第7章 windows相关

磁盘分区对齐详解与配置 - Windows篇

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-14*

许多系统管理员可能不曾听过"磁盘分区对齐"之说，甚至一些有经验的存储管理员对分区对齐也不甚了解。磁盘分区不对齐现象是什么，为什么会造成比较严重的性能下降？相反，配置正确的分区起始位置（Offset）设置会使存储系统发挥更大的性能潜力。

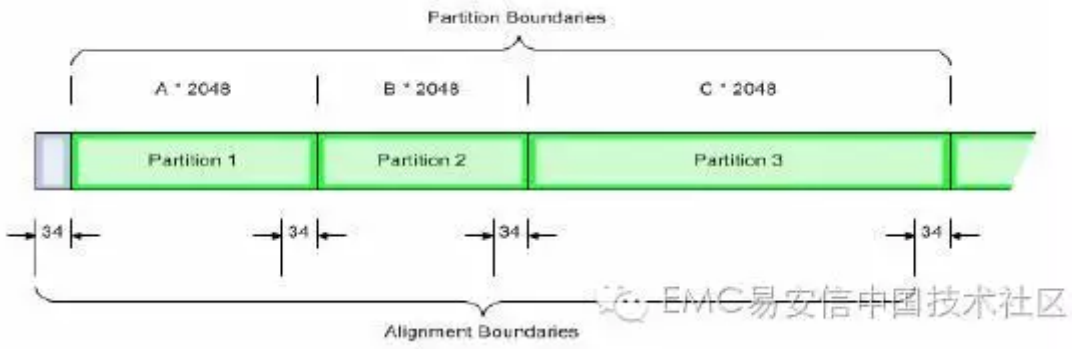
文章就磁盘分区对齐进行的介绍，并且给出了在Windows平台上如何配置的方法。

什么是磁盘分区对齐（Disk Alignment、Partition Alignment）：

Windows的磁盘有一种结构叫做Master Boot Record(MBR), 它的默认大小为63个Block（每个大小为512字节）。它的存在使得磁盘的初始位置和的磁盘上第一个分区的初始位置有63个Block的错位。如果磁盘的单个Track大于63个Block的话。这就会导致默认的初始的位置是从第64个开始。使文件系统的中的Track和位于磁盘中的两个Track之上。这种不对齐现象会导致存储系统的性能下降，原因是单个I/O请求会跨越多个磁盘上的Track，从而导致存储系统的额外性能开销。特别是对于一些随机I/O比较较大的应用程序，影响将更大。

而对于Windows 2003以后支持的GPT Disk，也会存在磁盘分区不对齐的现象，但是结构有所不同。如图1所示（图中单位为Block，512字节），所有的分区由1MB大小（2048 Block）构成，第一个分区从LBA 34开始，即17KB大小位置。这也就意味着所有的分区会有17KB的不对齐的情况发生。同样会导致I/O读写性能影响。

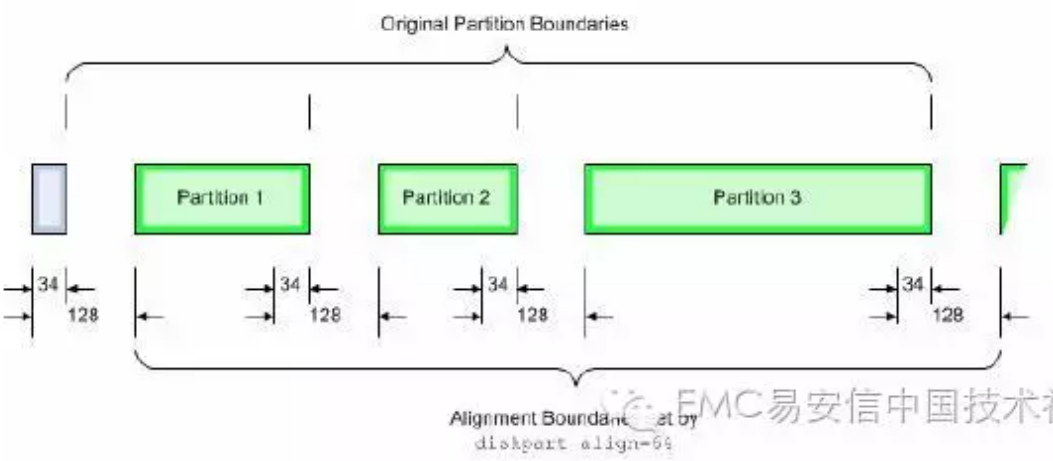
图1



Windows磁盘分区对齐配置：

配置磁盘分区对齐后，如下图所示四个分区的例子中，对比图1中不对称的情况，图2中这些Windows的分区结束地址与图1中的地址位置一致了。

图2



磁盘分区对齐操作，只针对Windows的Basic Disk。对于Windows 2008和Windows 2008 R2，无需对磁盘进行对齐操作，因为这个过程已经在操作系统划分分区的时候自动进行了，软件的升级还是给管理员们带来了不少便利的。对于Windows 2003和 2003 R2，以及Windows 2000，建议设置开始偏移量（starting offset）为64KB（128 block）。另外，对于Dynamic Disk类型，不能进行磁盘分区对齐操作。不过，如果原来的Basic Disk上已经进行了对齐操作的分区，会在转换后保留原来的对齐。

查看Windows磁盘分区对齐方法：

\1. 查看Basic Disk的分区对齐信息：

方法1：

在Windows的命令行下输入Diskpart命令。

C:>diskpart

选择对应的Disk

DISKPART> list disk （显示本机所有磁盘）

DISKPART> Select Disk X （x代表上面显示的从0开始的磁盘编号）

DISKPART> list partition （显示从1开始的所有的分区信息，在最右边有一个Offset/偏移量的值，如果它是8的倍数，说明你的硬盘分区是对齐的，如果不是，说明你的磁盘分区没有对齐）

