Jimmy May's Blog

SQL Server Performance, Best Practices, & Productivity

Disk Partition Alignment (Sector Alignment) for SQL Server: Part 4: Essentials (Cheat Sheet)

Jimmy May 4 Dec 2008 10:40 AM 12

The purpose of this post is to document Disk Partition Alignment Essentials. It is intended for engineers who are already familiar with disk partition alignment yet want a "cheat sheet".

As most of you know, partition alignment is an essential best practice. We are seeing I/O enhancements in the lab & at important real-life customer sites of 25% - 40% measured by a variety of metrics. Your mileage may vary. It will be sometime before Windows Server 2008 is ubiquitous & existing partitions are re-built. In the meantime, disk partition alignment will remain a relevant technology

Disk Partition Alignment is an essential foundation of high I/O performance. It must be performed on partitions prior to formatting & addition of user data. Existing partitions must be re-built; Data on existing volumes must be backed up, then restored after alignment & re-formatting.

Additional details are available in a previous post:
Disk Partition Alignment (Sector Alignment) for SQL Server: Part 1: Slide Deck
http://blogs.msdn.com/jimmymay/archive/2008/10/14/disk-partition-alignment-for-sql-server-slide-deck.aspx
In addition, stay tuned for the white paper sponsored by SQL CAT at www.sqlcat.com.

Attached Document
The attached doc is a compressed Word.doc version of this post.

Applies to:
Partition alignment is important for all categories of disks:

- MBR basic
- MBR dynamic
- GPT basic
- GPT dynamic

Three Values, Two Essential Correlations

Perform these calculations for each partition which must result in *integer* values:

```
Partition_Offset ÷ Stripe_Unit_Size
Stripe_Unit_Size ÷ File_Allocation_Unit_Size
```

Of the two, the first is far more important. Use the information below to divine this information.

Stripe Unit Size

Windows cannot reliably report stripe unit size. For local storage & DAS, vendor utilities should be able to provide the information. Otherwise, talk to your SAN man (or woman) to get the stripe unit size.

File Allocation Unit Size

Run this command for each drive to get the file allocation unit size:

```
fsutil fsinfo ntfsinfo c:
fsutil fsinfo ntfsinfo d:
```

etc…

Values should be 65536 bytes (64KB) for partitions on which SQL Server data or log files reside.

Starting Partition Offset Analysis for Existing Partitions

Basic Disks

Run this command to get the starting offsets for *basic disks*:

```
wmic partition get BlockSize, StartingOffset, Name, Index
```

wmic Output

```
C:\>wmic partition get BlockSize, StartingOffset, Name, Index
BlockSize  Index  Name                   StartingOffset
512        0      Disk #0, Partition #0  1048576
512        1      Disk #0, Partition #1  53688139776
```

```
512        2      Disk #0, Partition #2  161062322176
512        0      Disk #1, Partition #0  65536
512        0      Disk #2, Partition #0  32256
```

Dynamic Volumes

Analyzing basic dynamic disks is not quite as straightforward.

- The `wmic` command is NOT VALID for dynamic disks.
- Analysis for Windows dynamic & 3rd party dynamic disks is different.

Windows Dynamic Volumes

- The status of windows dynamic volumes requires dmdiag with the `-v` option: `dmdiag -v`
- The `-v` option generates two sections *if* Windows dynamic volumes are present:

`dmdiag -v` **Section**	**Column**
`---------- Dynamic Disk Information -----------`	`Rel Sec`
`---------- LDM Volume Information -----------`	`Rel Sectors column`

- Be mindful that the output is in units of sectors & requires conversion to bytes for purposes of the correlations cited above.

`dmdiag -v` **Partial Output**

```
---------- Dynamic Disk Information -----------
 DiskGroup: S0029170Dg0
  Group-ID: e60175bf-47ce-45e0-b725-9bcf73cc2a43
   ...<some columns redacted for brevity>...
   Sub Disk  Rel Sec   Tot Sec   Vol Type  DevName
   ========  =======   =======   ========  =========
   Disk1-01  128       16384     Simple    Harddisk4
   Disk1-02  16512     33527208  Spanned   Harddisk4
etc…

---------- LDM Volume Information -----------
   ...<some columns redacted for brevity>...
   Volume   Volume  Size     Total    Rel      Vol     Plex
   Name     Type    Sectors  Size     Sectors  State   State
   ======   ======  =======  =======  =======  ======  ======
   Volume1  Simple  16384    16384    128      ACTIVE  ACTIVE
   Volume2  Simple  16384    16384    128      ACTIVE  ACTIVE
etc…
```

