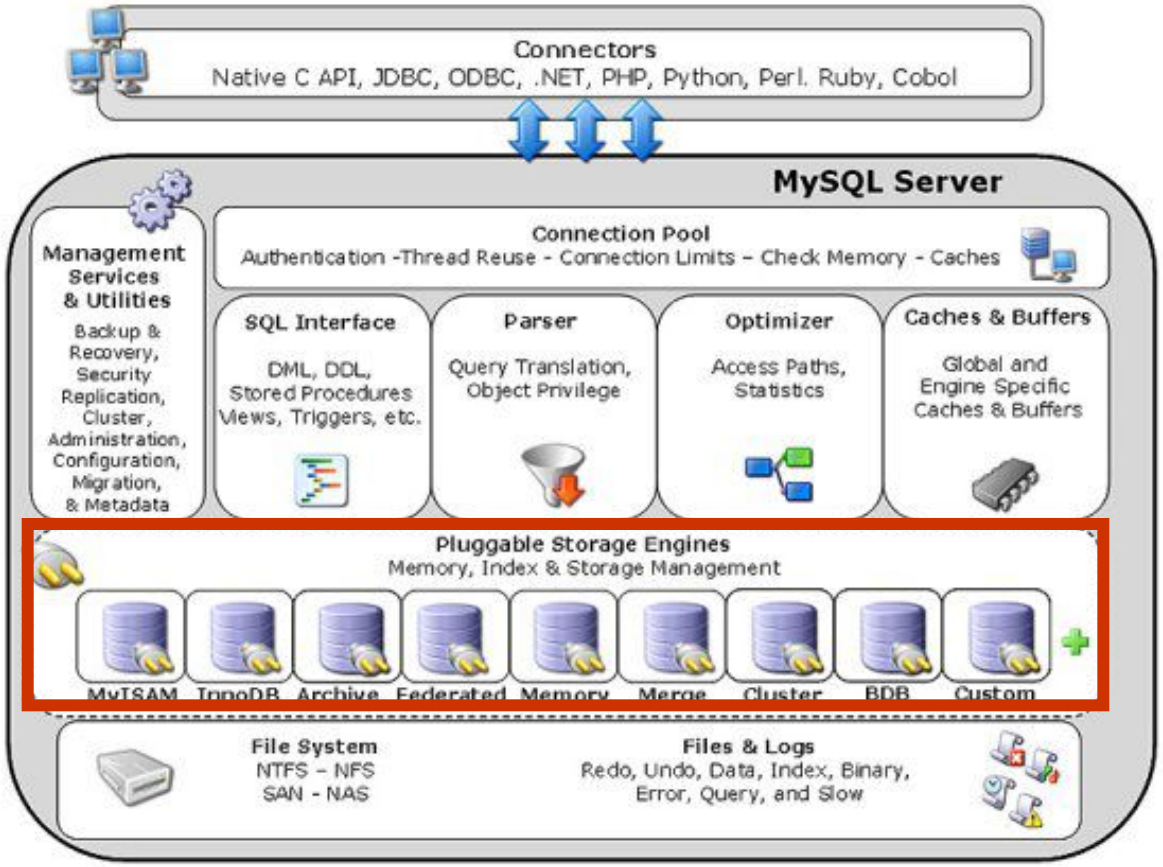
innodb介绍

平台技术中心-杨辰

mysql: 插件式的存储引擎

– myisam
– innodb
– TokuDB
– infobright
– ...



innodb存储引擎

- 特点:
  - 事务安全, 多版本读取, 行锁, 外键, ...
- innobase Oy公司开发, oracle收购
- mysql(>=5.5)默认存储引擎
- 衍生引擎: XtraDB

大纲

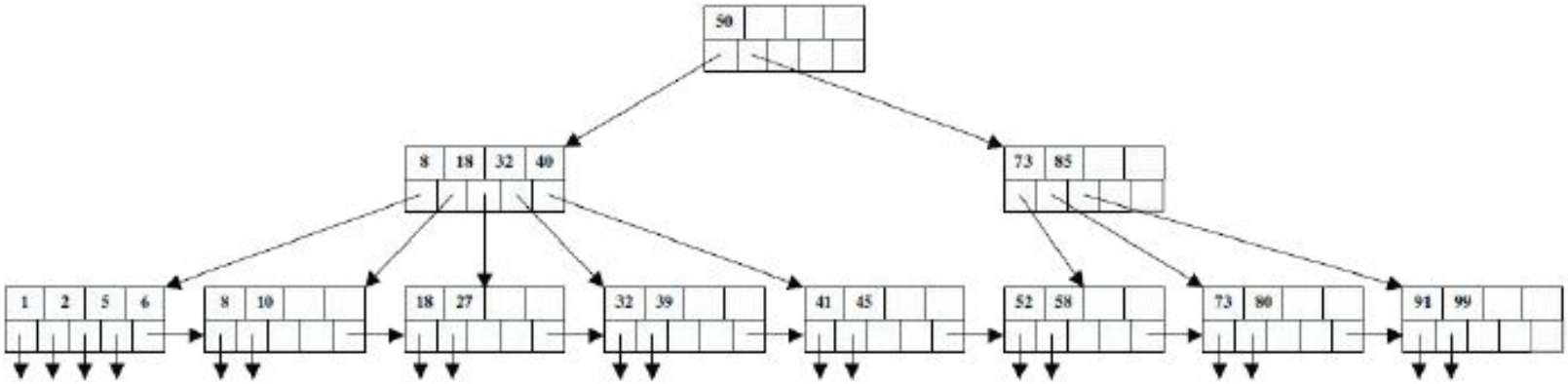
- innodb索引
- innodb的IO相关参数概念

索引

- 数据库索引类型
  - B-tree索引: 区间查找, 排序
  - Hash索引: O(1)查找
- innodb中的索引
  - 聚簇索引 (clustering index)
  - 二级索引 (secondary index)
  - 自适应哈希索引 (adaptive hash index)

