

Module 1: Performance Management Overview

Upon completion of this module, you should be able to:

- Discuss performance analysis issues
- Relate the basic performance measurements to I/O operations



This module focuses on the basic principles of performance management.

Why Analyze Performance?

- Short Term Problem Identification & Resolution
 - ▶ What is causing this slowdown?
 - ▶ What can be done to prevent it in the future?
- Better Resource Usage
 - ▶ What resources are going unused?
 - ▶ Where can I best allocate this new storage?
- Meeting Performance SLAs
 - ▶ Regular performance reports to show compliance
- Long Term Capacity Management
 - ▶ How much more can be added to present environment?
 - ▶ When will new hardware have to be added?



Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 2

There are several reasons why performance analysis is an important part of managing a complex data environment. Short term problem identification can often be performed simply by looking for abnormal data patterns. If delays are being experienced and one component is showing twice the normal I/O rate, it is probable that, that is causing the issue. Performance analysis can quickly trace this issue to its root cause.

As a medium term measure, performance analysis can be used to better allocate resources. When new drives must be mapped, new filesystems must be created, or new applications installed, a quick performance analysis of the components involved will show what drives or data channels are least used. Allocating the new traffic to the unused components helps minimize the performance impact.

Complex provider-consumer environments will often use Service Level Agreements (SLA) to define a guaranteed level of performance. Simple performance analysis reporting must then be put in place to show compliance with the SLAs.

In the long term, detailed performance analysis can help in traffic capacity management. By analyzing the current I/O patterns and predicting the maximum performance thresholds, you can predict how much longer your environment can grow. Good capacity management helps eliminate problems that make short term analysis necessary.

Capacity Management

Capacity Management =
Storage Capacity + I/O Capacity

- Capacity Planning – Storage Capacity
 - ▶ Analyzing storage allocation and utilization
 - ▶ Predicting future storage allocation needs
 - ▶ Tuning environment to avoid wasted storage
 - ▶ Allocating storage to increase capacity
- Performance Management – I/O Capacity
 - ▶ Analyzing performance measures
 - ▶ Predicting future I/O capacity needs
 - ▶ Tuning environment to improve performance
 - ▶ Adding hardware to increase performance capacity

ITIL

Technology
Information
Library

EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 3

Capacity management is an important part of IT Service Management, as defined by ITIL. Successful capacity management provides storage and processing resources when needed in a cost effective way.

In the storage arena, capacity management is usually identified with storage capacity planning. This is the provision of the storage needed for data and applications. Analysis of the current allocation and utilization must be done to detect shortages or wasted space. Past increases, and future business trends are considered to help predict future storage needs. Tuning, to recover wasted space, or allocating new storage are ways to ensure that sufficient storage is available for current and future needs.

While it is not always immediately associated with it, performance management is also an important element of capacity management. In performance management, the issue is not how much data can be stored, but how fast it can be accessed, and transferred to where it is needed. Analysis compares the current rates of data transfer to expected maximums. Tuning improves access speed using existing hardware. If tuning cannot provide the performance increase needed for current or future needs, additional hardware can be considered.

Performance Service Level Agreements

- **Performance Metrics:** Measures of Performance
- **Service Level:** Target Thresholds for Performance Metrics
- Service Level Agreements can be formally defined, or informal:
 - ▶ “The database must be able to sustain a minimum of 800 I/O per second during peak traffic conditions” (formal definition)
 - ▶ “Average response time for customer transactions on the A server must be less than 1 second” (formal definition)
 - ▶ “If the number of user complaints rises significantly, response time will have to be improved” (informal definition)
 - ▶ “The backup job used to take two hours, now it takes three or more” (informal definition)



Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 4

Service Level Agreements (SLA) can be made around performance. An SLA typically compares performance metrics to a service level. As long as the metrics meet the threshold defined by the service level, the performance SLA is met. Performance management seeks to maintain this situation.

Many organizations have well defined SLAs for performance. Thresholds for many metrics are written into contracts. Frequent measurement is done to ensure that metrics are within the specified limits.

