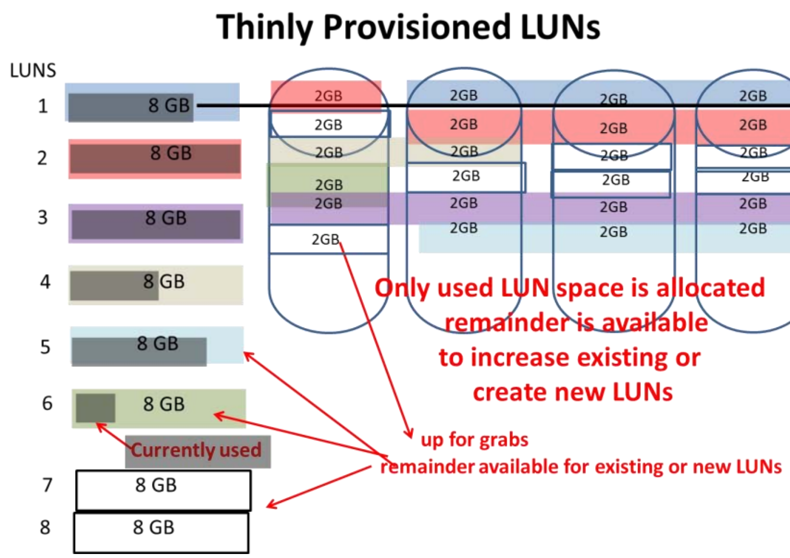

Figure 4. Thin Storage Provisioning

The unshaded portions of the LUNs represent space that has not been written to and is empty. Pre-allocation doesn't occur until the space is actually used so that space is available to the pool for other LUNs to use.

Thin provisioning easily enables over-allocation or oversubscription of actual storage space. If the goal is to save money, and not enough physical storage resources exist in the pool, when multiple logical file systems suddenly need to expand to accommodate activity, someone can be caught short. This is like a run on a bank. You might have an account with a stated amount of $10,000. But, if everyone goes to the bank at once to withdraw, there won't be enough cash on hand to give everyone their money. It physically isn't there. The bank only keeps enough cash on hand for reasonable operations. The rest of the money (for example, your money) is used to provide loans, investments, and mortgages. The thin-provisioned storage array works in much the same way. It saves money until the demand overreaches the actual resources, which is often far below the logical resources that you perceive are available. Figure 5 represents a simplistic illustration of oversubscription.