B+树结构

- ordered search tree
- large fanout to optimize disk IO
- split on insert



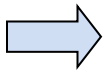
聚簇索引 & 二级索引

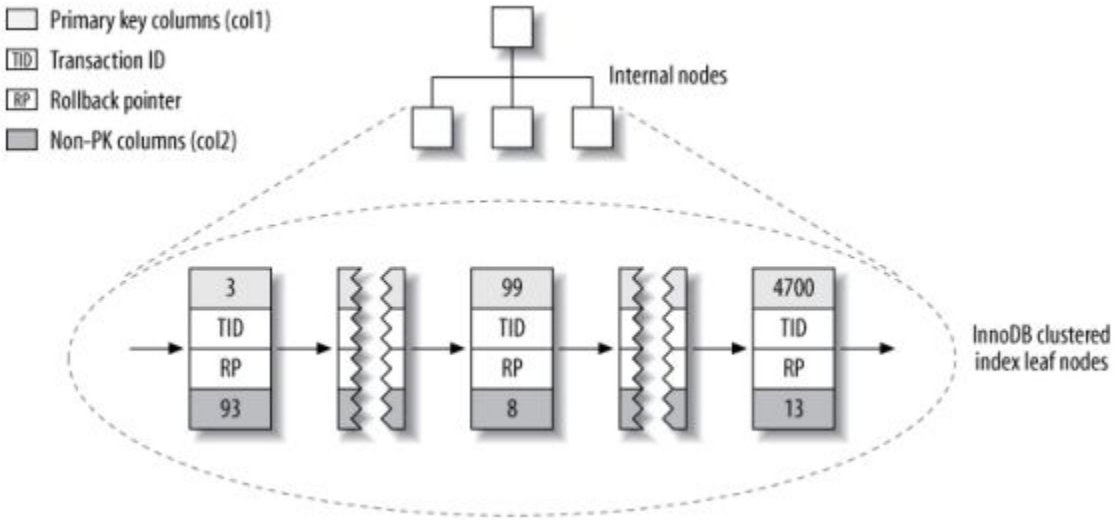
- 聚簇索引
  - primary key / unique index
  - 物理存储顺序按照聚簇索引, 按页排列
  - 叶节点即数据
- 二级索引
  - 指向聚簇索引
  - 两次查找

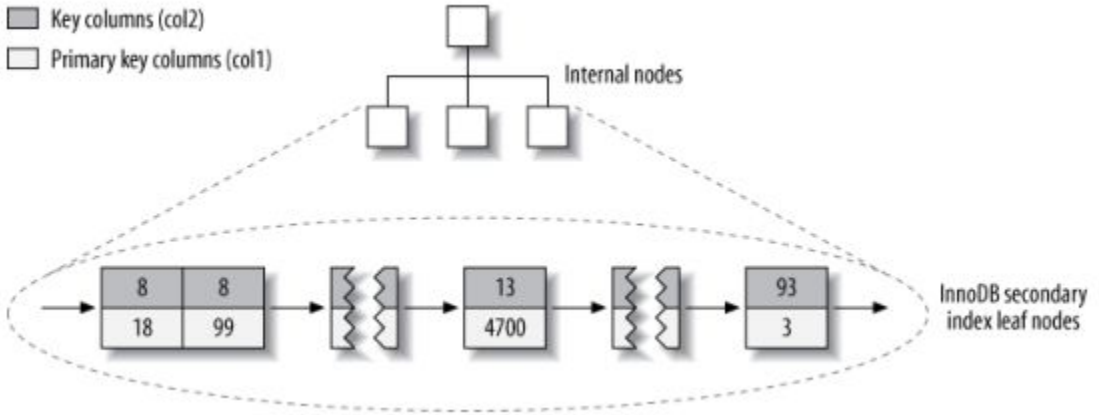
聚簇索引 & 二级索引 示例

```
CREATE TABLE layout_test (
   col1 int NOT NULL,
   col2 int NOT NULL,
   PRIMARY KEY(col1),
   KEY(col2)
);
```

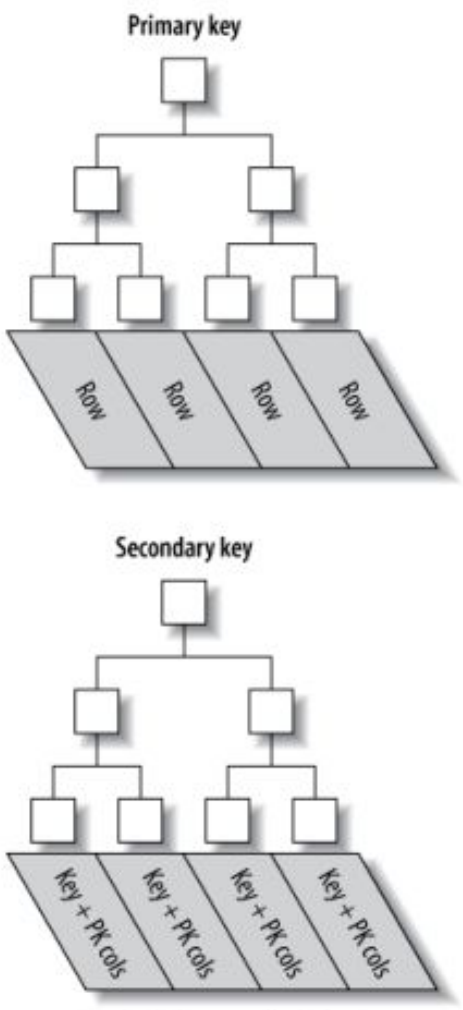
Row number	col1	col2
0	99	8
1	12	56
2	3000	62
9997	18	8
9998	4700	13
9999	3	93



