

SQLskills Immersion Event

IEPTO2: Performance Tuning and Optimization

Module 4: I/O Concepts and I/O Subsystems for DBAs

Jonathan Kehayias

Jonathan@SQLskills.com



Overview

- **Defining storage terminology**
 - IOPS, MB/sec, Latency
- **Magnetic vs. Solid State Drives (SSDs)**
- **RAID configurations**
- **Storage Area Networks**
- **Configuring drives in Windows**
- **Testing with Diskspd and Iometer**

IOPS

- **Number of read or write operations performed per second**
- **Size matters**
 - Smaller block sizes will allow higher I/Os per second than large blocks
 - Watch out for vendor configuration quotes
 - Stated IOPS are typically 4KB block sizes which is rare for databases
- **Important for OLTP workloads where transactions are small and reads/writes are typically single page and highly random**
- **Monitoring in Performance Monitor**
 - Physical Disk: Disk Transfers/Sec
 - Physical Disk: Disk Writes/Sec
 - Physical Disk: Disk Reads/Sec

Throughput (Bandwidth) in MB/sec

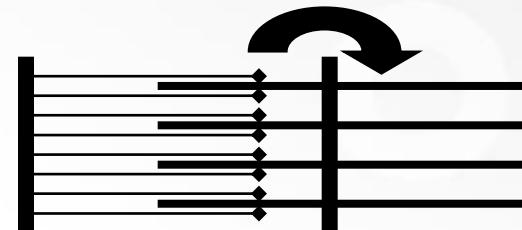
- **Measurement of the amount of data that can be transferred to or from the disk configuration for a average block size**
 - Rough approximation (number of operations) x (size of operation)
- **Size matters**
 - Larger block sizes will allow higher MB/sec at lower IOPS than small blocks
- **Important for data warehouse and other large scan implementations but generally not as important OLTP databases**
- **Very useful for testing appropriate functionality of multiple paths with SANs (covered in Module 3)**
- **Monitoring in Performance Monitor**
 - Physical Disk: Disk Bytes/Sec
 - Physical Disk: Disk Write Bytes/Sec
 - Physical Disk: Disk Read Bytes/Sec

Latency

- **How long it takes an operation to finish measured in milliseconds**
- **General SQL Server guidelines**
 - < 8ms: excellent
 - < 12ms: good
 - < 20ms: fair
 - > 20ms: poor
- **Reality is “It Depends”**
 - Block size matters, larger block I/O's take longer to complete
 - Should be workload and SLA-based to meet business requirements
 - Log writes should be as fast as possible (i.e. minimal latency)
- **Monitoring in Performance Monitor**
 - Physical Disk: Avg. Disk Sec/Transfer
 - Physical Disk: Avg. Disk Sec/Read
 - Physical Disk: Avg. Disk Sec/Write

Traditional HDD Platters

- Drives have multiple platters with read/write heads above and below each platter
 - The read/write heads float on a cushion of air created by the rotational speed of the disk platters known as an air bearing
 - “Like a 747 flying 500mph at 1/4 inch above the ground” – Jim Gray
- Multiple heads still have a fixed articulation point which prevents parallel I/O operations on a single disk
- Mechanical limitations affect performance



Traditional HDD

Performance and Latency

- The latency or response time in milliseconds it takes for the drive to begin to transfer data
- Response time = seek time + rotational latency
 - Seek time: the time it takes the read/write head to move from the current track to another track on the hard drive.
 - Rotational latency: the time it takes to rotate the disk platter to the correct position for access by the read/write head

HDD Spindle [RPM]	Average rotational latency [ms]
7,200	4.17
10,000	3.00
15,000	2.00

Solid State Drives (SSDs)

- SSDs have no moving mechanical components
- Data is stored in NAND based flash memory that is persistent with bit level addressing for hard disk emulation
- Access time depends electrical connections to solid state memory, making it very fast and consistent across cells
- General average seek time ranges between 0.08ms and 0.16ms
- NAND types
 - SLC: Single-Level Cell flash stores a single bit value with two states
 - Lower voltages results in longer endurance of 100,000 write cycles
 - MLC: Multi-Level Cell flash stores a double bit value with four states
 - Higher data density per cell at the trade off of higher voltages with reduced cell endurance of 10,000 write cycles
 - TLC: Triple-Level Cell flash stores a triple bit value with eight states
 - Higher data density per cell with reduced cell endurance of 5,000 write cycles
 - 3D MLC/TLC: Layers NAND for higher storage densities (up to 32TB in 2.5")

SSD Concepts (1)

- **Overprovisioning**
 - Reserves a permanent or temporary portion of the SSD capacity as working space for the controller
 - Improves performance and endurance of the NAND cells
- **Wear Leveling**
 - Distributes writes as evenly as possible across the cells to increase endurance and reduce a single cell from wearing out under heavy writes
 - May increase write amplification as static data is moved from a underwritten cell to a heavily written cells
- **Write Amplification**
 - Garbage Collection – data is written in pages but only erased in blocks
 - Used pages within a block are read and rewritten to another block to allow reclamation of the stale pages by erasing the current block

SSD Concepts (2)

▪ Flash Translation Layer

- Provides logical block mapping and garbage collection capabilities
- May be prone to “*flying writes*” due to firmware or controller bugs leading to database corruption on write
 - Customer database with GPT (guid partition database) data would show up using DBCC PAGE while investigating corruption reported by DBCC CHECKDB
 - First part of the page was all zeros followed by the GPT information as outlined for the GPT in MSDN

▪ OS Error 665 and Defragmentation

- OS file fragmentation can affect SQL Server databases stored on SSDs due to NTFS file attribute list limitations
 - Small autogrowths over time can fragment the file hitting the NTFS limitation
 - SQL Server 2014 can run on newer Refs file system without this limit
- <http://bit.ly/ssd-defrag>

RAID

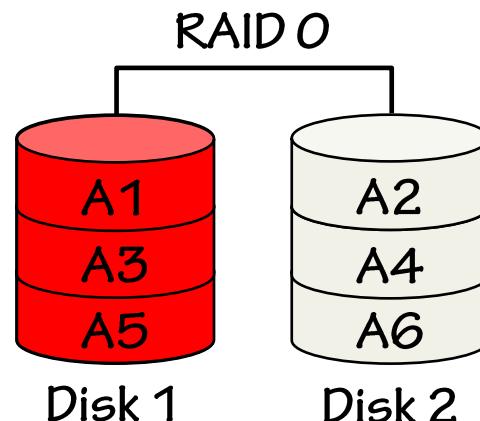
Redundant Array of Independent/Inexpensive Disks

- Combines multiple disk drive components into a single logical unit, distributing data across the disks for redundancy and improved performance
- Configurations or levels are named by the word RAID and a numeric value specific to the configuration for standardization
- Common RAID levels
 - RAID 0 - Striping
 - RAID 1 – Mirroring
 - RAID 5 – Striping with parity
 - RAID 6 – Striping with double-parity

RAID Levels: RAID 0

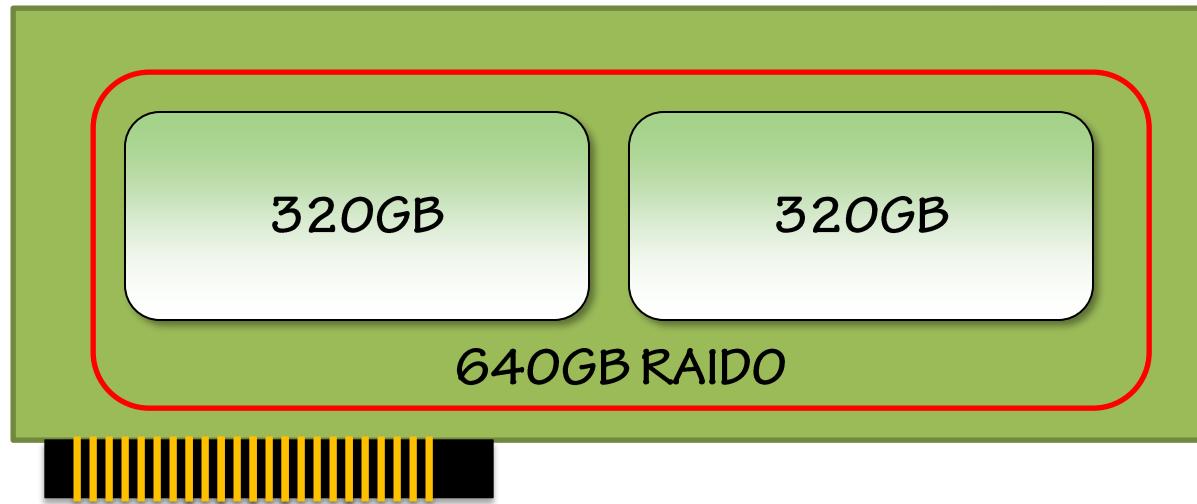
Striping

- **Usage scenarios:**
 - Data you don't care about
 - Performance matters more than data loss risk
- **Pros:**
 - Fast read and write performance
- **Cons:**
 - Total loss of data from single disk failure
- **Watch out for this with PCI-Express SSDs and IaaS implementations**



PCI-Express SSDs

- Some PCI-Express SSDs like the FusionIO ioDrive Duo 640 present two physical disks to Windows

