

夏日充电季

VIP年卡送100次下载

送3个月

立抢超低价

免费领AI权益，限时抽免单！

关闭

CSDN

博客

下载

学习

社区

GitCode

InsCodeAI

会议

搜CSDN

搜索

AI 搜索

登录

会员中心

消息

历史

创作中心

创作

傲骄鹿先生

博客等级

码龄9年

168

原创

1206

点赞

5447

收藏

1万+

粉丝

关注

私信

博文代码项目一键运行

项目源码一键打包带走

立即体验

SSL证书申请

HTTPS认证

单域名 20元

通配符 199元

🔒🔥SSL证书+单域名证书20元+通配符证书199元!🔥

广告

热门文章

Linux—vi命令详解 546235

Java中final、finally和finalize的区别 84072

Linux——scp的用法 51278

Java基础面试（核心重点） 48688

MySQL面试经典100题（收藏版，附答案） 24190

分类专栏

架构师进阶路线 24篇

1、架构师必备工具箱 7篇

2、架构师内功心法 12篇

3、分布式与高并发

4、展开全部 9篇

上一篇：MySQL优化（二）MySQL体系结...

下一篇：Linux CentOS8通过RPM方式安...

最新评论

IDEA @SpringBootApplication报错原因... 自由岛499: 谢谢 一点用都没有

aspose-words、itextpdf完美解决java将w... czl1sb: 为什么转出来的pdf字体有粗有细

centos7安装网卡驱动 傲骄鹿先生: 这个好像是一个大学的镜像，他们有没有维护我不确定，过一段时间再...

centos7安装网卡驱动 m0\_59151092: 大佬，现在这个网址用不了了吗😭

MySQL面试经典100题（收藏版，附答案） qq\_44151164: 有pdf文档吗？

大家在看

P1352 没有上司的舞会 (树型dp)

MCU外设初始化：为什么参数配置必须优先于使能

小杰python高级（four day）——matplotlib库 1

## MySQL优化（三）详细解读InnoDB存储引擎

原创 傲骄鹿先生 于 2021-04-17 15:32:32 发布 阅读量1.1k 收藏 2 点赞数 2 CC 4.0 BY-SA版权

分类专栏： MySQL 文章标签： mysql InnoDB存储引擎

MySQL 专栏收录该内容 11 篇文章 订阅专栏

写在前面：

我热爱技术，热爱分享，热爱生活， 我始终相信：技术是开源的，知识是共享的！

博客里面的内容大部分均为原创，是自己日常的学习记录和总结，便于自己在后面的时间里回顾，当然也是希望可以分享自己的知识。目前的内容几乎是基础知识和技术入门，如果你觉得还可以的话不妨关注一下，我们共同进步！

个人除了分享博客之外，也喜欢看书，写一点日常杂文和心情分享，如果你感兴趣，也可以关注关注！

微信公众号：傲骄鹿先生

### 目录

- 一、InnoDB体系架构

二、内存

三、线程

四、checkpoint

五、Master Thread工作方式

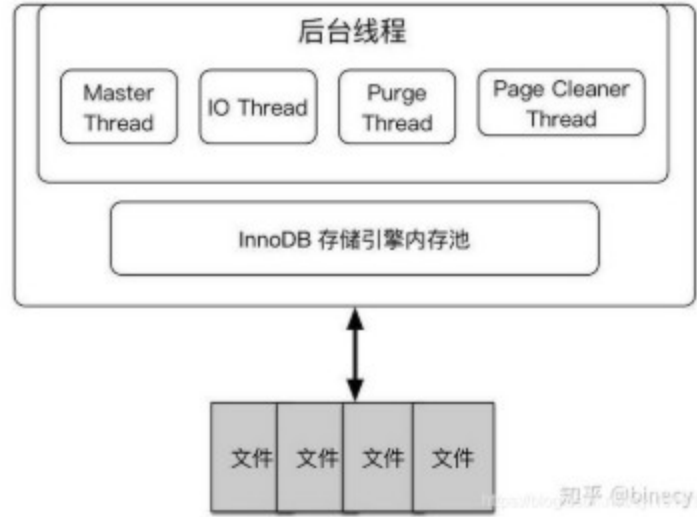
六、InnoDB关键特性

七、InnoDB的启动、关闭与恢复

InnoDB独立表空间，支持MVCC，行锁设计，提供一致性非锁定读，支持外键，插入缓冲，二次写，自适应哈希索引，预读使用聚集的方式存储数据，每张表的存储都是按主键顺序存放。