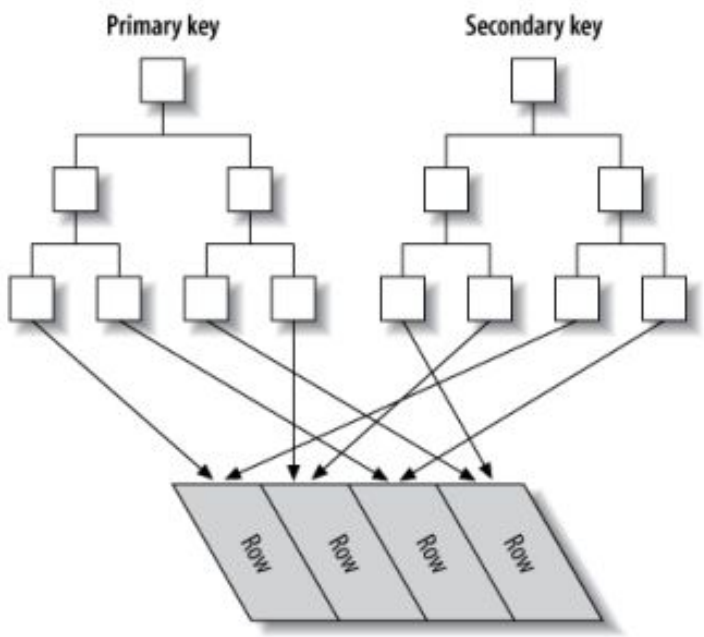




和MyISAM比较



InnoDB (clustered) table layout



MyISAM (nonclustered) table layout

聚簇索引对于insert启示

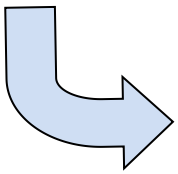
- 按照主键顺序插入
- 自增索引

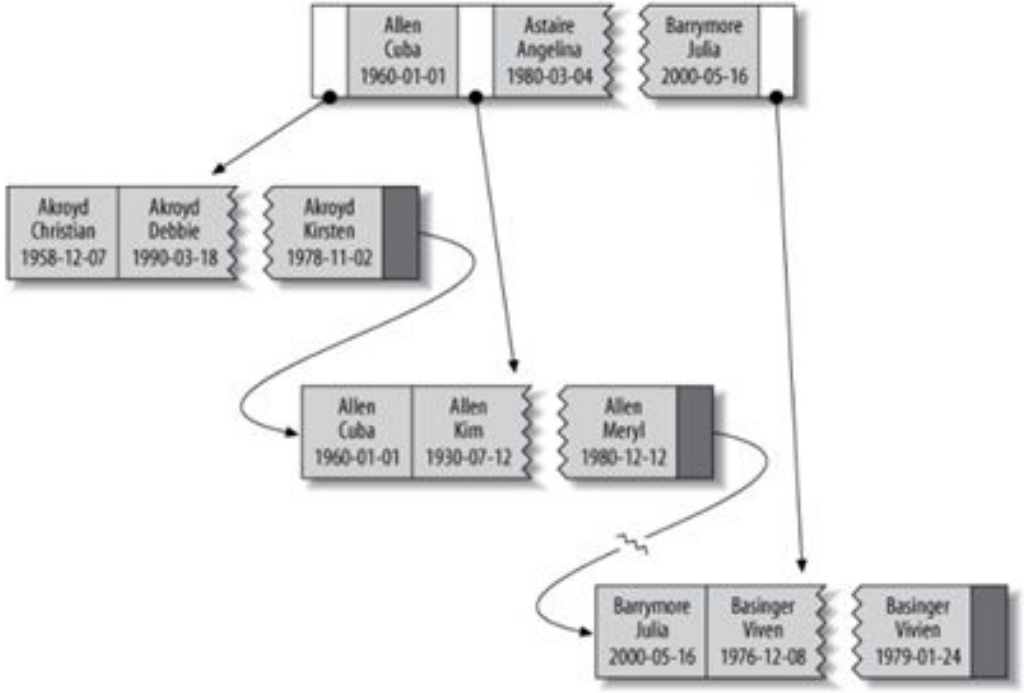
```
create table xxoo (
    id int unsigned not null auto_increment,
    ...,
    primary key (`id`)
)
```

- 自增索引的问题
  - 高并发情况下, 锁争用
    - innodb_autoinc_lock_mode
  - 主从同步

多列索引(multicolumn indexes)

```
CREATE TABLE People (
   last_name  varchar(50)    not null,
   first_name varchar(50)    not null,
   dob        date           not null,
   gender     enum('m', 'f')not null,
   key(last_name, first_name, dob)
);
```





列顺序很重要!

索引设计

- 选择主键
- 注意主键顺序
- 使用EXPLAIN观察执行计划
- 按照查询特点添加二级索引
- 例子:

```
CREATE TABLE `4399_channel_new.channel_game_week` (
  `recdate` date NOT NULL COMMENT '日期，为该周周一所在Y-m-d',
  `cid` int(11) NOT NULL COMMENT '渠道',
  `game` varchar(20) NOT NULL COMMENT '游戏缩写，_ALL_ 表示全部统计里',
  `manager` varchar(100) NOT NULL COMMENT '渠道负责人',
  `payment` decimal(18,2) NOT NULL DEFAULT '0.00' COMMENT '投入',
  `regcount` int(11) NOT NULL DEFAULT '0' COMMENT '注册数',
  `liveuser` int(11) NOT NULL DEFAULT '0' COMMENT '活跃注册数',
  `paypeople` int(11) NOT NULL DEFAULT '0' COMMENT '付费人数',
  `income` decimal(18,2) NOT NULL DEFAULT '0.00' COMMENT '付费金额',
  PRIMARY KEY (`recdate`,`cid`,`game`,`manager`),
  KEY `cid_recdate_index` (`cid`,`recdate`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='4399市场游戏周报'
```

