

SQLskills Immersion Event

IE2: Performance Tuning

Module 2: I/O for DBAs

Jonathan Kehayias
Jonathan@SQLskills.com



Overview

- Defining storage terminology
 - IOPS, MB/sec, Latency
- Magnetic vs. Solid State Drives (SSDs)
- RAID configurations
- Mount points
- Partition alignment
- Testing with SQLIO and IOMeter
- Direct-attached storage
 - Traditional vs. PCI-Express drives



2

© SQLskills, All rights reserved.
<http://www.SQLskills.com>



IOPS

- Number of read or write operations
- Size matters
- Higher is better
- Rough approximation in Performance Monitor
 - Physical Disk: Disk Transfers/Sec
 - Physical Disk: Disk Writes/Sec
 - Physical Disk: Disk Reads/Sec

Throughput (Bandwidth) in MB/sec

- Rough approximation:
(number of operations) x (size of operation)
- Higher is better
- Generally not as important as IOPS for OLTP databases, but is significant for databases that transfer large volumes of data
- Very useful for testing appropriate functionality of multipathing

Latency

- How long it takes an operation to finish measured in milliseconds
- **Monitoring**
 - Physical Disk: Avg. Disk Sec/Read
 - Physical Disk: Avg. Disk Sec/Write
- **General SQL Server guidelines**
 - < 8ms: excellent
 - < 12ms: good
 - < 20ms: fair
 - > 20ms: poor
- **Reality is “It Depends”**
 - Should be SLA-based to meet business requirements
 - Log writes should be as fast as possible (i.e. minimal latency)

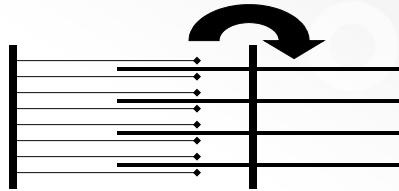
Traditional HDD Access Time

- The response time or amount of time 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

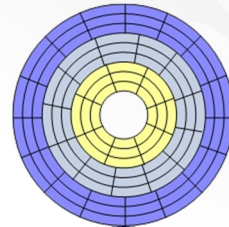
Traditional HDD Platters

- Drives have multiple platters with read/write heads above and below each platter
- “Disk head is like a 747 flying 500mph at 1/4 inch above the ground” – Jim Gray
- Multiple heads still have a single articulation point which prevents parallel I/O operations on a single disk



Traditional HDD Data Density

- Outer tracks have a higher data density than inner tracks due to the distance traveled per rotation
 - ~4 sectors vs. 3 sectors vs. 2 sectors
- Short stroking disks can increase the number of IOPS and decrease seek time by shortening the distance the read/write head must travel to change locations on the platter
 - Trade off better performance for lower storage usage on disk
 - 1TB 7200 RPM disk short stroked 300GB will have a faster seek latency than a 300GB 15000 RPM disk at full stroke



SSDs

SLC vs. MLC

- **SLC: Single-Level Cell flash stores a single bit value with two states 0 or 1**
 - Lower voltages results in longer endurance of 100,000 write cycles
- **MLC: Multi-Level Cell flash stores a double bit value with four states, 00, 01, 10, or 11**
 - Higher data density per cell at the trade off of higher voltages with reduce cell endurance of 10,000 write cycles

SSDs

Access Time

- SSD access time does not depend on mechanical moving parts, but electrical connections to solid state memory, so the access time is very fast and consistent across cells
- General average seek time ranges between 0.08ms and 0.16ms

RAID

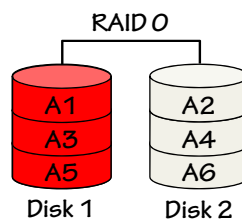
Redundant Array of Independent/Inexpensive Disks

- 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

RAID Levels: RAID 0

Striping

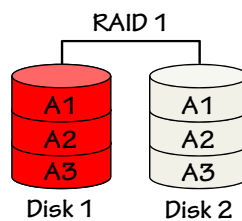
- **Usage scenarios:**
 - Data you don't care about
 - Performance matters more than data loss risk
- **Pros:**
 - Fast read and write performance
- **Cons:**
 - Total loss of data from single disk failure