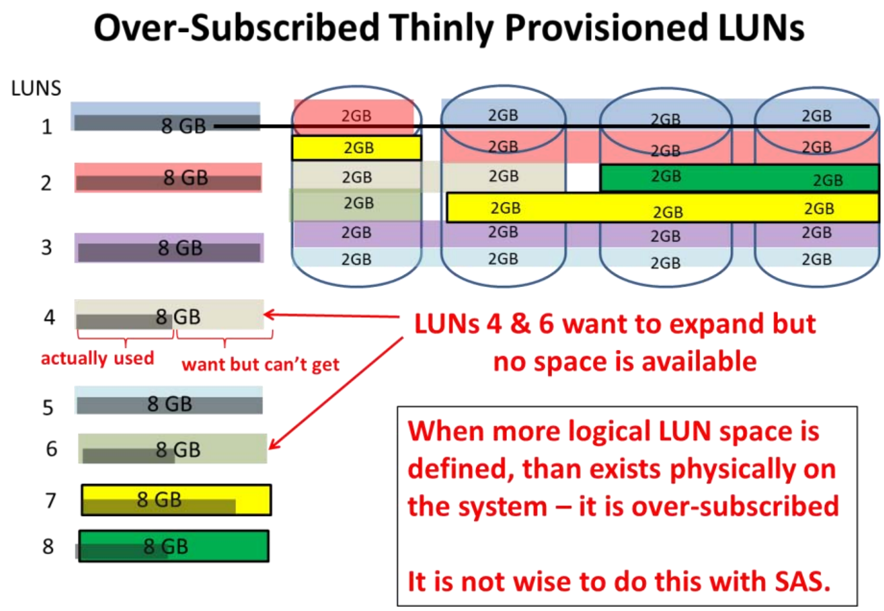

Figure 5. Oversubscribed,Thin Storage Provisioning

When Thinly Provisioned Becomes Thick

Given the goal of driving storage utilization upward to 100%, what starts out as a thinly provisioned system can, over time, resemble a thickly provisioned system. Several factors drive this—actual increased space usage, users not deleting unused or old data (insufficient cleanup), oversubscription, and inefficient space reclamation by the system.

When enough users use space to the logical limits of their LUNs, the system begins losing its reserve pool space. This can make a system resemble a thick provision as seen in Figure 6. If users don't clean up space on large file systems when data is no longer used, this can happen more quickly. In addition, the complex mechanisms that arrays use to reclaim file space deleted by the host are widely varying with varying effectiveness. Space reclamation of released space back to the thin pool can be an intricate process between the server host and the storage array. The activity of reclaiming space (with utilities that actually empty the released storage blocks so that they can be identified for reclaim) consumes array resources. This resource usage must be balanced with space needs in terms of balancing performance service levels with sufficiently available pool space. It is always a balancing act.

Continual file space management, space reclaim, and capacity planning must be done to ensure that thin pools don't become consumed and end up as pseudo-thick pools. The goal to drive higher storage utilization must never be fully reached. Reaching the goal will eventually fill up the array and cause issues. Thin pools typically buy more time than thick pools before total utilization occurs, but with a smaller investment in disk resources. Additional resources must be added as performance needs and capacity require.

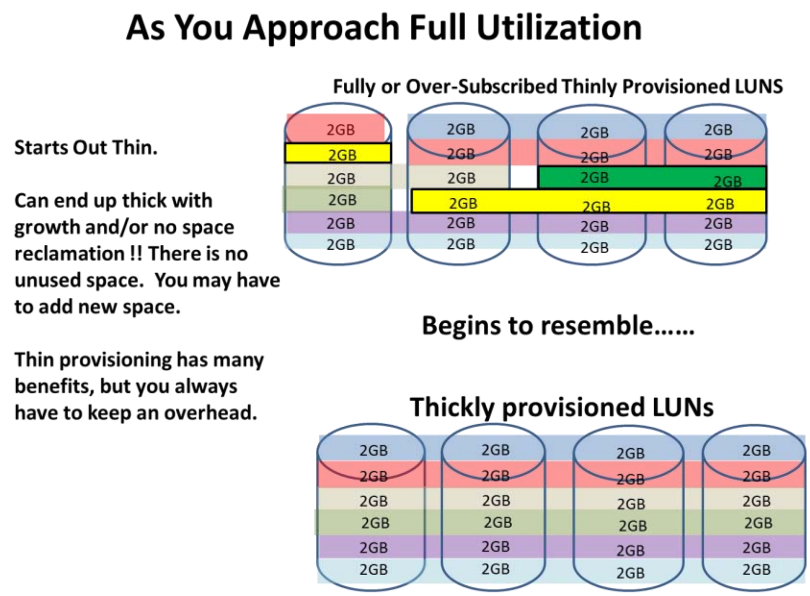

Figure 6. Provisioning over Time

STORAGE TIERS

The use of storage tiers has been ongoing. Before the proliferation of solid-state drives, storage administrators utilized storage tiers. Storage tiers previously consisted of primarily fiber channel drives (FC), Serial Attached SCSI drives (SAS), and Serial Advanced Technology Attachment drives (SATA).

Smaller, faster drives such as SAS and FC drives are preferred for SAS sequential usage. They provide a relatively good throughput (on average 20 - 25 MB per second per spindle under sustained mix-usage SAS load). These small and fast drives, when used in large quantities, yield high throughput for the file systems striped over them. The less expensive, larger, and typically slower, SATA drives are generally used for random workloads, non-critical performance, applications and backups. SATA drives striped in RAID 10 offer a cheap solution to providing IOPs performance for random workloads.