方法2：

使用WMIC命令，在Windows命令行下输入下列命令，命令输出的列StartingOffset为该分区的偏移量，该数值除以512则为扇区数目：

wmic partition get BlockSize, StartingOffset, Name, Index

2.查看Dynamic Disk的对齐信息：

使用dmdiag工具，下载地址；

执行从命令行执行dmdiag.exe –v

在命令行输出中，LDM Volume区域的RelSec列，该列显示的就是Dynamic Disk的起始扇区。

磁盘分区对齐配置方法：

对于Windows 2003 sp1以下版本，使用dispar命令来设置偏移量和分区对齐。步骤如下：

1. 安装在Windows Resource Kit后，在Performance Tools目录下，通过命令行运行diskpar命令：
2. 使用diskpar –s N命令，下面的例子中定义了配置一个20GB的磁盘的错位64KB的配置

```
> diskpar -s 2
Set partition can only be done on a raw device.
You can use Disk Manager to delete all existing partitions.
Are you sure drive 2 is a raw device without any partition? (Y/N) y

----Drive 2 Geometry Information ----
Cylinders = 2612
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 21484431360 (Bytes) = 20489 (MB)

We are going to set the new disk partition.
All data on this drive will be lost. Continue (Y/N)? y

Please specify starting offset (in sectors): 128
Please specify partition length (in MB) (Max = 20489): 20489
```


对于Windows 2003 sp1以上版本，dispar命令被dispart命令所替换。

1. 在CMD命令行运行Diskpart命令

   C:>diskpart
2. 选择对应的Disk

   DISKPART> Select Disk X

3. 创建分区设置偏移量为64KB（128 Block），下面的例子为创建一个1GB的分区。

   DISKPART> create partition primary size=1024

最后，进行分区对齐操作的时候需要注意：对齐操作需要在磁盘上写入数据之前完成，最好在磁盘刚刚映射到主机时进行。磁盘对齐操作必然损坏磁盘上的数据，所以如果有数据需先备份，操作的时候注意数据安全。

Windows Perfmon与Linux IOstat存储性能工具

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-21*

Windows的Perfmon和*nux的IOStat是操作系统自带的存储性能状态检测的工具。它们能收集绝大多数性能指标，文章给出了两个工具输出的参数详解。操作系统自带的工具也派大用场。

Windows Perfmon:

Perfmon的全称是Windows Peformance Monitor（“Perfmon”），是Windows操作系统自带一个性能监控工具，可以用来发现操作系统是否存在I/O瓶颈。基本上每个管理员都会用它。虽然他的功能强大，涵盖得面也比较广。不过其中参数众多，可能管理员在选择特定性能指标的时候无从下手。本文是着重在磁盘性能的性能参数入手，介绍一下Perform针对PhysicalDisk的部分。

Perfmon中和磁盘性能有关的计数器有以下几种：

% idle time

% disk time

% disk read time

% disk write time

Disk Bytes/sec

Disk Transfers/sec

Disk Reads/sec

Disk Writes/sec

Disk Read Bytes/sec

Disk Write Bytes/sec

Avg. disk sec/read

Avg. disk sec/write

Avg. disk sec/transfer

Avg. disk bytes/transfer

Avg. Disk Queue Length

Avg. disk read queue length

Avg. disk write queue length

Current Disk Queue Length

其中比较重要的参数有：

**
**

% Disk Time

% Disk Time说明在性能数据采样区间内，磁盘处于做读写状态的百分比。一个没有I/O压力的磁盘，这个值应该小于100%。理论上应该是% Disk Read Time 和% Disk Write Time这两个值的和。但是有时候这个值会远大于100%，不太准确。作为一个趋势线，供管理员参考。

% Idle Time

% Idle Time说明在采样区间内，磁盘处于空闲状态的百分比。相对于经常大于100%的% Disk Time，这个值却是准的。当磁盘处于空闲状态时，它的值是100%。当磁盘在满负荷做操作时，它的值是0%。所以我们可以通过这个值，反推出% Disk Time的真实值。

Disk Bytes/sec

Disk Bytes/sec指明每秒钟磁盘作的读和写的数量，以Bytes为单位。它是Disk Read Bytes/sec 和 Disk Write Bytes/sec的和。管理员可以根据链接的存储系统的磁盘速度和Raid类型，看看是否达到理论上限。关于各种磁盘的参考速度，可以参考《如何计算IOPS》

Avg. disk sec/transfer

磁盘每做一次读或者写的动作所花的平均时间。如果后端存储的缓存可用且没有压力，这个值理论上会比较低。

Avg. disk sec/read

磁盘每做一次读操作所花的平均时间。后端存储Read Hit的比例会影响到这个数值

Avg. disk sec/write

磁盘每做一次写操作所花的平均时间一样。存储系统的缓存会影响到这个数值给一些参考值，这个是一个通用的界定，管理员还需要针对自己的应用要求来衡量：

好: < 10 ms、 还能接受: 10ms ~ 20ms 、比较差：20ms ~ 50m、很差: > 50ms

Avg. Disk Queue Length

在一个时间点上，磁盘队列的长度，也就是发出的磁盘操作正在等待被磁盘处理的请求数目。例如前端应用发出一条读的请求，但是目标磁盘当时正在处理其他任务（大文件读写之类的）。那么这个新的读请求就会被放在磁盘队列里。这时候磁盘队列的值就是1。理论上讲，这个值不应该长时间地大于2，如果是的话，说明磁盘的相应速度已经很慢了，前端应用程序肯定会受到影响。

Linux IOStat：

IOStat *nux的系统命令用来收集操作系统存储端的输入输出相关的统计信息。简单易用，遇到存储相关问题的时候管理员最想想到的就是这个命令，它支持本地磁盘的同时也支持mount到操作系统的网络文件系统，例如NFS。

IOstat的基本输出包括以下几项(单位都是KB)：

Tps：指该磁盘设备上每秒的传输次数，这和存储系统的IOPS有点不同，根据不同的应用，每个IO的小大是不一样的，有可能多个逻辑请求会被合并成一个文件系统的IO。

kB_read/s: 每秒从该磁盘设备上读取的数据量

kB_wrtn/s: 每秒写入的数据量

kB_read：命令运行的区间内读取数据总量

kB_wrtn：命令运行的区间内写入数据总量

如果使用-X参数，iostat会输出更多信息

rrqm/s：每秒这个设备相关的读取请求有多少被合并了，例如文件系统发现不同的读取请求读取的是相同Block的数据，文件会将这个请求合并；

wrqm/s：每秒这个设备相关的写入请求有多少被Merge了。

rsec/s：每秒所读取的扇区数

wsec/：每秒所写入的扇区数。

r/s：每秒读取请求的数目；

w/s：每秒写入请求的数目；

await：每一个IO请求的处理的平均时间（单位是微秒毫秒）。类似于前文中介绍的Avg. Disk Queue Length，可以理解为IO的响应时间。

%util：在统计时间内所有处理IO时间，除以总共统计时间。和Windows中的% Idle Time。

检查存储性能的过程不能单单靠一个指标确定问题所在，管理员要综合考虑，某一个数值高并不代表肯定有存储性能问题存在。以Windows Perfmon为例，在磁盘负荷高峰的时候，可能会看到Avg. disk sec/transfer比较高的现象。但是这个时候Disk Bytes/sec是不是也很高就比较关键了。如果Disk Bytes/sec已经接近了磁盘的最高上限，那么这个问题就更加是一个磁盘超负荷的问题。如果Disk Bytes/sec不算很高，就应该考虑的是为什么磁盘的速度不理想，可以往后端存储的方向查找问题所在。调整Raid保护级别，存储系统内部的资源分配，创建条带等都是解决方案。

