

Database & Disk Configuration With SQL Server

ISSN 1941-7802

TechNote Number: SQL-DISK-005
www.computationpress.com

© 2007 Computation Press, LLC. and/or its Affiliates. All Rights Reserved. Reproduction and distribution of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Computation Press disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although Computation Press's research may discuss legal issues related to the information technology business, Computation Press does not provide legal advice or services and its research should not be construed or used as such. Computation Press shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The opinions expressed herein are subject to change without notice.

Database & Disk Configuration With SQL Server

by
Frank McBath
frankmcb@hotmail.com

Session Objectives

- Session Objective(s):
 - Demonstrate the value of disk optimization
 - Learn how to configure disk drives for optimal performance with SQL Server
- List out best practices for:
 - Monitoring
 - Throughput
 - File Layout
 - Disk Alignment

Presenter Bio

Global Database Technologist for MSFT-ORCL Alliance

Been at Microsoft for 10 years in Consulting, Operations, Technical Sales & Support

Author/Co-Author/Contributor/Presenter

- SQL Server Backup & Recovery, Prentice Hall
- SQL Server 2000 High Availability, Microsoft Press
- SQL Server 2000 Resource Kit, Microsoft Press
- SQL Server 2005 Administrators Companion, Microsoft Press

Database + Disk + Performance Blog

- <http://databasediskperf.blogspot.com/>

Preface: Many Factors Affect Disk I/O Perf

- There are myriad best practices & considerations for optimal disk I/O subsystem performance. Be mindful of all such factors:
 - RAID level
 - File allocation unit size
 - Number, size, & speed of disks
 - Configuration of HBAs & fabric switches
 - Network bandwidth
 - Cache on disk, controllers, & SAN
 - Whether disks are dedicated, shared, or virtualized
 - Bus speed
 - Number of paths from disk I/O subsystem to server
 - Driver versions for all components
 - Stripe size
 - Stripe unit size
 - Workload

Summary First

- Everything else is just supporting data, tools and some “how to” thoughts.
- **Key Points:**
 - Disk are usually the bottleneck
 - Conventional Disk Queue Length is ok, but generally a meaningless metric.
 - Disk response time of at least **20 ms for data** and **10 ms for log** are the **keys to performance**. This isn't really debatable for optimum performance.
- How you get there is what this slide deck is about.

The Myth

- I spent a lot of money. I will get great performance.

The Fact

- Performance is a combination of variables. Money is only one of them. Be prepared to put a lot of sweat into it. Quite often, inexpensive systems can get better performance due to better planning.

The Advice In This Deck

- It's not fruit salad... you can't pick and choose what you like and don't like...
- If you are not getting 20ms data and 10ms log... no amount of query tuning will really help you over time. It's physics.
- Often, disk architecture issues are only fixed by flattening and rebuilding your disk.
- Starting off right is the only answer.

Signs of Bad Performance

- **Obvious:**
 - Users complain
 - Long running queries
 - Blocking
 - Locks held too long
- **Indirect:**
 - Backups taking too long
 - Background jobs taking too long
 - Just not getting throughput you would like

Define Performance

- How many of you have defined a realistic SLA with the users? (*Everyone wants sub-second response...*)
- What can you do to avoid problems?
 - A bad query on any database is going to kill performance. 55G of IO per query is just that.
 - Is this a hard business rule?
 - All users can look at all contacts and sort by last name?
 - Do you have to use case insensitivity?
 - Can you change the code page to dictionary?

Bad Disk Configuration = Poor Scalability

- Anything runs with 5 users by sheer brute force... talk to me about 1000+ concurrently running.
- Poor disk architecture is the leading cause of non-scalability with SQL Server
- Talk to me now, talk to me later...
 - The longer you wait, the worse it gets. It just gets bigger and takes more time to fix.
 - Ex. Backups taking 24 hours... the rolling backup that never stops.
 - Restripping 200G is a lot easier than restripping 1TB.

SQL IO Basics

- The following URL's cover all the essentials for how SQL Server does IO:
 - <http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlIObasics.mspx>
 - <http://blogs.msdn.com/sqlcat/archive/2005/11/17/493944.aspx>

Modern Enterprise Disks Characteristics: My Base Line Disk Drive

- Maxtor Atlas 15K II
 - Ultra320 SCSI
 - Average 3ms, Max 8ms
 - Max Sustained 98 MB/sec
 - Burst 320 MB/sec
 - 8M Cache
 - 15K RPM

http://www.maxtor.com/files/maxtor/en_us/documentation/data_sheets/atlas_15kii_data_sheet.pdf

Why the SAN guys hate you...

- Drives only got bigger, not faster
- You ruin their TCO story.
 - You need 10x72G 15Krpm disk drives (0+1) for a 100G transaction log.
 - That's 620G of wasted space!!!
 - 720G space – 100G t-log = 620G wasted
- They can't "leverage" that BIG investment they got.
 - Leverage = Sharing
 - We don't like to share
- You leave tons of empty space around

Backup Performance Correlation

- Why look at backups?
 - To me, a long running backup can be indicative of larger architecture issues in your database.
- Disk latency can correlate to disk transfer rates...
- Hence, slow backups can be indicative of slow disk (ie. Latency issues).
 - Note, it is possible to have higher throughput on smaller IO size.
 - Ex. 30MB/sec on backup (1MB) and 50MB/sec on index seeks (8K). See with Avg. Disk Bytes/Read.
- The following slides are data points for comparison

Points of Reference

- My desktop
 - Single 10K Western Digital Raptor SATA
 - BACKUP DATABASE successfully processed 68113 pages in 7.373 seconds (**75.679 MB/sec**).
- Good
 - Microsoft OLTP Database 2TB with 72Gb 15K drives Raid10 on SAN:
 - BACKUP DATABASE successfully processed 244138625 pages in 5561.102 seconds (**359.638 MB/sec**).
- Better
 - BACKUP DATABASE successfully processed 302810923 pages in 6010.430 seconds (**412.720 MB/sec**).
- Great
 - BACKUP DATABASE successfully processed 251031988 pages in 3144.803 seconds (**653.921 MB/sec**).
- More detail on these numbers later...
- Now let's look at what I see every day...

Money Doesn't Fix Problems...

Bank in S. America: IBM Shark

- Poor disk configuration. One Big File. One LUN. RAID 5. 200ms disk access times. Single point of failure. 115GB Siebel database.

99 percent backed up.

Processed 14095840 pages for database 'BCCE_CCE', file 'BCC_Data' on file 1.

100 percent backed up.

Processed 9952 pages for database 'BCCE_CCE', file 'BCC_Log' on file 1.

BACKUP DATABASE successfully processed 14105792 pages in 4593.810 seconds (**25.154 MB/sec**).

name	db_size	owner	dbid
BANK_DB	129526.88 MB	SIEBEL	5

name	fileid	filename
BANK_DB_Data	1	E:\mssql2000\mssql\data\BANK_DB.mdf
BANK_DB_Log	2	E:\mssql2000\mssql\data\BANK_DB_log.ldf

Transport Company: NetApps

- Not enough spindles. Not enough HBAs. One big file. Everything on one LUN. RAID 5. 81GB PeopleSoft DB.