InnoDB是事物安全的MySQL 存储引擎，设计上采用了类似Oracle数据库的架构。通常来说，InnoDB存储引擎是OLTP应用中核心表的首选存储引擎，也正是因为其存在，才使得MySQL数据库变得更有魅力。

### 一、InnoDB体系架构



#### 1、后台进程：

- 1.1 Master Thread：核心线程，负责缓冲池的数据异步入盘，包括脏页刷新、合并插入缓冲、undo页回收等。

1.2 IO Thread：包括read thread 和writer thread，使用show variables like '%innodb\_%io\_thread%';查看。

1.3 Purge Thread：回收事务提交后不再需要的undo log，通过show variables like '%innodb\_purge\_threads%'; 查看。

1.4 Page clear thread：脏页的刷新操作，从master thread分离出来。

#### 2、InnoDB存储引擎有多个内存块，可以认为这些内存块组成了一个大的内存池，负责如下的工作：

- 维护所有进程/线程需要使用的多个内部数据结构

缓存磁盘上的数据，方便快速地读取，同时对磁盘文件数据修改之前在这里缓存

重做日志（redo log）缓存

InnoDB内存池主要有以下部分：

- 缓冲池

重做日志缓冲



aspose-words、itextpdf完美解决java将w...  
czl1sb: 为什么转出来的pdf字体有粗有细  
centos7安装网卡驱动  
傲骄鹿先生: 这个好像是一个大学的镜像，他们有没有维护我不确定，过一段时间再...  
centos7安装网卡驱动  
m0\_59151092: 大佬，现在这个网址用不了了吗 😭  
MySQL面试经典100题（收藏版，附答案）  
qq\_44151164: 有pdf文档吗？

大家在看

P1352 没有上司的舞会（树型dp）  
MCU外设初始化：为什么参数配置必须优先于使能  
小杰python高级（four day）——matplotlib库 1  
CANoe快速入门实战（第八章）-CAPL Test Module开发  
Android 13.0 第三方app接收不到开机广播问题的解决以及开机自启动功能实现一 3

最新文章

Ruoyi导出excel，导出对象的子列表  
阿里云centos7.9服务器磁盘挂载，切换服务路径  
Linux新建文件详解  
  
2024年 4篇      2023年 10篇  
2022年 10篇      2021年 21篇  
2020年 33篇      2019年 42篇  
2018年 12篇      2017年 55篇  
2016年 7篇