Windows存储管理之磁盘类型简介

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-16*

各种操作系统连接到存储系统之后，并且操作系统识别物理磁盘之后，需要对磁盘进行进一步配置。如果用户连接存储是的Windows Server，存储管理员势必需要了解Windows中的磁盘类型与文件系统。笔者从存储的角度总结了Windows磁盘与分区类型的特点，并对其内容进行介绍与分析，帮助管理员更好的了解Windows主机连接的存储和文件系统。

Windows的磁盘类型:

Windows的物理磁盘类型分为两种：

Basic Disk（基本磁盘） – 这个类型的物理磁盘可以被MS-DOS和所有的Windows操作系统所访问。Basic Disk可以包括最多四个主分区（Primary Partition），或者是三个主分区和一个扩展分区（Extended Partition）的逻辑磁盘（Logical Disk）。Basic不支持容错功能，可以在MBR和GPT创建磁盘。

Dynamic Disk （动态磁盘）– Dynamic Disk提供一些Basic Disk没有的功能，比如过将一个逻辑卷扩展到多个物理磁盘之上。Dynamic Disk使用隐藏的数据库来维护位物理磁盘上的动态卷。如果用户需要扩展一个逻辑磁盘到多个物理磁盘，需要使用Windows Disk Management和Diskpart.ext工具先将Basic disk转换为Dynamic Disk。Dynamic Disk支持在线创建（需要重启）和在线扩展逻辑卷。多份的元数据存储在磁盘中。简化管理，可以使用软Raid功能，Mirror、Spanned等等。

Windows磁盘的分区类型：

谈到磁盘结构，很有必要了解一下下面两个概念MBR和GPT。

MBR （Master Boot Record）物理磁盘上第一个扇区（Sector）,也叫作主引导扇区，是计算机卡机后访问磁盘时说必须要读取的首个扇区，它位于柱面0，磁头0，扇区1。Windows的MBR磁盘被分割成多个连续的区域叫做分区（Partition），每个分区的的信息都存储在MBR，即磁盘的首个扇区中，在MBR中定义了分区的起始位置和长度。只有一个主分区可以处于激活状态，且支持操作系统启动。

GPT（GUID Partition Table）一种由基于 Itanium 计算机中的可扩展固件接口 (EFI) 使用的磁盘分区架构。与主启动记录(MBR) 分区方法相比，GPT 具有更多的优点，因为它允许每个磁盘有多达128 个分区，理论上支持最大 18 EB卷大小，允许将主磁盘分区表和备份磁盘分区表用于冗余，还支持唯一的磁盘和分区ID (GUID)。 GPT是在windows使用大容量磁盘的选择。

下表是MBR和GPT对应的Windows操作系统信息：

	MBR	**GPT**
Windows操作系统版本	MS-DOS所有Windows版本	Windows 2003以上版本
硬件支持	32位CPU	64位CPU
最大支持单个逻辑卷	2TB	256TB
分区表拷贝数	一份	Primary和Backup两份分区表，支持checksum
最大支持分区数目	4个主分区或者3个主分区和一个扩展分区	128个分区
数据存储位置	存储在分区中	存储在分区，关键的Platform数据存储在对用户隐藏的分区中

下表为Basic和Dynamic Disk支持的Volume类型（MBR磁盘类型）：

Basic	**Dynamic**	**Volume类型**
支持	支持	Simple Volume
支持	支持	Spanned Volume
支持	支持	Striped Volume（Raid-0）
支持	支持	RAID-5 Volume
支持	支持	Mirrored Volume（Raid-1）

下表为Basic和Dynamic Disk支持的Volume类型（GPT磁盘类型），可以看到需要在Windows中实现软件Raid，需要将磁盘类型转换为Dynamic才可以。

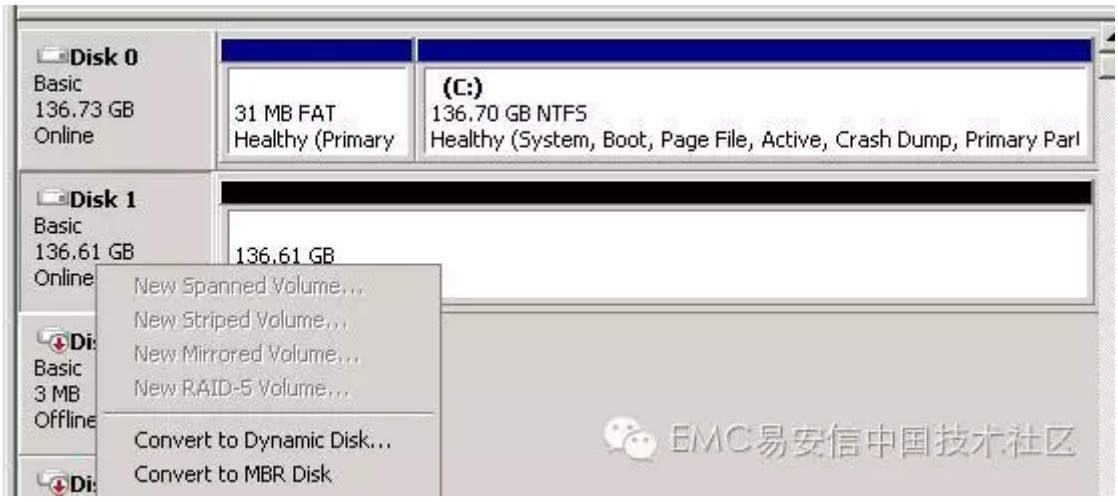
Basic	Dynamic	Volume类型
支持	支持	Simple Volume
	支持	Spanned Volume
	支持	Striped Volume（Raid-0）
	支持	RAID-5 Volume
	支持	Mirrored Volume（Raid-1）