RAID Levels: RAID 1

Mirroring

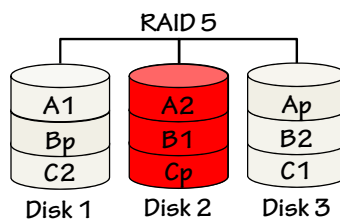
- **Usage scenarios:**
 - Important data that requires redundancy to protect from disk failures
- **Pros:**
 - Mirrored storage protects from single disk loss
- **Cons:**
 - Only provides half the storage
 - Write performance equal to one disk



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



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

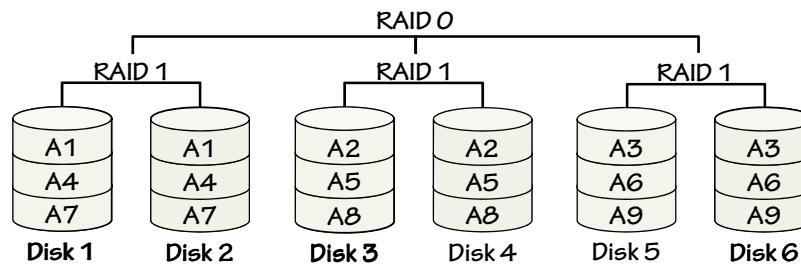
Nested RAID Levels: RAID 10 or 1+0 Striping Over Mirrored Pairs

- 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

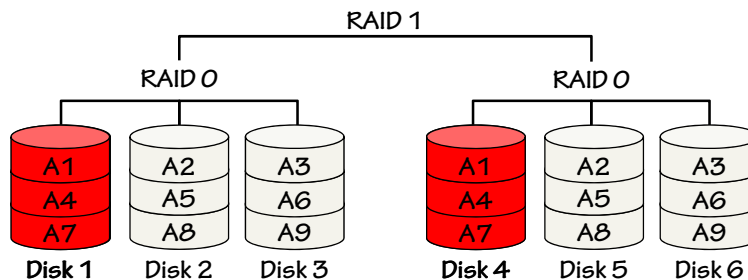
- Single disk loss keeps array available
- Multiple disk losses in different RAID 1 keeps array available
- Double disk loss in same RAID 1 results in array failure



Nested RAID Levels: RAID 01 or 0+1

Striping Over Mirrored Pairs

- 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

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)
 - 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%	

Disk 0		
Basic	System Reserved	(C:)
40.00 GB	100 MB NTFS	39.90 GB NTFS
Online	Healthy (System, Active, Primary Partition)	Healthy (Boot, Page File, Crash Dump, Primary Partition)

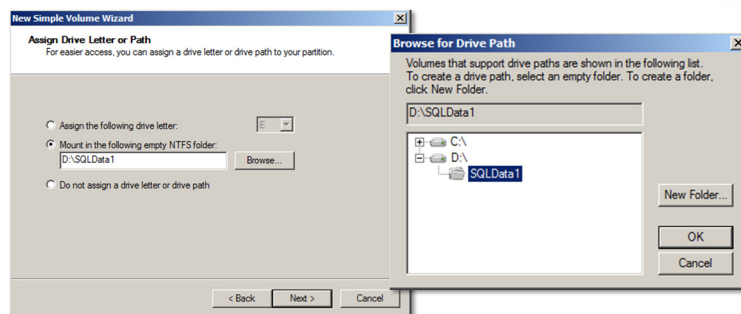
Windows Disk Types

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

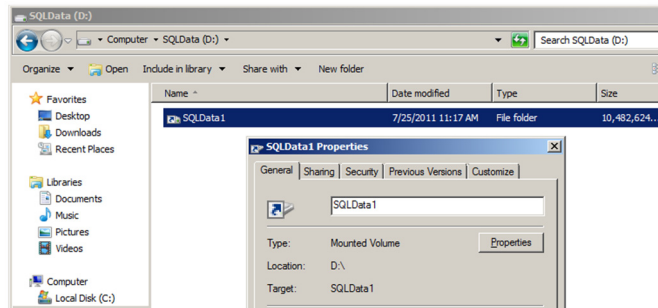
- **Provide a mechanism to exceed 26 volume limit based on available drive letters**
 - Additional volumes can be mounted as folders under an existing volume
- **May be required for large multi-instance clusters**



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



Demo

Mount points

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
 - $32256 \text{ bytes (31.5 KB)} / 65536 \text{ bytes (64 KB)} = 0.4921875$
 - Misaligned
 - $131072 \text{ bytes (128 KB)} / 65536 \text{ bytes (64 KB)} = 2$
 - Properly aligned
- **Windows Server 2008**
 - Uses 1024 KB (2048 sectors) alignment for new partitions created using Windows Server 2008
 - Partitions not created by Windows Server 2008 and previously created partitions may be misaligned
 - Even on Windows Server 2008, always check and validate