广告

烈火超变

5748/6879

【圣神】落星·破荒者

【武器】99阶 攻击+80%  
技能加持: 攻击力+50%  
暴击率+50%

火爆 狂热1区 点击换区

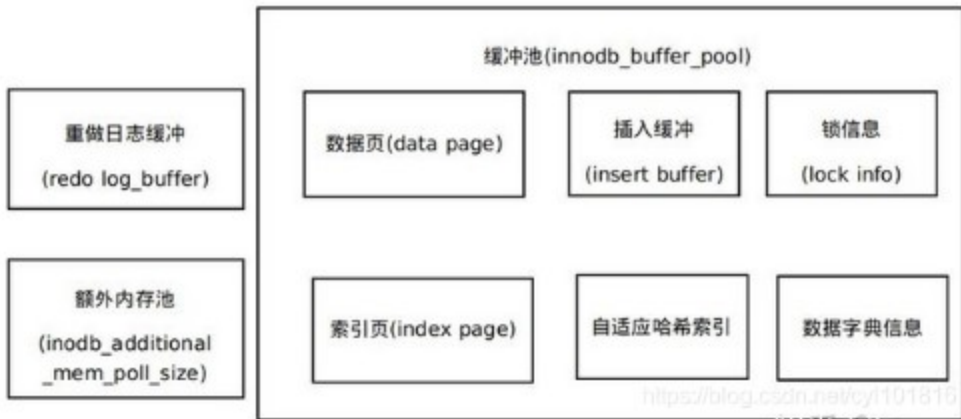
开始游戏

目录

- 一、InnoDB体系架构
- 二、内存
- 三、线程
- 四、checkpoint

二、内存

2.1 缓冲池



从上图来看，主要包括数据页、索引页、undo页、insert buffer、adaptive hash index、数据字典等，其中索引页和数据页占用多数内存。

配置innodb\_pool\_buffer\_instances将缓冲池分割为多个实例，减少内部竞争(比如锁)。

InnoDB是基于磁盘存储的，并将其中的记录按照页的方式进行管理。而缓冲池就是一块内存区域，主要缓冲数据页和索引页。  
InnoDB中对页的读取操作，首先判断该页是否在缓冲池中，若在，直接读取该页，若不在则从磁盘读取页数据，并存放在缓冲池中。  
对页的修改操作，首先修改在缓冲池中的页，再以一定的频率（Checkpoint机制）刷新到磁盘。参数：innodb\_buffer\_pool\_size设置缓冲池大小

缓冲池通过LRU（Latest Recent Used，最近最少实用）算法进行管理。最频繁使用的页在LRU列表前端，最少使用的页在尾端，当缓冲池不能存放新读取的页时，首先释放LRU列表尾端的页（页数据刷新到磁盘，并从缓冲池中删除）。  
InnoDB对于新读取的页，不是放到LRU列表最前端，而是放到midpoint位置（默认为5/8处）。这是因为一些SQL操作会访问大量的页（如全表扫描），读取大量非热点数据，如果直接放到首部，可能导致真正的热点数据被移除。

2.2 LRU list、free list、flush list

默认的缓冲页大小是16KB，使用LRU算法进行管理，新从磁盘加载的页默认加到LRU列表的midpoint处（尾端算起37%位置处）。通过show engine innodb status输出如下（部分）：

AI写代码 登录复制			
1	Buffer pool size	512	【缓冲池内存512*16K】
2	Free buffers	256	
3	Database pages	256	【LRU列表占用页】
4	Old database pages	0	
5	Modified db pages	0	
6	Pending reads	0	
7	Pending writes: LRU	0, flush list	0, single page
8	Pages made young	0, not young	0
9	0.00	young/s, 0.00	non-young/s
10	Pages read and created	0, writes	展开

LRU列表中的页被修改后变为dirty page，此时缓冲池中的页和磁盘不一致，通过checkpoint刷回磁盘，其中Flush list即为dirty page列表。

2.3 重做日志缓冲

重做日志先放到这个缓冲区，然后按一定频率刷新到重做日志文件。配置参数：innodb\_log\_buffer\_size，默认是8MB，

刷新规则：

- 1. Master Thread每秒将一部分重做日志缓冲刷新到重做日志文件
- 2. 每一事务提交时会将重做日志刷新到重做日志文件（如果配置了）
- 3. 重做日志缓冲区使用空间大于1/2