总结：

在Windows连接到存储之后，选择选择GPT分区类型与Dynamic Disk可试Windows存储管理灵活性性所有提高，GPT分区格式抛开了MBR最大2TB的容量限制，支持在线扩展，各种优势。动态磁盘不受分区数目限制。启用GPT和Dynamic Disk的方式很简单。

在Windows Server 2008/2008R2中，Server Manager – Storage – Disk Management – 右键需要转换的磁盘，如下图：

在Windows Server 2003/2003 R2中，Computer Management – Disk Management – 右键需要转换的磁盘，如下图：



Windows存储管理之磁盘结构详解

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-17*

在之前一篇《Windows存储管理之磁盘类型简介》中介绍了Windows的基本磁盘类型。本篇中我们将对GPT和MBR这两类磁盘类型的结构进行深入介绍。

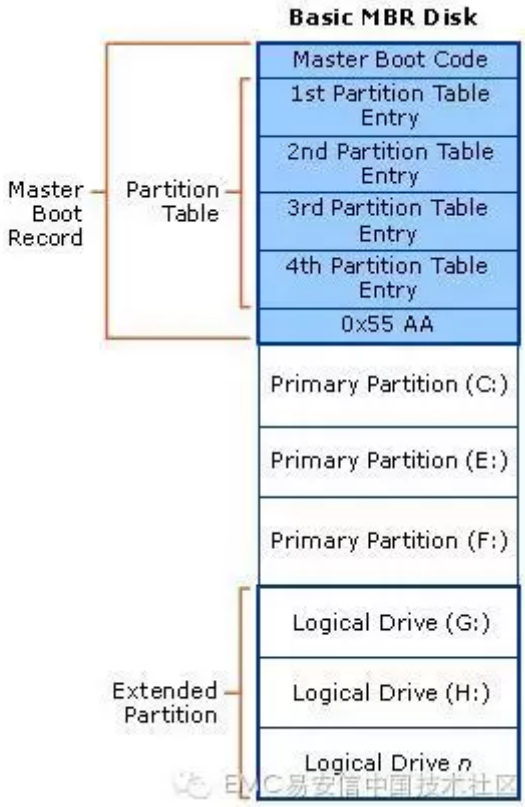
Windows磁盘结构:

Windows的主流磁盘结构分为MBR和GPT两种。MBR是早期Windows的唯一选择，但是随着物理磁盘的容量不断增大。GPT结构成为目前的主流，最大支持超过2TB的容量，提供容错，多分区支持，比MBR来的更加强大。

MBR （Master Boot Record ）磁盘结构:

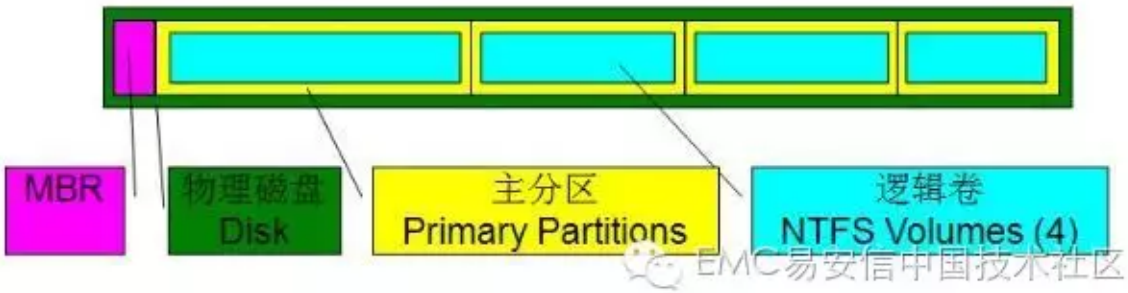
在Basic MBR Disk中的MBR中包含了几种信息。

- Bootstrap Code – 也叫Master boot code，它是一段可执行的代码，主要作用是，扫描活动分区的分区表，寻找活动分区的开始扇区，加载启动扇区到内存等功能。
- Disk Signature - Windows的所有物理磁盘都有一个磁盘签名的机制，如果没有签名windows则不能访问该磁盘的数据。当Windows新扫描到一个物理磁盘，尝试写入一些磁盘签名，用来标识这个磁盘。签名的长度为8个字节。然后会写入到第一个扇区，位置为0x01B8 - 0x01BB。签名存储在注册表的HKLM\SYSTEM\MountedDevices位置。
- Partition Table – 分区表，一个64字节的数据结构用来定义每个分区的起始位置。每个分区定义去大小为16个字节。因为这个设计，所以MBR的的扩展主分区最多只能支持4个。

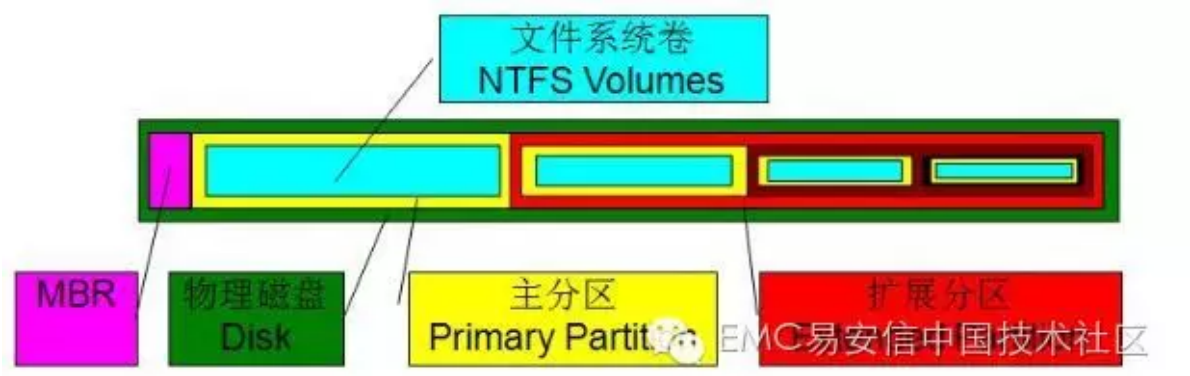


一个简单的Basic （MBR）的磁盘结构如下图所示，我可以看到最外层的是物理磁盘（Disk），在物理磁盘的最前端包含MBR（Master Boot Record），这个例子中，定义了一个分区和NTFS逻辑卷。

MBR Disk支持最大四个主分区（Primary Patition），如果创建多个主分区的，则结构如下。一个物理磁盘中包含四个主分区，每个主分区包含一个文件卷。



