Determining If an Application Is I/O Bound

Storage Performance Problems

Storage performance problems are the root of many issues an IT organization faces. The ability to deliver data for processing is a key component of application performance. Additional potential hardware limiters of performance such as processing power and networking, are increasing at exponential rates as Moore’s Law continues its march forward. As a result, storage performance problems are pervasive throughout organizations.

Nytro flash accelerator cards deliver accelerated performance and an overall lower cost per gigabyte PCIe flash solution. Purpose-built to meet the requirements of open source environments and software-defined storage, Nytro cards are designed to deliver leading endurance and reliability using low-cost flash optimized for low power and thermal characteristics.

With the broadest flash portfolio on the market, Nytro cards can support the performance and price requirements of a range of enterprise applications. Determining if an application can benefit from increased storage performance is the first step in determining where to deploy the Nytro solutions.

How an Application Interfaces with Storage

There are several physical properties that limit what performance is obtainable with a particular storage device. Understanding these limits is key to understanding impact at the application level of different storage options. There are three fundamental limits of a storage device:

- Bandwidth – the maximum throughput of a storage device based on the physical limitations of the channel that connects the storage to the server.
- Response time – the time required to process an I/O request, fetch the data and return it to the server.
- IOPS – A measure of the maximum concurrency of I/O that a storage device can handle.

An application interfaces with the storage by requesting I/Os to the storage devices. These can be small or large amounts of data (I/O size), random or sequential, reads or writes, and there is a limitation on the number of parallel I/O requests. Storage systems can have different limitations based on the random or sequential nature of the workload and the size of the I/O can determine whether the bandwidth or IOPS limit of the storage is the ultimate limit of the performance. Either of these limitations will show up the same way—as an increase in the response time of the storage above its best case. This is either because it is waiting on bandwidth to transmit data or it can’t handle the number of parallel requests, forcing wait time while backend resources free up.
Determining If an Application Is I/O Bound

The number of parallel I/O requests that an application submits to storage is fixed through either the coding of the application or the number of concurrent users. This sets the maximum parallel requests to the storage and is very important in determining if the storage is limiting performance or not. The queue of the application can be used with a version of Little’s Law (shown below) to determine the IOPS that are possible with a particular storage device. Simply put, a lower response time increases IOPS.

\[ \text{Queue Depth} \times \text{Response Time} = \text{IOPS} \]

**Determining If an Application Is I/O Bound**

An application is I/O bound if it spends more time waiting on I/O than other potential bottlenecks: CPU resources, the network, or number of users. I/O bottlenecks are common and there are many tools that can be used to see how much the application is waiting on storage compares to time spent waiting on other resources. If there is little or no time spent waiting on storage, faster storage will be of little benefit.

There are two easily measurable metrics that help determine if an application is I/O bound:

- **CPU use** – Faster storage accelerates an application by returning time to the processors, either directly by eliminating the time when processors are waiting for data, or indirectly by allowing users to be more productive and submit more work to a processor. If the processor use is high, there is not much time that can be returned with faster storage. Processor use >75% leaves little room for improvement with faster storage.

- **Disk queue** – The disk queue is a queue of I/O requests that are in-process or waiting to be serviced by a storage subsystem. The historical rule-of-thumb was to have a disk queue approximately equal to the number of disks in a system; otherwise the requests would get backed up waiting for disks to be available to process I/O. This rule was in place to assure that the response time of the storage subsystem would be close to the best-case response time that a disk could deliver. Now that flash storage options with lower response times are available, there is significant room for improvement. The rule-of-thumb can be revised to say that if the disk queue is greater than 1, there is potentially a storage I/O bottleneck. Anytime that this disk queue is less than 1, there is some period of time where there are no I/Os outstanding and the storage is idle. The higher the disk queue, the higher the potential performance improvement.