2.4 额外的内存池

内存堆，对InnoDB内部使用的数据结构对象进行管理。在对一些数据结构本身的内存进行分配时，需要从额外的内存池中申请，当该区域的内存不够时，会从缓冲池中申请。

三、线程

InnoDB存储引擎是多线程的模型，因此后台有多个不同的后台线程，负责处理不同的任务。

主要作用：

- 负责刷新内存池中的数据，保证缓冲池的内存缓冲的是最近的数据
- 已修改的数据文件刷新到磁盘文件
- 保证数据库发生异常的情况下InnoDB能恢复到正常状态。



五、Master Thread工作方式

六、InnoDB关键特性

七、InnoDB的启动、关闭与恢复

InnoDB运行时主要有以下线程：

Master Thread

负责将缓冲池中的数据异步刷新到磁盘，保证数据的一致性，包括脏页的刷新，合并插入缓冲（INSERT BUFFER），UNDO页的回收等。

IO Thread

负责AIO请求的回调处理。

参数：innodb\_read\_io\_threads, innodb\_write\_io\_threads

Purge Thread

事务提交后，undo log可能不再需要，由Purge Thread负责回收并重新分配的这些已经使用的undo 页。

注意：Purge Thread需要离散地读取undo页。

Page Cleaner Thread

InnoDB 1.2.x引入，将Master Threader中刷新脏页的工作移至该线程，如上面说的FLUSH LRU LIST Checkpoint以及Async/Sync Flush Checkpoint。

四、checkpoint

当每次执行update、delete等语句更改记录时，缓冲池中的页与磁盘不一致，但是缓冲池的页不能频繁刷新到磁盘中（频率过大性能低），因此增加了write ahead log策略，当事务提交时先写重做日志，再修改内存页。当发生宕机时通过重做日志来恢复。checkpint解决以下问题：

- （1）减少重做日志大小，缩减数据恢复时间。
- （2）缓冲池不够用时将脏页刷回磁盘。
- （3）重做日志不可用时将脏页刷回磁盘（如写满）。

show variables like 'innodb\_max\_dirty\_pages\_pct'; (默认75%)来控制innododb强制进行checkpoint。

若每个重做日志大小为1G，定了了两个总共2G，则：

asyn\_water\_mark = 75 % \* 重做日志总大小。

syn\_water\_mark = 90 % \* 重做日志总大小。

- （1）当checkpoint\_age < asyn\_water\_mark时则不需要刷新脏页回盘。
- （2）当syn\_water\_mark < checkpoint\_age < syn\_water\_mark 时触发ASYNC FLUSH。
- （3）当checkpoint\_age>syn\_water\_mark触发sync flush，此情况很少发生，一般出现在大量load data或bulk insert时。

五、Master Thread工作方式

Master Thread具有最高的线程优先级，内部由多个循环组成：主循环（loop），后台循环(backgroup loop)，刷新循环(flush loop)，暂停循环(suspend loop)，Master Thread根据数据库运行状态在以上循环切换。

Master Thread主要流程伪代码如下：

AI写代码 登录复制

```
1 void master_thread() {
2     goto loop;
3     // 主循环
4     loop:
5     for(int i = 0; i < 10; i++) {
6         // 每秒一次操作
7         thread_sleep(1)
8         // 日志缓冲刷新到磁盘，即使这个事务没提交
9         do log buffer flush to disk
10        // 合并插入缓冲（如要前一秒IO次数 展开 InnoDB认为IO压力很小，执行该操作）
```

如上所示，主循环有两大操作，每秒操作和十秒操作。

InnoDB1.0.x优化：

在每秒操作中，Master Thread每次最多刷新100个脏页（脏页比例超过innodb\_max\_dirty\_pages\_pct），合并20个插入缓冲，如果在写入密集的应用，处理速度可能太慢了。

从InnoDB 1.0.x开始，提供了通过innodb\_io\_capacity参数

每秒操作中合并插入缓冲数量为innodb\_io\_capacity \* 5%  
刷新脏页数量为innodb\_io\_capacity  
而默认innodb\_max\_dirty\_pages\_pct参数值从90调整为75

引入以下参数innodb\_adaptive\_flushing：自适应刷新  
脏页比例小于innodb\_max\_dirty\_pages\_pct，也会刷新一定量的脏页（由InnoDB控制刷新策略和数量）  
innodb\_purge\_batch\_size：控制每次full purge回收Undo页，默认还是20

InnoDB1.2.x优化：

- InnoDB空闲时，执行原来的10秒一次操作，繁忙时，执行原来的每秒一次操作
- 刷新脏页操作，分离到单独Page Cleaner Thread

六、InnoDB关键特性

InnoDB关键特性包括：



- Insert buffer （插入缓冲）
- double write （两次写）
- adaptive hash index （自适应哈希索引）
- Async IO （异步IO）
- Flush neighbor page （刷新临近页）