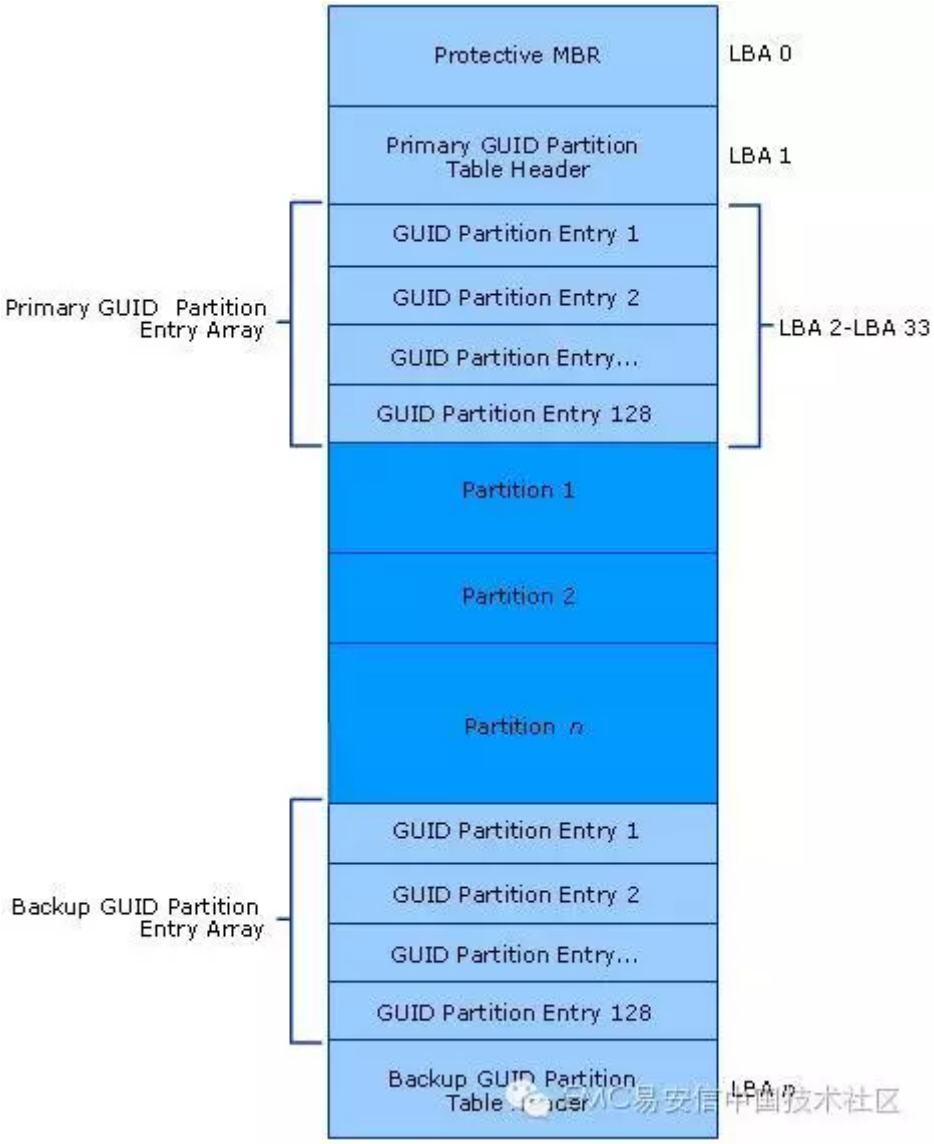
如果启用了扩展分区，则在扩展分区中可以包含多个逻辑卷。



GPT （GUID - Globally Unique Identifer）磁盘结构：

截止201年，大多数操作系统都支持GPT GPT Disk在主的MBR中包含几个内容，GPT Disk的分区表包括以下几个内容。在MBR硬盘中，分区信息直接存储于主引导记录(MBR)中（主引导记录中还存储着系统的引导程序）。但在GPT硬盘中，分区表的位置信息储存在GPT头中。但出于兼容性考虑，硬盘的第一个扇区仍然用作MBR，之后才是GPT头。

- Protective MBR - 和MBR在Partition Table中包含主分区信息不同的是，GPT Disk在磁盘的第一个扇区（Sector）为“Protective MBR”，它位于LBA0（通用的存储寻址方式大小为512每单位）这个位置上。它包含的内容为磁盘的分区信息和初始的BIO启动器。这是为了兼容性的考虑，保证一些遗留的MBR磁盘工具可以识别到GPT Disk。
- Partition Table Header – 分区表头定义了一些磁盘上可使用的块，同时还定义了组成分区表的Partition Entries数目和大小（大小通常为128个字节）。GPT支持的64位版本的Windows Server 2003以上版本，支持创建最大128个分区，每个分区记录大小为128个字节。在分区表头中还记录了磁盘的GUID，用来记录自身的大小与位置以及备用GPT表头的位置（位于磁盘的最后一个扇区）。同时还包括CRC32的校验值。
- Partition entries – GPT Disk用简单直接的条目来描述分区。最初的16个字节用来标识分区类型。第二个16直接用来记录改分区唯一的GUID。接下来三个8字节的记录分别描述的初始LBA地址，结束LBA地址和分区属性。最后72个字节为分区名。单个分区记录大小为128个字节。通常Partition entries会从LBA2地址开始。
- 最后，为了为了减少分区表损坏的风险，GPT在硬盘最后保存了一份分区表的副本。



Windows磁盘MBR结构详解

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-22*

在之前的文章 Windows存储管理之磁盘结构详解 中介绍了Windows的磁盘结构和MBR。本文将对Windows Basic Disk中的MBR的结构进行介绍，帮助读者更好的了解Windows系统中的磁盘存储。

更多信息

Windows Basic Disk中的MBR:

Master Boot Record作为磁盘中最重要的数据结构，在磁盘分区的时候会被创建。MBR中包括几个部分，一段可执行的代码叫做Master Boot Code，Disk Signature以及磁盘的分区表。在MBR的末端还有一直为0x55AA值大小为两个字节的Sector Marker的签名字段。这个字通常还标注了extend boot record（EBR）和启动扇区（boot sector）的结束。