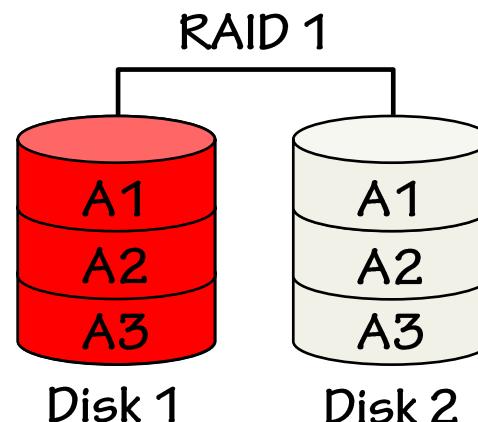


- To get full capacity from the card requires Windows RAID 0

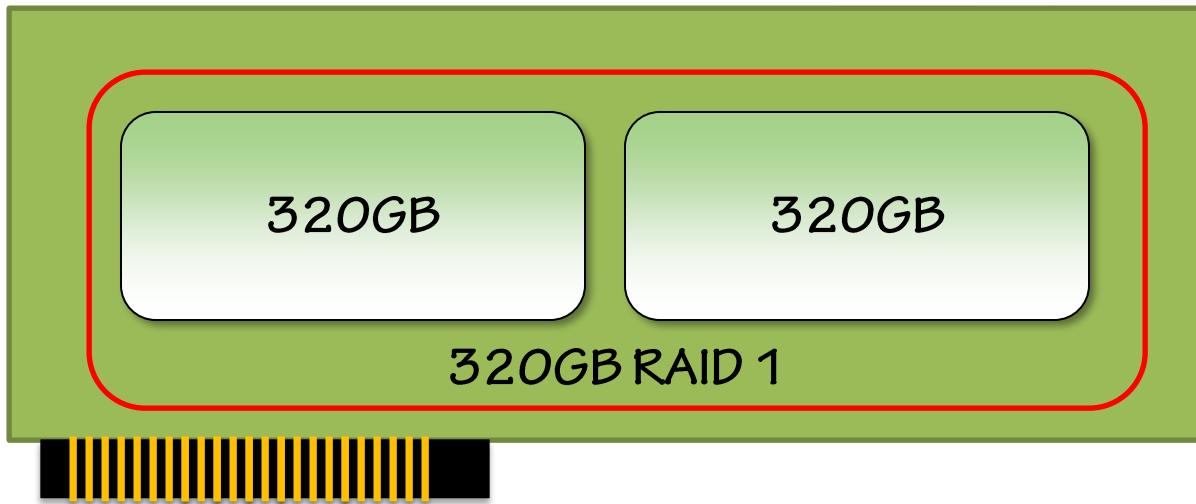
RAID Levels: RAID 1

Mirroring

- **Usage scenarios:**
 - Important data that requires redundancy to protect from disk failures
- **Pros:**
 - Mirrored storage protects from single disk loss
- **Cons:**
 - Only provides half the storage
 - Write performance equal to one disk

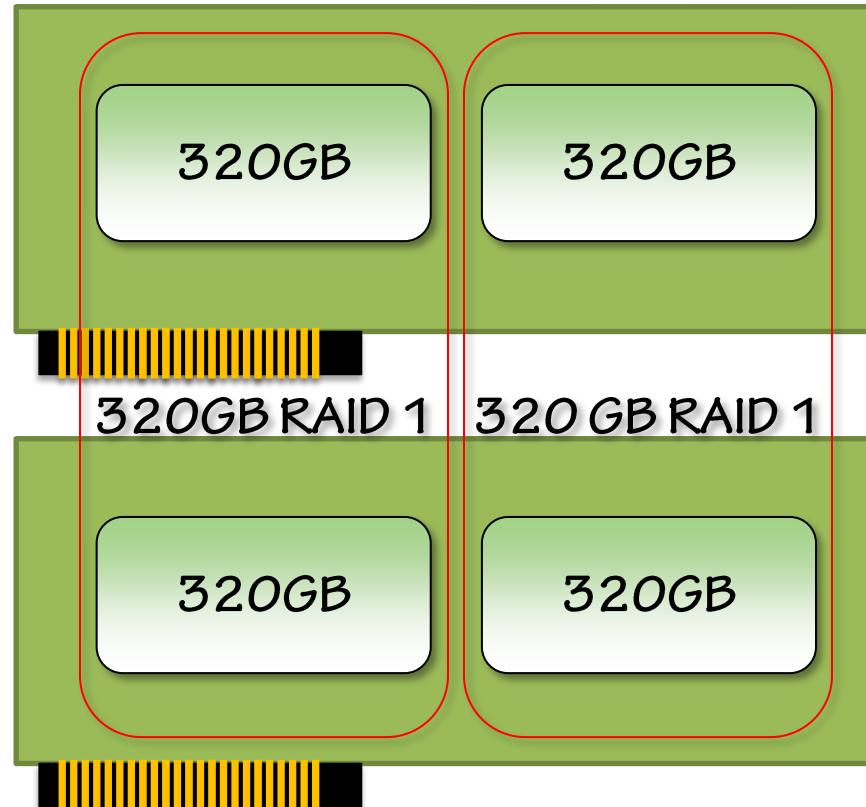


PCI-Express SSDs and RAID 1



- RAID 1 using a single card still does not provide redundancy against a controller failure on the card or PCIe slot failures

PCI-Express SSDs

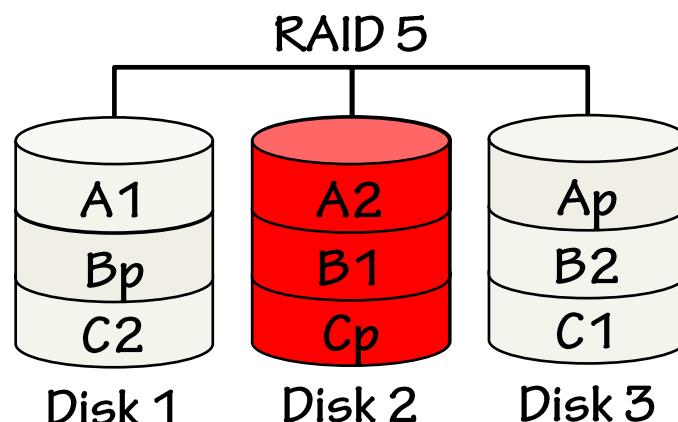


- 640GB capacity with full redundancy requires two cards and RAID 1 across the cards