Partition Alignment Tools

- **Windows Server 2000**
 - Diskpart.exe
 - Works with MBR partitions only
 - Alignment set by number of sectors
- **Windows Server 2003+**
 - DiskPart.exe
 - Works with MBR and GPT partitions
 - Alignment set by size in KB
- **WMI**
 - To get current offset information
wmic partition get StartingOffset, Name

Using Diskpart

```
diskpart
list disk
select disk <DiskNumber>
create partition primary align=<Offset_in_KB>
assign letter=<DriveLetter>
format fs=ntfs unit=64K label="<label>" nowait
```

This slide is a reference for the demo!

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!!

Extending a Partition

- Grow an existing disk partition to use additional space
- Use when a SAN LUN is extended or VMDK/VHD grown

```
C:\Windows\system32\diskpart.exe
Microsoft DiskPart version 6.1.7600
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: W7SQL2008R2

DISKPART> select disk 2
Disk 2 is now the selected disk.

DISKPART> list partition
Partition ###  Type              Size      Offset
-----
Partition 1   Primary          10 GB     1024 KB

DISKPART> select partition 1
Partition 1 is now the selected partition.

DISKPART> extend
DiskPart successfully extended the volume.

DISKPART> _
```

Demo

Windows disks and partition alignment

Testing Disk Configurations

SQLIOSim

- **Verifies that the I/O subsystem functions correctly under high load**
 - Not a tool for performance benchmarking the configuration
 - Test the system against pull-the-plug scenarios
 - Some I/O controllers and SATA drives fake disabling the write cache and can cause lost writes from power failures
- **Generates data files and simulates SQL Server I/O patterns without requiring a SQL instance**
 - Uses a SQLOS-like scheduling layer
 - Checks correctness based on page checksums
 - Monitors “stuck” I/Os under high load (requests taking longer than 15 seconds)

Benchmarking Storage

SQLIO 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

Using SQLIO to Benchmark Storage

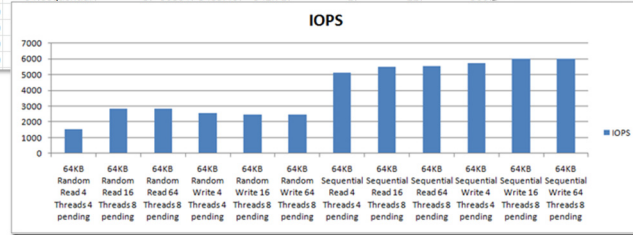
- `sqlio -kw -t16 -s10 -o8 -fsequential -b8 -BH -LS -Fparam.txt`
- `-k` R = read/w= write
- `-t` number of threads to use
- `-s` test duration in seconds
- `-o` number of outstanding I/O requests
- `-f` random/sequential (I/O type)
- `-b` I/O block size in kilobytes
- `-B` H = hardware caching
- `-LS` capture disk latency information
- `-F` parameters file
- `-64` use 64-bit memory operations

Reading SQLIO Results

```
sqlio v1.5.SG
using system counter for latency timings, 14318180 counts per second
16 threads writing for 60 secs to file d:\MSSQL\testfile1.dat
    using 8KB random I/Os
    enabling multiple I/Os per thread with 8 outstanding
    buffering set to use hardware disk cache (but not file cache)
using current size: 2048 MB for file: d:\MSSQL\testfile1.dat
initialization done
CUMULATIVE DATA:
throughput metrics:
I/Os/sec: 18527.91
MBs/sec: 144.74
latency metrics:
Min_Latency(ms): 0
Avg_Latency(ms): 6
Max_Latency(ms): 4900
histogram:
ms: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24+
%: 78 6 8 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2
```

Parsing SQLIO Output Into Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	Threads	Operation	Duration	IOSize	IOType	PendingIO	FileSize	IOPS	MBs/Sec	Min_Lat(ms)	Avg_Lat(ms)	Max_Lat(ms)
2	4	Read	120	64	Random	4	36864	1529.14	95.57	0	9	851
3	16	Read	120	64	Random	8	36864	2832.42	177.02	8	44	658
4	64	Read	120	64	Random	8	36864	2825.51	176.59	1	180	965
5	4	Write	120	64	Random	4	36864	2579.65	161.22	0	5	1451
6	16	Write	120	64	Random	8	36864	2487.32	155.45	1	50	1716
7	64	Write	120	64	Random	8	36864	2474.58	154.66	1	206	3008
8	4	Read	120	64	Sequential	4	36864	5117.07	319.81	0	2	587
9	16	Read	120	64	Sequential	8	36864	5483.46	342.71	1	22	506