SLAs can also be informal, unwritten expectations. A system that had been functioning at a particular speed, but has suddenly slowed down can be considered to have violated an informal service agreement. Whenever users complain of poor access speed, it can be considered a request for performance (capacity) improvement. The expectation of a certain level of service delivery is just as important as written SLA agreements in the eyes of a customer.

Styles of Performance Management: Reactive Style

- No ongoing program
- Waits for the phone to ring
- Solves problems by trial and error or by experience
- May have tools, but only turns them on when problems arise



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

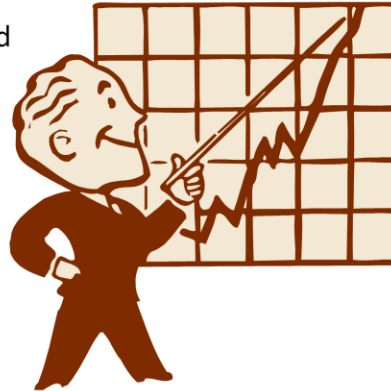
Module 1: Performance Management Overview 5

As with any business process, there are many ways to approach performance management. A **reactive** analyst simply reacts to problems as they arise, typically when a client or customer complains about poor performance. The reactive analyst then springs into action, using whatever tools and experience are available to troubleshoot the problem. Performance delays are felt by the customers until the problem is resolved.

Clearly, performance problems will be noticed by the customers frequently with this style. An impression of disorganization and poor planning will be given, especially if the same problem occurs more than once.

Styles of Performance Management: The Casually Observant

- No ongoing program
- Available real-time data
- Notices real-time trends
- Able to recognize workload patterns and trends
- Must be there to solve problems



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 6

A casually observant analyst monitors the current state on a regular basis to spot problems. Regular observation shows the limits at which the environment can operate before performance problems occur. As performance measures increase toward levels known to cause problems, the analyst can try to fix the situation. If there is sufficient lead time, the problem might be solved before poor performance is noticed by the customers.

This is an improvement over the reactive style, but it is far from perfect. Although the analyst knows from experience how to spot performance problems, he must be actively monitoring the environment to spot them. Some alerting mechanism might be put in place to monitor the thresholds in the analyst's absence, but the lead times in active IT environments are often very short. The problem might have become severe by the time the alert is acted on by the analyst.

Without recording the past performance history, the analyst has no way to identify traffic growth, except by intuition. As a business grows, data center traffic will gradually increase. Long-term analysis of the trends will show the increasing rate of performance problems over time.

Styles of Performance Management: The Actively Observant

- Ongoing data collection program in place
- Management reports include trend analysis
- Partition and allocate resources by business applications
- SLAs are enforced
- Organization is measured by quality of service



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 7

An analyst who is actively observant monitors the current and past state of the environment. Regular performance data is collected and archived, and regular reports are produced to show the growth trends. Consistency in analysis quickly shows any abnormalities.

An analyst with this level of knowledge can partition the resources to business applications based on their performance capacity. Under-utilized components can be assigned to applications that are expected to generate a lot of traffic, and over-utilized components can be omitted from consideration for additional uses.

Once performance is regularly considered and measured by business unit, Service Level Agreements (SLA) can be established and enforced. A typical SLA will be based on one of the quality of service measures discussed later in this module (typically response time). The agreement guarantees that a performance measure will remain below a chosen threshold. Continuous monitoring and reporting on this value proves or disproves SLA validity.