90 percent processed.

Processed 9958064 pages for database 'FSPRD', file 'FS88PRD_Data' on file 1.

100 percent processed.

Processed 4312 pages for database 'FSPRD', file 'FS88PRD_Log' on file 1.

BACKUP DATABASE successfully processed 9962376 pages in 1040.035 seconds (**78.470 MB/sec**).

name	fileid	filename	filegroup	size	growth
ABCCLOSE_Data	1	D:\MSSQL\Data\ABCCLOSE_DATA.mdf	PRIMARY	74572288 KB	512000 KB
ABCCLOSE_Log	2	D:\MSSQL\Data\ABCCLOSE_LOG.ldf	NULL	619520 KB	102400 KB
PSFTPRD_Data	1	D:\MSSQL\Data\PSFTPRD.mdf	PRIMARY	88335360 KB	512000 KB
PSFTPRD_Log	2	D:\MSSQL\Data\PSFTPRD_log.ldf	NULL	7544320 KB	102400 KB
Northwind	1	D:\MSSQL\data\northwnd.mdf	PRIMARY	3712 KB	10%
Northwind_log	2	D:\MSSQL\data\northwnd.ldf	NULL	4224 KB	10%
abcde_Data	1	d:\MSSQL\data\abcde_Data.MDF	PRIMARY	1792 KB	10%
abcde_Log	2	d:\MSSQL\data\abcde_Log.LDF	NULL	3456 KB	10%
master	1	D:\MSSQL\data\master.mdf	PRIMARY	25408 KB	10%
mastlog	2	D:\MSSQL\data\mastlog.ldf	NULL	22144 KB	10%
modeldev	1	D:\MSSQL\data\model.mdf	PRIMARY	1280 KB	10%
modellog	2	D:\MSSQL\data\modellog.ldf	NULL	3136 KB	10%
MSDBData	1	D:\MSSQL\data\msdbdata.mdf	PRIMARY	359040 KB	10%
MSDBLog	2	D:\MSSQL\data\msdblog.ldf	NULL	1024640 KB	10%
pubs	1	D:\MSSQL\data\pubs.mdf	PRIMARY	2176 KB	10%
pubs_log	2	D:\MSSQL\data\pubs_log.ldf	NULL	3840 KB	10%
tempdev	1	D:\MSSQL\data\tempdb.mdf	PRIMARY	2004992 KB	153600 KB
templog	2	D:\MSSQL\data\templog.ldf	NULL	61952 KB	10240 KB

Government: EMC DMX & Superdome

- Issue: Concatenated RAID volumes causing sequential IO. 1.6TB Siebel DB.

100 percent backed up.

BACKUP DATABASE successfully processed 202986177
pages in 47070.773 seconds (**35.326 MB/sec**)

Before & After: Fixed

Issue: One big RAID device (0+1). One big file. Everything on one drive. Not enough drives or channels. Starved on disk. Mixed SCSI speeds.

Fix: Put on another DAS and use two channels. Use RAID 5 for data and 0+1 for tempdb and log. Give more disk for data. Use multiple data files.

Cost: \$6,556

Baseline from 10/27/06:

BACKUP DATABASE successfully processed 8259522 pages in 638.561 seconds (**105.960 MB/sec**).

After database work 3/4/07:

BACKUP DATABASE successfully processed 9497102 pages in 313.501 seconds (**248.165 MB/sec**).

Results: **134% increase in throughput performance.**

The Solution

Description	Quantity	Unit price	Sales price
HP Smart Array 642 controller (RAID)	1	\$499.00	\$499.00
-Configurable- HP Modular Smart Array 30 Single Bus	1	\$6,057.00	\$6,057.00
HP Modular Smart Array 30 Single Bus			
Single I/O module			
12 ft VHDCI to VHDCI SCSI cable			
AC power cords with IEC-C13 plug			
Redundant power supply			
Mounting hardware			
Documentation CD			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			
HP 36.4GB Pluggable Ultra320 SCSI 15,000 rpm (1') Universal Hard Drive			

Subtotal:	\$6,556.00
¹(Estimated) Tax:	\$512.49
(UPS Ground) Shipping and handling:	\$56.28
Grand total:	\$7,124.77
²Business lease cost:	\$177.67

RAID Levels

- Why is all this stuff important?
 - HA & Performance
- More vocabulary & concepts...What is RAID?
- Why use it?
- How does one properly use it?
- Reference:
 - <http://www.storagereview.com/guide2000/ref/hdd/perf/raid/levels/single.html>

Common RAID Levels & Why

- Most Common
 - RAID 0
 - RAID 1
 - RAID 0+1
 - RAID 5
 - RAID 6 for SATA
- Pros & Cons

RAID: Number of Disk Usable

- RAID 0
 - All disk used
 - High performance, no protection.
- RAID 1
 - 50% of disk used
- RAID 5
 - $(\text{Number of Drives} - 2) / (\text{Number of Drives})$
 - Good protection, good read performance, fair write performance.
- RAID 6
 - $(\text{Number of Drives} - 2) / (\text{Number of Drives})$
 - Typically used with less reliable and inexpensive SATA drives
- RAID 0+1
 - 50% of disk used
 - Maximum protection and high performance.

RAID: Starved on Spindles

- See how number of disk in RAID set effect pure read throughput.
- HP MSA 60, 4 x 72GB 15K SAS, P800 SmartArray with 512MB cache.
- **RAID 0:** No Parity. All disk used.
 - BACKUP DATABASE successfully processed 15018185 pages in 355.591 seconds (**345.984 MB/sec**).
- **RAID 6:** Two parity disks. Only 2 disk used.
 - BACKUP DATABASE successfully processed 10059329 pages in 395.086 seconds (**208.577 MB/sec**).
- **RAID 5:** One parity disk. Only 3 disk used.
 - BACKUP DATABASE successfully processed 10172177 pages in 295.428 seconds (**282.066 MB/sec**).

