

# 20230320钉钉一面

[八股](#)

[JVM内存模型](#)

[JVM内存回收机制](#)

[线程](#)

[MySQL索引](#)

[Python](#)

[项目](#)

[python爬虫](#)

[代码注入](#)

[mongodb部署](#)

[算法](#)

## 八股

### JVM内存模型

### JVM内存回收机制

- 1. 主要发生在哪个区
- 2. 垃圾回收过程，以CMS为例？围绕对象的生命周期来讲？
- 3. 新生代分为什么区？新生代变为老年代的阈值

### 线程

线程创建方式：runnable、线程池、callable

<https://segmentfault.com/a/1190000022878543>

### MySQL索引

- 1. 索引数据结构
- 2. b+树和b树区别
- 3. **MySQL数据页概念？页分裂？**

#### 数据页合并（Page Merging）

数据页合并是InnoDB为了优化存储空间使用而进行的一种操作。当一个数据页中的记录被删除，腾出的空间允许其他记录使用时，这个页的空间利用率会降低。InnoDB定义了一个阈值，称为MERGE\_THRESHOLD（默认情况下是页大小的50%）。当一个数据页的使用率低于这个阈值时，InnoDB会考虑将这个页与相邻的页（前一个或后一个）合并，前提是相邻的页也有足够的空间。

合并操作的结果是，两个页中的记录被合并到一个页中，而另一个页则变为空页，可以用于存储新的数据。这个过程可以减少页的碎片化，提高存储效率。页合并通常在删除或更新操作中发生，并且与当前页的相邻页相关。

## 数据页分裂（Page Splitting）

数据页分裂是在插入或更新数据时可能发生的一种操作。当一个数据页已经满了，无法再容纳更多的记录时，InnoDB需要创建一个新的数据页来存储额外的数据。如果下一个应该存储新记录的数据页也已满，InnoDB会执行页分裂操作。

页分裂的过程大致如下：

1. 创建一个新的数据页。
2. 判断当前页可以从哪里进行分裂（记录行层面）。
3. 移动记录行，将一部分数据移动到新的页中。
4. 重新定义页之间的关系，确保B+树的逻辑顺序。

分裂操作后，原来的页和新创建的页都会包含一部分记录，并且它们在B+树中的位置会相应调整。这个过程确保了数据的物理存储顺序与逻辑顺序的一致性，尽管物理上这些页可能位于不同的区（extent）。

4. 主键索引、普通索引、唯一索引、联合索引、覆盖索引
5. 联合索引的使用场景（eg：在众包测试平台中，根据工人id和报告发布日期来查询对应的报告。）
6. 索引的缺点
7. B+树能存放多少数据？

在Innodb存储引擎里面，**最小存储单元是页，而一个页的大小默认是16KB**。一个节点（叶子节点或非叶子节点）的大小就是一页。一条数据必须是放在一个节点中，**不能拆开到多个节点存储**。而一般一条数据大概1KByte，那么一页能容纳的大概16条，即一个节点容纳不止一条数据，而且是多达16条！！**所以大小采用最小存储单元足够用了~**

非叶子结点：

假设主键类型为bigint，占用8Byte，指针可以设置为占用6Byte，总共14Byte。这样就可以算出一个非叶子节点大概可以

存放 $16\text{KByte}/14\text{Byte}=1170$ 个“主键+指针”的组合。  
假设主键类型为int，占用4Byte，指针可以设置为占用6Byte，总共10Byte。这样就可以算出一个非叶子节点大概可以存放 $16\text{KByte}/10\text{Byte}\approx 1600$ 个“主键+指针”的组合。

叶子节点：

在B+树中，真正的数据是只存储在叶子节点中的。这里我们假设我们的一行数据大小是1K，那么我们一个叶子节点就可以存 **$16\text{KByte}/1\text{kByte}=16$ 条（行）数据。**

主键为bigint（约2000w）：

2层B+树的话：可以存放 $1170\text{个}\times 16\text{条}=18720\text{条（行）数据}$ 。

3层B+树的话：可以存放 $1170\text{个}\times 1170\text{个}\times 16\text{条}=21902400\text{条（行）数据}$ 。

主键为int（约4000w）：

2层B+树的话：可以存放 $1600\text{个}\times 16\text{条}=25600\text{条（行）数据}$ 。

3层B+树的话：可以存放 $1600\text{个}\times 1600\text{个}\times 16\text{条}=40960000\text{条（行）数据}$ 。

## Python

python用过的数据结构

## 项目

### python爬虫

1. 使用时遇到什么困难
2. ip地址被ban有什么解决方法？ip代理池

### 代码注入

有什么量化的结果？

### mongodb部署

有什么量化的结果？为什么要自动化部署？

## 算法

零钱兑换，不限制次数