- Even in the absence of the `-v` switch, the output is fairly verbose. There are many sections which purport to report starting partition offset. As I understand it, the Logical Disk Manager (LDM) "spoofs" tools built for basic disks so they won't overwrite dynamic volumes. *The output of these sections is NOT to be trusted for dynamic volumes*. For example, the following reveals the classic default misaligned value of 32,256 bytes. However, the Partition Type 0x42 betrays that this is a dynamic volume & thus the starting partition offset information in this section is NOT reliable:

```
::: !!!UNRELIABLE for Starting Offsets of Dynamic Volumes!!! :::
---------- Partition Table Info Disk 1 ----------
...<redacted for brevity>...
       Starting       Partition Hidden  Partition Partition
  Offset (bytes)   Length (bytes) Sectors Number    Type (HEX)
          32,256  268,431,980,544      63 0         0x42
               0                0       0 1         0x00
               0                0       0 2         0x00
               0                0       0 3         0x00
...<redacted for brevity>...
```

MBR 3rd Party Dynamic Volumes

Dynamic volumes created by 3rd party vendors may not be properly interpreted by dmdiag -v. For example, Veritas dynamic disks require their proprietary tools to determine whether existing volumes are aligned.

GPT basic & dynamic

I haven't tested this on GPT disks and look forward to doing so.

Implementation: DISKPART Example

Here's an example of performing partition alignment, providing a drive letter, & initiating an async format:

```
C:\diskpart
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
```

Note that the format command is available only in Windows Server 2008 & Vista.

Microsoft Documentation

This the mother of all articles on disk partition alignment. It's by PFE Robert Smith:

Disk performance may be slower than expected when you use multiple disks in Windows Server 2003, in Windows XP, and in Windows 2000
http://support.microsoft.com/kb/929491
This is the best white paper I've seen on this topic. It's by SQL CAT member Mike Ruthruff. It is mercifully brief &, as our friends at Guinness might say, simply "Brilliant!":
Predeployment I/O Best Practices
SQL Server Best Practices Article
http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/pdpliobp.mspx
Cited above, my deck:
Disk Partition Alignment (Sector Alignment) for SQL Server: Part 1: Slide Deck
http://blogs.msdn.com/jimmymay/archive/2008/10/14/disk-partition-alignment-for-sql-server-slide-deck.aspx
Again, stay tuned for the white paper from SQL CAT.

Acknowledgements

Among the many engineers who have assisted, my special thanks to Microsoft Sr. Dev Lead Deborah Jones for her help in the divination of dynamic volumes.

Administrivia

Jimmy May, MCDBA, MCSE, MCITP: DBA + DB Dev | Senior Performance Consultant: SQL Server
A.C.E.: Assessment Consulting & Engineering Services
http://blogs.msdn.com/jimmymay
Performance is paramount: Asking users to wait is like asking them to leave.

Disk Partition Alignment (Sector Alignment) for SQL Server - Part 4 - Essentials (Cheat Sheet) -- Jimmy May.zip

Comments

SQL Server and Cloud Links for the Week | Brent Ozar - SQL Server DBA

5 Dec 2008 8:01 AM

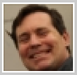
PingBack from http://www.brentozar.com/archive/2008/12/sql-server-and-cloud-links-for-the-week-3/

Larry Chesnut

5 Jan 2009 10:43 PM

Jimmy, this is a great discussion, but I am stumped by one data point. You showed us how to obtain the Partition_Offset and File_Allocation_Unit_Size, but you did not show us how to find the Stripe_Unit_Size.

Could you post something on how to do this?



Jimmy May

6 Jan 2009 9:57 PM

@Larry Chesnut

Larry, for DAS, you can usually use utilities installed on the server to which the disks are attached. For SAN, Windows cannot offer a reliable way to provide the stripe unit size. Thus you're theoretically at the mercy of your SAN admin.

However, it's not as bad as it sounds. Most contemporary implementations likely have stripe unit sizes which are compatible with a 1024KB (1MB) offset & a 64KB file allocation unit. For example, how often would we run into anything less than a 64KB stripe unit size? And in my experience all stripe unit sizes larger than 64KB are always multiples thereof.

In fact, one of the reasons that Windows Server 2008 implements a 1024KB starting partition offset is because of its durablity in terms of likelihood of "getting it right". That is a starting partition offset of 1024KB is very likely to correllate well with common values for stripe unit size (& file allocation unit size):

Partition_Offset ÷ Stripe_Unit_Size

Stripe_Unit_Size ÷ File_Allocation_Unit_Size

Specifically 1024KB starting partition offset correlates with stripe unit sizes of 64KB, 128KB, 256KB, 512KB, & 1024KB.