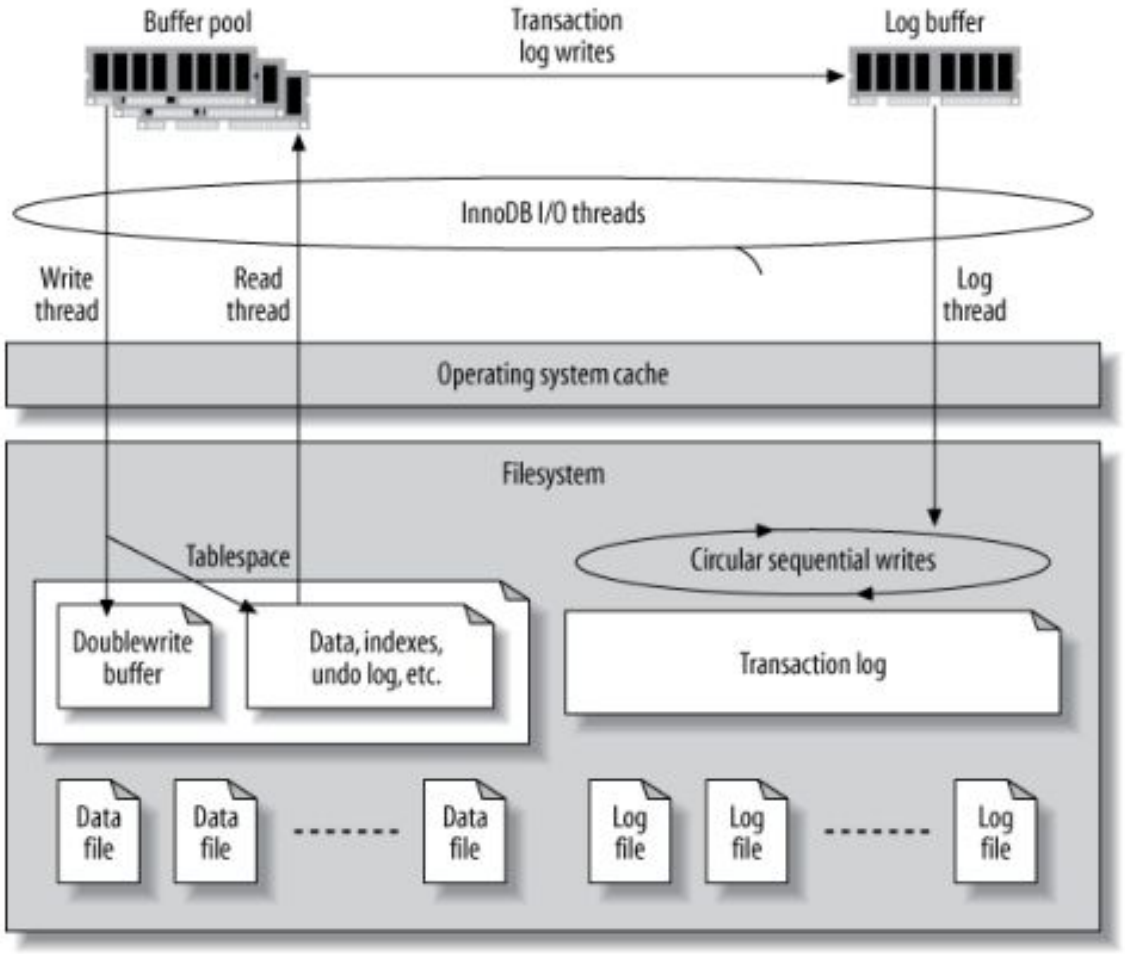
自适应哈希索引(Adaptive Hash Indexing)

- 缓存中自动构建
- 覆盖表部分常用的索引
- innodb_adaptive_hash_index 开关(默认ON)

