

50讲索引：如何在海量数据中快速查找某个数据



在第48节中，我们讲了MySQL数据库索引的实现原理。MySQL底层依赖的是B+树这种数据结构。留言里有同学问我，那**类似Redis这样的Key-Value数据库中的索引，又是怎么实现的呢？底层依赖的又是什么数据结构呢？**

今天，我就来讲一下索引这种常用的技术解决思路，底层往往会依赖哪些数据结构。同时，通过索引这个应用场景，我也带你回顾一下，之前我们学过的几种支持动态集合的数据结构。

为什么需要索引？

在实际的软件开发中，业务纷繁复杂，功能千变万化，但是，万变不离其宗。如果抛开这些业务和功能的外壳，其实它们的本质都可以抽象为“对数据的存储和计算”。对应到数据结构和算法中，那“存储”需要的就是数据结构，“计算”需要的就是算法。

对于存储的需求，功能上无外乎增删改查。这其实并不复杂。但是，一旦存储的数据很多，那性能就成了这些系统要关注的重点，特别是在一些跟存储相关的基础系统（比如MySQL数据库、分布式文件系统等）、中间件（比如消息中间件RocketMQ等）中。

“如何节省存储空间、如何提高数据增删改查的执行效率”，这样的问题就成了设计的重点。而这些系统的实现，都离不开一个东西，那就是**索引**。不夸张地说，索引设计得好坏，直接决定了这些系统是否优秀。

索引这个概念，非常好理解。你可以类比书籍的目录来理解。如果没有目录，我们想要查找某个知识点的时候，就要一页一页翻。通过目录，我们就可以快速定位相关知识点的页数，查找的速度也会有质的提高。

索引的需求定义

索引的概念不难理解，我想你应该已经搞明白。接下来，我们就分析一下，在设计索引的过程中，需要考虑到的一些因素，换句话说就是，我们该如何定义清楚需求呢？

对于系统设计需求，我们一般可以从**功能性需求**和**非功能性需求**两方面来分析，这个我们之前也说过。因此，这个问题也不例外。

1.功能性需求

对于功能性需求需要考虑的点，我把它大致概括成下面这几点。

数据是格式化数据还是非格式化数据？要构建索引的原始数据，类型有很多。我把它分为两类，一类是结构化数据，比如，MySQL中的数据；另一类是非结构化数据，比如搜索引擎中网页。对于非结构化数据，我们一般需要做预处理，提取出查询关键词，对关键词构建索引。

数据是静态数据还是动态数据？如果原始数据是一组静态数据，也就是说，不会有数据的增加、删除、更新操作，所以，我们在构建索引的时候，只需要考虑查询效率就可以了。这样，索引的构建就相对简单些。不过，大部分情况下，我们都是对动态数据构建索引，也就是说，我们不仅要考虑到索引的查询效率，在原始数据更新的同时，我们还需要动态地更新索引。支持动态数据集的索引，设计起来相对也要更加复杂些。

索引存储在内存还是硬盘？如果索引存储在内存中，那查询的速度肯定要比存储在磁盘中的高。但是，如果原始数据量很大的情况下，对应的索引可能也会很大。这个时候，因为内存有限，我们可能就不得不将索引存储在磁盘中了。实际上，还有第三种情况，那就是一部分存储在内存，一部分存储在磁盘，这样就可以兼顾内存消耗和查询效率。

单值查找还是区间查找？所谓单值查找，也就是根据查询关键词等于某个值的数据。这种查询需求最常见。所谓区间查找，就是查找关键词处于某个区间值的所有数据。你可以类比MySQL数据库的查询需求，自己想象一下。实际上，不同的应用场景，查询的需求会多种多样。

单关键词查找还是多关键词组合查找？比如，搜索引擎中构建的索引，既要支持一个关键词的查找，比如“数据结构”，也要支持组合关键词查找，比如“数据结构 AND 算法”。对于单关键词的查找，索引构建起来相对简单些。对于多关键词查询来说，要分多种情况。像MySQL这种结构化数据的查询需求，我们可以实现针对多个关键词的组合，建立索引；对于像搜索引擎这样的非结构数据的查询需求，我们可以针对单个关键词构建索引，然后通过集合操作，比如求并集、求交集等，计算出多个关键词组合的查询结果。

实际上，不同的场景，不同的原始数据，对于索引的需求也会千差万别。我这里只列举了一些比较有共性的需求。

2.非功能性需求

讲完了功能性需求，我们再来看，索引设计的非功能性需求。

不管是存储在内存中还是磁盘中，索引对存储空间的消耗不能过大。如果存储在内存中，索引对占用存储空间的限制就会非常苛刻。毕竟内存空间非常有限，一个中间件启动后就占用几个GB的内存，开发者显然是无法接受的。如果存储在硬盘中，那索引对占用存储空间的限制，稍微会放宽一些。但是，我们也不能掉以轻心。因为，有时候，索引对存储空间的消耗会超过原始数据。

在考虑索引查询效率的同时，我们还要考虑索引的维护成本。索引的目的是提高查询效率，但是，基于动态数据集构建的索引，我们还要考虑到，索引的维护成本。因为在原始数据动态增删改的同时，我们也需要动态的更新索引。而索引的更新势必会影响到增删改操作的性能。

构建索引常用的数据结构有哪些？

我刚刚从很宏观的角度，总结了在索引设计的过程中，需要考虑的一些共性因素。现在，我们就来看，对于不同需求的索引结构，底层一般使用哪种数据结构。

实际上，常用来构建索引的数据结构，就是我们之前讲过的几种支持动态数据集的数据结构。比如，散列表、红黑树、跳表、B+树。除此之外，位图、布隆过滤器可以作为辅助索引，有序数组可以用来对静态数据构建索引。