Styles of Performance Management: The Proactive Style

- Maintains an end-to-end view
- Uses planning tools to design for growth



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 8

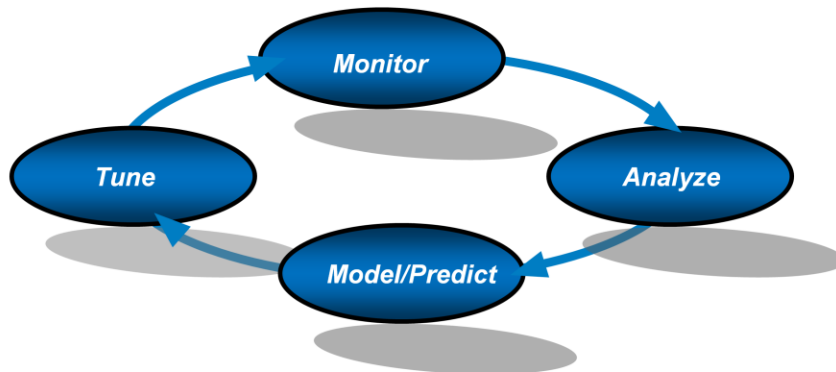
A completely proactive analyst maintains an end-to-end view of the environment. Hosts, SAN, and storage are all considered in the analysis. Several monitoring tools might be required to view performance of this varied environment.

Specialized planning tools are also used for growth modeling. Such tools can estimate the effects of additional workload, giving the analyst a way of predicting the effects of future growth.

This level of management is beyond the scope of this course. Our focus will be on Symmetrix analysis alone, using analysis tools that have no modeling capability.

Performance Lifecycle

- Basic performance monitoring to collect raw metrics
- Analyze performance issues
- Predict future capacity needs based on past performance measurements and resource demands
- Institute changes to tune the environment



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 9

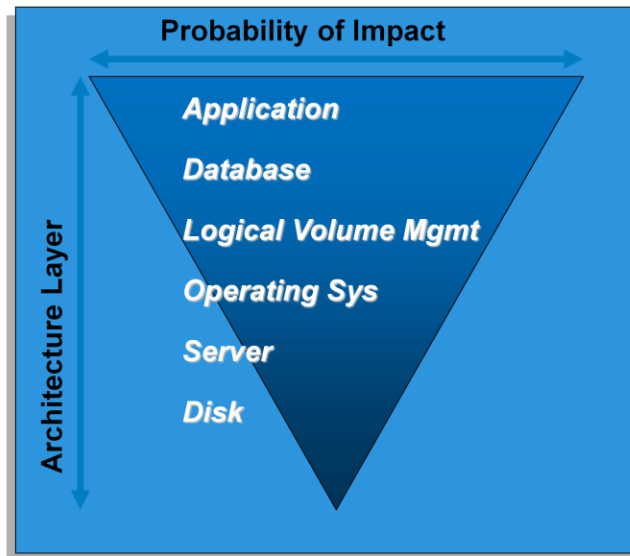
Performance management, like all service management, should be an ongoing process. Continuous improvements should be made to counter the ever increasing demands for services that every IT department faces. Anticipating performance issues in this way reduces the number of emergencies that send IT staff scrambling.

Basic performance archiving is a type of monitoring; most performance tools can monitor the environment on a continuous basis. Relevant performance metrics can be incorporated into periodic reports that show the status of the storage environment. These reports are analyzed to detect repeated trends or issues. Solutions to these performance issues can be proposed and modeled. Performance histories can be consulted to predict future I/O capacity needs, and those needs are also planned for. Finally, the necessary changes are made to tune the environment.

After a change has been made, the results must be observed, and if necessary, improved upon. The process repeats again at this point.

Performance Hierarchy

- Storage array least likely to be source of performance issue
- More opportunities to optimize performance higher in the Architecture



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 10

There are many factors that influence the speed of data access, as the figure in this slide illustrates. Data transfers through user applications, databases, volume managers, operating systems, server hardware, and finally storage systems. Any of these components can create inefficiencies and bottlenecks and, therefore, reduce performance.

The figure also roughly approximates the relative probability of impact on performance. The larger parts of the pyramid, toward the top, are more likely to contribute to performance problems. Look for changes and warnings on the top before looking at the components toward the lower level of the pyramid.

Storage is shown as the smallest of the components. Storage problems are few compared to the other sources of performance issues. Sophisticated storage arrays are designed with performance in mind. Unfortunately, advanced storage systems are often difficult to monitor and improve compared to other components, making it important to proactively manage them.

This class covers the Symmetrix performance issues only. While it may be a small part of the performance hierarchy, it is a complex storage array that requires very detailed knowledge to fully analyze.