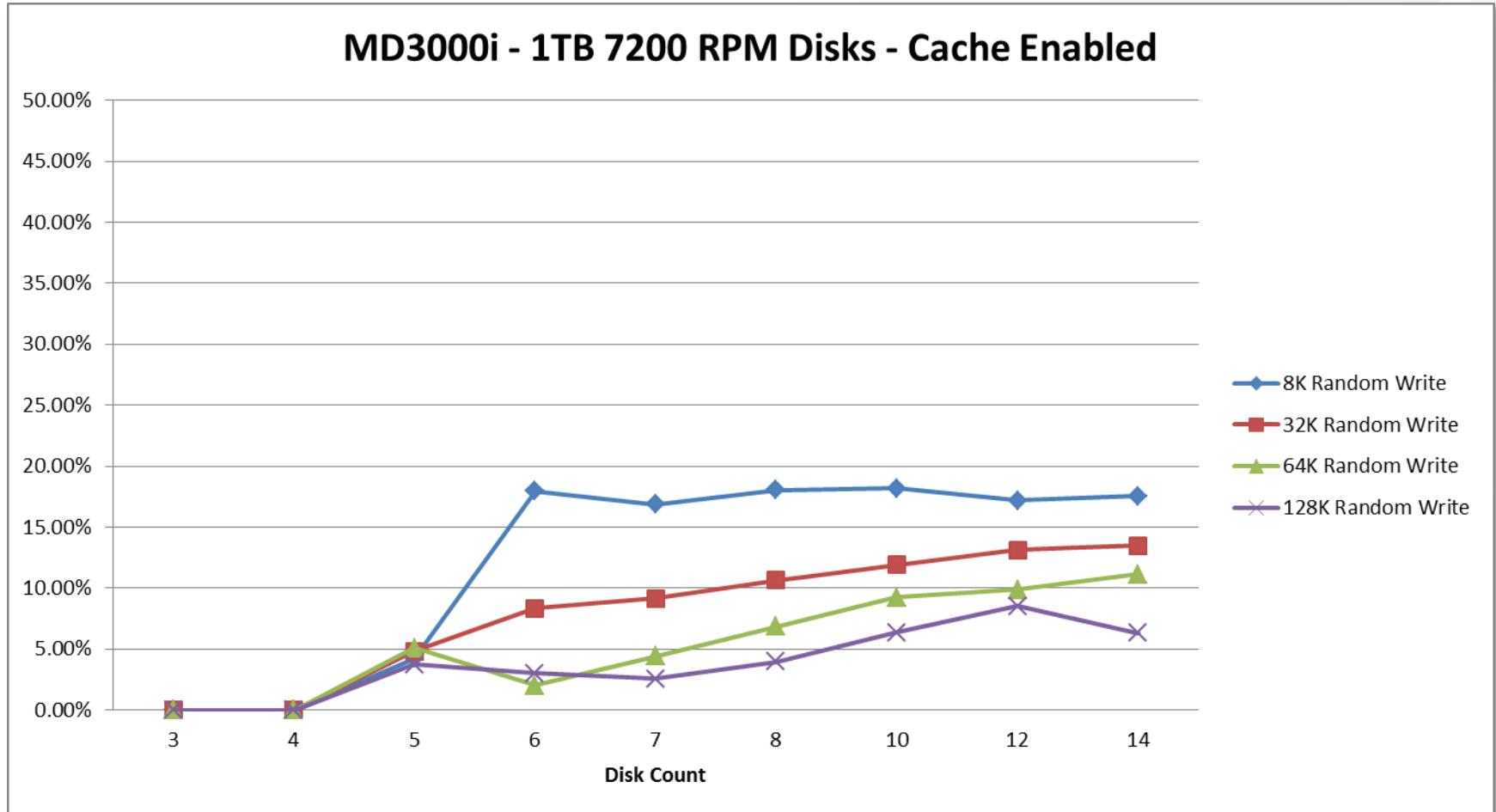
RAID Levels: RAID 5

Striping with Parity

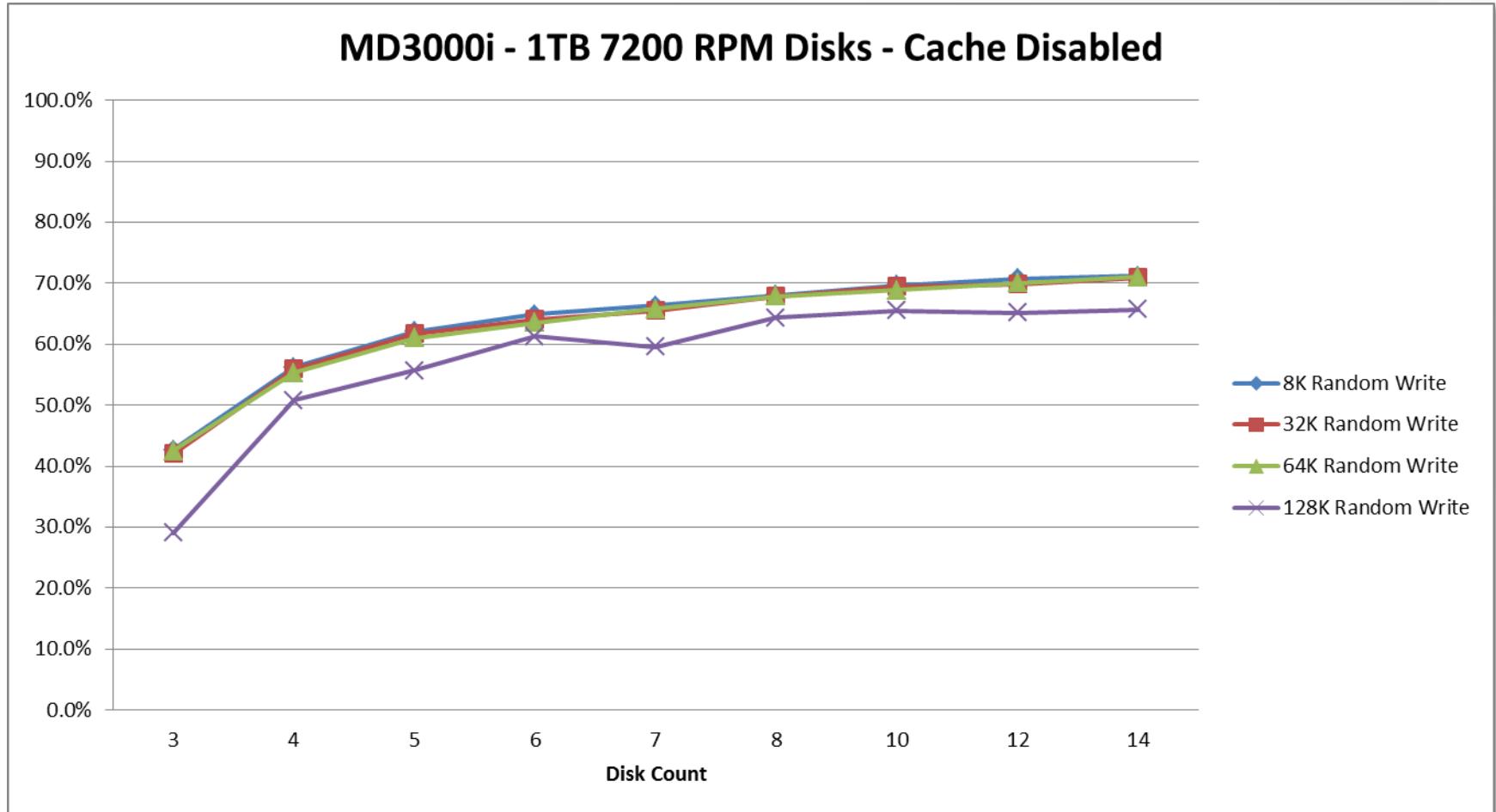
- **Usage scenarios:**
 - Where reads exceed writes, data files
- **Pros:**
 - Maximize available capacity from minimal number of disks with redundancy
- **Cons:**
 - Protection from single disk failure only as loss of two disks results in total data loss
 - Write penalty for parity calculation
 - Significant performance impact occurs when degraded



RAID 5 Degraded Performance



RAID 5 Degraded Performance



Nested RAID Levels

- Implement one level of RAID on top of another level of RAID
- Can be used to create better redundancy as well as higher levels of performance
- Common nested RAID levels are
 - 10 or 1+0: striping over mirrored pairs
 - 01 or 0+1: mirroring over striping
 - 50 or 5+0: striping over single-parity striping
 - 60 or 6+0: striping of double-parity striping
 - 100 or 10+0 or 1+0+0: striping over a striped set of mirrored pairs

Nested RAID Levels: RAID 10 or 1+0

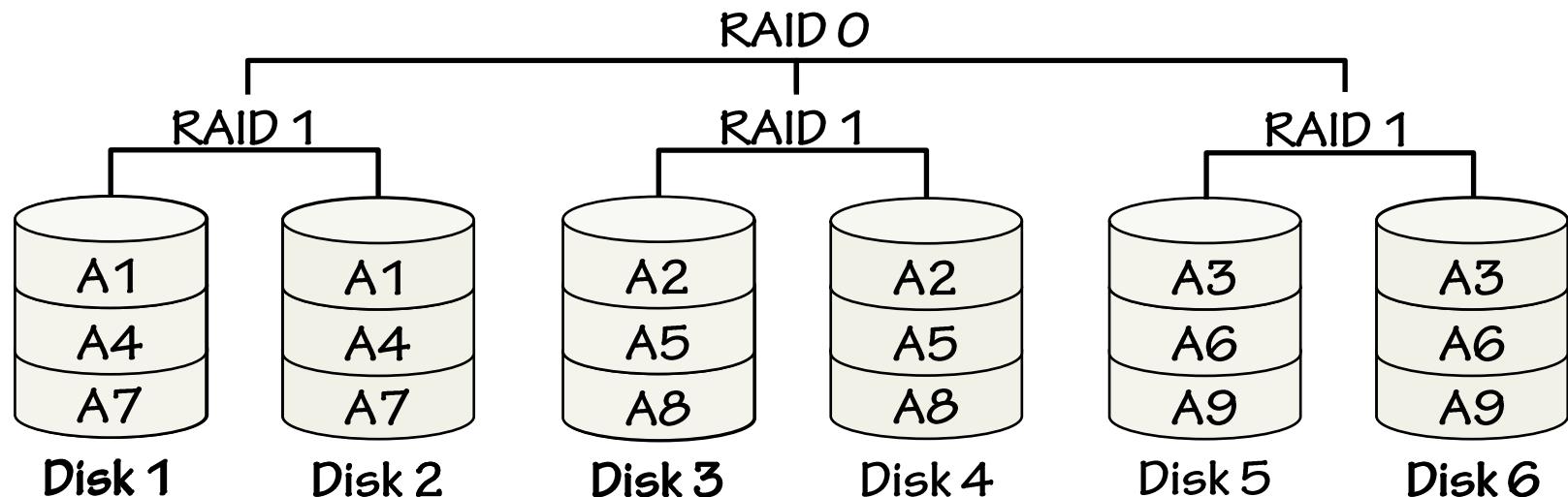
Striping Over Mirrored Pairs

- **Usage scenarios:**
 - Transaction log files, heavy write data files
 - Redundancy is more important than cost
- **Pros:**
 - Fast read and write performance
 - Supports multiple disk failures as long as two of the failed disks aren't in the same RAID 1 pair
- **Cons:**
 - Doubles the cost of storage

