[MBR & GPT](http://blog.csdn.net/wzsy/article/details/7372210)

MBR 和 GPT 的问题已经走到了非常现实的阶段-3T硬盘开始普及。MBR 由于对 3T 硬盘寻址无法支持，终于要开始被弃用了。但仅仅只是“开始”，我感觉至少10年之内 MBR 还不会消失。恐怕当GPT适应不了需求的时候，MBR才会消失。  
  
一、MBR 的失误  
说“失误”牵强了一点。我们不能要求当年只有5M的硬盘就扣出2M来放寻址信息。计算机永远比你想象的要发展的快。  
首先你要有一些基本的硬盘的MBR知识，如果你不熟悉，可以先在 wiki 看一看[硬盘的基本结构](http://zh.wikipedia.org/wiki/%E7%A1%AC%E7%9B%98)。  
在了解了磁道、柱面、扇区之后，再看下面的 MBR 结构就很了然了：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 基本MBR结构 | | | | |
| 偏移量(位置) (within sector) | | 长度 (in bytes) | 解释 |  |
| 十进制 | 十六进制 |  |
| 000 – 445 | 000 – 1BD | 446 | 引导程序 |  |
| 446 – 509 | 1BE – 1FD | 64 | 主分区表 |  |
| 510 – 511 | 1FE – 1FF | 2 | 引导记录 |  |

表 1.

MBR 永远占用了第一个扇区，通常为512bytes大小。在这512bytes中被MBR本身的程序用去了446bytes。于是，真正描述分区信息的就只有 64 bytes。而根据4个主分区的设计，每一个分区能用到的只有 16bytes. 那么！这16bytes是如何工作的？

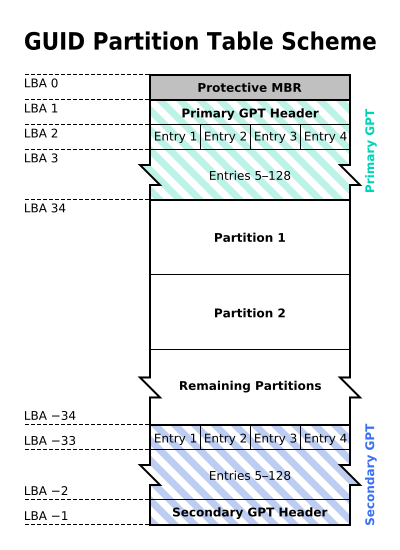
|  |  |  |
| --- | --- | --- |
| 16-byte分区表结构 | | |
| Relative Offsets (within entry) | 长度  (bytes) | 内容 |
| 0 | 1 | 引导标记 (80h =active) |
| 1 – 3 | 3 | CHS 开始位置 |
| 4 | 1 | 分区类型描述 |
| 5 – 7 | 3 | CHS 结束位置 |
| 8 – 11 | 4 | 启始扇区 |
| 12 – 15 | 4 | 分区大小 (扇区单位) |

表 2.

在表2中，有2个“开始位置”，我们分开理解。  
硬盘的历史过程中，第一个遇到的问题就是 8G 瓶颈。这就是因为在 [CHS](http://zh.wikipedia.org/wiki/CHS) 描述方式中，描述起点和终点均为3个 byte。我们假设是 FF FF FF ，所以是 16\*16\*16\*16\*16\*16\*512=8589934592 这就是 8G 瓶颈的由来了。  
但是今天 CHS 方式几乎已经不用了。后面还有 4 个 bytes 的“启始” 是给 LPA 方式用的。他的极限就是 8589934592 \* 256 =2199023255552 这就是 2T 的限制由来了。  
在硬盘容量的历史中，有很多因素决定了它的限制。BIOS，ATA（IDE），文件系统。[百度上的这篇文章](http://baike.baidu.com/view/1329184.html" \l "sub1329184)简单的介绍了各个阶段的限制。

二、GPT 的思路

GPT （[GUID](http://zh.wikipedia.org/wiki/GUID) Partition Table）属于 EFI 方案中的一部分。虽然很早就开始制定这个标准，但到今天，还是有很多系统不支持 GPT。

首先，引用一张wikipedia上的结构图，来了解一下 GPT 的基本结构。  


GPT 为了兼容 MBR，LBA 0 依旧保留了MBR的结构。在GPT工作是，会优先读取 GPT (LBA1) 内容。如果没有 GPT 内容，则认为这是一块MBR磁盘。再从LBA 0 读取MBR。

在硬盘末尾，GPT 备份了一份，这样当GPT出错时，可以快速的从硬盘末尾恢复。LBA -1 (负1) 表示倒数第一块 LBA。

从LBA 2 到 LBA 33 ，一共预留了 128 个分区表空间。 GPT 支持在一块硬盘上创建 128 个分区（真有这种需求？）。所以每一个分区可以使用 128 bytes 的空间。