大纲

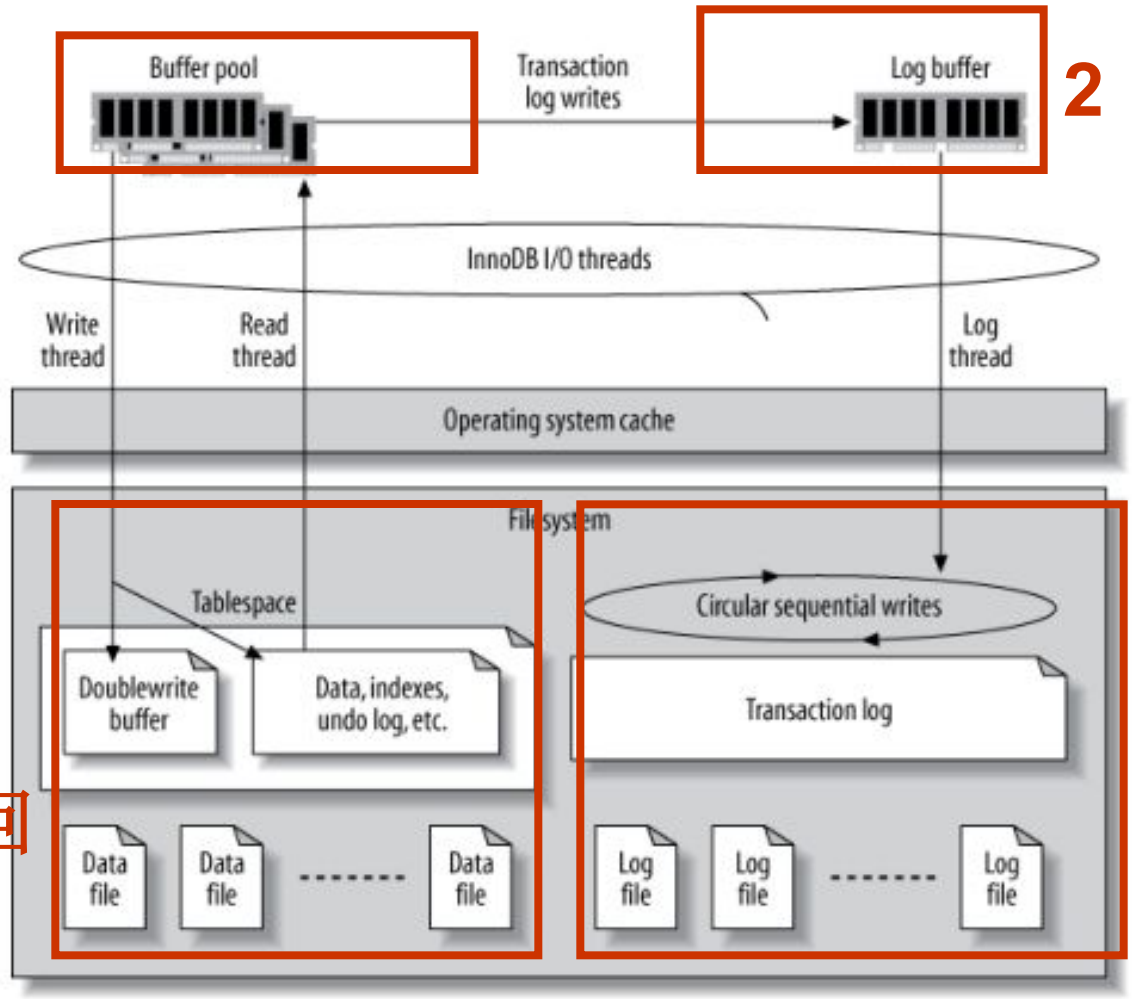
- innodb索引
- innodb的IO相关参数概念

InnoDB结构


Buffer pool
Transaction log writes
Log buffer
InnoDB I/O threads
Write thread
Read thread
Log thread
Operating system cache
Filesystem
Tablespace
Circular sequential writes
Doublewrite buffer
Data, indexes, undo log, etc.
Transaction log
Data file
Data file
Data file
Log file
Log file
Log file

InnoDB结构

4 缓存

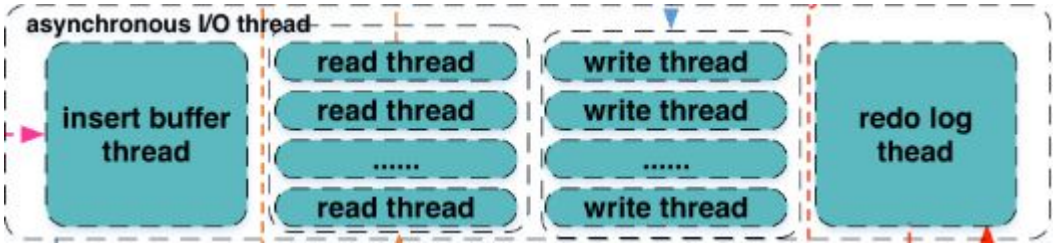


2 日志缓存

3 表空间及脏页写回

1 日志

Innodb IO threads



- log thread
- insert buffer thread
- read thread * N
  - 参数 innodb_read_io_threads
- write thread * N
  - 参数 innodb_write_io_threads

日志(innodb redo log)

- WAL (write ahead logging)
- 为什么不直接写数据页?
- LSN (Log Sequence Number)
  - 数据库状态的一个快照
- 崩溃恢复时回放
- checkpoint, why?
  - 日志大小有限
  - 加快崩溃恢复速度
  - 减少脏页, 增加可用缓存空间

和binlog区别

- mysql, 而不是存储引擎, 记录binlog
- 记录内容不同
  - binlog 逻辑日志
  - innodb redo log 每个页的物理更改情况
- 写入时间不同
  - binlog 仅在事务提交时写

日志相关参数

- innodb_log_file_size 单个日志文件大小
- innodb_log_files_in_group 日志文件数量
- 日志总大小
  - 太小 更频繁的触发checkpoint
  - 太大 崩溃恢复时间长

日志缓存

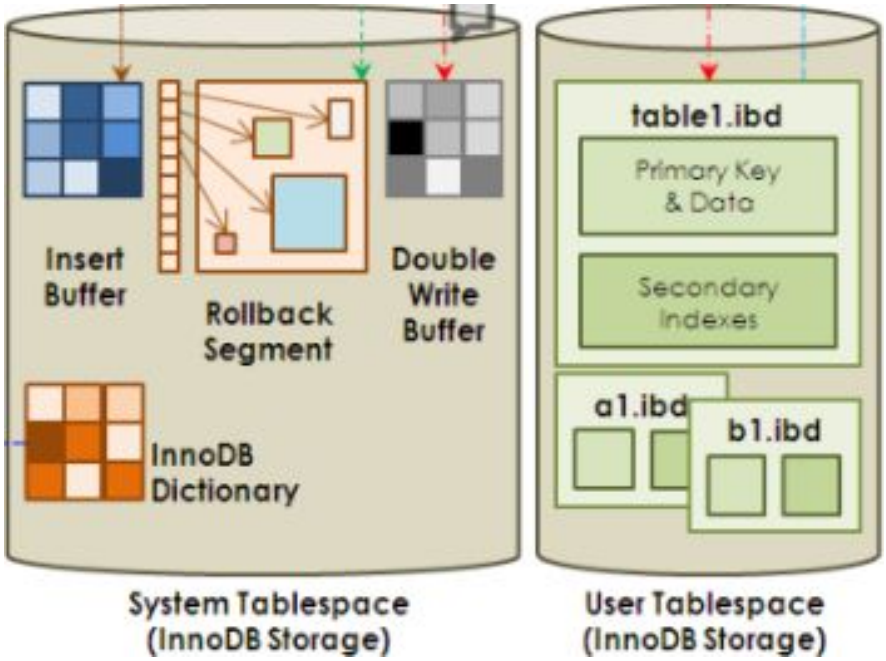
- 提高日志写入性能
- innodb_log_buffer_size 参数
- 日志写回触发:
  - 日志缓存满了
  - 每秒触发
  - 事务提交时
- innodb_flush_log_at_trx_commit 参数
  - 0 事务提交时不写回
  - 1 事务提交时写回 + fsync (默认)
  - 2 事务提交时写回