Access Time Comparison

8 KB Transfer	Time	x 1,000,000
CPU to RAM, 166 MHz 64 bit bus	6 μ s	6 sec
HBA to FA, 100 MB per sec.	78 μ s	1 min 18 sec
Symmetrix DMX-4 Director to Cache (2.5 GHz serial)	26 μ s	26 sec
DA to Drive, 73 GB 15,000 rpm, average seek 3.6 ms, latency 2.0 ms	5,700 μ s	1 hr 35 min

Note: Although drive access is the lowest point on the performance hierarchy, inefficiencies there can cause dramatic slowdowns when compared to the other components.



Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 11

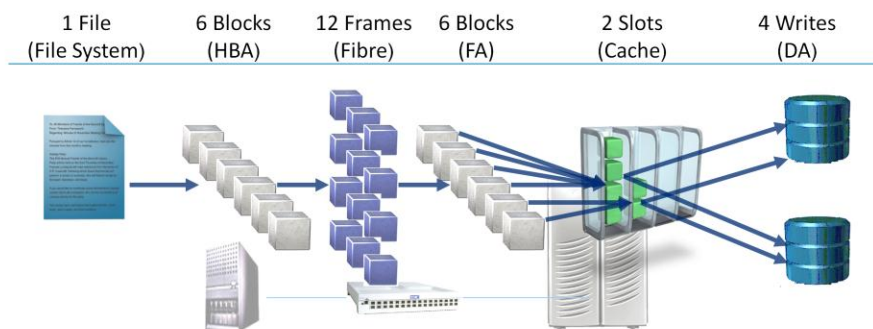
Some sample times required for data transfers are presented in this slide. To make these times more understandable, the times are also shown scaled up, so that one microsecond is equivalent to one second.

It is clear from this chart that drive access times are much slower than purely electronic access. Many of the performance issues and solutions presented here are centered on reducing the number of drive accesses.

All of the time estimates here are based solely on the numbers shown. The 78 μ s value for transferring 8 KB of data from a Host Bus Adapter (HBA) to a Fibre Channel Adapter (FA) is calculated by simply dividing 8 KB by 102,400 KB per second (the listed FA speed in KB). As we will discuss in just a few pages, a real I/O operation requires some setup and termination, which lengthens the time taken for the overall task.

What is an I/O?

- An I/O is a complete transfer, or unit of work, between end points
- I/O might be broken into smaller units, or re-combined depending on the protocol of the end points



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 12

An I/O is a unit of work, or a complete transfer, between two end points. There are many end points in a single pathway of the Enterprise storage environment. Each one uses a different protocol, which may break the received I/O into multiple I/Os before passing it on to the next end point. When considering an I/O at a particular level, always keep in mind that more than one complete transfer might be taking place at a different level to transmit that same data.

The illustration shows how a single file might be broken up during the process of storing it on a Symmetrix. The file will frequently be broken up into multiple I/Os (or blocks) when the File System Manager transfers it to the Host Bus Adapter (HBA). The HBA perceives the activity as a series of smaller work units.

The HBA will conform to Fibre Channel protocol, and transfer each I/O it receives as a series of frames of up to 2 KB to the next connectivity device. The frames are routed through the fabric to the fibre adapter (FA) on a Symmetrix, where they are reassembled into the original I/O form seen by the HBA.

The FA will transfer the data into Symmetrix cache slots. For this example, let's assume that the file data must be stored into two slots. The illustration shows each received I/O being transferred only once to cache, but if the cache is mirrored, two transfers are needed for each.

The cache slots will be treated as single units from this point; a disk director (DA) will transfer each slot to the physical drive (called "destaging") in one write operation. Depending on how the data is protected, several drive operations might be required to store it. This illustration shows RAID-1 protection; each slot is transferred to two different physical drives.