Nested RAID Levels: RAID 10 or 1+0

Striping Over Mirrored Pairs

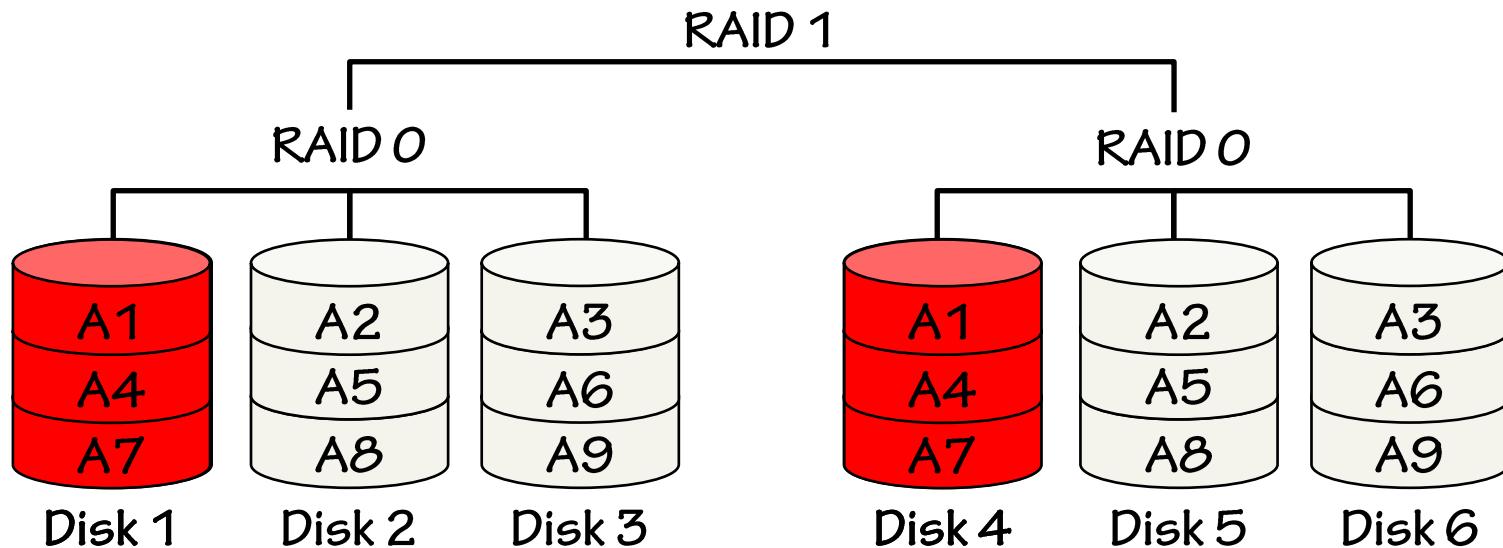
- Single disk loss keeps array available
- Multiple disk losses in different RAID 1 keeps array available
- Double disk loss in same RAID 1 results in array failure



Nested RAID Levels: RAID 01 or 0+1

Mirroring Over Striping

- IS NOT THE SAME AS RAID 10/1+0
- Single disk loss keeps array available
- Double disk loss in different RAID 0 results in array failure



Other RAID Configurations

- **RAID 6**
 - Similar to RAID 5 but provides double parity disks allowing for up to two disks to fail and the array remains available
 - Common recommendation for some SAN vendors
- **RAID 50, RAID 100, etc**
 - Specialized RAID configurations that stripe other RAID levels to increase performance

DAS Benefits

- Inexpensive to implement and achieve high performance configurations for SQL Server
- Ease of configuration requires minimal knowledge of underlying storage specifics
- Storage dedicated to SQL Server simplifying monitoring and troubleshooting of performance

DAS Disadvantages

- **Does not support clustering before SQL Server 2012**
 - Some exceptions exist where a single array may be connected by serially-attached SCSI (SAS) to two servers redundantly
 - MD3200
- **Requires appropriate IOPS, MB/sec, and storage size considerations up front**
 - LUN configurations cannot be dynamically reconfigured once set up (some exceptions exist)
 - Beware of controller and interface bottlenecks that limit the available throughput from the disks in the array

Storage Area Networks (SANs)

- “A storage area network (SAN) is any high-performance network whose primary purpose is to enable storage devices to communicate with computer systems and with each other” (Storage Networking Industry Association)
 - Does not specify interconnect technology
 - Does not specify types of storage devices

What Do You Think of as a SAN?



Scale out Compellent

OR



Equallogic PS6210XS

- Reality is: both are SANs and have very different capabilities!

SAN Advantages

- **Shared storage**
 - Increases disk utilization
 - Reduces management by making it easier to create new volumes and dynamically allocate storage
 - Create diskless servers that boot from SAN only
- **Advanced features**
 - Mirroring, snapshots, continuous data protection, clustering and geo-clustering only offered by SANs
- **Performance**
 - Almost unlimited number of spindles, controllers, and cache can be put together to meet the requirements

SAN Disadvantages

- **Unpredictable performance**
 - When you share your disks, controllers, and fiber switches between dozens of servers it is very difficult to have predictable performance
- **Higher latency**
 - Distance the I/Os have to travel; added layers of switches, cabling and ports
 - PCI Bus → HBA → FC switches → FC ports → array processors → disks
- **Limited bandwidth**
 - FC now 16Gb/s max
 - iSCSI 10Gb/s max
 - Note: 1Gb/s = 128MB/s
- **Cost**

SAN Components

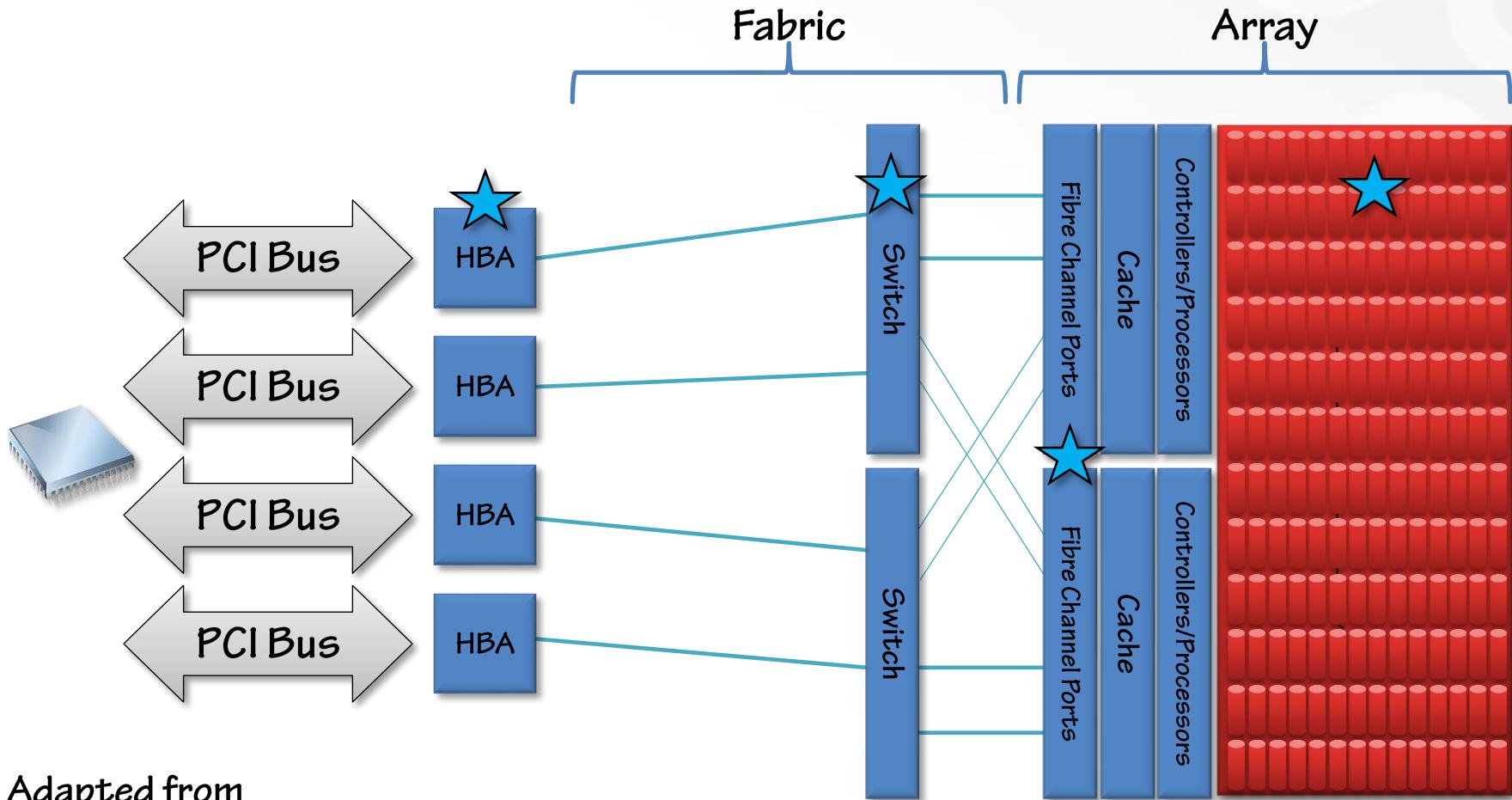
Basics

- **Drives**
 - Physical interface, size, rotation speed, and RAID configuration level
- **SAN storage controllers**
 - FA port/iSCSI port count and speeds
 - Cache size and configuration
- **SAN fabric**
 - Number of switches, their port speeds and path configurations
- **Host bus adapter**
 - Queue depth, port speeds, pathing/multipathing

Multipathing

- Primary purpose of multipathing configurations is to provide redundancy and protection
- Secondary to redundancy is to improve performance
 - Just having multiple paths doesn't guarantee a performance improvement
 - Some SANs only allow one path per LUN
 - iSCSI allows one connection per target
 - Achieving performance improvements may require changes to the physical database layout

Path to the Drives: SAN



Adapted from
slide deck by
Thomas Kejser
(with permission)

★ Denotes possible bottleneck point
Slowest point in the configuration is the limiting factor