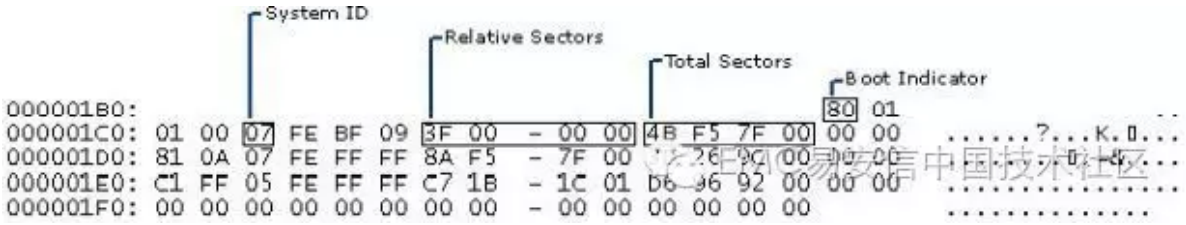
Master Boot Code代码主要完成下列几项活动。

1. 扫描活动分区的分区表
2. 找到活动分区的起始扇区位置
3. 将一个启动扇区的拷贝从活动分区载入到内存
4. 将控制权转移到启动扇区上的执行代码

如果Master Boot Code不能完成这些功能，Windows系统就会抛出一些错误，比如“Invalid partition table”、“Error loading operating system”、“Missing operating system”从而提示相应的步骤中发生了错误。

Basic Disk中的分区表：

在Basic Disk中的Partition Table是一个64个字节的数据结构用来定义物理磁盘上的分区类型与位置的，独立于操作系统。每个分区表的记录是16个直接长度，最大包括四条记录，每条记录从预先定义的起始位置。下面的例子显示一段MBR的记录，其中包括显示了一个三个分区记录，起始位置分别是0x01BE、0x01CE、0x01DE。图中还显示了，分区记录中几个关键的字段。0x01C2是System ID，用来定义逻辑卷的类型，图中07就是表示Installable File System（NTFS）。0x01C6开始的四个字节是Relative Sectore，表示了逻辑卷的起始位置。0x01CA开始的四个字节显示了整个逻辑卷的扇区总数。Boot Indictor显示了是否分区为活动分区。



Windows GPT磁盘GUID结构详解

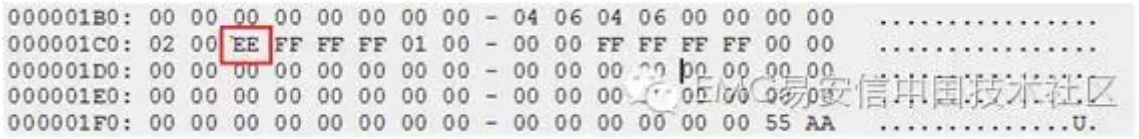
原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-23*

前一篇 **Windows磁盘MBR结构详解** 中我们介绍了Basic Disk中的Master Boot Record结构。GPT Disk作为Windows 2003以后引入的分区结构。使用了GUID分区表结构，它与MBR相比好处是支持更大和更多的分区，提高容错。本文介绍了GUID分区表的结构和各个字段的含义。

更多信息

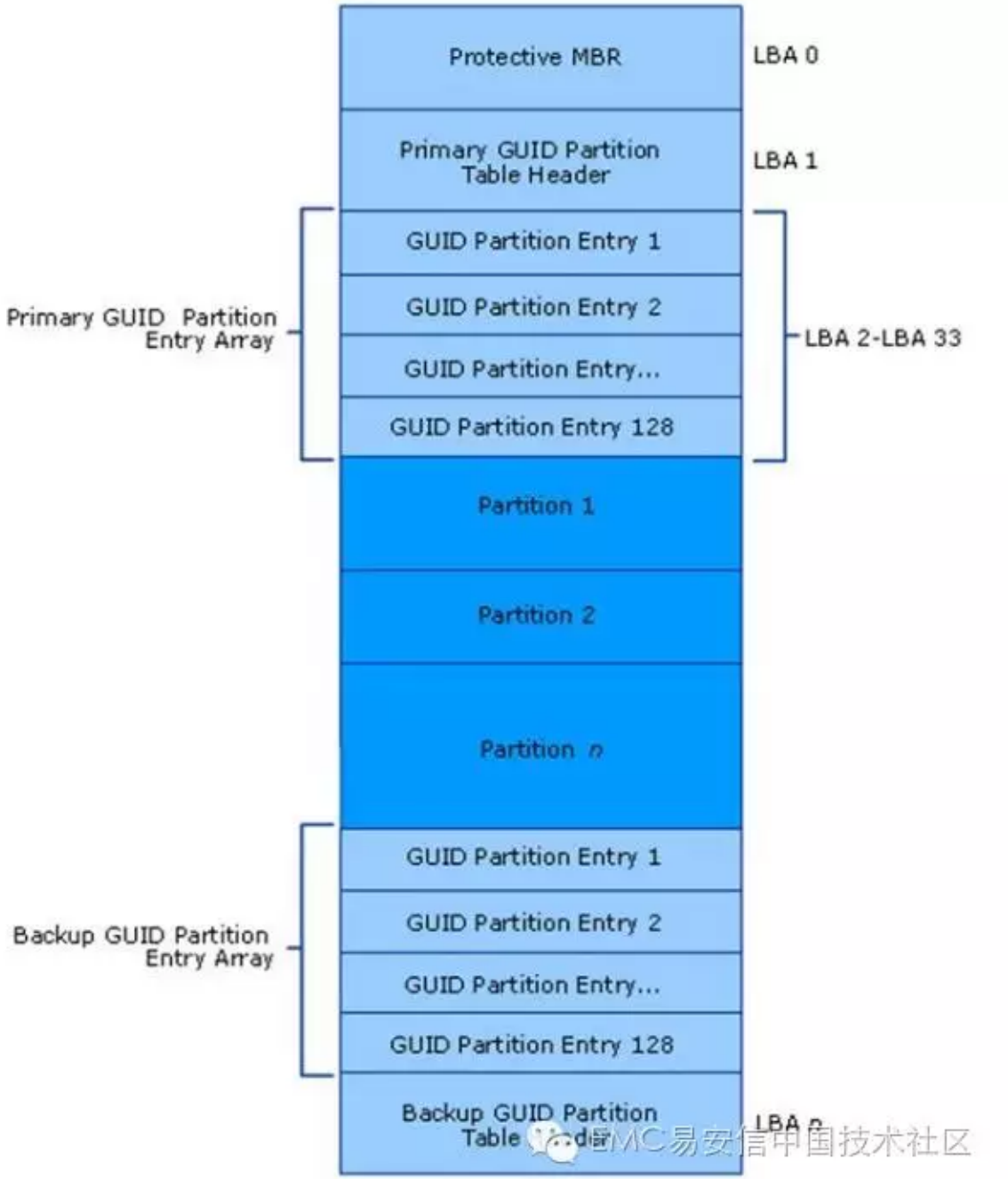
GPT Disk 的Protective MBR:

GPT Disk的结构中，第一个LBA位置（LBA 0）存放的是Protective MBR，随后LBA1的位置才是GPT的GUID分区表头。Protective MBR的作用是为了兼容性考虑，阻止一些遗留的MBR磁盘工具破坏GUID分区表。这种在LBA 0的位置存放Protective MBR的结构是基于Extensible Firmware Interface（EFI）规格。Protective MBR和MBR拥有同样的结构，同时还包括一条包含值为0xEE的System ID的分区记录（如下图所示），这个值标注了该分区为GPT分区，如果包含该分区的磁盘被移动到Windows 2000中或者被一些遗留的磁盘工具访问，则该分区会被标注晨GPT Protective分区，不能被删除。



Globally Unique Identifier Partition Table(GUID)结构:

如下图所示，组成GPT Disk的GUID记录有以下几种：



GUID Partition Table Header： GPT Header定义了分区记录所用到的Logical Block Address的区域。同时还定义了分区在磁盘上的位置、它自身的GUID、以及一组CRC32的校验值。Primary GPT Header位于磁盘的LBA1位置，紧跟着Protective MBR。Backup GPT Header位于磁盘最后一个磁道之上。下图是GPT Header中包含几个字段：

```
00000000: 45 46 49 20 50 41 52 54 - 00 00 01 00 5C 00 00 00   EFI PART....\...
00000010: 27 6D 9F C9 00 00 00 00 - 01 00 00 00 00 00 00 00   'm..............
00000020: 37 C8 11 01 00 00 00 00 - 22 00 00 00 00 00 00 00   7.......".......
00000030: 17 C8 11 01 00 00 00 00 - 00 A2 DA 98 9F 79 C0 01   .............y..
00000040: A1 F4 04 62 2F D5 EC 6D - 02 00 00 00 00 00 00 00   ...b/..m........
00000050: 80 00 00 00 80 00 00 00 - 27 C3 F3 00 00 00 00 00   ........'.......
00000060: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00   ................
```

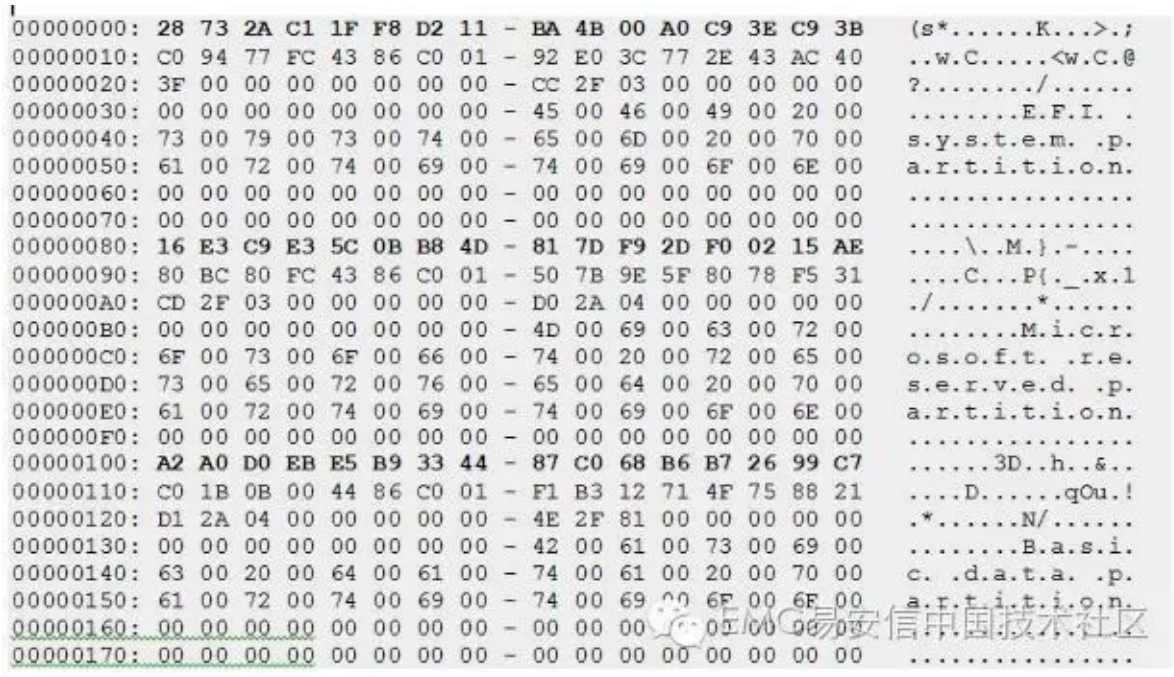
- 首位0x00是为长度8个字节的签名档，这个值必须为固定值，用来定义EFI的兼容性。

- 0x08开始有四组长度为4个字节的字段，Revision标注EFI规格的版本信息、Header Size记录了整个GPT Head的大小，也为固定值、CRC32的校验信息、Reserved预留位。
- 随后0x18开始是5个长度为8个字节的字段和一个长度为16字节字段，Primary LBA记录了Primary GPT Head的位置、Backup LBA记录Backup GPT Header的位置、First Usable LBA记录了第一个分区的起始位置，例如64位的Windows Server 2003，它的起始位置是34、Last Usable LBA记录的分区结束的位置。Disk GUID长度为16个字节，用来标识分区表头和磁盘、 Partition Entry LBA记录GUID分区记录的位置，固定为LBA2。
- 从0x50开始为3个长度为4字节的记录，Number of Partition Entries记录最大的磁盘分区数，例如64位的Windows Server是128个、Size of Partition Entry记录了每个GUID分区记录大小，它的值为128字节、Partition Entry Array CRC32记录一组分区记录的校验值。
- 最后从0x5C位置开始长度为420字节的预留空间，值都为0。

GUID Partition Entry Array：类似MBR中的分区表，GUID partition entry array中包含了磁盘上每个分区的记录。比如64位的Windows Server 2003创建了一个Array值为16384字节，则第一个可用的数据大于等于LBA34. GUID Partition Entry Array也在分区最后存在一个副本，位置是在最后一个可用的LBA之后，GUID Partition table header之前。

GUID Partition Entry： GUID Partition Entry长度为128个字节，用来定义单个分区结构。每个GUID Partition entry从Partition Type记录开始。长度为16个字节的Partition Type GUID，类似MBR磁盘分区表中的System ID，它定义了分区中包含的数据和分区的作用。下图中是一个典型的GPT Disk的GUID Partition entry array记录，这个图中显示了三种分区记录