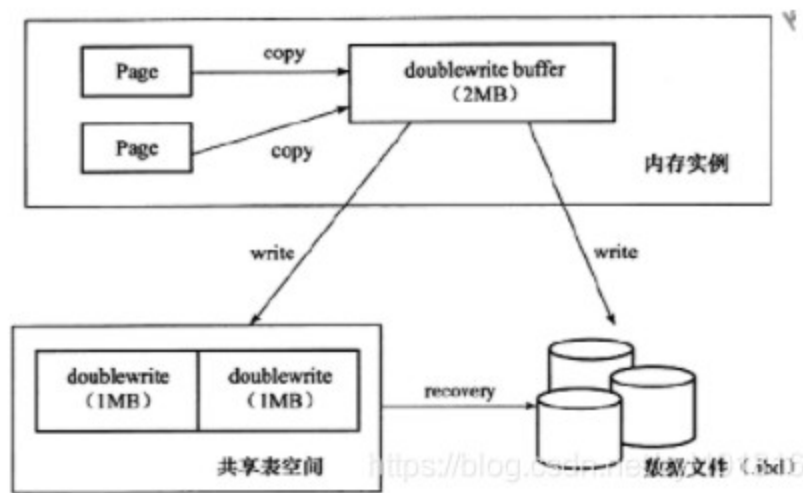
Insert buffer

若插入按照聚集索引primary key插入，页中的行记录按照primary存放，一般情况下不需要读取另一个页记录，插入速度很快（如果使用UUID或者指定的ID插入而非自增类型则可能导致非连续插入导致性能下降，由B+树特性决定）。如果按照非聚集索引插入就很有可能存在大量的离散插入，insert buffer对于非聚集索引的插入和更新操作进行一定频率的合并操作，再merge到真正的索引页中。使用insert buffer需满足条件：

- （1）索引为辅助索引。
- （2）索引非唯一。（唯一索引需要从查找索引页中的唯一性， 可能导致离散读取）

Double write

Doubel write保证了页的可靠性，Redo log是记录对页(16K)的物理操作，若innodb将页写回表时写了一部分(如4K)出现宕机，则物理页将会损坏无法通过redolog恢复。所以在apply重做日志前，将缓冲池中的脏页通过memcpy到doublewrite buffer中，再将doublewrite buffer页分两次每次1MB刷入共享表空间的磁盘文件中（磁盘连续，开销较小），完成doublewrite buffer的页写入后再写入各个表空间的表中。



当写入页时发生系统崩溃，恢复过程中，innodb从共享表空间的doublewrite找到该页的副本，并将其恢复到表空间文件中，再apply重做日志。

Adaptive hash index

Innodb根据访问频率对热点页建立哈希索引，AHI的要求是对页面的访问模式必须一样，如连续使用where a='xxx' 访问了100次。建立热点哈希后读取速度可能提升两倍，辅助索引连接性能提升5倍。

通过show engine innodb status\G;查看hash searches/s, 表示使用自适应哈希，对于范围查找则不能使用。

Async IO

用户执行一次扫描如果需要查询多个索引页，可能会执行多个IO操作，AIO可同时发起多个IO请求，系统自动将这些IO请求合并（如请求数据页[1, 2]、[2,3]则可合并为从1开始连续扫描3个页）提高读取性能。

刷新临近页

InnoDB提供刷新临近页功能：当刷新一脏页时，同时检测所在区(extent)的所有页，如果有脏页则一并刷新，好处则是通过AIO特性合并写IO请求，缺点则是有些页不怎么脏也好被刷新，而且频繁的更改那些不怎么脏的页又很快变成脏页，造成频繁刷新。对于固态硬盘则考虑关闭此功能（将innodb\_flush\_neighbors设置为0）。

七、InnoDB的启动、关闭与恢复

innodb\_fast\_shutdown

该值影响数据库正常关闭时的行为，取值可以为0/1/2（默认为1）：

【为0时】：关闭过程中需要完成所有的full purge好merge insert buffer，并将所有的脏页刷新回磁盘，这个过程可能需要一定的时间，如果是升级InnoDB则必须将此参数调整为0再关闭数据库。

