

后端开发常问面试题集锦

——算法问题



欢迎做 Java 的工程师朋友们加入 Java 高级架构进阶群；群内有技术大咖指点难题，还提供免费的 Java 架构学习资料（里面有高可用,高并发,高性能及分布式,Jvm 性能调优,Spring 源码,MyBatis,Netty,Redis,Kafka,Mysql,Zookeeper,Tomcat,Docker,Dubbo,Nginx 等多个知识点的架构资料）

群名称:Java高级架构进阶@群
群 号:963944895

AVL 树、红黑树、BB+树和 Trie 树的比较

AVL 树

最早的平衡二叉树之一。AVL 是一种高度平衡的二叉树，所以通常的结果是，维护这种高度平衡所付出的代价比从中获得的效率收益还大，故而实际的应用不多，更多的地方是用追求局部而不是非常严格整体平衡的红黑树。当然，如果场景中对插入删除不频繁，只是对查找特别有要求，AVL 还是优于红黑的。

使用场景：Windows 对进程地址空间的管理用到了 AVL 树。

红黑树

平衡二叉树，通过对任何一条从根到叶子的简单路径上各个节点的颜色进行约束，确保没有一条路径会比其他路径长 2 倍，因而是近似平衡的。所以相对于严格要求平衡的 AVL 树来说，它的旋转保持平衡次数较少。用于搜索时，插入删除次数多的情况下我们就用红黑树来取代 AVL。

使用场景

- 广泛用在 C++ 的 STL 中。map 和 set 都是用红黑树实现的
- 著名的 linux 进程调度 Completely Fair Scheduler, 用红黑树管理
- 进程控制块
- epoll 在内核中的实现，用红黑树管理事件块
- nginx 中，用红黑树管理 timer 等
- Java 的 TreeMap 实现

B 树、B+树

它们特点是一样的，是多路查找树，一般用于数据库系统中，为什么，因为它们分支多层数少呗，都知道磁盘 IO 是非常耗时的，而像大量数据存储在磁盘中所以我们要有效的减少磁盘 IO 次数避免磁盘频繁的查找。

使用场景

- 主要用在文件系统以及数据库中做索引等，比如 Mysql: B-Tree Index in MySql
- 像 mysql 的数据库定义是可以指定 B+ 索引还是 hash 索引。

Trie 树

又名单词查找树，一种树形结构，常用来操作字符串。它是不同字符串的相同前缀只保存一份。相对直接保存字符串肯定是节省空间的，但是它保存大量字符串时会很耗费内存。

类似的有

前缀树(prefix tree)，后缀树(suffix tree)，radix tree(patricia tree, compact prefix tree)，crit-bit tree（解决耗费内存问题），以及前面说的double array trie。

使用场景

- 字符匹配

前缀树：字符串快速检索，字符串排序，最长公共前缀，自动匹配前缀显示后缀。

后缀树：查找字符串 s1 在 s2 中，字符串 s1 在 s2 中出现的次数，字符串 s1, s2 最长公共部分，最长回文串。

radix tree：linux 内核，nginx。

- IP 选路

也是前缀匹配，一定程度会用到 trie