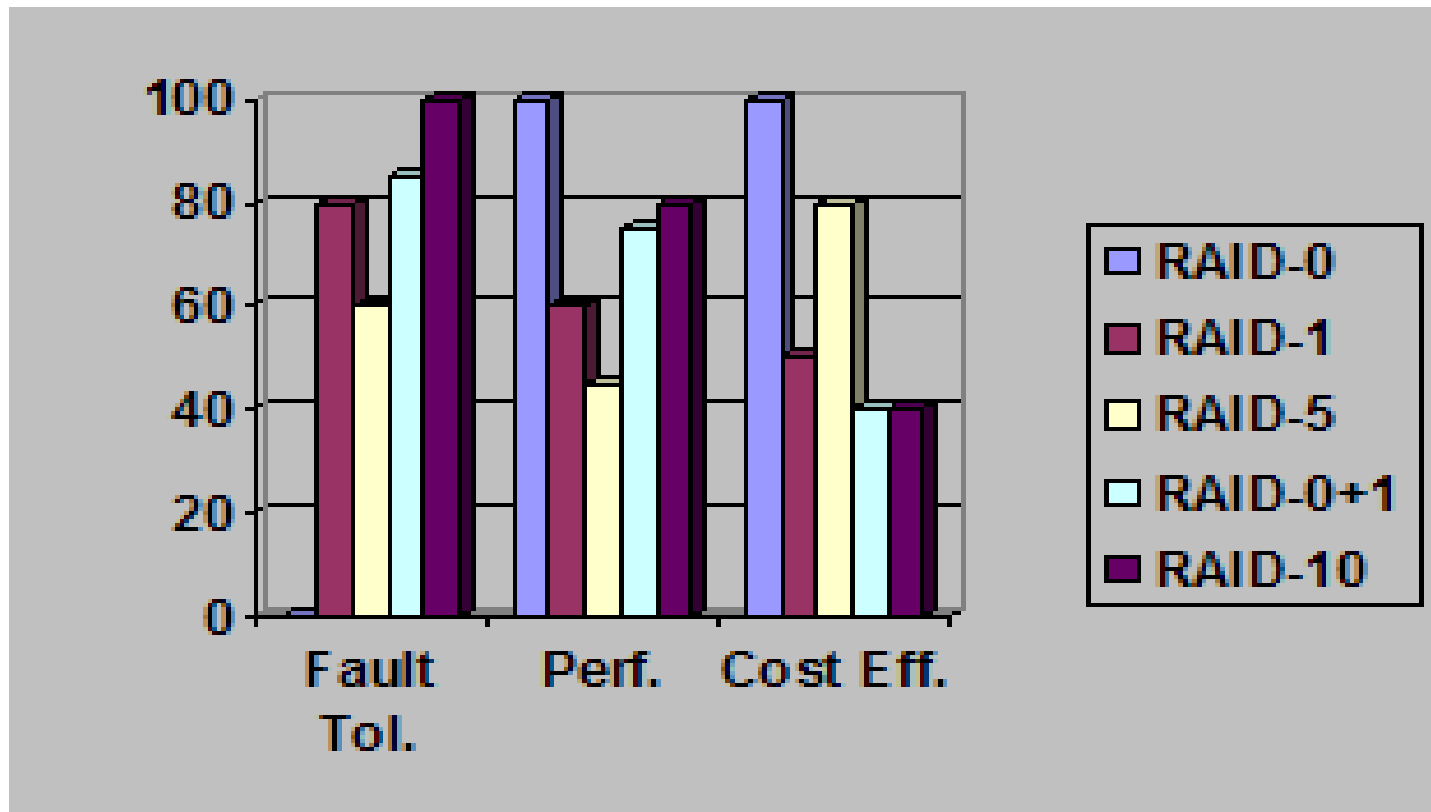
RAID Performance Penalties

- Loss of performance for RAID overhead
- Applies against each disk in the RAID
- The penalties are:
 - RAID-0 = None
 - 1, 0+1, 10 = 20%
 - 2 = 10%
 - 3, 30 = 25%
 - 4 = 33%
 - 5, 50 = 43%

Problems With RAID

- What goes wrong...
 - People slice it wrong
 - Wrong database files on wrong RAID type
 - Parity calculation on write intensive files
 - TEMPDB, Transaction Log
- Starved Resources
 - No one ever got fired for saying “Use RAID 0+1”
 - 95%+ OLTP are SELECT
 - Data files benefit from RAID 5
- Concatenated RAID volumes

The Best RAID For The Job



Basic Important Monitoring Concepts

Latency vs. Queue Length

- Disk Queue Length

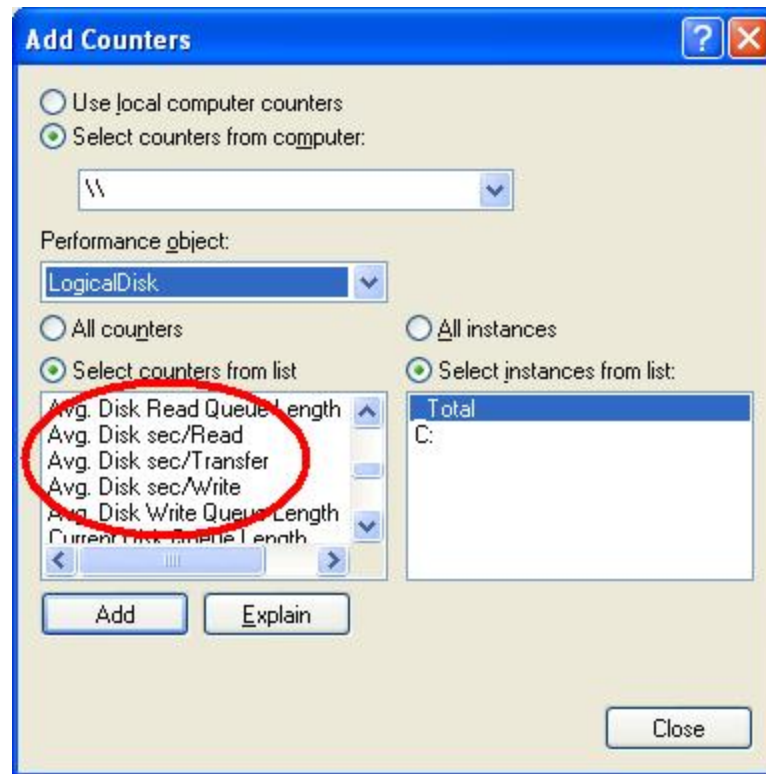
- Queue length is ambiguous without detailed knowledge of the underlying configuration.
 - For example, do I have 3 physical spindles or 30? A queue of 60 may be ok if I have 30 disk drives. Unusable if I have only 3.
- SQL Server with throttle back the transaction log if queue length too big. Hence, queue will never be large and is misleading.

- Disk Transfer Times

- More accurate because at the end of the day, it's speed that counts regardless of the number of physical disk you have.

Perfmon Disk Counters

- Here's where the counters are...



Note: EMC & Veritas use their virtual disk counters

Disk Queue Length (I)

- Traditionally, this is a good measure for finding bottlenecks.
- *Problem is, in a SAN environment, it can be impossible to understand good from bad.*
 - Multiple disks can participate in a stripe/LUN/logical drive letter. And very few people ever seem to know their disk layout.
- How many of you can give me a drive mapping by disk, HBA and switch right now?
 - Without that, queue length is really meaningless to me.

Disk Queue Length (II)

- **Rule of Thumb: More than 2 per physical disk means problems.**
 - For example, a disk queue length of 10 is not bad if you have 5 physical disk behind it...
 - But... if you had a disk queue length of 10 and only 2 physical disks for the F: drive, then you would have a problem.
- Remember that the log will never have more than 3. Again, deceptive counter to watch.