**Perfmon (Windows)**

The performance monitor (Perfmon) tool in Windows allows the recording of a large number of counters on a periodic basis for analysis. There are a few counters that can be used to quickly identify if the storage is a bottleneck. Collecting all of the “PhysicalDisk” counters as well as “Processor Information->% Processor Time” is enough to get a basic indication of whether the storage or the processor is the bottleneck for an application. Figure 1 shows the Processor and PhysicalDisk counters from a system with an I/O bottleneck.
In this system, the processor time is clearly not a bottleneck, and the Average Disk Queue Length, at 217, is a clear indication of an I/O bottleneck. Due to so many I/O commands waiting for their chance to be serviced the “Avg. Disk sec/ Transfer” (response time) at 264ms is dramatically higher than what a nonstressed disk system can deliver.
The same workload on a higher performing storage system is shown below in Figure 2.

The impact of storage with a lower response time has a number of immediate effects: reduced disk queuing, lower I/O response time, a dramatic increase in IOPS and bandwidth, and an increase in processor use. This final effect is the return of time that processes were idling waiting on I/O for active processing. In this case, the reason that the processor time did not increase further was that a bandwidth limitation was reached with the storage. With the lower response time delivered by faster storage, this server was able to run more than an order of magnitude faster.
Determining If an Application Is I/O Bound

IOstat (Linux)

IOstat is a very common utility on Unix systems provided by the sysstat RPM. The key characteristics of a workload can be examined to determine if there is a storage bottleneck: processor use, storage queuing, response time, and bandwidth. One of the outputs of “iostat -xk 2” is shown on an I/O bound system in Figure 3:

```
 avg-cpu: %user %nice %system %iowait %steal %idle
          0.06  0.00  0.18  7.91  0.00  91.85

 Device: rrqm/s wrqm/s r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
   sda  0.00  0.00  0.00  1.50  0.00  6.00  8.00  0.01  7.00  2.33  0.35
   sdb  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
   sdc  0.00  0.00  1345.00 609.00 10760.00 4872.00 16.00 15.95 8.19 0.51 100.00
```

In this system, the response time of the storage (the combination of wait time and service time), just under 9ms, is reasonable for a disk-based storage system. However, the low processor use and constant disk queuing makes it clear that storage is the bottleneck. This type of workload is an example of an application with a parallelism limit. This type of application can possibly lead to countless arguments within an IT department. The storage subsystem is not the performance limiter, but rather how the application is able to interact with it. The pre-SSD solution of adding additional spindles to the storage system would have no impact on performance, as the workload is not hitting the limits of the parallelism of the existing storage system. If it was, the response time would be considerably higher as it was in the Perfmon example. However, it is still possible to reduce the response time of the storage by moving to storage that has a much lower best case response time. This same workload on higher performance storage is shown in Figure 4:

```
 avg-cpu: %user %nice %system %iowait %steal %idle
          0.76  0.00  3.12  7.54  0.00  88.58

 Device: rrqm/s wrqm/s r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
   sda  0.00  0.50  0.00  3.00  0.00  14.00 9.33  0.00  1.00  0.50  0.15
   sdb  0.00  0.00 31191.5 13449.0 249536.0 107592.0 16.00 14.34 0.32 0.02 100.00
   sdc  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
```

In this system, the response time of the storage (the combination of wait time and service time), just under 9ms, is reasonable for a disk-based storage system. However, the low processor use and constant disk queuing makes it clear that storage is the bottleneck. This type of workload is an example of an application with a parallelism limit. This type of application can possibly lead to countless arguments within an IT department. The storage subsystem is not the performance limiter, but rather how the application is able to interact with it. The pre-SSD solution of adding additional spindles to the storage system would have no impact on performance, as the workload is not hitting the limits of the parallelism of the existing storage system. If it was, the response time would be considerably higher as it was in the Perfmon example. However, it is still possible to reduce the response time of the storage by moving to storage that has a much lower best case response time. This same workload on higher performance storage is shown in Figure 4.
Determining If an Application Is I/O Bound

The higher performance storage enabled a dramatic increase in performance; 20 times the data is being processed in the same time as before. In this example, the lower response time (at just over 300µs) dramatically increased the performance of this application even with its limits to I/O parallelism.

Where To Go From Here

Common operating system tools can be quickly used to determine if an application is storage I/O bound. The presence of ample idle CPU resources and constant active requests to the storage system (I/O queuing) is a clear indication of a storage bottleneck. Now that PCIe-based flash technology is available, switching to storage that has a lower response time and the ability to handle high IOPS is a straightforward proposition.