Understanding SAN I/O Capacity

- The available bandwidth of a SAN is limited to the lowest available bandwidth bottleneck in the configuration
- Example scenario
 - HBAs: 2 x 8Gb/s = ~1,600MB/s (max)
 - Switch: 2 x 8Gb/s = ~1,600MB/s (max)
 - SAN has 4 x 2Gb/s FA Ports = ~800MB/s (max)

Understanding SAN I/O Capacity

Scenario 1

- HBAs: $2 \times 8\text{Gb/s} = \sim 1,600\text{MB/s}$ (max)
- Switch: $2 \times 8\text{Gb/s} = \sim 1,600\text{MB/s}$ (max)
- SAN ports: $4 \times 2\text{Gb/s FA ports} = \sim 800\text{MB/s}$ (max)
- Cache: $1 \times 1,600\text{MB/s} = \sim 1,600\text{MB/s}$ (max)
- Disks: $80 \times 20\text{MB/s} = \sim 1,600\text{MB/s}$ (max)

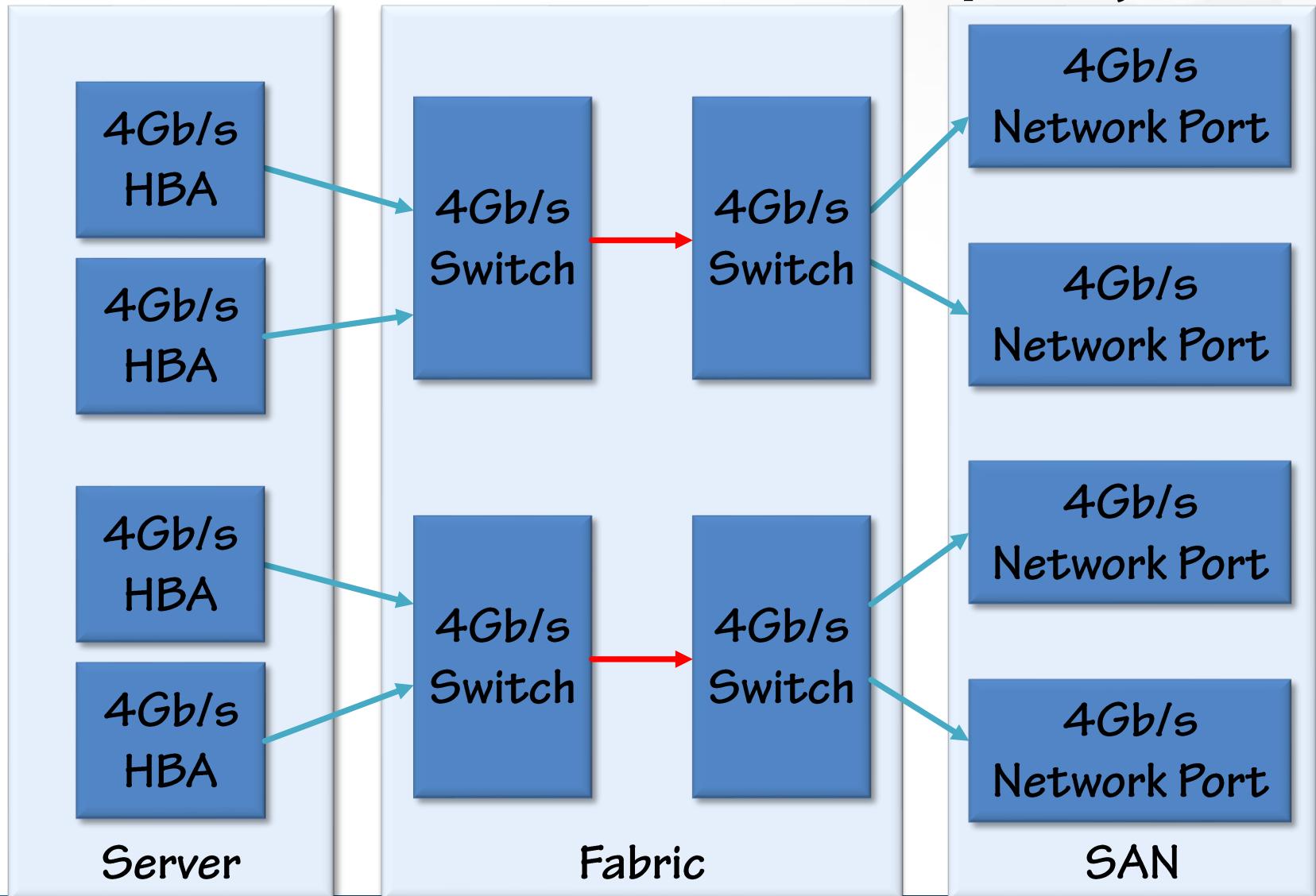
- Maximum bandwidth is limited by the SAN ports

Understanding SAN I/O Capacity

Scenario 2

- HBAs: $2 \times 8\text{Gb/s} = \sim 1,600\text{MB/s}$ (max)
- Switch: $2 \times 8\text{Gb/s} = \sim 1,600\text{MB/s}$ (max)
- SAN ports: $4 \times 8\text{Gb/s FA ports} = \sim 3,200\text{MB/s}$ (max)
- Cache: $1 \times 1,600\text{MB/s} = \sim 1,600\text{MB/s}$ (max)
- Disks: $40 \times 20\text{MB/s} = \sim 800\text{MB/s}$ (max)
- Maximum bandwidth is limited by the disks

Understand Bottlenecks Completely



iSCSI vs. Fibre Channel

Best Practices

- **Use the fastest interface you can:**
 - 16Gb/s FC, 10Gb/s iSCSI, 8Gb/s FC, 4Gb/s FC, 2Gb/s FC, 1Gb/s iSCSI or FC
- **The slower the interface the more important appropriate multipathing becomes to achieving the needed throughput**
- **Know the limitations of the controller's connections**
 - You can probably connect more than you are currently using

iSCSI Considerations

- **Network design**
 - Dedicated hardware including HBAs, switches, and cabling
 - TCP Offload Engine (TCOE) enabled NICs or iSCSI HBAs for servers
 - Use Jumbo Frames end-to-end on iSCSI network to improve throughput for larger packet sizes
- **Single path per target may require considerations for LUN and database file layout to achieve the optimal I/O configuration**
- **May require intervention to bring multiple paths back online after failure**

iSCSI Performance Characteristics

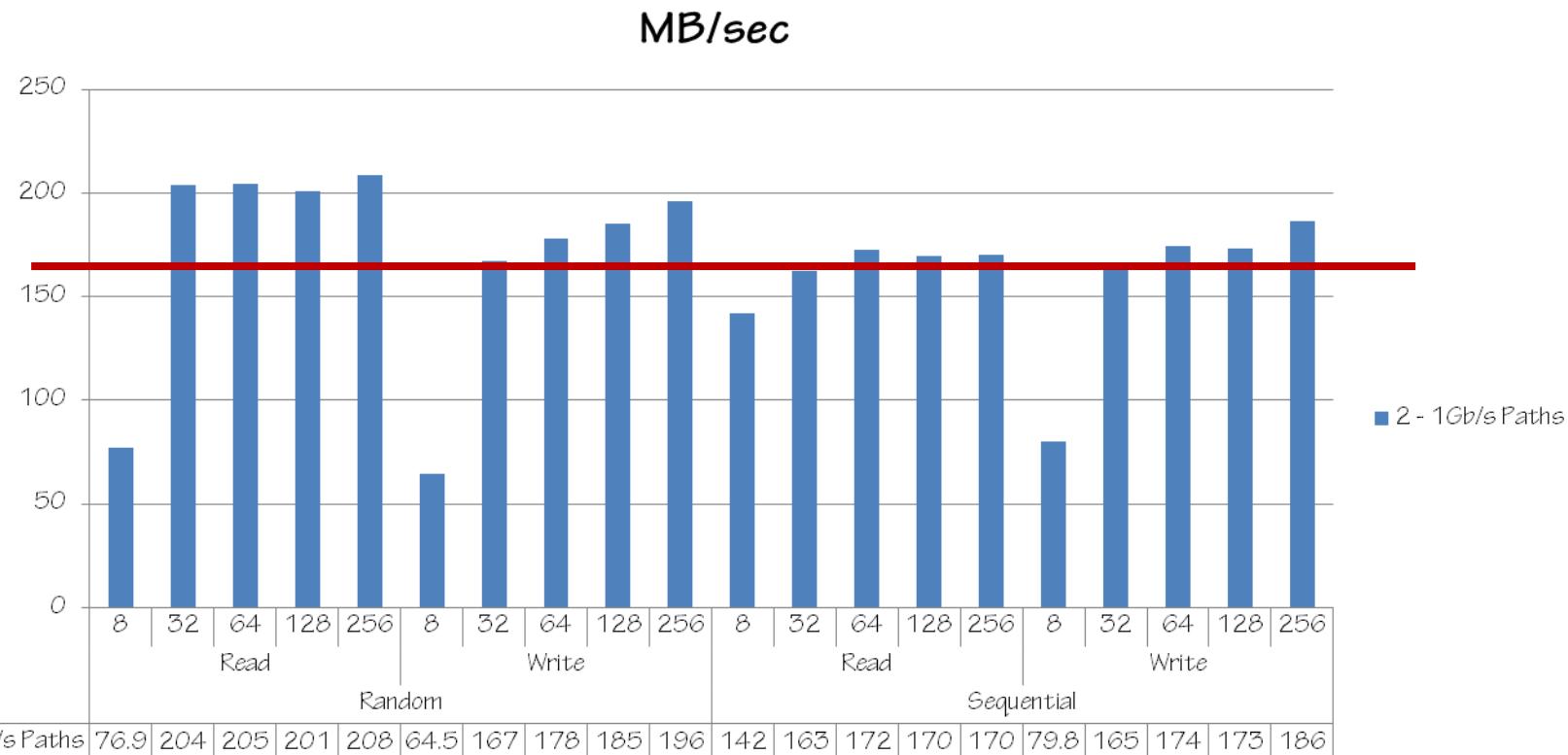
- **Equallogic PS-6100XS**

- 7 x 400GB SSD + 17 x 600GB 10K RPM
 - RAID 6
 - Two controllers – 4GB cache, 4 x 1Gb/sec per controller