Latency: Targets to Aim For (I)

- Latency (avg. sec/transfer) should not be beyond **20ms for the data** partitions and less than **10ms for transaction log** partitions
 - PERFMON counters to monitor response times
 - Logical Disk -> Avg. Sec/Transfer
 - Logical Disk -> Avg. Disk Queue/Sec
- **More than 10ms and you risk the log backing up.**

READ from Disk: 20ms

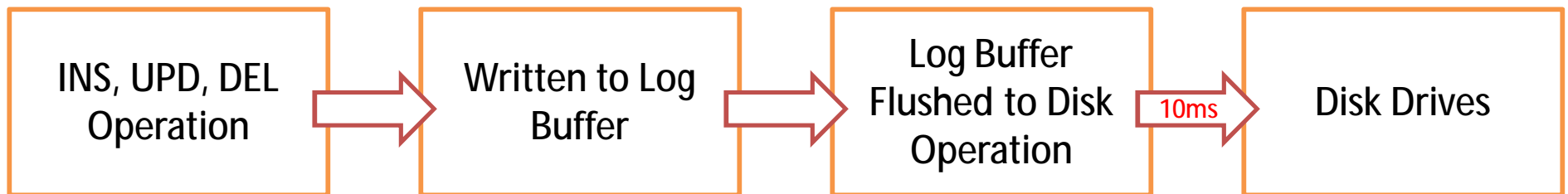


95% to 98% OLTP are SELECT

Disk & The Transaction Log

- Attention: Due to SQL Server's architecture the maximum queue for a dedicated transaction log partition (recommended) will not be higher than 3.
 - Therefore Avg. Disk Queue/sec is not very helpful for analyzing performance issue on the transaction log partition
 - Log stalls hold up transaction commit and can lead to significant concurrency issues
- From SQL Dev:
 - The 32-bit product can have up to 8 writes outstanding; 64-bit, 32 out. *However*, once the first write is issued (perhaps in response to a commit), successive writes will only take place if the (next) 64KB log buffer fills up.

CRUD to Disk: 10ms



2% to 5% OLTP are INSERT, UPDATE, DELETE

Disk Time Outs & Databases

Database Time Outs & What To Do About Them

As of SQL Server 2000 SP4, informational messages are written out to the errorlog which enumerate stalls in the disk. The disk may be a single disk or an entire storage subsystem. The message coming out of SQL Server when you have the issue looks like this:

```
SQL Server has encountered 4507 occurrence(s) of IO requests
taking longer than 15 seconds to complete on file [D:\Program
Files\Microsoft SQL Server\MSSQL\Data\BIGDB_Data.MDF] in
database [BIGDB] (7)
```

While the message is “informational”, the fact is your database is having very serious performance issues.

I wrote a TechNote that talks about how databases get IO issues, what they look like, monitoring, and how to architect a solution to solve them.

<http://www.computationpress.com/>

Disk Design & SQL Server

Basics 101

- The basics still apply even with a SAN
 - Many people think, “Oh, I got a SAN... it will do everything for me...Let’s just drop it all on F:”
- Separate Data, Log, and Tempdb
- Have an HA strategy in place
 - Not just backups, RAID, etc... but also how to put your whole app back together.
- Run your production database in FULL recovery mode
- **Technical Note on Siebel & SQL Server Disk Layout: 14 pages**
 - Technical Note 9: Best Practices for Microsoft SQL Server Disk Layout with Siebel Applications
 - http://72.32.28.196/assets/pdf/TechNote_9_SQL_Server_Disk_Layout_for_Siebel.pdf

Think Big

- Usually the volume and the growth of databases are underestimated.
- Growth rate higher than expected.
- Mid-size companies grow between 2 to 5GB/week in SAP R/3 databases
- Bigger companies see 10 – 12 GB/week (for example, Microsoft)
- Large customers can grow 20 – 30 GB/week
- Never shrink the size of a SQL Server database.

Best Number and Size of SQL Server Data Files (I)

- Performance: Ideal ratio of no. of CPU to no. of data files should be 1:1
 - Dual core processor (AMD Opteron, Intel 'Montecito' are 2 processors)
 - Hyperthreading could be calculated as 2 processors as well
 - Most customers productive run 1 : 2 ratio with great performance
 - No need to change currently good running system to achieve this ratio
 - Keep in mind for new installations

Best Number and Size of SQL Server Data Files (II)

- Other aspects:
 - Available storage hardware like # of arrays or partitions and amount of storage space
 - Estimated database growth during production
- Huge data files can create problems in handling (setting up sandbox systems, copying, etc)
- Too many data files can increase monitoring overhead
- Most customers between 4 and 24 data files

Number of Files for TempDB

- To ensure high concurrency for TempDB, make sure that the number of files (equal size) are equal to the number of CPUs in the SQL Server.
 - <http://support.microsoft.com/kb/328551/>

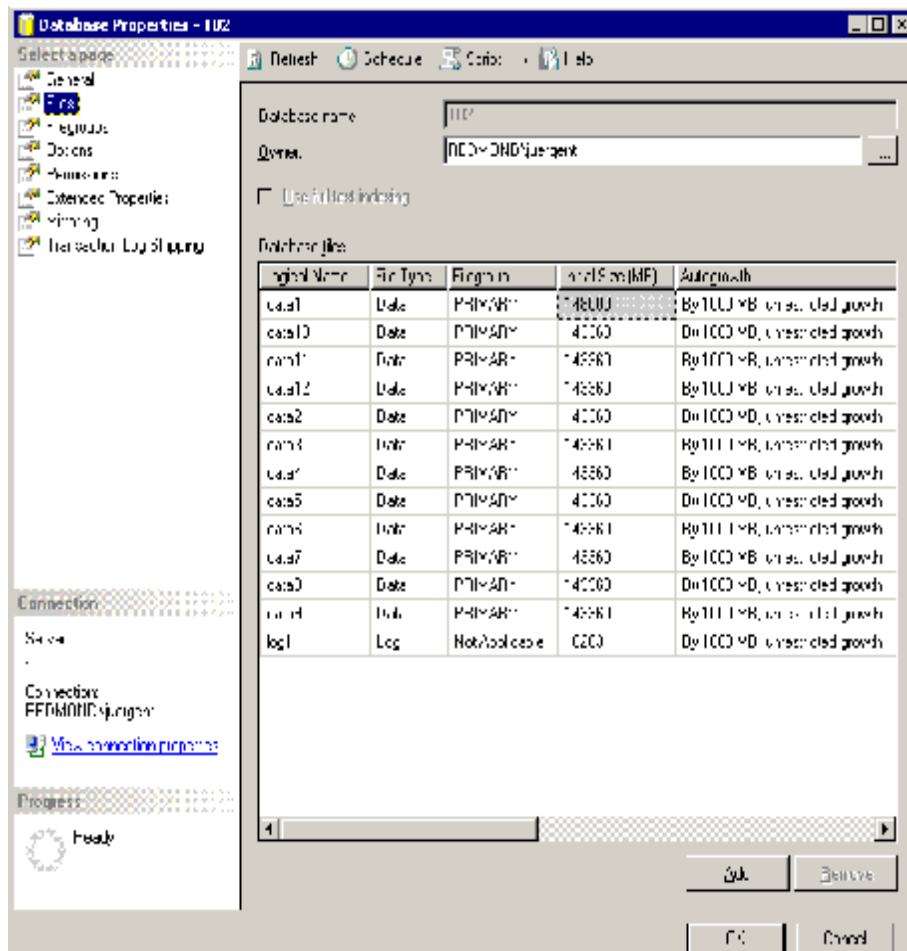
SQL Server and tempdb

- Different products stress tempdb very differently. For example, read uncommitted snapshot isolation vs. OLTP.
- SAP R/3: Moderate usage of tempdb
 - Some joins, little aggregation, small sorts
- SAP BW: Heavy usage of tempdb
 - Big hash joins, huge aggregations, large sorts
- Siebel:
 - Big sorts, complicated joins