表空间(table space)

- 数据的物理存储结构
- 配置参数
  - innodb_data_home_dir
  - innodb_data_file_path
    - ibdataX 只增不减
    - ibdata1:1G;ibdata2:1G;ibdata3:1G:autoextend
  - innodb_file_per_table 每个表单独文件存储
    - 5.6.6后默认开启

表空间内容

- system tablespace
  - insert buffer, 用于缓存二级非唯一索引写入
  - 事务回滚记录
  - double write buffer
  - data dictionary 元信息
- user tablespace
  - 聚簇索引+数据
  - 二级索引



表空间文件格式

- 格式
  - Antelope
  - Barracuda 压缩表, 动态列结构
  - ...
- 参数
  - innodb_file_format
  - 需打开innodb_file_per_table

DROP TABLE触发缓存锁

- http://bugs.mysql.com/bug.php?id=51325
- DDL触发缓存锁
- 解决办法innodb_lazy_drop_table
- 慎线上DDL!

脏页(从缓存)写回

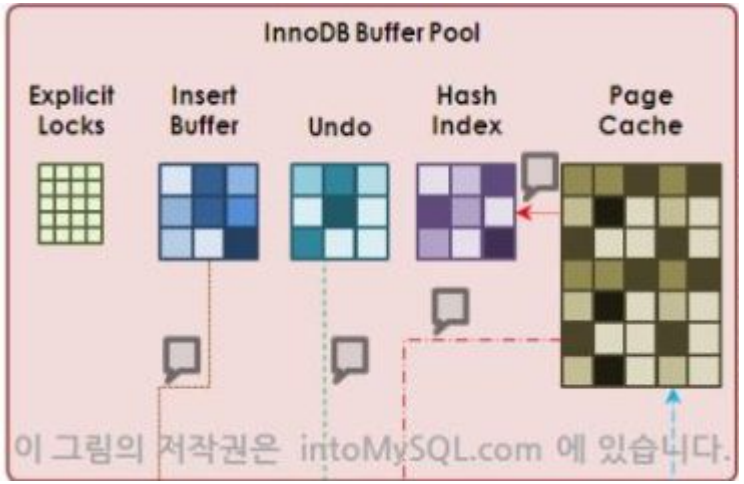
- 由checkpoint触发
  - 缓存不够了
  - 日志不够了
  - 脏页比例太高了
    - innodb_max_dirty_pages_pct
  - innodb_adaptive_flushing
  - ...
- innodb_flush_method
  - O_DIRECT 避免OS的IO缓存
  - ...

doublewrite buffer

- 数据页每次写回, 两次写操作
  - 写入doublewrite buffer
  - 写入真正页面
- 目的:
  - 保证数据页原子性写入
- 开关参数innodb_doublewrite

缓存(buffer pool)结构

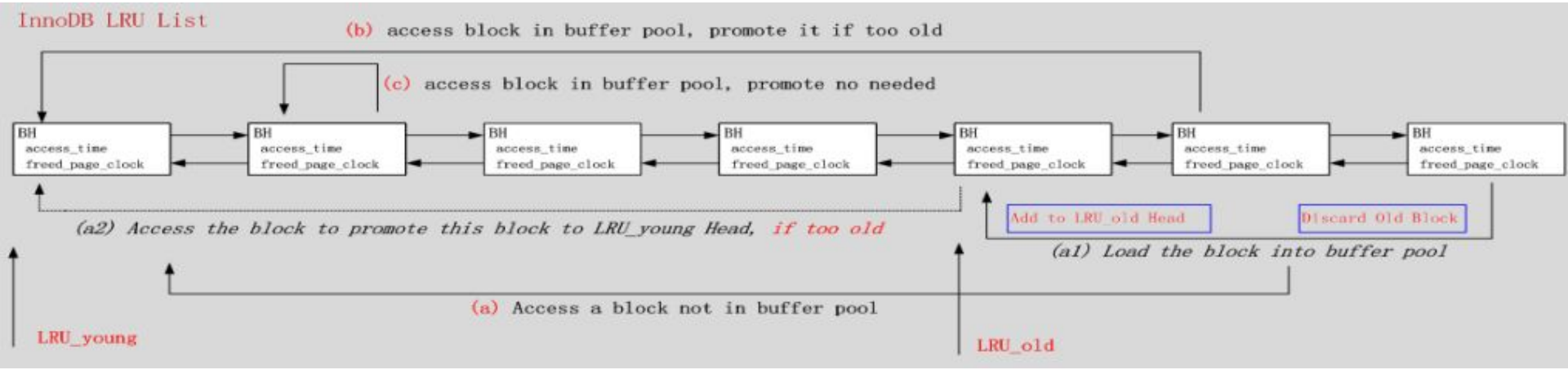
- 什么在缓存里?
  - adaptive hash index
  - insert buffer cache
  - undo buffer cache
  - page cache
- 参数
  - innodb_buffer_pool_size
    - 非常重要 一般设到 60%-80% 系统内存大小
  - innodb_buffer_pool_instances
    - 多实例, 缓解内存锁争用