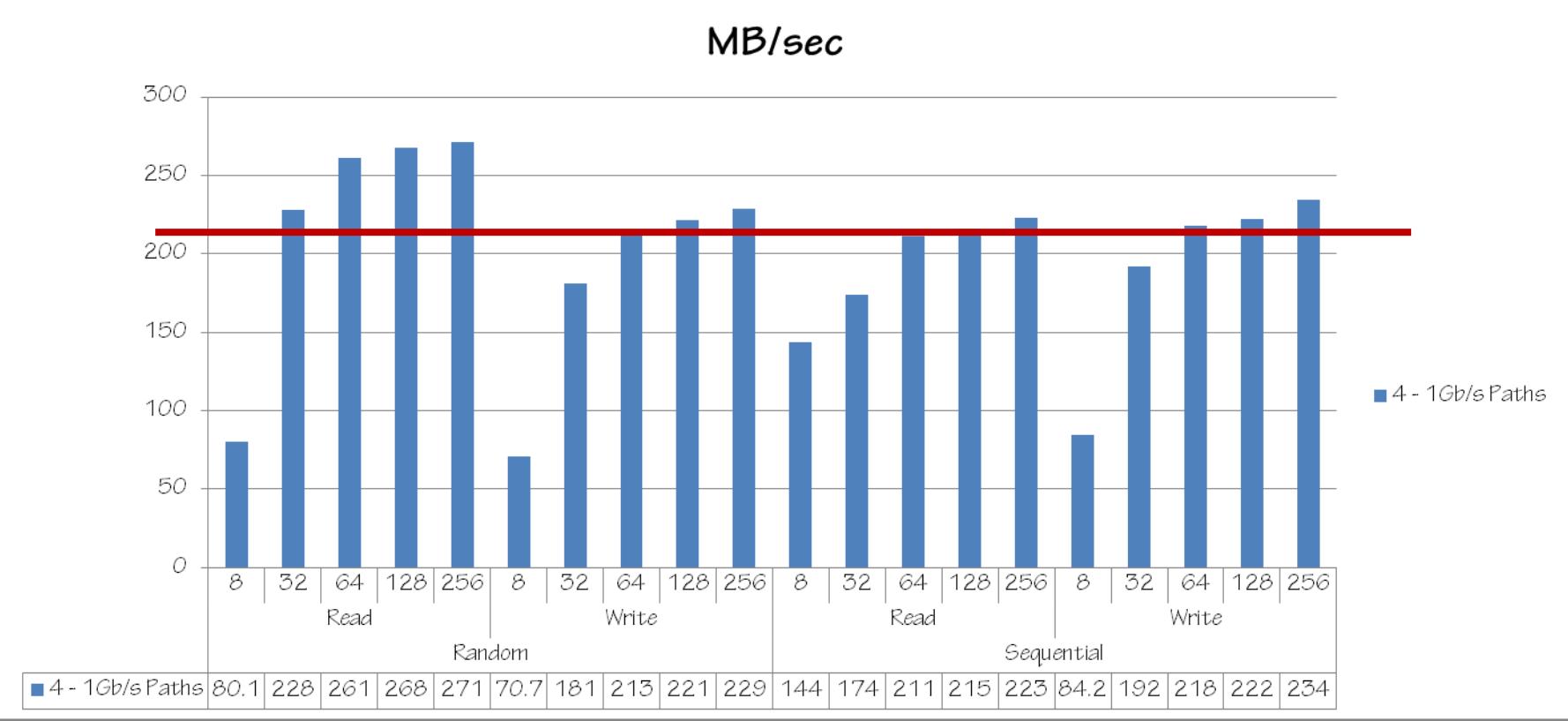
- **Equallogic PS-6110XS**

- 7 x 400GB SSD + 17 x 600GB 10K RPM
 - RAID 6
 - Two controllers – 4GB cache, 4 x 10Gb/sec per controller