第一部分{28732AC1–1FF8–D211–BA4B–00A0C93EC93B}为EFI System分区、中间部分{16E3C9E3–5C0B–B84D–817D–F92DF00215AE}为Microsoft Reserved分区，最下面的一个{A2A0D0EB–E5B9–3344–87C0–68B6B72699C7}是Windows Basic Disk中的一个主分区。



在每个部分的中间位置主要记录了一些字段。Partition Type GUID之后的0x10位置开始，是一个长度为16字节的Unique Partition GUID记录用来标识每条记录的唯一性。0x20开始为3个长度为8个字节的字段，Starting LBA和Ending LBA分别记录了分区的起始和结束的位置、Attribute Bit描述了分区是如何使用的，例如是否为隐藏和只读等等、最后0x38开始的72个字节字段用来Unicode的分区的名字，名字

最长为36个字符。

存储SCSI锁解读

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-06-24*

SCSI锁是多台主机用来操作LUN的基本机制。在Windows存储环境中，当多台Windows主机需要访问一个LUN的情况下，例如Windows Cluster环境，就会用到SCSI锁。本文就SCSI锁的作用和类型，以及Windows Cluster 2003/2008中使用到的SCSI锁进行介绍。

SCSI锁的作用:

在一个共享存储的环境下，多台前端主机可能会同时访问同一台存储设备，如果此时多台主机在同一时点上对一个LUN进行写操作，那么可想而知这个LUN将不知道哪个数据先写，哪个数据后写。为了防止这种情况发生而导致的数据损坏，所以就有了SCSI锁的概念。通过SCSI Reservation机制来进行SCSI锁的操作，目前绝大多数的磁盘都支持'SCSI reservvation命令'。如果一台主机给磁盘传输了一条SCSI Reservation命令，则这个磁盘对于其他的主机就处于锁定状态。如果有其他的主机给已经被锁定的磁盘发送读写请求，则会收到'reservation conflict'报错信息。如果保留SCSI锁的主机崩溃，或者其他主机给磁盘发送'break reservation或者reset target命令，用来解除SCSI锁。然后，第二个主机发送I/O请求之前需要重新发送SCSI Reservation命令给磁盘。

SCSI锁的分类:

SCSI锁有两种类型：SCSI-2 Reservation和SCSI-3 Reservation。一个LUN上只能存在一种类型的SCSI锁。

SCSI-2 Reservation只允许设备被发出SCSI锁的Initiator访问，也就是主机的HBA。比如主机1上的HBA1对访问的LUN加上SCSI-2锁，此时即使主机1的HBA2也无法访问这个LUN。所以SCSI-2 Reservation也被称为Single Path Reservation。

SCSI-3 Reservation（Persistent Reservation）是使用PR Key来对磁盘进行加锁。通常一台Host会有唯一的PR Key，不同的主机对应的PR Key也不同。所以一般SCSI-3 Reservation通常被应用在多通路的共享环境下面。这里SCSI-3 Reservation也称之为Persistent Reservation。

Windows Cluster中的SCSI锁：

Windows 2003集群中使用SCSI-2 reserve/release命令。作为非持久的reservation，所以集群中的一台节点会持有SCSI-2 Reservation的锁，然后每过3秒会重新刷新一次。如果故障转移发生，则切换节点主机会在相应的磁盘上放置SCSI-2 Reservation然后维护SCSI锁。如果所有节点主机上的集群服务都会关闭，Reservation也不会保留。

Windows 2008集群中使用SCSI-3 persistent reservation机制。如果磁盘从主机上没有正确移除，集群使用的磁盘（Cluster Disk）会保留着这些Reservation。锁对应的SCSI锁会一直存在于相应的磁盘之上，即使集群服务被关闭或者磁盘对于主机取消掩饰（unmasked）。所以，有些时候需要强行移除磁盘上的Reservation。

在Windows环境中，移除Reservation移除的命令是（disknumber可以在Windows的Disk Management中找到）

CLUSTER NODE /CLEARPR:disknumber

Windows扩展逻辑卷的操作方法

原创 EMC中文技术社区 戴尔易安信技术支持 *2016-07-01*

如果一个Windows的逻辑卷原有空间用尽了，操作系统提供了几种操作方式，将逻辑卷扩展到未分配的磁盘空间。本文介绍了在Windows操作系统平台，如何使用diskpart.exe命令对逻辑卷在未分配的空间上进行扩展的方法。

扩展逻辑卷的前提条件:

diskpart是一个Windows平台自带的管理磁盘，分区和逻辑卷的命令行工具。diskpart命令可以同时用于Basic Disk和Dynamic Disk。diskpart的extend命令可以在保留数据的前提下用来合并未分配的空间到现有逻辑卷。使用extend命令之前，需要先满足一些条件：

1. 逻辑卷必须为NTFS文件系统。
2. 对于Basic Volume，未分配的空间的扩展区必须是在同一个磁盘上的连续空间。
3. 对于Dynamic Volume，未分配空间必须是同一个操作系统中其他的Dynamic Disk。
4. 对于系统卷和启动卷不支持这个操作。

5. 如果需要扩展的分区上含有Windows的page file，需要管理虚拟内存的page file才能继续进行操作。

操作方法：

\1. 扩展数据分区和逻辑卷，首先需要先选择对应的逻辑卷，然后定义扩展数据的大小。步骤如下：

- 命令行输入 diskpart.exe
- 输入 list volume 显示现有的逻辑卷
- 输入 select , 选择需要扩展的逻辑卷
- 输入extend [size=n] [disk=n] [noerr] , size的单位为MB、disk参数指定需要扩展数据的磁盘，只针对Dynamic Disk的扩展，如果不指定则默认是当前的磁盘、noerr适用于脚本的批量处理，如果定义这个参数，遇到错误信息，diskpart会继续运行。如果没有定义noerr参数，系统抛出的错误就会导致diskpart命令终止。

当extend命令完成以后，会收到提示信息diskpart成功扩展了逻辑卷，操作完成。

\2. 扩展启动分区（只是适用于Windows Server 2008以上版本）

- 点击Start，然后点击Server Manager
- 展开Storage，然后选择Disk Management
- 右键需要扩展的卷，点击Extend Volume
- 然后根据向导扩展启动分区
- 需要注意的是，扩展启动分区，必须在同一个磁盘的连续未分配空间上进行。