缓存换页

- 请求磁盘新页, 且内存中可用页不够, 必须换出页以腾出空间
- LRU算法
  - least recently used
  - 换出最近最少使用的页

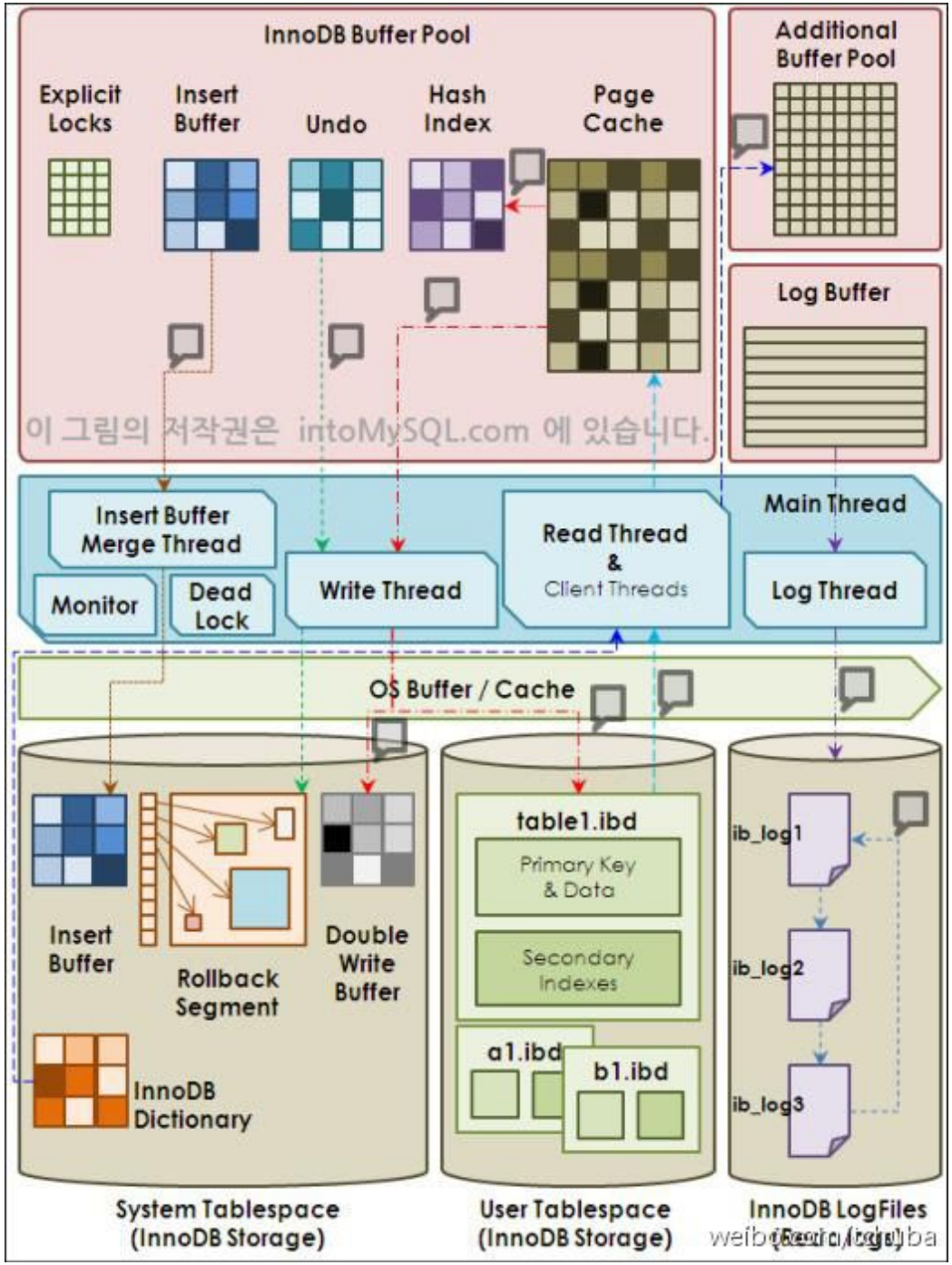
LRU缓存换页详解



- LRU 双向链表, 两部分
  - LRU_young 频繁访问的页
  - LRU_old 不经常频繁访问的页
- Block Header (BH)
  - access time 页第一次被真正访问的时间
  - freed_block_clock 参数 决定是否移动位置
- 页换入和提升流程

LRU相关参数

- innodb_old_blocks_pct
  - 默认 3 / 8
- innodb_old_blocks_time
  - 第一次访问页, 不提升至LRU_young, 只设置access_time
  - 第二次访问页, 当前时间与access_time差值大于此参数, 提升; 否则不提升
  - 避免扫表操作对于缓存造成的抖动
  - 默认0


InnoDB Buffer Pool
Explicit Locks
Insert Buffer
Undo
Hash Index
Page Cache
Additional Buffer Pool
Log Buffer
이 그림의 저작권은 intoMySQL.com 에 있습니다.
Insert Buffer Merge Thread
Monitor
Dead Lock
Write Thread
Read Thread & Client Threads
Main Thread
Log Thread
OS Buffer / Cache
Insert Buffer
Rollback Segment
Double Write Buffer
InnoDB Dictionary
table1.ibd
Primary Key & Data
Secondary Indexes
a1.ibd
b1.ibd
ib_log1
ib_log2
ib_log3
System Tablespace (InnoDB Storage)
User Tablespace (InnoDB Storage)
InnoDB LogFiles