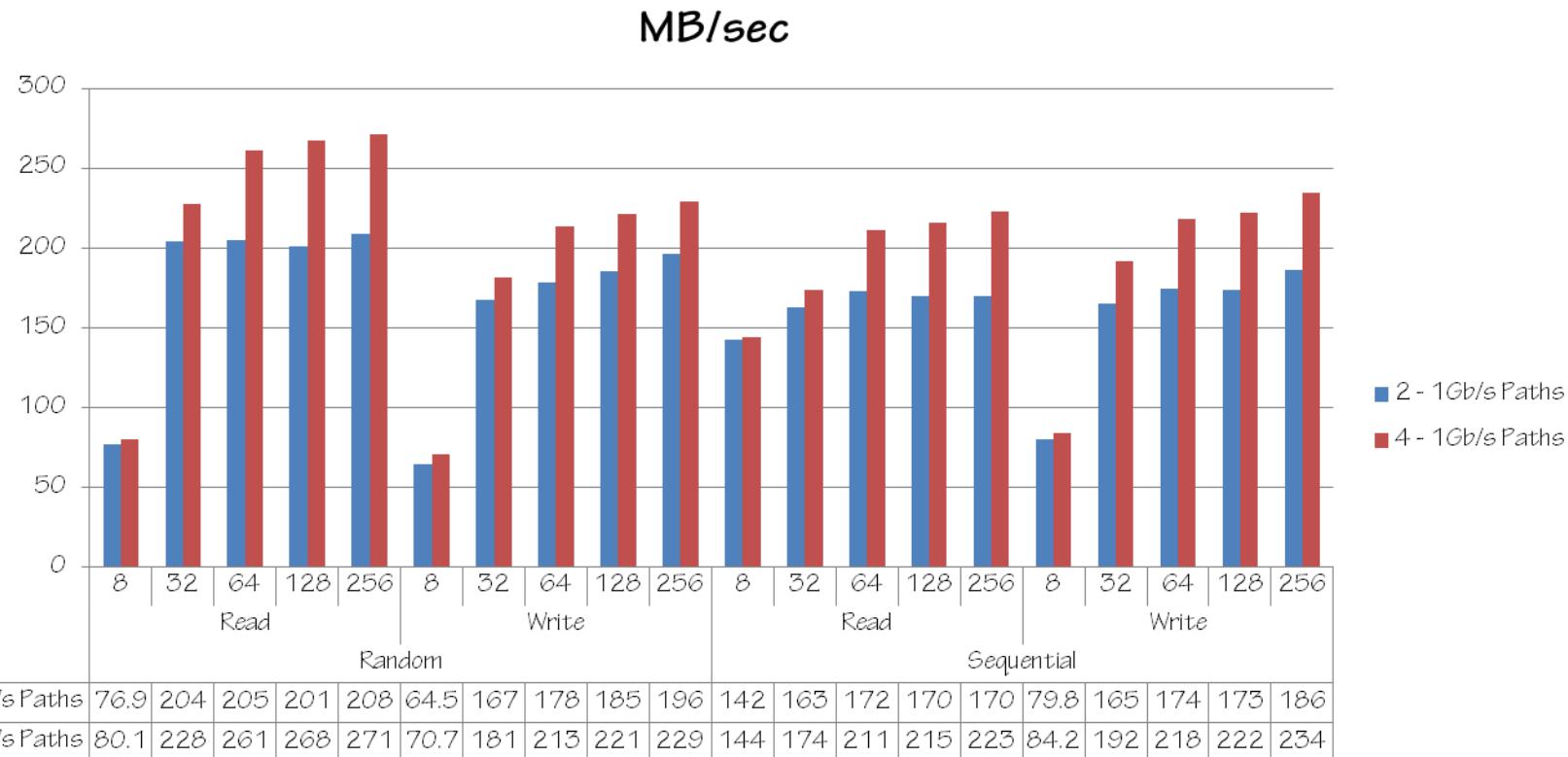
PS-6100XS – Two 1Gb/s Paths



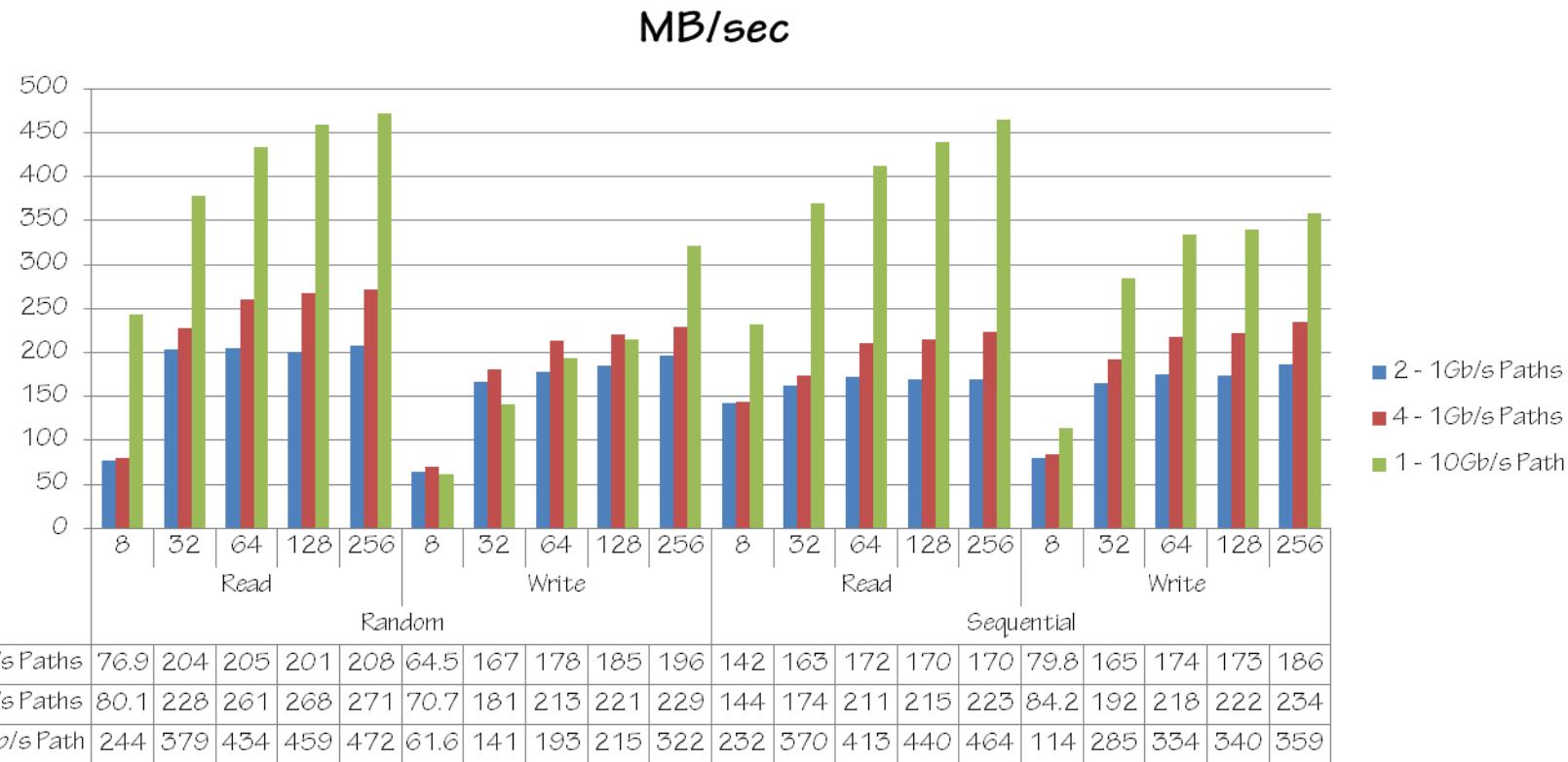
PS-6100XS – Four 1Gb/s Paths



PS-6100XS – Side-by-Side Comparison



Comparison to PS-6110XS – 10Gb/s iSCSI



Real World Customer Example

- **EMC CX4-240 SAN with 240 15K RPM disks**
 - Each shelf of drives contains 16 15K RPM 300GB disks – total of 15 shelves
 - Three shelves of drives (total 48) dedicated to SQL Server
 - 4 disk RAID 10 for system database data files
 - 4 disk RAID 10 for system database log files
 - 8 disk RAID 10 for main database data files
 - 4 disk RAID 10 for main database log files
 - 8 disk RAID 10 for tempdb data files
 - 4 disk RAID 10 for tempdb log file and secondary database log file
 - 6 disk RAID 10 for second database data file
 - 7 disk RAID 5 for backups
 - 3 hot spares (one per shelf)
 - Avg. Latency for writes 70-200ms
 - Avg. Latency for reads 140-260ms

Real World Customer Example (2)

- Same EMC CX4-240 SAN with 240 15K RPM disks
 - Each shelf of drives contains 16 15K RPM 300GB disks – total of 15 shelves
 - Three shelves of drives (total 48) dedicated to SQL Server
 - 3 hot spares (one per shelf)
 - 28 disk RAID 5+0 (EMC internal striping of 4 disk RAID 5) for database data files
 - 10 disk RAID 10 for log files
 - 7 disk RAID 5 for backups
 - Avg. Latency for writes 20-40ms
 - Avg. Latency for reads 20-50ms
- Identical workloads and the same total number of disks but striped so that performance is significantly improved
- This also is EMC's recommended configuration for SQL Server

Storage Spaces Direct (S2D) Windows Server 2016

- Storage Spaces Direct uses local-attached drives to create a converged or hyper-converged environment that supports failover clustering and shared volumes across independent servers
 - Converged = SMB3
 - Hyper-converged = CSVs
- Hyper-converged configurations require 4GB RAM per TB of drive capacity for cache and S2D internal usage
- Requires Windows Server 2016 Datacenter and a minimum of 2 nodes with a maximum of 16 nodes
 - All servers must have the same drive types
 - 2 drive minimum for cache per server if using caching
 - 4 drive minimum for capacity per server (2 for virtual machines)

Cloud Storage (Azure and AWS)

- Cloud infrastructure performance is based entirely on the configuration appropriately being sized to requirements
- Maximum IOPS limits per VM and per disk volume are independent of each other
 - AWS EC2 instance type maximum I/O throughput limits (<https://sqlskills.com/help/aws-ebs>)
 - Azure VM size and type IOPS limits (<https://sqlskills.com/help/azvmio>)
- Achieving appropriate storage throughput usually requires striping multiple volumes together using Windows Storage Spaces (not S2D)
 - Must use PowerShell for configuration to set column count (# disks to stripe data across) – THIS IS INCREDIBLY IMPORTANT!!! Column count cannot be changed after VDISK creation.

VMware Considerations

■ Controller types

- LSI Logic SAS
 - ESXi 4.x and later – max of 60 virtual disks per controller
 - Common for OS boot device – not recommended by VMware for I/O intensive workloads
- VMware Paravirtual SCSI
 - ESXi 4.x and later – max of 256 virtual disks per controller
 - Device queue depth limited to 64 and controller queue depth limited to 254 by default
 - Can be increased to 254 per device and 1024 per controller
<https://kb.vmware.com/s/article/2053145>
 - Max of 4 per VM – distribute VMDKs across multiple controllers
- NVMe
 - Requires appropriate hardware support
 - High performance access to non-volatile memory storage devices

Storage Design

Shared vs. Dedicated Spindles

- **Shared spindles allows for higher utilization of the underlying storage capacity**
 - Requires significant consideration up front to minimize the performance impact of sharing spindles
 - One application (often SQL Server, Exchange, or other OLTP solution) can cause the majority of I/O demand
 - I/O demanding applications should never share the same physical spindles
- **Dedicated spindles have lower utilization of storage capacity but predictable performance**
 - Some SANs only offer shared storage configurations
- **Virtualization**
 - Underlying VHD/VMDK appears dedicated but may be on shared LUN physically

Windows Disk Types

Basic

- Default and most common disk type to use in Windows for SQL Server
- Can use master boot record (MBR) or GUID-partition table (GPT) partition types
 - MBR limit to 2TB in size
- Volumes can be extended if contiguous free space exists physically

Disk Management Volume List + Graphical View										
Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead	
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash D...)	39.90 GB	25.60 GB	64 %	No	0%	
(D:)	Spanned	Dynamic	RAW	Healthy	19.99 GB	19.99 GB	100 %	No	0%	
(E:)	Striped	Dynamic	RAW	Healthy	19.99 GB	19.99 GB	100 %	No	0%	
(F:)	Mirror	Dynamic	RAW	Healthy	10.00 GB	10.00 GB	100 %	Yes	50%	
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary ...)	100 MB	72 MB	72 %	No	0%	

The graphical view shows two main components. On the left, 'Disk 0' is listed with a total capacity of 40.00 GB, being used as a 'System Reserved' volume (100 MB NTFS) which is healthy. On the right, the 'C:' volume is shown with a capacity of 39.90 GB NTFS, also marked as healthy. The background of the graphical area has diagonal stripes.