SAP & SQL Server: tempdb

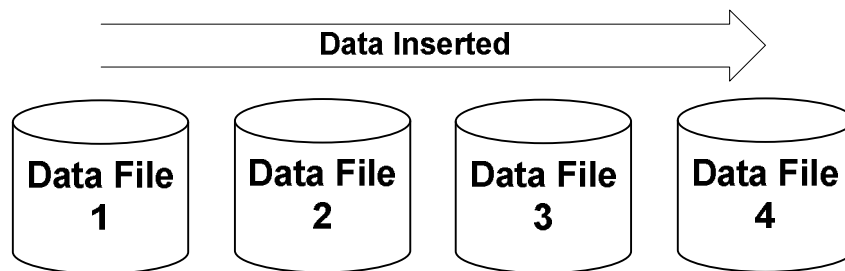
- SAP R/3:
 - 1-2 GB usually enough
 - Several to one dozen disk drives is usually good enough.
- SAP BW:
 - 2 times the space of the largest table. For example, fact table or ODS/DSO object.
 - 1:1 number of tempdb data files to BW datafiles.

How many Log files for databases?



- Only one log file makes sense, since the log is written strictly sequentially
- The initial size of the transaction log should be 5GB for a productive installation
- To avoid the creation of too many 'Virtual Log Files' the 'autogrow' feature should be set to 50%

SQL Server Proportional Fill Feature



- Data is added round robin to the data files and is proportionally spread even across them. This is to avoid hot spots.
- Despite the 'autogrow' setting these data files should be grown manually
- If one data file fills up 'autogrow' will be disabled and the data files will be filled sequentially

Limitations of Storage Adapters

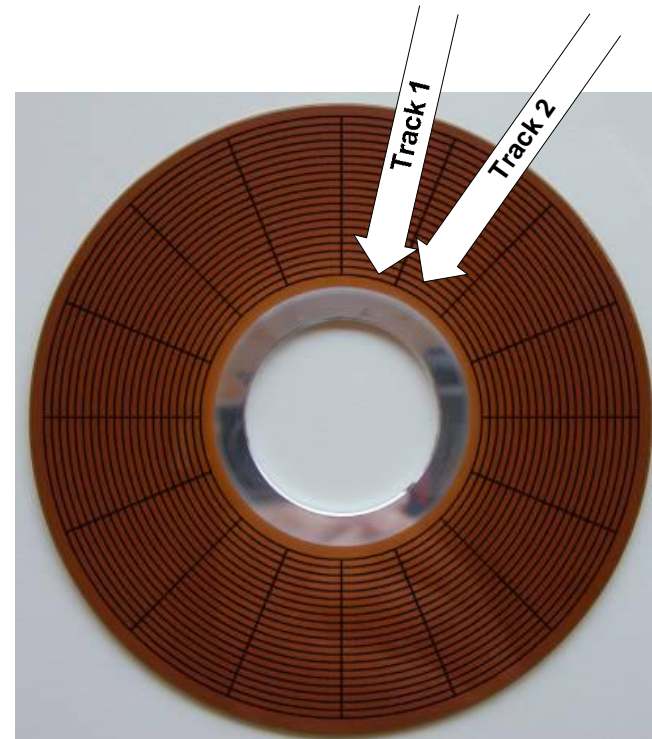
- Remember: 200 Sequential IO/s per second of SQL Server can be handled by one disk drive
 - Translates into 12.8 MB/sec
- You will need 20 disk drives for heavy sequential I/O.
 - Translates into 256 MB/sec
 - Requires about 4x1GBit HBAs
 - Requires about 2x2GBit HBAs
 - At least one, preferably 2, Ultra SCSI320 adapters or channels.
 - For example, an HP SmartArray Controller may have 4 channels on one card.

Disk... The Grungy Details

Disk Geometry

- In this picture, you can conceptually see the disk laid out as sectors and tracks.
- Due to misalignment, data at the end of one track overlaps onto the next track.
- Thus, 1 read request is split over 2 tracks causing 2 IO's.

Data is Split Over 2 Tracks Causing 2 IO's
for 1 Request



<http://www.storagereview.com/guide2000/ref/hdd/geom/tracks.html>

The Problem:

Windows has 63 Hidden Sectors

- Output of diskpar (Windows 2000 Resource Kit)

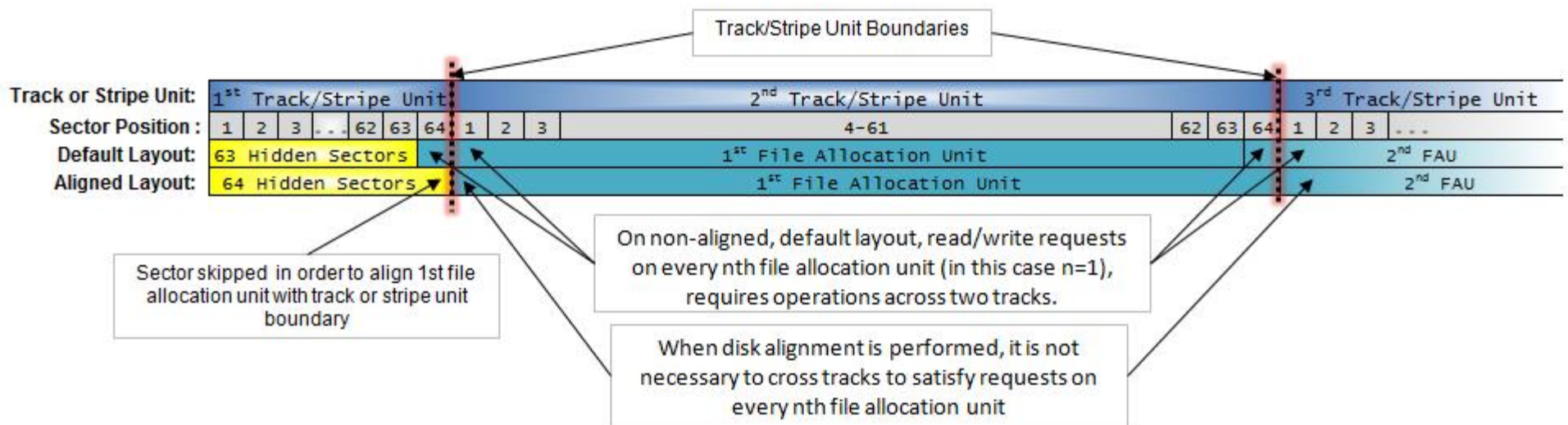
```
C:\>diskpar -i 0
---- Drive 0 Geometry Infomation ----
Cylinders = 12161
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 100027630080 (Bytes) = 95393 (MB)
---- Drive Partition 0 Infomation ----
StatringOffset = 32256
PartitionLength = 49319424
HiddenSectors = 63
PartitionNumber = 1
PartitionType = de
```