来看一看 LBA 1 的结构吧，有了 512bytes 的巨型空间，都用不掉了。

|  |  |  |
| --- | --- | --- |
| GPT头的基本结构 | | |
| 起始字节 | 长度 | 内容 |
| 0 | 8字节 | 签名（”EFI PART”, 45 46 49 20 50 41 52 54） |
| 8 | 4字节 | 修订（比如，1.0，值是 00 00 01 00） |
| 12 | 4字节 | 分区表头的大小（单位是字节，通常是92字节，即 5C 00 00 00） |
| 16 | 4字节 | 分区表头前3项（第0－15字节）的[CRC32](http://zh.wikipedia.org/wiki/CRC32" \o "CRC32)校验，如果值正在计算，则是 0 |
| 20 | 4字节 | 保留，必须是 0 |
| 24 | 8字节 | 当前LBA（这个分区表头的位置） |
| 32 | 8字节 | 备份LBA（备份分区表头的位置） |
| 40 | 8字节 | 第一个可用于分区的LBA（主分区表的最后一个LBA + 1） |
| 48 | 8字节 | 最后一个可用于分区的LBA（备份分区表的最后一个LBA − 1） |
| 56 | 16字节 | 硬盘GUID（在[类UNIX](http://zh.wikipedia.org/wiki/%E9%A1%9EUNIX" \o "类UNIX)系统中也叫[UUID](http://zh.wikipedia.org/wiki/UUID)） |
| 72 | 8字节 | 分区表项的起始LBA（对于主分区表来说，这个值是 LBA 2） |
| 80 | 4字节 | 分区表的数量 |
| 84 | 4字节 | 一个分区表的大小（通常是128） |
| 88 | 4字节 | 分区串行的CRC32校验 |
| 92 | \* | 保留，剩余的字节必须是0（对于512字节LBA的硬盘即是420个字节） |

此表来源为 [中文 Wikipedia](http://zh.wikipedia.org/wiki/%E5%85%A8%E5%B1%80%E5%94%AF%E4%B8%80%E6%A8%99%E8%AD%98%E5%88%86%E5%8D%80%E8%A1%A8), 稍微修改了两个字，便于理解。

通过这张表，程序可以获知分区数量，第一个分区在哪个位置，以及GPT头是否正确（CRC32）。在这里着重解决的是分数数量问题，以及分区表的安全性问题。而分区的大小问题，则依赖了 GPT 的分区表：

|  |  |  |
| --- | --- | --- |
| GPT分区表项的格式 | | |
| 起始字节 | 长度 | 内容 |
| 0 | 16字节 | 分区类型GUID |
| 16 | 16字节 | 分区GUID |
| 32 | 8字节 | 起始LBA（[小端序](http://zh.wikipedia.org/wiki/%E5%B0%8F%E7%AB%AF%E5%BA%8F)） |
| 40 | 8字节 | 末尾LBA |
| 48 | 8字节 | 属性标签（如：60表示“只读”） |
| 56 | 72字节 | 分区名（可以包括36个[UTF-16](http://zh.wikipedia.org/wiki/UTF-16)（小端序）字符） |

重点就在这里了，在描述分区位置的时候，使用了8个字节，其最大值为 FF FF FF FF FF FF FF FF 。那么，假设每一个LBA 为512 bytes。其能描述的大小为 8589934592 TB = 8 ZB 。关于这个数字我算下来的确是在 ZB 级别的。但是搜索到的文章说法不一：

[突破2TB限制 3TB硬盘装操作系统实战](http://tech.sina.com.cn/h/2011-04-12/05441713987.shtml) 9.4ZB （GPT分区表采用8个字节即64bit来存储扇区数，因此它最大可支持264个扇区。）  
[关于windows server 2003最大支持硬盘容量的问题](http://social.microsoft.com/Forums/zh-CN/windowsserversystemzhchs/thread/413073fd-b169-44a2-9dbb-5648d8e03c07) 18EB

对于上面这些数字，18EB 恐怕是 LBA 数量，忘记乘 512 bytes 了。9.4ZB 暂时不知道是怎么来的。但这个数字暂且不做定论。因为[微软的 FAQ](http://msdn.microsoft.com/en-us/windows/hardware/gg463525.aspx) 也提到了，这只是理论值，谁知道下一个瓶颈在哪里呢。

Q. How big can a GPT disk be?

A. In theory, a GPT disk can be up to 2^64 logical blocks in length. Logical blocks are commonly 512 bytes in size.

The maximum partition (and disk) size is a function of the operating system version. Windows XP and the original release of Windows Server 2003 have a limit of 2TB per physical disk, including all partitions. For Windows Server 2003 SP1, Windows XP x64 edition, and later versions, the maximum raw partition of 18 exabytes can be supported. (Windows file systems currently are limited to 256 terabytes each.)