【为1时(默认)】：不需要full purge和merge insert buffer，但会将缓冲池中的脏页写回磁盘。

【为2时】：不需要full purge和merge insert buffer，也不会将缓冲池中的脏页写回磁盘，而是将日志写入日志文件中，后续启动时recovery。

innodb\_force\_recovery

参数innodb\_force\_recovery直接影响InnoDB的恢复情况。

默认值为0：进行所有的恢复操作，当不能进行有效恢复（如数据页corrupt）则将错误写入错误日志中。

某些情况下不需要完整的恢复造成，则可定制恢复策略，参数innodb\_force\_recovery还可以设置为6个非零值：1-6，大的数字表示包含了前面所有小数字表示的影响。有以下几种：

- 1（SRV\_FORCE\_IGNORE\_CORRUPT）：忽略检查到的corrupt页。
- 2（SRV\_FORCE\_NO\_BACKGROUND）：阻止Master Thread线程运行，如果master thread需要进行full purge操作，这样会导致crash。
- 3（SRV\_FORACE\_NO\_TRX\_UNDO）:不进行事务的回滚操作。



- 4（SRV\_FORCE\_NO\_IBUF\_MERGE）:不进行插入缓冲区的合并操作。
- 5（SRV\_FORCE\_NO\_UNDO\_LOG\_SCAN）:不查看undo log，这样未提交的事务被视为已提交。
- 6（SRV\_FORCE\_NO\_LOG\_REDO）:不进行redo操作。

在设置了innodb\_force\_recovery大于0后可对表进行select/create/drop操作，但不能进行insert update和delete等DML。如有大事务未提交，并且发生了宕机，恢复过程缓慢，不需要进行事务回滚则将参数设置为3以加快启动过程。

 **傲骄鹿先生** 沉淀，成长，分享，让自己和他人都有所收获

 微信公众号 >

mysql io thread\_MySQL IO线程及相关参数调优weixin\_42355387的博客1173

一、关于一个SQL的简单的工作过程1、工作前提描述1、启动MySQL，在内存中分配一个大空间innodb\_buffer\_p...

MySQL——Innodb存储引擎的索引原理详细解读weixin\_41438466的博客458

0 前言 回想大二上数据库应用，老师在讲索引的时候，是像下面这么说的 索引就像一本书的目录。而当用户通过...

MySQL启动停止重启脚本及my.cnf配置参数优化详解\_mysql启动指定my.cnf...8-7

innodb\_read\_io\_threads = 4 # innodb使用后台线程处理数据页上的读写 I/O(输入输出)请求,根据你的 CPU 核数来...

MySQL 数据库的优化\_mysql my.cnf配置优化7-21

用来指定 innodb tablespace 文件,如果我们不在 my.cnf 文件中指定innodb\_data\_home\_dir 和innodb\_data\_file\_p...

Mysql5.7 InnoDB线程u010648194的博客2666

一、查看线程 在performance\_schema数据库中的， threads表可以查看。 select thread\_id,name,type,processlist...

MySQL之InnoDB存储引擎深度解析 最新发布aa\_hdkf\_vg的博客961

InnoDB存储引擎是当前应用最广泛的存储引擎之一，以出色的事务处理能力、数据完整性保障以及高并发性能，...

MySQL配置文件my.cnf详解7-24

#innodb\_buffer\_pool\_size 一致 可以开启多个内存缓冲池,把需要缓冲的数据hash到不同的缓冲池中,这样可以并行...

mysql my.cnf配置详解\_mysql 在my.cnf 配置listener7-31

mysql my.cnf配置详解 本文详细解析MySQL配置文件的各项参数,涵盖基础设置、日志管理、主从复制、InnoDB...

innoDB 储存引擎 IO Thread 后台线程 innodb\_read\_io\_threads 和 in...weixin\_40896800的博客8367

在innodb储存中大量使用了AIO（Async IO）来处理些IO请求，这样可以极大提高数据库性能。而IO Thread 的主...