The advent of solid-state drives (SSD), also referred to as flash drives, brought significant performance improvement in SAS Read I/O operations and some improvement in SAS Write I/O operations. For more information, see http://support.sas.com/resources/papers/proceedings10/FAQforStorageConfiguration.pdf. Many SAS shops have used them in their arrays. Recent improvements in the garbage-collection activities on some SSD models have further improved the Write I/O performance for SAS and have made this higher-priced technology more attractive as a higher-performance alternative.

Modern arrays have storage tiering software that allows initial placement of data on one or more tiers of storage (SATA, FC, SAS, or SSD). The software allows automated (based on rules at setup) promotion or demotion of data on tiers based on actual performance and sustained I/O demand. Coupled with thin provisioning and automated data management, automated tier promotion offers extreme flexibility with relatively little administrative effort. The rules for promotion and demotion are tunable, which allows flexibility based on experience. SAS data tends to be large enough that the automated decision to promote to faster storage devices (for example, from FC to SSD) and the associated sub-LUN block-level migration has, in some instances, not occurred before noticeable latency. While providing benefits, automated tiering isn't perfect, and SAS workloads should be placed on the minimum tier that is acceptable for the prescribed workload's general performance.

CONSIDERATIONS TO MEET THE CHALLENGES

When you read the "General SAS I/O Considerations" and "SAS Work-Specific Considerations" sections above, it is academic to understand the potential for performance issues if file system construction, thin provisioning, and storage tier usage are not architected and managed properly. It isn't that those technologies aren't advantageous in many circumstances because they certainly are. In cases where SAS operations are housed on mixed arrays with other applications, provisioning, tuning, or architectural compromises are often made to benefit the whole. However, that might not serve SAS I/O profiles well. Considerations might need to be made depending on the throughput performance requirements of your SAS applications that alter the typical architecture, setup, and usage of thin provisioned arrays. These considerations include, but are not limited to the following:

- You should not provision performance-dependent work on larger, slower drives (for example, 500 GB or larger 10K SATA drives) unless you have employed a random access workload (e.g. SAS OLAP, random traversal of heavily indexed datasets, or SAS statistical PROCEDURES that have random I/O behavior). In the cases of a heavily random access pattern, we have found SATA drives completely acceptable.
- SATA drives are inexpensive and provide high capacity, but they lower the critical spindle count necessary to aggregate I/O throughput across a physical stripe set. This is crucial for large-block, sequential I/O patterns of most SAS usage. When these drives become full, their slow spin rates and high seek times exacerbate the throughput issue for sequential workloads. They are best suited for random SAS workloads.
- If you are using thin provisioning, avoid oversubscription of the pool spaces available to SAS file systems. Because of the ad hoc nature of SAS I/O demand, rapid extension of the space might be necessary and it needs to be immediately available on requisite storage. This is especially true for SASWORK areas.
- Plan underlying LUN placement carefully. Thin pools can be precarious for SAS workloads. You might not be able to avoid physical drive sharing in pools supporting applications that are not from SAS. If you do share drives, monitor them closely because heavy random I/O profiles can have a detrimental effect on SAS resources used by sequential SAS processes.
- If you choose to use thin pools, carefully consider pool size and construction when provisioning SAS LUNs. On many arrays, large striped volumes (or metavolumes) are needed across most or all of the physical disks in the array to garnish drive throughput, front-end and back-end adapter throughput, and balancing. In these cases, multi-pathing I/O is used.
- If your previous SAS storage system was on a large array, performance is critical. If your previous storage system was barely adequate on thickly provisioned storage, you might need to consider not using thin pools for file systems related to SAS, especially SASWORK. If thin pools are used, avoid oversubscribing.
- If you are using thin pools, work with your storage vendor to understand how zero-space reclamation will be handled in your system. This activity affects how quickly unused storage is released back to the pool and on what schedule and basis. Also, ensure that users keep file systems clean. Do not allow unused data to accumulate. We have several customers who began with thin pools, only to fill them over a short time. This created problems because they never cleaned up their file systems and ran out of pool space.
- If you are using multiple layers of virtualization in your storage assembly (for example, IBM SAN Volume Controller (SVC)) on top of an already virtualized array, ask your storage vendor team what path to implement.
- Read and consider the advice of the vendor-specific arrays listed in the "Recommended Reading" section.