- By default, for years Windows instantiated 63 hidden sectors in all new partitions.
- These hidden sectors contain the master boot record (MBR).
- Note the typos:
 - "StatringOffset" instead of "StartingOffset".
 - "Infomation" instead of "Information"

The Problem: 63 Sectors < 64 Sectors

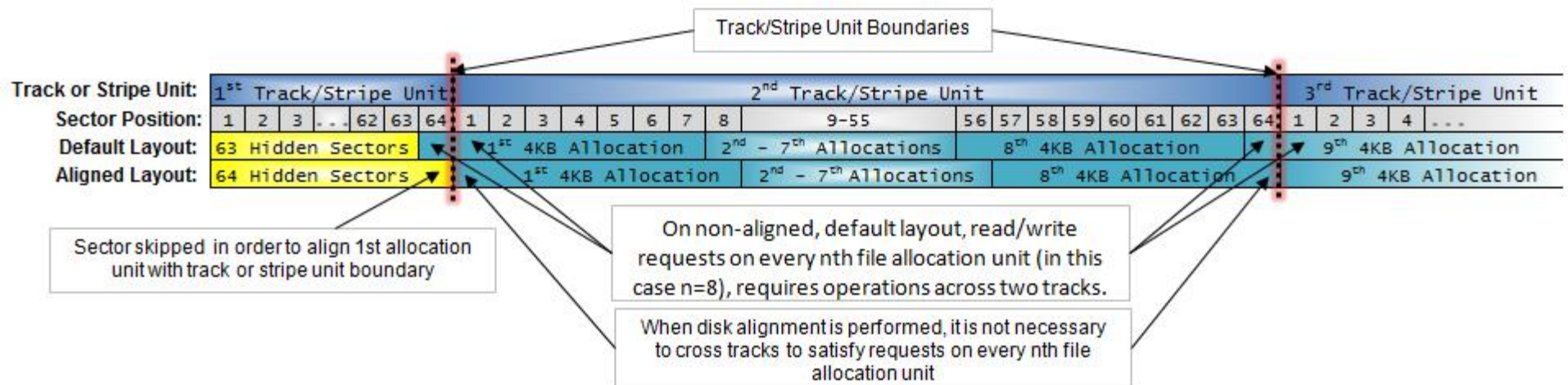
- Classically there were 64 sectors per track.
- So on the inner first track we're one shy.
- When Windows writes the first chunk of data to a new partition:
 - It writes the first 512B to the 64th sector of the first track.
 - It writes the rest to the second track.
- This single *write* request requires access of both tracks.
- To *read* this data requires accessing both tracks.
- Partition alignment remedies this situation.
- Let's look at a picture...

Partition Alignment Graphic: Default vs. Optimized for SQL Server



- *This is a very simplified graphic*
- Senior Technology Architect
 - The worst scenario? Random operations using 64K IO and 64K chunk size. One sector off and you are hitting two disks for every IO thus halving the random performance potential.
- *Note: On a RAID array this means accessing two different stripe units on two separate disks.*

Partition Alignment Graphic: Default vs. Aligned



- *This is a very simplified graphic*
- This *very simplified* graphic corresponds to the default NTFS file allocation unit of 4KB.

Source: Jimmy May

Everyone Recommends It, Exchange, Oracle, HP, EMC... Anything That Is Disk Intensive...

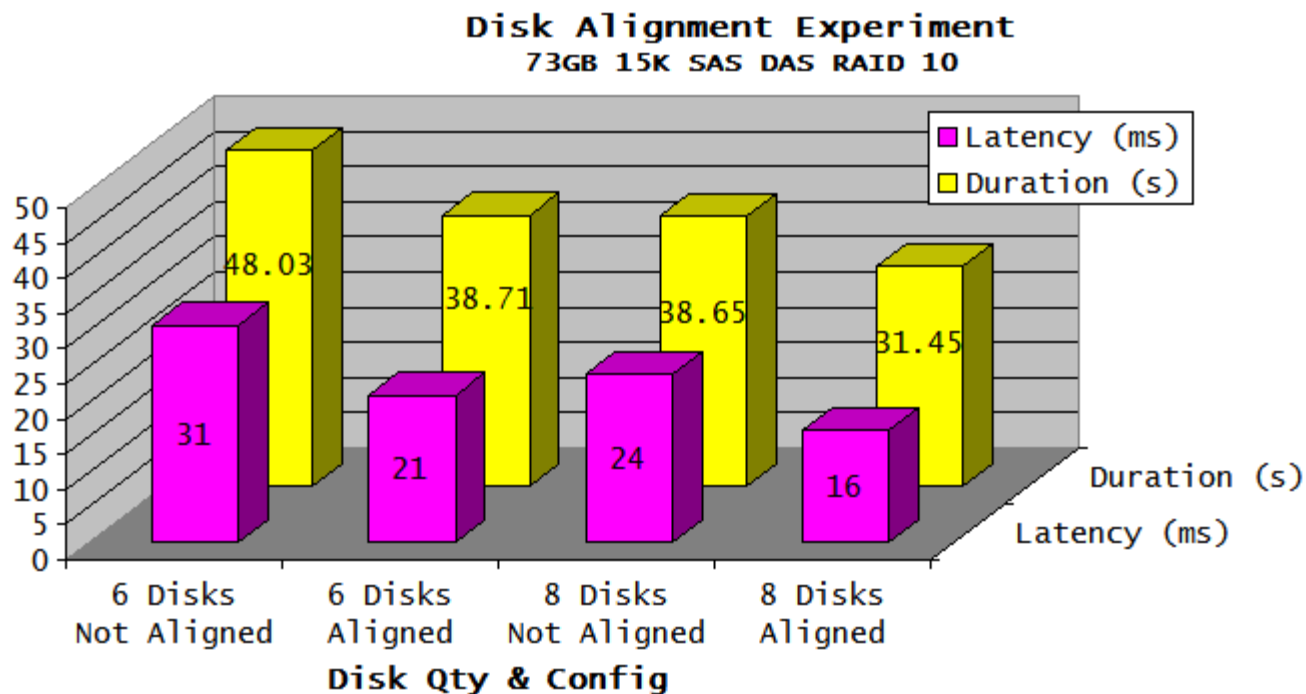
- How Disk Drives Work (geometry, etc...):
- <http://www.storagereview.com/guide2000/ref/hdd/hist.html>
- Microsoft:
- <http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/pdpliobp.msp?pf=true>
- http://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/en-us/Default.asp?url=/resources/documentation/Windows/2000/server/reskit/en-us/prork/pree_exa_oori.asp
- <http://www.microsoft.com/technet/prodtechnol/exchange/2003/library/optimizestorage.msp>
- <http://www.microsoft.com/resources/documentation/windows/2000/professional/reskit/en-us/part6/proch30.msp>
- <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/diskpart.msp>
- <http://support.microsoft.com/default.aspx?scid=kb;en-us;q300415>
- 3rd Party:
- <http://h71019.www7.hp.com/ActiveAnswers/downloads/Exchange2003EVA5000PerformanceWhitePaper.doc>
- http://www.emc.com/pdf/techlib/exchange_2000_symmetrix.pdf
 - See Page 21
- http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=PSD_OI040301_CW02&prodTypeId=12169&prodSeriesId=377751&locale=en_US
- <http://seer.support.veritas.com/docs/268701.htm>

