Paper 416-2012

Storage 101: Understanding Storage for SAS® Applications

Bob Augustine, Hewlett-Packard, Downers Grove, Il, USA

ABSTRACT

MOST OF THE PERFORMANCE PROBLEMS WE SEE HAVE TO DO WITH I/O. MOST OF THE TIME, WHEN DEPLOYING SAS APPLICATIONS IF YOU GET THE STORAGE PERFORMANCE RIGHT YOU GET THE SAS PERFORMANCE.

THIS DISCUSSION WILL INCLUDE:
- **RULES OF THUMB FOR I/O REQUIREMENTS.**
- **SAS I/O CHARACTERISTICS.**
- **VARIOUS FILE SYSTEMS AND THEIR PERFORMANCE.**
- **HOW TO TEST FILE SYSTEM PERFORMANCE PRIOR TO LOADING SAS.**
- **HOW TO MEASURE FILE SYSTEM PERFORMANCE.**
- **UNDERSTAND SHARED FILE SYSTEM PROTOCOLS AND PERFORMANCE. EXAMPLES INCLUDE CIFS, NFS, STORNEXT, AND OTHERS.**

INTRODUCTION

SAS performance is heavily dependent on storage. The purpose of this presentation is to share experiences with customer issues, benchmarking and POCs.

We will then drill down into how to determine what type of performance your configuration will deliver.

We will then discuss how to determine how to emulate SAS I/O. While this is not a perfect science, it can be a useful tool for tuning a storage/file system combination before SAS software is loaded and configured. Because each modification is destructive in nature, it's important to be able to make each test execute with a minimum of effort and time between tests.

RULES OF THUMB FOR I/O REQUIREMENTS

This is a moving target. When I started working with SAS 3 ½ years ago, we planned for the requirement of 25MB/sec/core. At the time, there were murmurings of this number going to 50MB/sec/core. Today this number is trending higher and is typically set at 75MB/sec/core.

We have seen instances when 50MB/sec/core will be fine, but there are other more I/O intensive and less CPU intensive SAS operations that require upwards of 135MB/sec/core.

For the sake of planning purposes, let's use 75MB/sec/core. If you have a server with 2 X 2-core processors, then you should plan for 300MB/sec for the server. If you have a server with 2 X 6-core processors, likewise you should plan for 900MB/sec.

SAS I/O CHARACTERSTICS

Generally speaking all SAS I/O happens via the operating system's file system buffer cache.

SAS I/O is generally 8K chunks and it's sequential in nature. Further SAS I/O is, in general, a 50% read, 50% write mix, that can occasionally get as read intensive as 60% read and 40% write.

This compares with a typical OLTP environment being 80% read, 20% write and mostly random.

Because of this, general performance characterization of storage systems using I/O Per Second (IOPS) is not applicable to SAS I/O. We then use MB/sec or GB/sec as the I/O metric of choice.

VARIOUS FILE SYSTEMS AND THEIR PERFORMANCE

Various file system deliver various performance with SAS applications. The file systems tested on Linux include EXT3, EXT4, XFS.

We have found that EXT4 is almost twice as fast as EXT3 and XFS is about 10-15% faster than EXT4. Because of this we recommend XFS for all SAS deployments on RHEL 5.5 and later.

HOW TO TEST I/O PERFORMANCE PRIOR TO LOADING SAS

Since standard storage metrics most often do not apply to SAS, there is a need to try different configurations to determine which will yield the best performance.

The types of things that enter into performance is the number of LUNs per storage controller, the file system stripe size, whether separate file systems are to be used for SAS Work and SAS Data and determining which file system type is best for your storage and OS configuration.

Since changing any of the above parameters is destructive to the underlying data stored on the file system, we need to have a way to attempt emulating SAS I/O characteristics.

Warning: While the following is a good way of providing insight regarding how a specific tune will affect I/O, it is nowhere near a perfect science. It will get you reasonably close, but most of our experience indicates that actual SAS I/O will not mimic what you have found with the following commands. As an example, we did a test using a 32-core DL580 server. Using the below tests, we observed 2.5-2.8GB/sec. However, when we loaded and ran SAS tests, we found the I/O performance to be between 3.2 and 3.5GB/sec and in fact it went as high as 4GB/sec for extended periods of time.

The following is a script that we used to tune that I/O subsystem:

```
#!/bin/bash

dd if=/asuite/output/file1 of=/dev/null bs=1M &
dd if=/asuite/output/file2 of=/dev/null bs=1M &
dd if=/asuite/output/file3 of=/dev/null bs=1M &
dd if=/asuite/output/file4 of=/dev/null bs=1M &
dd if=/asuite/output/file5 of=/dev/null bs=1M &
dd if=/asuite/output/file6 of=/dev/null bs=1M &
dd if=/dev/zero of=/asuite/output/file7 bs=1M count=51200 &
dd if=/dev/zero of=/asuite/output/file8 bs=1M count=51200 &
dd if=/dev/zero of=/asuite/output/file9 bs=1M count=51200 &
```

HOW TO MEASURE FILE SYSTEM PERFORMANCE

There are several methods to measure file system performance.

Some arrays have software which will tell you how much I/O is being delivered to the server. When this is available, we recommend its utilization.

Whether this is available or not, we recommend watching file system performance by utilizing the iostat command. There is, however, one issue that needs to be addressed. Since Linux utilizes a multi-level I/O structure, if you utilize iostat with no parameters, you will be double or triple counting the I/Os. Because of this we use the following script, but it does need to be modified for your specific installation. If you add more LUNs, you need to expand the script.

```
#!/bin/bash

iostat -d -m dm-0 \
dm-1 \
dm-2 \
dm-3 \
dm-4 \
dm-5
```

UNDERSTAND SHARED FILE SYSTEM PROTOCOLS AND PERFORMANCE

When needing access to a single file system by multiple servers there needs to be a traffic cop in place to ensure that processes on the various servers do not step on each other. Because free space lists are kept in main memory, the allocation of free space on one server could cause that server to write over space already allocated on another server. A prime example of multiple server utilization is SAS Grid.

Customers have deployed CIFS and NFS as relatively low performing shared file systems. However, as the performance of servers have increased and also as the number of Grid nodes have increased, NFS and CIFS are performance challenged.

When HP did a test of SAS Grid in the fall of 2011, we tested the performance of StorNext against the performance of XFS. Because it is possible with SAS Grid to have each server access their own, non-shared SAS Work area, we needed to determine if StorNext was faster than XFS. What we found was that StorNext out performed XFS by 10-15% during out tests.

CONCLUSION

Once all the ramifications of good I/O performance are understood, it is easier to assemble the needed parts required to provide a well performing, cohesive I/O subsystem. After the subsystem is deployed, it is also possible to monitor that subsystem to determine if its performance is degrading over time and if so, what needs to be done about it.

CONTACT INFORMATION

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

Name: Bob Augustine
Enterprise: Hewlett-Packard
Address: 2001 Butterfield Road, Suite 700
City, State ZIP: Downers Grove, Il 60515
Work Phone: 630.561.2956
E-mail: bob.augustine@hp.com