我们知道，**散列表**增删改查操作的性能非常好，时间复杂度是 $O(1)$ 。一些键值数据库，比如Redis、Memcache，就是使用散

列表来构建索引的。这类索引，一般都构建在内存中。

红黑树作为一种常用的平衡二叉查找树，数据插入、删除、查找的时间复杂度是 $O(\log n)$ ，也非常适合用来构建内存索引。Ext文件系统中，对磁盘块的索引，用的就是红黑树。

B+树比起红黑树来说，更加适合构建存储在磁盘中的索引。B+树是一个多叉树，所以，对相同个数的数据构建索引，B+树的高度要低于红黑树。当借助索引查询数据的时候，读取B+树索引，需要的磁盘IO次数非常更少。所以，大部分关系型数据库的索引，比如MySQL、Oracle，都是用B+树来实现的。

跳表也支持快速添加、删除、查找数据。而且，我们通过灵活调整索引结点个数和数据个数之间的比例，可以很好地平衡索引对内存的消耗及其查询效率。Redis中的有序集合，就是用跳表来构建的。

除了散列表、红黑树、B+树、跳表之外，位图和布隆过滤器这两个数据结构，也可以用于索引中，辅助存储在磁盘中的索引，加速数据查找的效率。我们来看下，具体是怎么做的？

我们知道，**布隆过滤器**有一定的判错率。但是，我们可以规避它的短处，发挥它的长处。尽管对于判定存在的数据，有可能并不存在，但是对于判定不存在的数据，那肯定就不存在。而且，布隆过滤器还有一个更大的特点，那就是内存占用非常少。我们可以针对数据，构建一个布隆过滤器，并且存储在内存中。当要查询数据的时候，我们可以先通过布隆过滤器，判定是否存在。如果通过布隆过滤器判定数据不存在，那我们就没有必要读取磁盘中的索引了。对于数据不存在的情况，数据查询就更加快速了。

实际上，有序数组也可以被作为索引。如果数据是静态的，也就是不会有插入、删除、更新操作，那我们可以把数据的关键词（查询用的）抽取出来，组织成有序数组，然后利用二分查找算法来快速查找数据。

总结引申

今天这节算是一节总结课。我从索引这个非常常用的技术方案，给你展示了散列表、红黑树、跳表、位图、布隆过滤器、有序数组这些数据结构的应用场景。学习完这节课之后，不知道你对这些数据结构以及索引，有没有更加清晰的认识呢？

从这一节内容中，你应该可以看出，架构设计离不开数据结构和算法。要想成长为一个优秀的业务架构师、基础架构师，数据结构和算法的根基一定要打稳。因为，那些看似很惊艳的架构设计思路，实际上，都是来自最常用的数据结构和算法。

课后思考

你知道基础系统、中间件、开源软件等系统中，有哪些用到了索引吗？这些系统的索引是如何实现的呢？

欢迎留言和我分享，也欢迎点击“[请朋友读](#)”，把今天的内容分享给你的好友，和他一起讨论、学习。

数据结构与算法之美

为工程师量身打造的数据结构与算法私教课

王争

前 Google 工程师



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

精选留言



Jerry银银

我对索引的理解

索引真是个好东西。索引的英文名字叫：index，记住这个英文单词，会让我们更容易记忆和联想它到底是什么。在实际的编程中，index这个单词，真是到处可见。例如：数组的下标就是index

如果用一句话描述“索引”的作用，那会是什么？我想是这样：索引是用来辅助查找，用计算机专业术语叫：Addressing(寻址)

现实世界中，我们的查找会存在两种场景：

1. 从局部信息，查询与其相关的整体信息
2. 从整体信息中查询局部信息

怎么理解呢？

搜索引擎需要查询一个网页中是否存在某个关键词以及通过某个关键词查询包含它的所有网页。

索引的应用

正是因为计算机大部分工作都是在Addressing，所以，在计算机中，索引到处存在。小到操作系统虚拟内存到真实内存的映射，就是索引嘛，大到分布式系统、网络，都是这个原理。

以上，我对索引的理解有点“广义”。我觉得数据结构和算法如此重要，它体现计算机精髓的地方方便于此。





es中的单排索引其实用了trie树，对每个需要索引的key维护了一个trie树，用于定位到这个key在文件中的位置，然后直接用有序列表直接去访问对应的documents，区块链拿以太坊来说吧，存储用的leveldb，数据存储用的数据结构是帕特利夏树，是一种高级的trie树，很好的做了数据的压缩，消息中间件像kafka这种，会去做持久化，每个partition都会有很多数据，会有大量数据存储在磁盘中，所以每个partition也会有个索引，方便去做快速访问

2019-01-21 09:40



Jerry银银

今天音频朗读帅哥把MySQL读成了 my s q l，早上起来听音频，萌了\\(//▽//)\\

2019-01-21 12:39

作者回复

官方读法就是 S Q L 哈

2019-01-22 16:35



one

希望老师能讲讲二级索引（从V查K）这块，一直搞不清楚，没有自己写过。还有空间数据结构的range现在也很火，比如uber，滴滴常用的，面试常考。

2019-01-21 09:32



往事随风，顺其自然

可以讲讲es 到排序索引结构原理和数据结构？

2019-01-22 08:18



在路边鼓掌的人

昨天刚学了操作系统的多级页表，应该也是比较经典的索引了

2019-01-22 07:21



纯洁的憎恶

理论联系实际，融会贯通。

2019-01-21 10:49



三木子

everything

2019-01-21 10:13



传说中的成大大

这一节就高深了....

2019-01-21 10:08



『LHCY』

es的倒排索引

2019-01-21 09:08