An exception might be some HP installations I've run into. I'm currently collaborating with an HP architect & will let you know what we determine.

It turns out that much vendor documentation is contradictory on this topic. However, many vendors unequivocally support disk partition alignment as a best practice, e.g., EMC, Hitachi, Dell, & Veritas.

If you gather any before-&-after metrics, I'd be interested in seeing them. I'm also available to for performance reviews, implementation, etc.

Stay tuned for the white paper I've written for SQL CAT which is in final review.

Let me know if you have any questions, & good luck!

Jimmy May, Aspiring Geek

Kendal Van Dyke
18 Feb 2009 10:12 PM

Really good stuff, thanks for putting this together. If you're curious, I ran a variety of tests against local storage and DASD and reported the results in a 7 part series here: http://kendalvandyke.blogspot.com/2009/02/disk-performance-hands-on-series.html

Jimmy May, Aspiring Geek: SQL Server Performance, Best Practices, Productivity, etc.<br><img src="http://img156.imageshack.us/img156/6808/xparentacelogoli1.gif" border="0"/>
11 Mar 2009 9:34 AM

My multi-talented, java-enabled yet Windows-empowered friend Alik Levin of www.practicethis.com (among

Frank
15 Apr 2009 8:28 AM

Hi Jimmy,

One question I've always had, but never knew where to ask. Your site seems to be the most complete reference for this topic so I'll give it a shot here:

If your partition offset is 1024KB-aligned, but your allocation unit size is only 64KB, then wouldn't your files eventually end up only 64KB-aligned anyway?

My thinking is that when you create a file, Windows doesn't care about the partition offset. All it does is finds some free clusters and allocates them to a file. Even after the file is created, any file extensions can be allocated anywhere on the partition. The only thing keeping these files or "extents" aligned is the 64KB allocation unit size. I looked at APIs such as CreateFile for anything to tell Windows how to align the file, but there is nothing like that.

So I guess my question is, what is the point of aligning the partition to 1024KB?

Jimmy May, Aspiring Geek: SQL Server Performance, Best Practices, Productivity, etc.<br><img src="http://img156.imageshack.us/img156/6808/xparentacelogoli1.gif" border="0"/>
8 May 2009 2:15 PM

I recently collaborated with Microsoft PFE Daniel Janik to create a template to make the case for disk

Alexander Gladchenko

12 May 2009 3:53 AM

Недавно Кевин Кляйн в очередной раз поднял тему выравнивания размеров кластера и блока, проблему, которая

Gregory Chernis
25 Jun 2009 8:54 AM

http://serverfault.com/questions/31304/how-to-create-dynamic-volume-in-stripe-configuration-aligned-to-1024

Thank you for starting the topic. With a ton of help from Evan Anderson, I scratched the surface of what happens with dynamic disks on Server 2003 SP2.

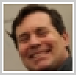
LarryVB
5 Aug 2009 12:42 PM

Jimmy, I am trying to get my head around all this and have initiated some testing with some disappointing results. I have created a new filegroup on a drive that has the starting offset and cluster size set to 64k. On this filegroup I placed a table with 82 million records (about 90GB) and then performed several operations of insert, delete and alter on about 5k records. Please note this drive is the only drive connected to this server with non-default values, all the other drives are still at the set-up default values. This new drive with this new filegroup has no other files on it. This scenario performs 14% slower than the drives with the defualt set-up values.

In a nutshell, with everything else being equal (RAID, fragmentation, disk type and speed, etc.) the defualt out-performs the drive with sector alignment and the cluster set at 64k.

I might have expected no gain, but certainly not a degraded scenario.

Any thoughts?



Jimmy May
5 Aug 2009 1:36 PM

@Frank: I just saw your comment, I'll have to address it later. But I'm wondering if you're confusing file allocation unit size & starting partition offset. Please contact me: jimmymay at microsoft dot com.

@LarryVB:

Speaking of apples & oranges, have you actually done the very same experiements without alignment on that drive?

By drive, do you mean a RAID array? Or literally a single piece of h/w?

Is this on a SAN? What mfg & model?

I could ask numerous other questions... Please reply to my email with details & I'll post the results of our discussion.

Karl
25 Mar 2010 10:40 AM

ISA 2006 with Advanced Logging (MSDE) on a 2nd physical disk should benefit from partition alignment, too. Thanks for the 'cheat sheet' guide!