Parts of an I/O



- One I/O consists of four parts or stages:
 - ▶ **Negotiation:** Communication used by both end points to agree to and manage the operation
 - ▶ **Header:** Identifying or addressing information (Added CRC or parity bits can be considered “header” also)
 - ▶ **Data:** Actual transferred data
 - ▶ **Acknowledgement:** Final communication to terminate the action
- Negotiation, Header, and Acknowledgement components are typically fixed in size, regardless of I/O size
- A change in the I/O size typically means a change in the Data component only

EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 13

Transferring a single I/O between any points involves more than just transferring the data. In every data transfer protocol, several other tasks must be performed when sending an I/O. These other tasks add time to the overall process and increase the amount of data on the transmission channel. They can be thought of as additional parts or components of the I/O transfer process.

The components are:

Negotiation, Acknowledgement: Both end points must agree to the transfer and manage the operation. This includes any “handshaking” tasks that processors use to schedule and organize the activity. Most protocols require some “setup” negotiation to start the I/O, as well as a final “finish” message to terminate it.

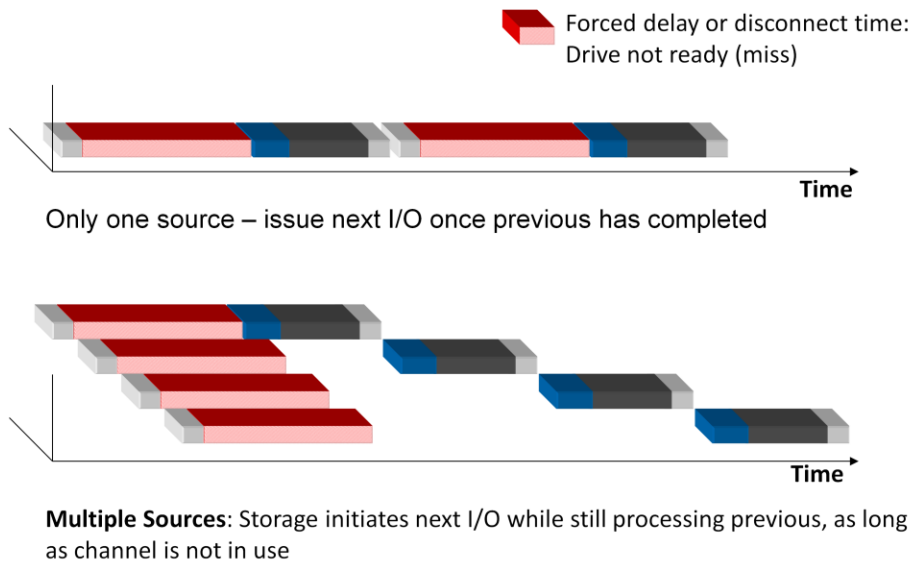
Header: Some identifier or address value that the receiving end point uses to properly utilize the data. Any CRC or parity bits used to error-check the data can be considered “header.” Also, this overhead adds to the information load on the channel.

Data: The actual data. All of the other components are added to the I/O during transfer and removed before the data can be utilized.

The negotiation, header, and acknowledgement components are largely fixed in size, regardless of I/O size. A 2KB I/O requires about the same negotiation and header information as a 32KB I/O. A change in the I/O size typically means a change in the data component only.

The illustration shows the parts of an I/O on a time graph. At the start of the I/O, some time is taken for the negotiation, then time is taken for the I/O header, etc.

Storage Array Delays in a Multi-Source Environment



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 14

The previous slide showed the parts of an I/O as a single continuous sequence of events. This is how an I/O will be processed much of the time, but in other cases, delays might cause a more complex situation. A forced delay will often be introduced by the array when the I/O cannot immediately be processed; this will typically happen in a read or write miss. While the array is preparing for the transfer (locating the data on a drive, making space available in cache, etc.), the I/O channel is unused.