MySQL利用多核处理器 innodb\_read\_io\_threadsBicction2929

在工作中可能遇到这样的情况，随着业务的增长，用户量也在逐渐增长，终究有一天，一到高峰期，数据库服务...

MySQL8--my.cnf配置文件的设置\_mysql8 my.cnf8-6

innodb\_flush\_sync = 0 innodb\_flush\_neighbors = 0 #CPU多核处理能力设置,假设CPU是4颗8核的,设置如下 #读...

linux /etc/my.cnf 参考配置(其中mysql安装路径需要批量修改为当前数据...8-8

本文详细解析MySQL 5.7版本的各项配置参数,包括服务端、客户端、事务隔离级别、数据库引擎与字符集、连接...

MySQL调优之服务器参数优化实践小小默：进无止境3426

\*\*关联博文： \*\* [MySQL中常见的优化策略详解](https://blog.csdn.net/J080624/article/details/53199410) 接上文[M...

【MySQL】一文带你理清InnoDB引擎的<内部架构>（内存结构，磁盘结构...YYDsis的博客1313

一.架构1.内存结构1.缓冲池：Buffer Pool2.更改缓冲区：Change Buffer.自适应哈希索引：Adaptive Hash index4....

mysql数据库部署-标准my.cnf配置\_mysql my.cnf配置7-18

MySQL从8.0开始undo 表空间管理已经发生了改变,在5.7版本中一旦MySQL初始化以后,就不能再改变undo表空间...

mysql之InnoDB Buffer Pool 深度解析与性能优化qq\_43460315的博客1763

InnoDB Buffer Pool 是 MySQL InnoDB 存储引擎中至关重要的内存组件，它作为磁盘数据页和 CPU 之间的缓冲...

在MySQL 5.6中，如何正确配置和优化InnoDB存储引擎，以提高数据操作效率和事务处理性...11-07

要充分挖掘MySQL 5.6中InnoDB存储引擎的性能潜力，首先需要确保正确配置。InnoDB存储引擎以其强大的事务...

【MySQL-23】万字总结<InnoDB引擎>——【逻辑存储结果&架构(内存结...YYDsis的博客1643

总结/概要一.逻辑存储结构二.架构1.内存结构1.缓冲池：Buffer Pool2.更改缓冲区：Change Buffer3.自适应哈希索...

【mysql】关于IO/内存方面的一些优化weixin\_33743248的博客428

这里使用的是mysql Ver 14.14 Distrib 5.6.19, for Linux (i686) using EditLine wrapper 一、mysql目录文件 ibdata...

Mysql之InnoDB多线程详解qq\_38571892的博客2438

InnoDB存储引擎是多线程的模型，因此其后台有多个不同的后台线程，负责处理不同的任务。主要分为：Master ...

开源数据库 - mysql - innodb线程weixin\_46325250的博客1579

MySQL的InnoDB存储引擎是一个多线程的模型，其后台包含多个不同的线程，负责处理不同的任务。这些线程共...

mysql参数优化sinat\_36355775的博客3785

mysql一些优化参数： 1.innodb\_write\_io\_threads和innodb\_read\_io\_threads 假如CPU是2颗8核的，那么可以设...

MySQL 参数设置优化LZN的博客825

文章目录一、MySQL 体系结构二、内存调优2.1 QueryCache2.2 Buffer Cache2.2.1 innodb\_buffer\_pool\_size2.2....

MySql 配置优化weixin\_43871535的博客5202

优化思路 针对MySQL优化，无非是涉及到内存、IO、CPU的优化，该文档将依次从这三个方面来介绍。问题定...

架构设计：系统存储（6）——MySQL数据库性能优化（2）395

=====（接上文《架构设计：系统存储（5）——MySQL数据库性能优...

InnoDB 性能调优 – 优化 MySQL 数据库的关键 InnoDB 变量taobao2024api的博客557

InnoDB 是 MySQL 的核心存储引擎，即使在最具挑战性的生产环境中也以其可靠性和性能而闻名。要真正优化 In...