innodb_buffer_pool_size

innodb_buffer_pool_instance

innodb_old_blocks_pct

innodb_old_blocks_time

innodb_max_dirty_pages_pct

innodb_adaptive_flushing

innodb_flush_method

innodb_data_home_dir

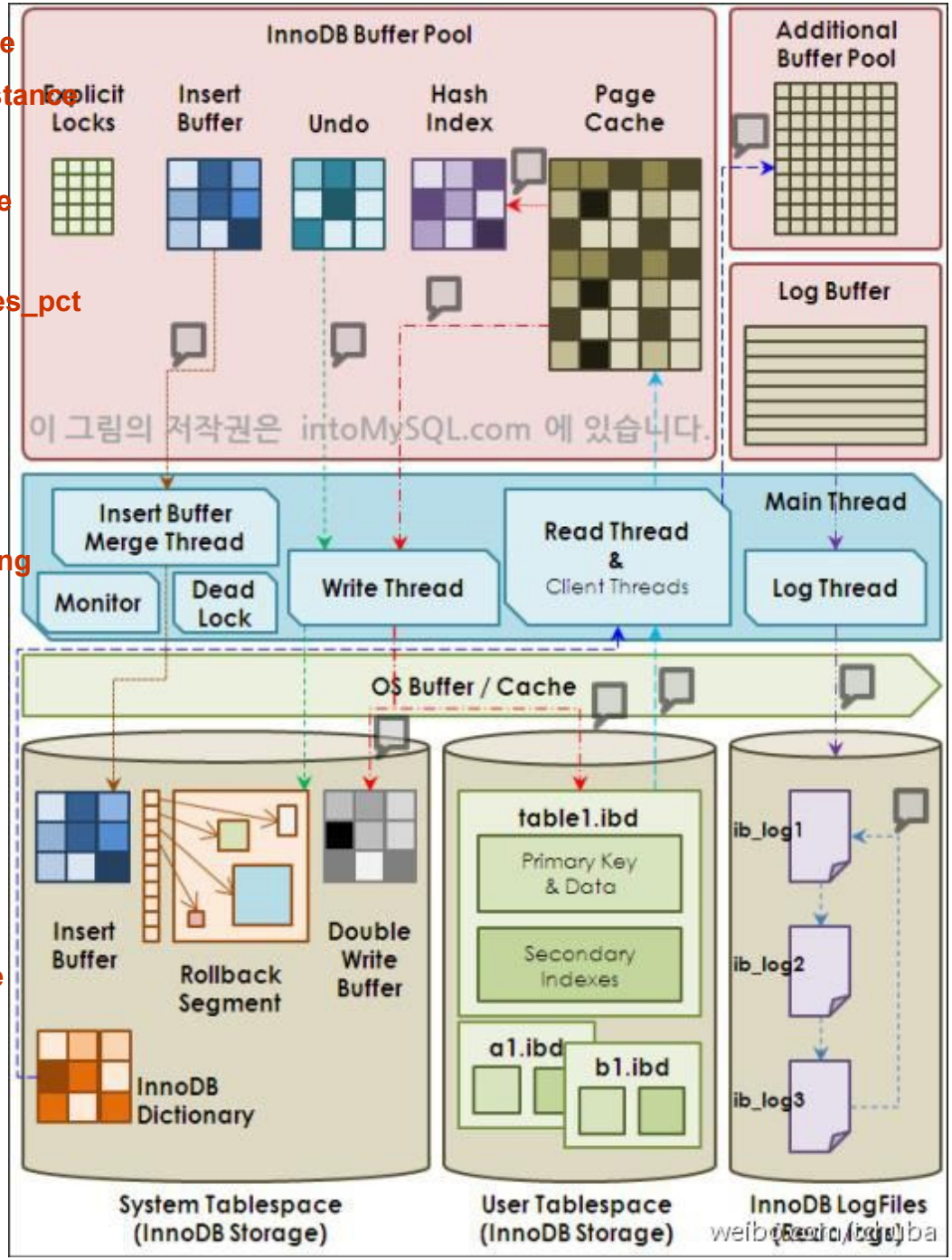
innodb_data_file_path

innodb_file_per_table

innodb_lazy_drop_table

innodb_doublewrite

innodb_file_format



innodb_additional_mem_po

innodb_log_pool_size

innodb_flush_log_at_trx_co

innodb_read_io_threads

innodb_write_io_threads

innodb_log_file_size

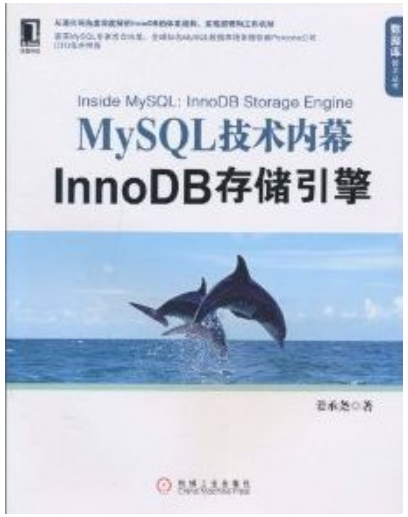
innodb_log_files_in_group

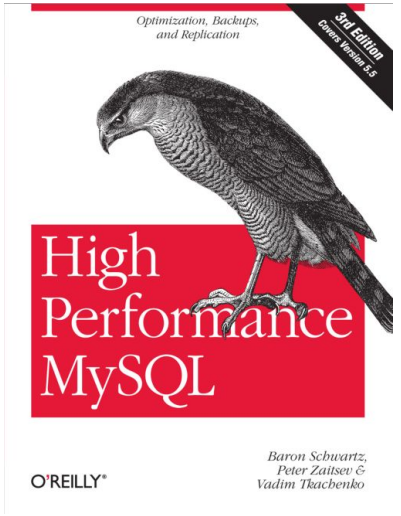
大纲

- innodb索引结构
  - 聚簇索引/二级索引/和MyISAM区别
  - 多列索引顺序
- innodb的IO相关参数概念理解
  - log
  - log buffer
  - table space
  - buffer pool

参考资料

- MySQL reference manual
- High Performance MySQL
- MySQL内核-Innodb存储引擎 姜承尧
- 何登成 博客





感谢大家的时间