# 背景

哈希（hash）是一种非常快的查找方法，在一般情况下这种查找的时间复杂度为O(1)，即一般仅需要一次查找就能定位数据。而B+树的查找次数，取决于B+树的高度，在生产环境中，B+树的高度一般为3~4层，故需要3~4次的查询。

InnoDB存储引擎不支持哈希索引，因此引入自适应哈希索引。

# 概述

InnoDB存储引擎会监控对表上各索引页的查询。如果观察到建立哈希索引可以带来速度提升，则建立哈希索引，称为自适应哈希索引（Adaptive Hash Index，AHI）。

AHI是通过缓冲池的B+树页构造而来，因此建立的速度非常快，而且不需要对整张表构建哈希索引。InnoDB存储引擎会自动根据访问的频率和模式来自动地为某些热点页建立哈希索引。

根据InnoDB存储引擎官方文档显示，启用AHI后，读取和写入速度可以提高2倍，辅助索引的连接操作性能可以提高5倍。因此，AHI是非常好的优化模式，其设计思想是数据库自优化（self-tuning），即无需DBA对数据库进行人为调整。

## 限制

哈希索引（不是自适应哈希）的限制：

1、哈希索引只能用来搜索等值的查询，如select \* from table where index\_col=xxx。而对于其他查询类型，如范围查找，是不能使用哈希索引的。

2、无法用于排序

3、有冲突可能

4、MySQL自动管理，用户无法干预。

AHI有一个要求，即对这个页的连续访问模式必须是一样的。例如对于（a,b）这样的联合索引页，其访问模式可以是如下情况：

1、WHERE a=xxx

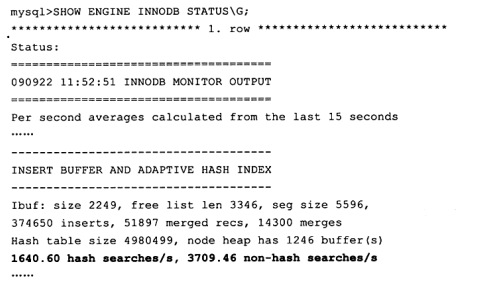
2、WHERE a=xxx AND b=xxx

访问模式一样指的是查询的条件一样，若交替进行上述两种查询，那么InnoDB存储引擎不会对该页构造AHI。此外AHI还有如下要求：

1. 以该模式访问了100次
2. 页通过该模式访问了N次，其中N=页中记录\*1/6

## 启动

通过命令SHOW ENGINE INNODB STATUS可以看到当前AHI的使用状况。



用户可以通过观察SHOW ENGINE STATUS的结果及参数innodb\_adaptive\_hash\_index来考虑是禁用或启动此特性，默认AHI为开启状态（建议关闭，意义不大）。可以通过 set global innodb\_adaptive\_hash\_index=off/on 关闭和打开该功能。

当在配置文件中启用了参数innodb\_adaptive\_hash\_index后，数据库启动时会自动创建槽数为innodb\_buffer\_pool\_size/256个的哈希表。例如，对当前参数innodb\_buffer\_pool\_size设置为10MB，则启动时InnoDB存储引擎会创建一个有10M/256=40 960个槽的自适应哈希表。

# 特点

## 优点

　1、无序，没有树高

　2、降低对二级索引树的频繁访问资源

　　　索引树高<=4，访问索引：访问树、根节点、叶子节点

　3、自适应

## 缺点

　 1、hash自适应索引会占用innodb buffer pool；

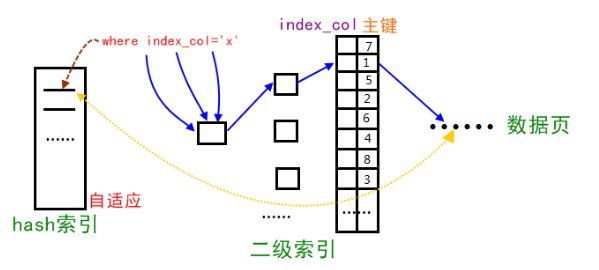
　 2、自适应hash索引只适合搜索等值的查询，如select \* from table where index\_col='xxx'，而对于其他查找类型，如范围查找，是不能使用的；

　　3、极端情况下，自适应hash索引才有比较大的意义，可以降低逻辑读。

# 原理

InnoDB存储引擎中自适应哈希索引使用的是散列表（Hash Table）的数据结构。

散列函数：数据库中一般采用除法散列的方法。



InnoDB存储引擎使用哈希算法对字典进行查找，其冲突机制采用链表方式，哈希函数采用除法散列方式。对于缓冲池页的哈希表来说，在缓冲池中的Page页都有一个chain指针，它指向相同哈希函数值的页。而对于除法散列，m的取值为略大于2倍的缓冲池页数量的质数。

例如：当前参数innodb\_buffer\_pool\_size的设置大小为10MB，则共有640个16KB的页。那对于缓冲池页内存的哈希表来说，需要分配640×2=1280个槽，但是1280不是质数，需要取比1 280略大的一个质数，应该是1399，所以在启动时会分配1399个槽的哈希表，用来哈希查询所在缓冲池中的页。哈希表本身需要20个字节，每个槽需要4个字节，因此一共需要20+4×1399=5616个字节。其中哈希表的20个字节从innodb\_additional\_mem\_pool\_size中进行分配，4×1399=5596个字节从系统申请分配。因此在对InnoDB存储引擎进行内存分配规划时，也应该规划好哈希表这部分内存，这部分内存一般从系统分配，没有参数可以控制。对于前面我们说的128GB的缓冲池内存，则分配的哈希表和槽一共需要差不多640MB的额外内存空间。

那InnoDB存储引擎对于页是怎么进行查找的呢？上面只是给出了一般的算法，怎么将要查找的页转换成自然数呢？

InnoDB存储引擎的表空间都有一个space号，我们要查的应该是某个表空间的某个连续16KB的页，即偏移量offset。InnoDB存储引擎将space左移20位，然后加上这个space和offset，即关键字K=space<<20+space+offset，然后通过除法散列到各个槽中。