Tools

- An updated version of the Disk Partition tool for Windows Server 2003 is available
 - <http://support.microsoft.com/default.aspx?scid=kb;en-us;923076&sd=rss&spid=3198>
 - “Disk alignment is a required optimization and must be applied by OEMs during Setup. Disk alignment provides a significant increase in system performance. Failure to perform disk alignment can decrease performance by 10 to 15 percent in RAID array systems.”

Performance Impact

- Disk Alignment Experiment
 - Latency & Duration on RAID 10, 64KB file allocation unit
 - 6 disks vs. 8 disks
 - Not Aligned vs. Aligned
- 6 *aligned* disks performed as well as 8 *non-aligned* disks.
- Thus, efficiencies of ~30% were achieved.



Three Best Practices Together: Disk Alignment + 64KB NTFS+ RAID 10

- Rebuild entire disk I/O subsystem compliant w/ 3 best practices.
- Results:
 - Serial throughput: 31%
 - Random Write latency: 200%-250%
 - Random Read latency: 10%
 - Sequential Write IOPs by File: 120%
 - Random Write IOPs by File: 10-50%
 - Random Write MB/s: 100%
 - Sequential Write MB/s: 140%

Good News / Bad News

- Good news:
 - Partition alignment is simple to perform.
- Bad news:
 - Partition alignment must be done prior to disks being formatted.
- This is great if you have a new SAN.
- But it might be painful to convert large amounts of existing data on misaligned partitions.
 - LUN migration tools provided by SAN vendor
 - But IO is still IO...

Partition Alignment Template

```
diskpart
list disk
select disk <DiskNumber>
create partition primary align=<Offset_in_KB>
assign letter=<DriveLetter>
format fs=<file-system> label =<"label ">
    unit=<FileAllocationUnitSize> nowait
```

- Note: If necessary, use `diskmgmt.msc` to map partition numbers to drive letters.
- Note: The `nowait` option forces the command to return immediately while the format continues.
- Note: The format command from within `diskpart` & from the command line are syntactically distinct.

Partition Alignment Example

```
C: \di skpart
```

```
Microsoft DiskPart version 6.0.6000
```

```
On computer: ASPIRINGGEEK
```

```
DISKPART> list disk
```

```
DISKPART> select disk 3
```

```
DISKPART> create partition primary align=1024
```

```
DISKPART> assign letter=E
```

```
DISKPART> format fs=ntfs unit=64K
```

```
label="MyFastDisk" nowait
```

Common Partition Offsets

- The classic, default *misaligned* offset
 - 32,256 bytes (*less than 32KB*)
- The following is a common re-defined offset—but it doesn't always correlate well with stripe size
 - 32,768 bytes (*exactly 32KB*)
 - *More on this later!*
- The following are common *usually valid* aligned offsets
 - 65,536 bytes (*exactly 64KB*)
 - 131,172 bytes (*exactly 128KB*)
 - 1,073,741,824 bytes (*exactly 1MB*)

Checking The Alignment

- WMI
- WMI C

WMI Script

- Save the following text as GetPartitionOffsets.vbs

```
' GetPartiti onOffsets. vbs
strComputer = "."
Set obj WMI Servi ce = GetObject ("wimngmts: \\" & strComputer &
"\root\CIMV2")
Set col Items = obj WMI Servi ce. ExecQuery( _
"SELECT * FROM Wi n32_Di skPartiti on", , 48)
' Wscri pt. Echo "-----"
Wscri pt. Echo "Wi n32_Di skPartiti on instance"
For Each obj Item in col Items
    Wscri pt. Echo "Di skIndex: " & obj Item. Di skIndex & " -- Name: "
    & obj Item. Name & " -- StartingOffset: " &
    obj Item. Start ingOffset
Next
```

- Execute via command line

```
C: \>cscript GetPartiti onOffsets. vbs
```

WMI Script: Output

- Execute via command line

```
C: \>cscript GetPartitionOffsets.vbs
```

```
Microsoft (R) Windows Script Host Version 5.6  
Copyright (C) Microsoft Corporation 1996-2001.  
All rights reserved.
```

```
Win32_DiskPartition instance
```

```
DiskIndex: 0 -- Name: Disk #0, Partition #0  
-- StartingOffset: 32256
```