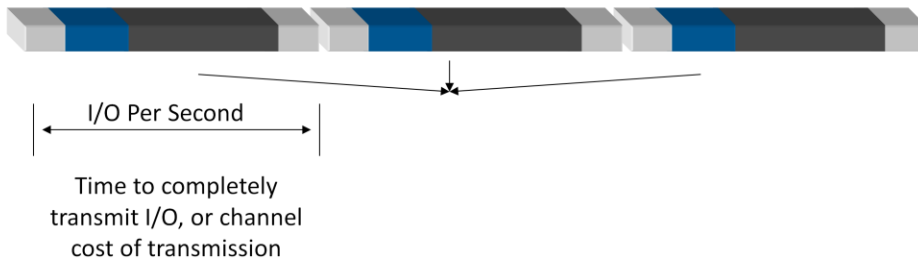
If only one host or process source is using that data channel, the delay will turn into wasted time. Each I/O will only be issued when the one before it is completed, so no alternative sequencing can occur. This is illustrated in the top diagram on this slide. Note that only two I/Os are serviced.

If multiple hosts or processes are using the same channel, the delay time can be used more efficiently. Under the SCSI protocol used for data transfer, when one source is informed of a delay, it must be disconnected from the channel until notified that the data is ready. During this time, a different source can initiate or service a different I/O. As shown in the bottom diagram on this slide, this allows the channel to be used nearly all the time, resulting in more overall I/Os per second. Note that since no more than one actual data packet can be transferred on the channel at a time, the diagram shows only one negotiation, header, data, or acknowledgement happening at once. Ultimately, this makes for a longer total response time for some of the I/Os, even as the overall system shows an improvement in data processing.

This concept is true in the SAN that connects the hosts to the array, as well as within the Symmetrix itself. Testing Symmetrix performance by sending a single-source I/O stream does not take full advantage of this multi-source optimization.

I/O Per Second

- **I/O Per Second (IOPS):** Number of transfers per second
- An important measure for small I/O, especially when large numbers must be transferred to meet service levels (e.g. database)



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

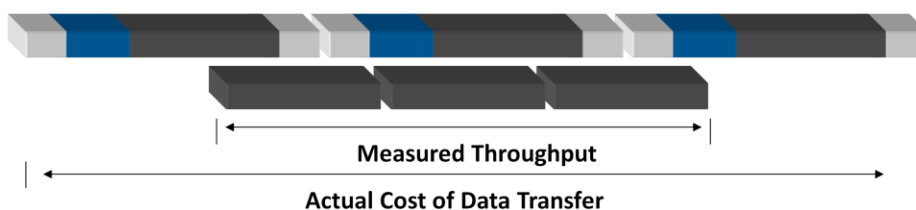
Module 1: Performance Management Overview 15

I/O per second (IOPS) measures the number of transfers per second between end points. The most common end points considered for this measurement are a host and its storage. As data is broken into blocks and transferred to and from drives, each is counted as an I/O. IOPS can also be measured at more focused points in the data transfer, such as between a caching system and the physical drives.

IOPS is a good measure when the I/O sizes are small, especially when large numbers must be transferred to meet service levels. This is often true in OLTP (On Line Transaction Processing) environments; a high IOPS measure indicates that many database transactions (typically one or more I/O each) can be serviced per second.

Throughput

- **Throughput:** Volume of Data transferred per second (negotiation and header ignored)
- An important measure for large I/O, especially when a large volume of data must be transferred in a specified time (e.g. backup)
- Frequently used as the capacity rating of channel technologies, (e.g. 2/4/8/16 Gbps Fibre)
 - ▶ This is a theoretical maximum for all traffic on the channel, including headers and negotiation



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

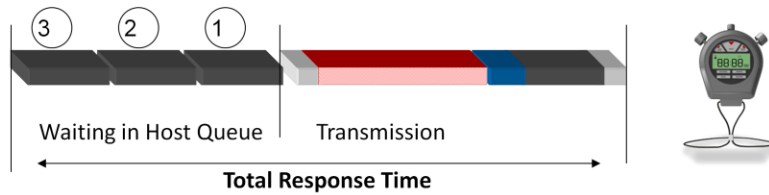
Module 1: Performance Management Overview 16

Throughput measures the volume of data transferred per second through an I/O channel. All commonly used performance measurement tools report only the data transferred when measuring throughput, since this is the “useful” part of the I/O. The header and negotiation “overhead” are ignored, even though bytes of content must be transferred across the channel to complete these tasks as well.