- <http://www.sqlskills.com/blogs/jonathan/post/Parsing-SQLIO-Output-to-Excel-Charts-using-Regex-in-PowerShell.aspx> (<http://bit.ly/NhPBJ2>)

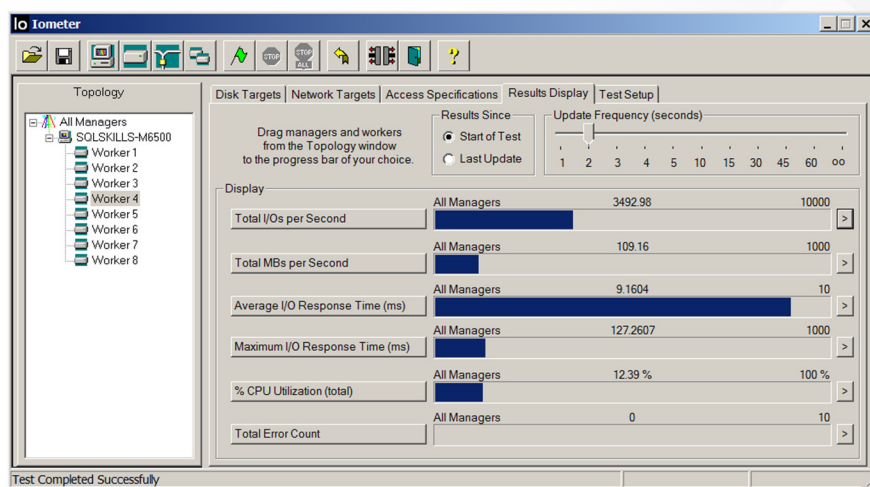
Demo

Using SQLIO

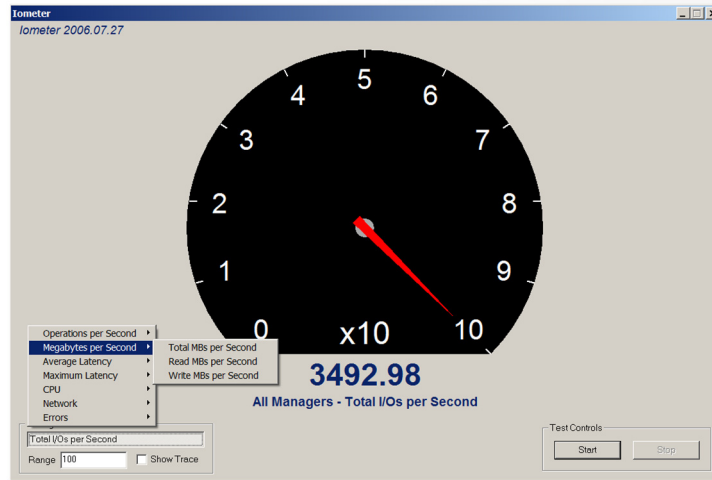
Using Iometer to Benchmark Storage

- **Uses two independent programs for benchmarking**
 - Iometer: graphical user interface
 - Dynamo: load generator
 - Multiple instances of the load generator can be run from multiple nodes to generate higher load on the SAN
 - Can test network I/O to determine how much bandwidth is needed
 - Different workers can perform different I/O tests simultaneously
- **Multi-platform tool that works against RAW disks as well as empty partitions**
 - Test performance of RAW disk and then compare to partitioned performance to check for significant differences
- **Outputs results to CSV file and graphically on screen**

Iometer Dashboard



Iometer Dashboard Meter



Demo

Using Iometer

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 like the MD3000/3000i array which can be attached by serially-attached SCSI (SAS) to two servers redundantly
- 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

Review

- **Defining storage terminology**
 - IOPS, MB/sec, Latency
- **Magnetic vs. Solid State Drives (SSDs)**
- **RAID configurations**
- **Mount points**
- **Partition alignment**
- **Testing with SQLIO and IOMeter**
- **Direct-attached storage**
 - Traditional vs. PCI-Express drives

Questions?