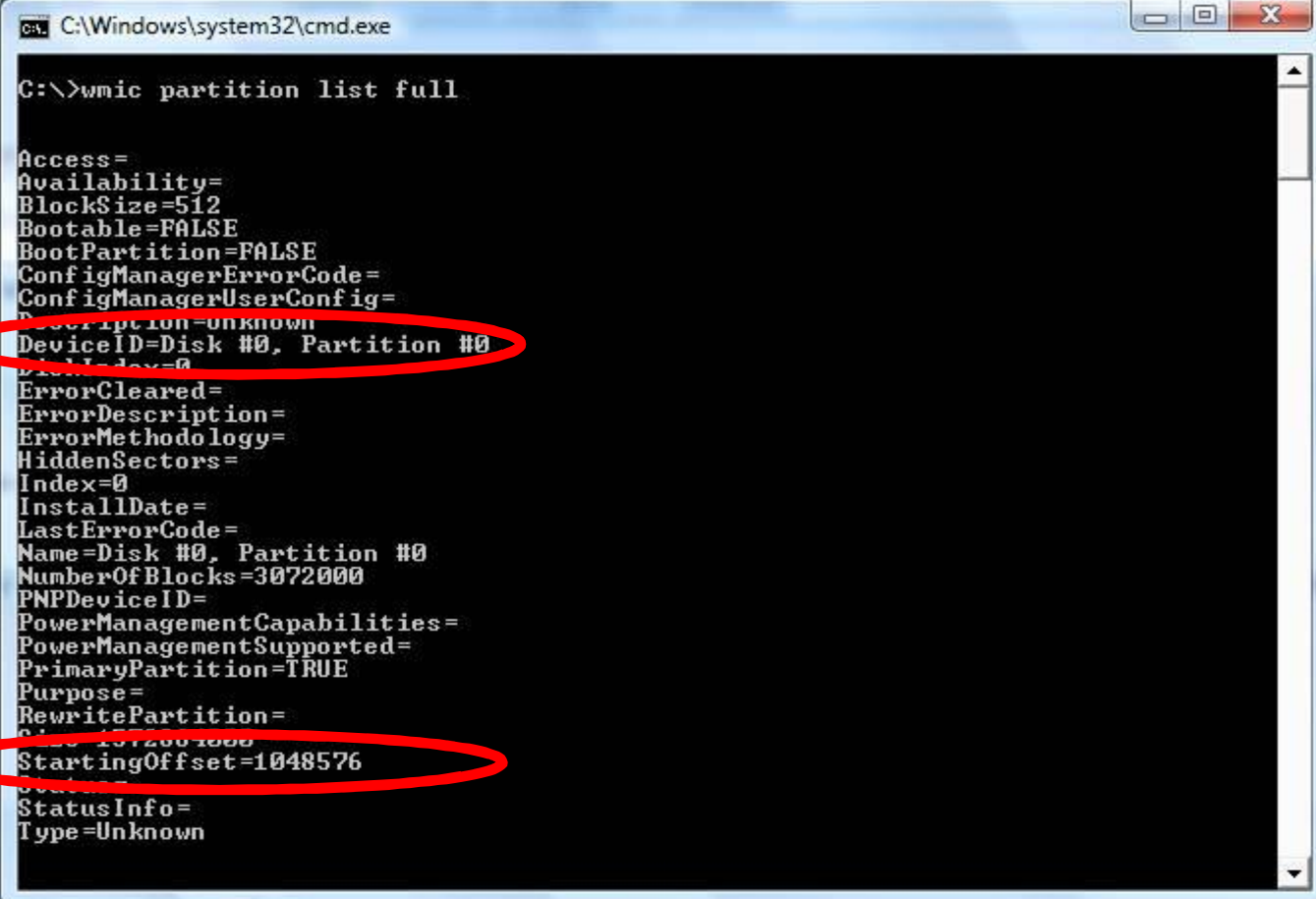
```
DiskIndex: 1 -- Name: Disk #1, Partition #0  
-- StartingOffset: 32256
```

```
DiskIndex: 2 -- Name: Disk #2, Partition #0  
-- StartingOffset: 1048576
```

WMIC

- Execute via command line

C:\>wmic partition list full



```
C:\Windows\system32\cmd.exe

C:\>wmic partition list full

Access=
Availability=
BlockSize=512
Bootable=FALSE
BootPartition=FALSE
ConfigManagerErrorCode=
ConfigManagerUserConfig=
Description=unknown
DeviceID=Disk #0, Partition #0
ErrorCleared=
ErrorDescription=
ErrorMethodology=
HiddenSectors=
Index=0
InstallDate=
LastErrorCode=
Name=Disk #0, Partition #0
NumberOfBlocks=3072000
PNPDeviceID=
PowerManagementCapabilities=
PowerManagementSupported=
PrimaryPartition=TRUE
Purpose=
RewritePartition=
StartingOffset=1048576
StatusInfo=
Type=Unknown
```

Getting the file allocation unit size

- `fsutil fsinfo ntfsinfo c:`
- **Default** NTFS 4KB format

```
C: \>fsutil fsinfo ntfsinfo c:
NTFS Volume Serial Number :          0x3a16ff9d16ff5879
Version :                             3.1
Number Sectors :                       0x000000000a2397ff
Total Clusters :                       0x00000000014472ff
Free Clusters :                        0x000000000025b76a
Total Reserved :                       0x00000000000051f0
Bytes Per Sector :                     512
Bytes Per Cluster :                    4096
Bytes Per FileRecord Segment :         1024
Clusters Per FileRecord Segment :      0
Mft Valid Data Length :                0x0000000007c90000
...
```

Getting the file allocation unit size

- `fsutil fsinfo ntfsinfo e:`
- You need it to be NTFS 64KB format

```
C:\>fsutil fsinfo ntfsinfo s:
```

```
NTFS Volume Serial Number :      0x328e659b8e6557fd
Version :                          3.1
Number Sectors :                    0x0000000012a187ff
Total Clusters :                    0x00000000025430f
Free Clusters :                     0x000000000253da8
Total Reserved :                    0x0000000000000000
Bytes Per Sector :                  512
Bytes Per Cluster :                 65536
Bytes Per FileRecord Segment :      1024
Clusters Per FileRecord Segment :   0
Mft Valid Data Length :             0x0000000000010000
...
```

IMPORTANT: Correlation of Partition Offset & Stripe Unit Size

- The following formula must result in an integer value:

$$\text{Partition_Offset} \div \text{Stripe_Unit_Size}$$

- Note: Ask your SAN Admin for Stripe Unit Size
 - Example A
 - Partition Offset = 32KB
 - Stripe Unit Size = 64KB
 - $32\text{KB} / 64\text{KB} = 0.5$
 - *Even though the offset was changed from the default 31.5KB, the partition is still misaligned!*
 - Example B
 - Partition Offset = 64KB
 - Stripe Unit Size = 64KB
 - $64\text{KB} / 64\text{KB} = 1.0$
 - *Good!*

Another Correlation: Stripe Unit Size & File Allocation Unit Size

- The stripe unit size must correlate with the file allocation unit size
 - *Here, too, an integral value is required*
 - Example A
 - Stripe Unit Size = 64KB
 - File Allocation Unit Size = 64KB
 - $64/64 = 1.0$
 - *Good!*
 - Example B
 - Stripe Unit Size = 256KB
 - File Allocation Unit Size = 64KB
 - $256/64 = 4.0$
 - *Good!*
- This alignment is usually not a problem.
- The problem is that the separate *partition alignment* must be done first.

Summary Steps for Alignment

- Be mindful of the important relationships between
 - Partition Offset
 - Stripe Unit Size
 - File Allocation Unit Size
- The following is a summary of the steps required:
 - Partition Offset must be explicitly re-defined using `diskpart` from the default 31.5KB to an appropriate value such as 64KB, 128KB, or 1MB
 - The result of the following must *both* result in integer values:
$$\text{Partition_Offset} \div \text{Stripe_Unit_Size}$$
$$\text{Stripe_Unit_Size} \div \text{File_Allocation_Unit_Size}$$

Help is on the Way: Vista & Windows Server 2008

- Partition alignment done by default
 - Windows Vista
 - Windows Server 2008
 - The default for disks over 4GB is 1MB.
 - The setting is found here:

HKLM\SYSTEM\CurrentControlSet\Services\VDS\Alignment

- Note: Pre-existing partitions won't automatically be aligned after an upgrade to Windows Server 2008—they must be done manually—just like existing partitions today.

Questions?