Throughput is a useful measure when I/O is large, especially when the time taken to transfer the overall volume of data needs to be minimized. Backup is a good example of this sort of activity. The total number of individual I/Os is not a concern, the total time to move the backup data archive is.

A “throughput” value is often used to rate the speed of a channel: 2/4/8/16 Gbps Fibre, etc. This is more accurately termed “bandwidth.” This number is a theoretical maximum for all traffic on the channel, including headers and negotiation. Since performance measurement tools report the data volume and ignore header and negotiation, measured throughput cannot reach the rated maximums, in practice. Additionally, processors at the end points will often require some calculation time between requests, further ensuring that the maximum ratings cannot be achieved under real conditions.

Response Time



- **Response Time:** The total time taken for an I/O transfer, including queuing, negotiation, header, and data
- An important measure when each transfer must be completed in a specified time to meet service levels
 - ▶ Key drive performance measure in Mainframe communities
- Because response time includes the data component, larger I/Os have longer response times than smaller I/Os under similar circumstances

EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 17

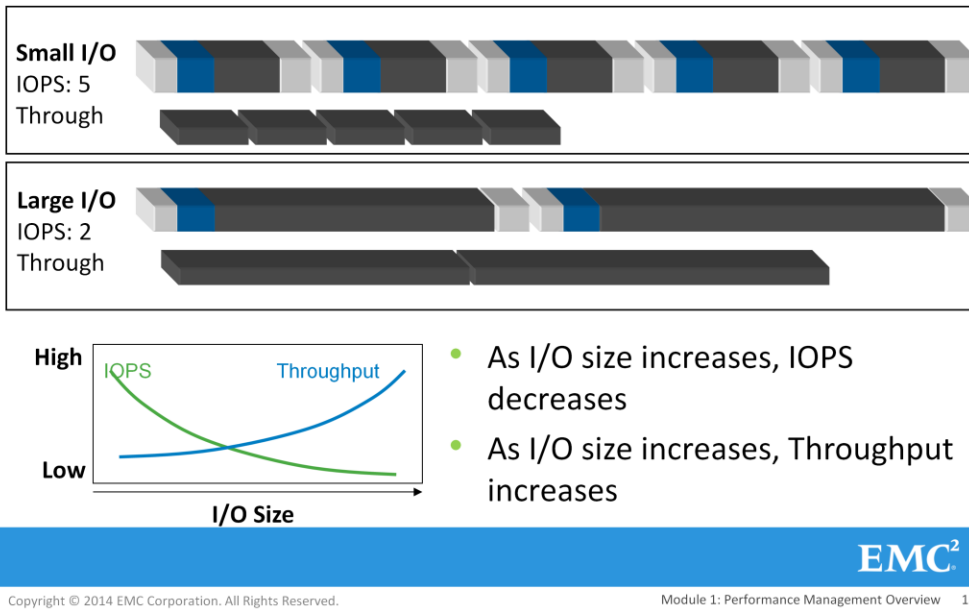
Response time measures the total time taken to completely transfer an I/O. This is typically measured from the time the filesystem or application issues the I/O request until the I/O acknowledgement occurs. Any queuing at the host and delays in the SAN or array will therefore be included here. Some response time measures might only show the time spent in one part of the path, such as the Symmetrix sampled average response time measures. In this case, queuing and delay measures will still be included if they occur strictly within the Symmetrix itself.

Much like the I/O per second measure, response time is a critical factor when each I/O must be processed quickly to meet service levels. It is also a good overall measure. A low response time implies a high IOPS and throughput since more I/O can travel through the channel per second. Response time is typically the primary measure of drive access speed in Mainframe communities.

Take care when comparing response time across systems, or even across applications. Since it naturally takes longer to transfer a larger I/O than a small one, size affects response time. Higher response time of one system does not necessarily mean poorer performance; the one system may simply have a larger I/O size.

