



4. Insert/Change Buffer 查看

```
mysql> show engine innodb status\G
-- -----忽略其他输出-----
INSERT BUFFER AND ADAPTIVE HASH INDEX
-----
Ibuf: size 1, free list len 0, seg size 2, 0 merges -- 这里为0，可能是Buffer Pool足够大。
-- 数据页都缓存存在内存中了，就用不到buffer了

merged operations:
insert 0, delete mark 0, delete 0
discarded operations:
insert 0, delete mark 0, delete 0
```

- 1. **seg size** : 页 的数量，例如当前页为8K，则 seg\_size \* 8K 就是Insert/Change Buffer使用的内存大小；
- 2. **merges** : 合并了多少 页；
- 3. **merged insert** : 插入了多少条 记录；
  - insert / merges 就是插入的效率（插入一条记录，就要读取一次页）；
- 4. **discarded operations** : 应该是很小的值，或者为0；当记录写入到Insert/Change Buffer后，对应的表被删除了，则相应的Buffer中的记录就应该被丢弃；

使用Insert/Change Buffer的前提时，需要 使用随机IO，这时才放入Buffer中，如果页已经在Buffer Pool（内存）中，就 不需要 使用Insert/Change Buffer了

5. Change Buffer

- 从MySQL 5.5以后，改名为Change Buffer，表示不仅仅适用于insert。
  1. Insert
  2. Delete-Marking（标记删除）
  3. Purge（清除）
  4. `innodb_change_buffering = all`
    - all
    - none（禁用）
    - inserts
    - deletes
    - changes = (insert & delete-marking)
    - purge

三. Adaptive Hash Index（自适应Hash索引）

- 搜索的时间复杂度
  - B+树 O(T)，只能定位到该记录所在的页；
  - 哈希表 O(1)，可以直接定位到记录；
- 可以自己判断是否是活跃的页，如果是活跃的页，可以自动做Hash，管理员无需人工干预；
- MySQL5.6版本后，官方 不推荐 使用自适应Hash索引
  - CPU 使用率变高，但是性能没有提升；
- MySQL5.7中增加了 `innodb_adaptive_hash_index_parts`，增加分片，以减少竞争；
- 只对等值的操作有意义；

四. Flush Neighbor Page (FNP)

- 刷新 脏页所在区（extent）的 所有脏页，合并IO，随机转顺序的优化；
  - 写入的数据太多
  - 如果业务确实是频繁更新，那刷新也会很频繁
- 对传统机械磁盘有意义；
  - `innodb_flush_neighbors=0|1|2`（>=MySQL 5.6）
  - 0：表示关闭该功能
  - 1：表示刷新一个区内的脏页
  - 2：表示刷新几个 连续 的脏页
- SSD建议关闭该功能；

```
mysql> show variables like "%flush_neigh%";
+-----+
| Variable_name | Value |
+-----+
| innodb_flush_neighbors | 2 | -- 非SSD建议使用2
+-----+
1 row in set (0.00 sec)
```