If you are considering deploying on tiered storage and using automated tiering, consider the following:

- Initial placements for SASWORK data should not include low-performing tiers (for example, capacity provisioning on smaller numbers of large, slower SATA drives).

- Using FC or SAS tiers for most SAS sequential application I/O is encouraged. As with thick provisioning, ensure that enough spindles exist to provide adequate throughput performance.
- Be very careful about placing SAS in thin-provisioned pools. If your workload is large and performance dependent, you might want to avoid this altogether and use very wide-striped metavolumes instead.
- Using SSD tier storage typically works well for SAS. Write I/O performance has improved, and SSDs can be a good but expensive alternative.
- If you are considering automated tier promotion, it is generally wise to sequester SAS workloads that require high performance and service levels to faster tiers (SAS, FC, and SSD).
- You should consider less expensive SATA for random workloads, and minimally used data or batch applications that do not require high-response time or strict service-level windows.

CONCLUSION

Modern array technology has incorporated virtualization, which enables thin provisioning and offers automated storage tiering. Modern technology is intended to combat high array costs by driving a higher utilization of resources and automating much of the array administration. Unfortunately, the goal of driving higher utilization is often diametric to having sufficient throughput resources quickly available for large-block I/O applications such as SAS.

When using storage arrays, the advice in this paper should be carefully considered, and the specific vendor papers listed in the “Recommended Reading” section should be reviewed. In many instances, new technologies can be used effectively. In other cases, new tecnologies must be mitigated with appropriate architecture changes and usage.

Work with your storage vendor to ensure that you are using its storage technology most effectively and that you are providing high performance to your SAS applications. There is no substitute for careful planning and testing to ensure that adequate performance will be provided.

REFERENCES

Intellimagic. 2012. “The IntelliMagic Online Class for IBM SVC/V700 Performance Analysis” (online class). http://www.intellimagic.net/intellimagic/home.

EMC Corporation. 2009. EMC Technical Note P/N 300-009-149 REV A02: *EMC Symmetrix V-MAX Best Practices*. EMC Corporation.

Crevar, Margaret, Leigh Ihnen, Harry Seifert, Frank Bartucca, and Vishwanathan Krishnamurthy. 2011. “Storage best practices: SAS 9 with IBM System Storage and IBM Power Systems; Considerations for optimal storage layout Version 3.0”. IBM Corporation.

IBM. “IBMStorWize V7000 Version 6.4.1 Getting Started.” e-Learning module. http://pic.dhe.ibm.com/infocenter/storwize/ic/index.jsp?topic=%2Fcom.ibm.storwize.v7000.learning%2Ftbird_learning_tutorials.html.

Evans, Chris. “Enterprise Computing: Thin Provisioning and The Cookie Monster!” The Storage Architect, Storage, Virtualisation, Social Networking, November 17, 2009. http://thestoragearchitect.wordpress.com/2009/11/17/enterprise-computing-thin-provisioning-and-the-cookie-monster/.

RECOMMENDED READING

- http://support.sas.com/kb/42/197.html contains many papers related to performance monitoring, hardware architecture and tuning, and best practices for SAS I/O workloads.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Tony Brown
SAS Institute Inc.
15455 N. Dallas Parkway
Dallas, TX 75001
Work Phone: +1(214) 977-3916
Fax: +1 (214) 977-3921
tony.brown@sas.com

Margaret Crevar
SAS Institute Inc.
100 SAS Campus Dr
Cary NC 27513-8617
Work Phone: +1 (919) 531-7243
Fax: +1 919 677-4444
margaret.crevar@sas.com