IOPS, Throughput, and I/O Size Relationship

- IOPS, Throughput, and I/O Size are closely related



The I/O per second, throughput, and I/O size measures are closely related.

When small I/Os are transferred, less time is taken on each I/O, increasing the number that can be moved in a given time period. The IOPS measure is typically large for small I/Os. However, the data payload is small when compared to the header and negotiation factors. So, while many I/Os are being exchanged, the total data throughput is small.

When large I/Os are transferred, more time is taken on each I/O, decreasing the IOPS measure. However, a larger percentage of the channel resources are spent on data rather than header and negotiation, which increases the overall throughput. This is illustrated in the block diagrams above. The combined data parts of the large I/O example is roughly 50% larger than the small I/O example.

The bottom graph in the illustration shows this relationship. As the I/O size being transferred increases, the IOPS measure drops. If the number of transfers per second is the important measure of performance in this environment, then smaller I/O size is beneficial. Databases typically set a size for their I/O transfers that maximizes this benefit.

The graph also illustrates how throughput will increase as the I/O size increases, even as the total number of I/Os goes down. File copy and backup operations will take advantage of this relationship by increasing the I/O size.

Utilization

Clock Tick Measures:

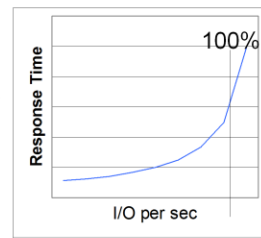
- Count busy and idle clock tick cycles of processors, disks, etc.
- Accurate measure of object utilization



||||| Busy
Vs.
||||| Idle

Expected Value Measures:

- Divide key performance indicator over max benchmarked
- Adjusted for component model and type, but not other performance factors



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

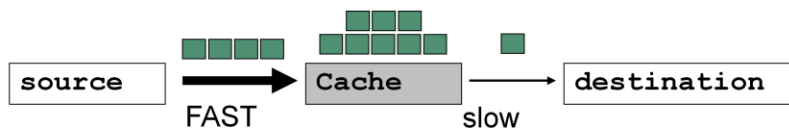
Module 1: Performance Management Overview 19

Utilization is a measure based on clock ticks. Clock tick measures are available for many components in a Symmetrix array, and are considered the most accurate way to measure utilization. Each clock tick that the processor, disk, or other object can be doing something is counted as either busy or idle depending on its activity. If the object has many idle ticks, it is under-utilized. If all of the ticks are busy on the other hand, it cannot possibly be doing more. By comparing the number of busy and idle ticks, you can express the utilization of the object.

Utilization measures in Performance Viewer are presented as % Busy and % Idle. The DA metrics also include % Non-IO Busy measures, which take into account background activity like FAST swaps, Optimizer moves, Snap copies, etc. Note that these metrics are calculated and reported differently than some of the other tools you may use. Some of the other tools and older versions of Enginuity used expected value measures and calculated the metric based on benchmarked performances.

Caching

- Cache buffers transfer between end points, often with different response times
- Gives source faster response time than destination can deliver
- Speed reverts to slower destination speed when cache runs out



EMC²

Copyright © 2014 EMC Corporation. All Rights Reserved.

Module 1: Performance Management Overview 20

Many components in a storage environment use caching to improve performance. Host applications and file system managers cache I/Os before sending them to the physical drive; a Symmetrix caches I/Os from host ports before sending them to the drive subsystem.

Cache is simply a region of memory where some number of I/Os can be stored until they are passed to the next destination. In the diagram on this slide, cache buffers the I/Os from a source to a destination (in any real environment, devices on either end will both send and receive). Frequently, the response time of the cache is much lower than the response time of the destination, so that caching the I/Os gives much better performance than just sending to the destination. Cache space is always limited, and obviously cannot be used once it fills up.

Module 1: Summary

Key points covered in this module:

- Performance analysis issues
- Key performance measure definitions:
 - ▶ I/O per second
 - ▶ Throughput
 - ▶ Response Time



This module covered basic principles of performance management.

This slide is intentionally left blank.



This slide is intentionally left blank.