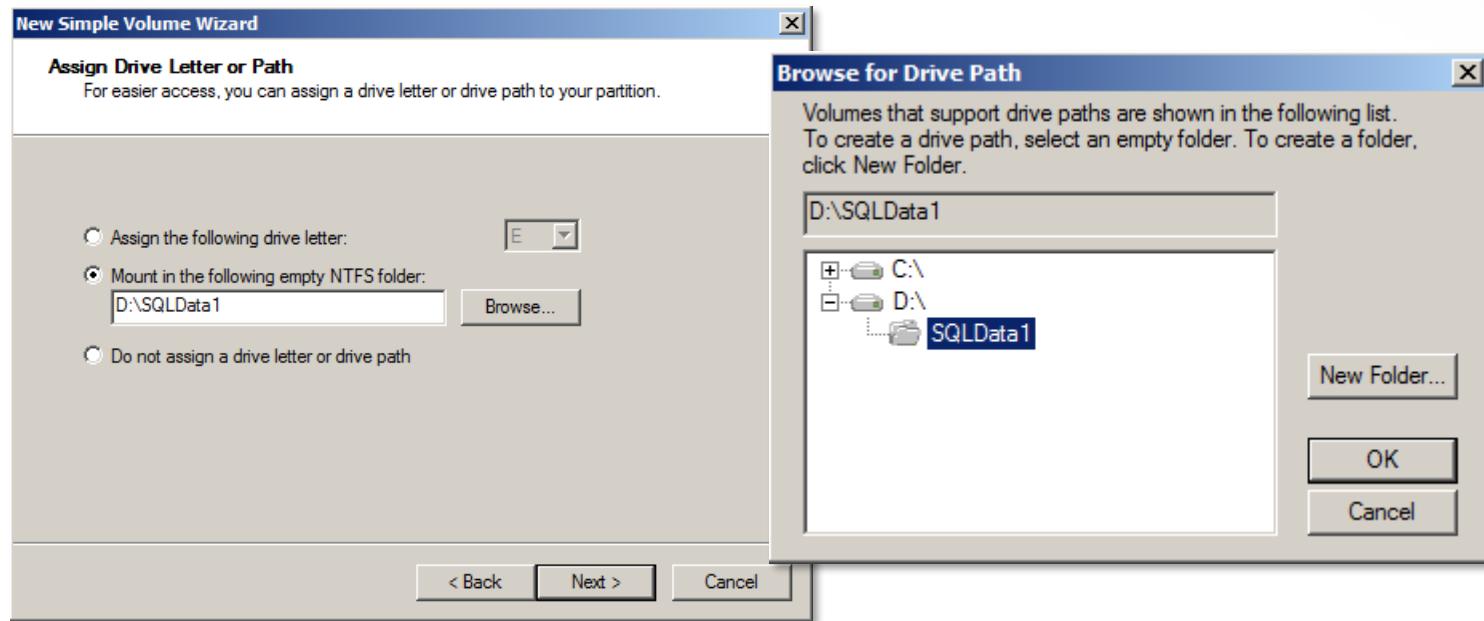
Windows Disk Types

Dynamic

- **Software RAID**
 - Create RAID 0, RAID 1, and RAID 5 volumes
- **Use with SQL:**
 - Mirrored (RAID 1)
 - Use with Fusion-io or other PCI-X SSDs to provide redundancy across multiple cards
- **Do not use:**
 - Striped (RAID 0)
 - No redundancy
 - Spanned
 - Data written to disks sequentially filling each disk before writing to the next disk in the span
 - Striping with parity (RAID 5)
 - High CPU overhead for software-parity calculations

Mount Points

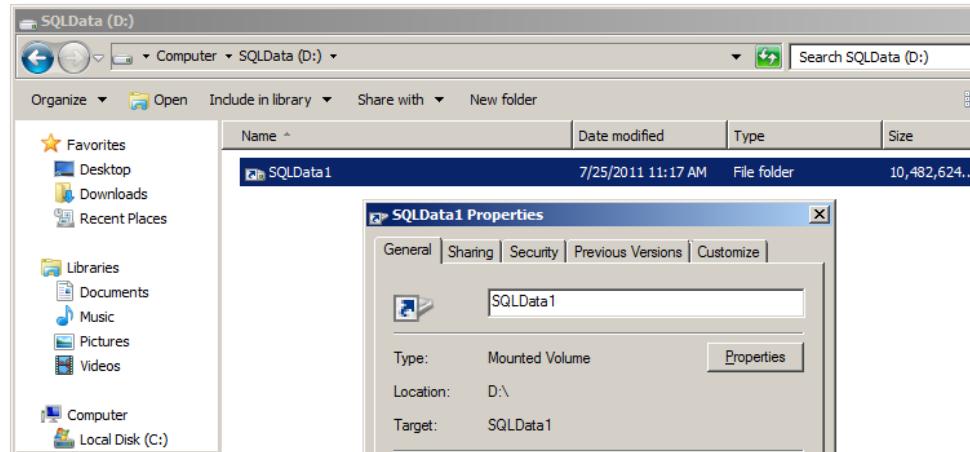
- Provide a mechanism to exceed 26 volume limit based on available drive letters
 - Additional volumes can be mounted as folders under an existing volume
- May be required for large multi-instance clusters



Mount Points

Monitoring Complications

- Mount point support is not built into most monitoring tools requiring custom solutions to monitor space usage of the mounted volumes
- Mount points properties provide the target volume information for matching to Disk Management
- `sys.dm_os_volume_stats` DMF in SQL Server 2008 R2 onwards will return mounted volume information per database file



Partition Alignment

Background

- **Disks report the first 63 sectors as hidden**
 - This reserved portion is where the master boot record is stored
 - Equates to 31.5KB of disk space: (512 bytes per sector) x (63 sectors)
- **Prior to Windows Server 2008, partitions were offset by 31.5KB resulting in misalignment of the partition with stripe units, disk controller, and cache segment lines**
 - Misalignment can reduce performance by 30% or more depending on the array stripe size and allocation unit size
 - Plenty of evidence from many, many customers

Partition Alignment

Determining Offset

- **(partition offset) divided by (stripe unit size)**
 - Should result in a whole number
 - Common stripe sizes are 64KB, 128KB, 256KB and 512KB
 - $32,256 \text{ bytes (31.5 KB)} / 65,536 \text{ bytes (64 KB)} = 0.4921875$
 - Misaligned
 - $131,072 \text{ bytes (128 KB)} / 65,536 \text{ bytes (64 KB)} = 2$
 - Properly aligned
- **Windows Server 2008**
 - Uses 1,024 KB (2,048 sectors) alignment for new partitions
 - Previously created partitions may be misaligned
 - Even on Windows Server 2008, always check and validate
 - Command Line: `wmic partition get StartingOffset, Name`
 - Manually create partitions using DiskPart to set offset when necessary
 - Uncommon stripe sizes

NTFS Allocation Unit Size

- Also known as 'NTFS cluster size'
- Sets the smallest unit of consumption on NTFS volume for files
- Can only be set when the volume is formatted
- Default size: 4KB
 - Optimized for storage of lots of small files
- For SQL Server a 64KB allocation unit size should be used to match the size of a single extent
 - Improves read-ahead performance
 - Reduces the number of split I/Os
 - Does not allow the disk to use NTFS compression
 - Plenty of empirical evidence to prove this
- This is not as critical as partition alignment for performance!!

Benchmarking Storage

Diskspd and IOMeter

- **Use test files that are at least twice the size of the installed cache size**
 - Small files against a large cache = cached I/O
 - Disabling cache = not reality during production
 - Exceeding cache size forces physical disk I/Os
- **Test different I/O sizes (4KB, 8KB, 64KB, 256KB)**
- **Test different workload types (read/write, sequential/random)**
- **Test impact of thread count and outstanding I/Os**

Considerations During Testing

- **Test normal configuration and impact of degraded RAID**
 - Degraded performance can be significantly worse than normal configuration and may not meet SLA
- **If testing a shared SAN, ensure that the SAN is under its normal workload during the tests**
- **“Pull-the-plug” tests**
 - Ensure that multipathing works as expected when one of the paths is removed
 - Test for path stickiness after restoration of a failed path to determine if the path resumes normal operation or not
 - Ensure that write caching is battery backed and doesn’t lead to lost write or data corruption issues

Key Takeaways

- Configure storage based on performance and redundancy requirements, not strictly on capacity
- The type of workload and data being stored on a disk array should dictate the RAID level being used
 - Don't use RAID 5 for heavy writes (e.g. transaction logs, tempdb), use RAID 10 instead to provide better redundancy and eliminate parity overhead
- PCI-Express SSDs require extra cards with RAID 1 across the cards for true redundancy

Key Takeaways

- Follow the SAN vendor's recommended best practices for SQL Server
- Don't assume SAN-based storage performance problems are automatically a problem with the SAN itself, remember there are a lot of components between the server and the actual storage
 - Watch out for the "I can ping it" network guy, the SAN admin hates him too
- Collect performance metrics in Performance Monitor and bring your latency numbers to the SAN admin to show where performance is affecting SQL Servers workload to troubleshoot
- Don't over-think SAN storage configuration and try to slice the storage up following "best practices" for isolation, it might be better to use a larger pool of disks, dedicated to SQL Server, than to have smaller isolated groups

Additional Resources

- Storage Area Network Essentials <http://amzn.to/1sLIEJy>
- SQL Server Best Practices for I/O Whitepaper <http://bit.ly/1rwtbYN>
- Storage Top 10 Best Practices <http://bit.ly/YHxhmJ>
- SQL Server SAN Best Practices <http://bit.ly/1nARo27>

Additional Resources

- NAND Basics: Understanding the Technology Behind Your SSD
<http://bit.ly/1sLIKRI>
- SLC, MLC or TLC NAND for Solid State Drives <http://bit.ly/YD88tz>
- The SSD Endurance Experiment <http://bit.ly/1pmthPy>
- Disk Partition Alignment Best Practices for SQL Server
<http://bit.ly/1uWKaE2>

Review

- **Defining storage terminology**
 - IOPS, MB/sec, Latency
- **Magnetic vs. Solid State Drives (SSDs)**
- **RAID configurations**
- **Storage Area Networks**
- **Configuring drives in Windows**
- **Testing with Diskspd and Iometer**

Questions?

