# 概述

InnoDB是事务安全的MySQL存储引擎，是OLTP应用中核心表的首选存储引擎。

# 体系架构

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，作用包括：

1. 维护所有进程/线程需要访问的多个内部数据结构；
2. 缓存磁盘上的数据，方便快速地读取，同时在对磁盘文件的数据修改之前

在这里缓存；

1. 重做日志（redo log）缓冲。

后台线程的主要作用是负责刷新内存池中的数据，保证缓冲池中的内存缓存的是最近的数据。此外将已修改的数据文件刷新到磁盘文件中，同时保证在数据库发生异常的情况下InnoDB能恢复到正常运行状态。

## 后台线程

### Master Thread

Master Thread是一个非常核心的后台线程，主要负责将缓存池中的数据异步刷新到磁盘，保持数据的一致性，包括脏页的刷新、合并插入缓冲（INSERT BUFFER）、UNDO页的回收等。

### IO Thread

在InnoDB存储引擎中大量使用了AIO（Async IO）来处理写IO请求，这样可以极大提高数据库的性能。而IO Thread的工作主要是负责这些IO请求的回调（call back）处理。

### Purge Thread

事务被提交后，所使用的undolog可能不再需要，因此需要Purge Thread来回收已经使用并分配的undo页。

用户可以在MySQL数据库的配置文件中添加innodb\_purge\_threads=1命令来启用独立的Purge Thread。在InnoDB1.1版本中，即使将innodb\_purge\_threads设为大于1，InnoDB存储引擎启动时也会将其设为1。从InnoDB1.2版本开始，InnoDB支持多个Purge Thread，这样做的目的是为了进一步加快undo页的回收。同时由于Purge Thread需要离散地读取undo页，这样也能更进一步利用磁盘的随机读取性能。

### Page Cleaner Thread

Page Cleaner Thread作用是将之前版本中脏页的刷新操作都放入到单独的线程中来完成，其目的为了减轻Master Thread的工作及对于用户查询线程的阻塞，进一步提高InnoDB存储引擎的性能。

## 内存

### 缓冲池

InnoDB存储引擎是基于磁盘存储的，并将其中的记录按照页的方式进行管理。因此，可以将其视为基于磁盘的数据库系统（Disk-base Database）。在数据库系统中，由于CPU速度与磁盘速度之间的差异，基于磁盘的数据库系统通常使用缓冲池技术来提高数据库的整体性能。

对于InnoDB存储引擎而言，其缓冲池的配置通过参数innodb\_buffer\_pool\_size来设置。

存储池中缓存的数据页类型有：索引页、数据页、插入缓存（insert buffer）、自适应哈希索引（adaptive hash index）、InnoDB存储的锁信息（lock info）、数据字典信息（data directory）等。

### LRU List/Free List/Flush List

通常来说，数据库中的缓冲池是通过LRU（Latest Recent Used，最近最少使用）算法来进行管理的。即最频繁使用的页在LRU列表的前端，而最少使用的页在LRU列表的尾端。当缓冲池不能存放新读取到的页时，将首先释放LRU列表中尾端的页。

### 重做日志缓冲

### 额外的内存池

# CheckPoint技术

为了避免在缓冲池到磁盘之间发生数据丢失的问题，当前事务数据库系统普遍都采用了Write Ahead Log策略，即当事务提交时，先写重做日志，再修改页。当由于发生宕机而导致数据丢失时，通过重做日志来完成数据的恢复。这也是事务ACID中D（Durability持久性）的要求。

# 关键特性

## 插入缓冲

## 两次写

## 自适应哈希索引

## 异步IO

## 刷新邻接页

# 启动/关闭/恢复