

AI代码助手限时免费！你的代码手替

文档 建议反馈 控制台

登录

首页 学习 活动 更多

文章/答案/技术大牛

分类

社区首页 > 专栏 > 史上最强算法论战：请不要嘻哈，这是哈希

史上最强算法论战：请不要嘻哈，这是哈希

发布于 2018-05-21 05:01:50 看 1K 回复 0

▲ 举报

文章被收录于专栏： [大数据文摘](#)

大数据文摘

转自 | 知象科技

微信 | 知象科技

论战主角之一龙博：知象科技CEO，公司正在招人，有意应聘者请投简历到 career@briphant.com，欲了解龙博及知象科技，请点击文末“阅读原文”。

这是“美丽互联”微信群里的一次算法论战，感谢书记员硅谷寒（梁寒）精彩的说书般的整理。硅谷寒是“湾区评论(valleytalk)”和36氪的专栏作者。

文章很长，但超级干货，值得收藏！

虽然下面的文字略有嘻哈的感觉，但我还是希望您在阅读之后，能够本着严肃的态度，来审视一番当今天下最有用的 [数据结构](#) ~哈希表 (hash table)。因为，有的地方就有江湖，有数据的地方就有哈希。

2015年，6月7日~6月9日，在史上最强计算机技术微信群里，发生了一场史诗级别的算法论战。

此战，因参与者阵容之强，脑洞之大，足可彪炳微信群聊史册。无论你是技艺精湛的码农，还是拼杀股市的大妈，这次讨论都可令你受益匪浅。好了，在你准备浏览下述老少咸宜男女通杀的内容之前，请先记下这个微信群的大名~“美丽互联”。

每场战争都有一个原因，管它是神仙打架，还是明星撕B。这场算法论战也无例外，起因嘛，很简单很money：上证指数突破5000点大关！

龙博下战书

上海证交所的白老师（白硕）在群聊里稍微夸奖了一下龙博（知象CEO）当年设计的一个 [大数据](#) 查询系统，于是龙博兴奋地变身“龙霸”，在群里点燃了导火索。

龙博：虽然哈希算法是我做的事情里面极小的一部分，但确实是值得骄傲的一个小成就。我拿这个题目考了很多人，谷歌的，微软的，百度的，基本没有及格的，哈哈。曾经有一个谷歌的技术总监要挑战，方案直接被我判了零分。

【书记员（硅谷寒）注：龙博自我吹捧之形，已跃然纸上。要知道，群里谷歌微软的人不要太多，“帮主”本人就是Google出来的，刚好也混了个技术总监的头衔。】

龙博（开始下战书了）：这个哈希算法和数据结构设计也有普适性，交易所内存大部分数据都用这个算法和数据结构进行管理。当上交所的方案是标准答案，看你们的答案与标准答案差异有多大。我就不提示了。有兴趣就挑战，做一个完整的方案出来，给我...

添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动均衡分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使

软件提示：请不要咋反外挂上使月

相关课程

[腾讯云向量数据库-RAG七天](#)

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

持仓类型（一位字符，A~F），
股东代码（10位字符，第1位A~Z，后面9位是数字0~9），
股票代号（short类型）

被定位到的持仓记录是一个64bit的值（value）

帮主带头应战，哎...

帮主：就完美哈希了，剩下就是数组问题了，因为是有限集合和数值是预先知道的。【书记员注：这是本群陈帮主，搜索领域大牛。帮主在第一时间跳出来应战】

龙博：你计算一下你需要数据结构所需的内存资源，再评估一下运算开销，是否能达到性能要求。哈希算法本身没什么大不了，上交所现在用的哈希算法平均查找次数为1.1。关键是数据结构设计。【书记员注：龙博在此处已经提示了设计的关键点，但可惜的是，之后参与论战的人都忽略了。】

帮主：1.（准备阶段）将已知的所有的KEY值传给PHF或MPHF生成算法，生成PHF或MPHF以及相应的数据；
2.（使用阶段）调用已有的PHF或MPHF以及相应的数据快速计算哈希值并进行相应操作。使用MPHF，那么我们可以分配一个Long transactions[几亿]，数组。

剩下的就是数组操作了。删除就是置零，其它就是修改Long的值。完毕。谢谢。

【书记员注：帮主一下子抛出来两个英文缩写PHF、MPHF，着实把龙博吓了一跳，其实这就是“完美 哈希函数”和“最小完美哈希函数”】

龙博：你把所有可能的KEY都放进去了想想你要用多少内存？所以我给你零分。我们总共只有一亿条记录。

帮主：根据前一天的纪录，算一次MPHF，其它新增的放另一个 hashtable。如果说一天都是新增的，那么我们一个小时（x分钟），重算一次MPHF。其实龙博和我也没什么分歧，最后是不是转化为设计一个好的hash function？

【书记员注：不得不说，帮主是天分极强的人，其实，他现在的方案已经很接近最终上交所的方案了，只不过缺少了一点点“画龙点睛”的东西。下面，另一个大牛，独孤虎，要登场了。确切地说，是“大牛团队”，因为独孤虎不是一个人在战斗，他拉上了自己公司里的整个团队，来解答龙博的题目。】

独孤虎炫目登场，杯具了！

独孤虎：我给出的答案是：

1) 采用hash：根据股票代码，将请求hash到特定CPU线程；
2) 采用数组+hash+SkipList+hash：持仓类型数量固定，在每个线程(CPU)采用一个10个元素的数组，每个元素为针对于一个持仓类型的Skip List，Skip List的元素的key(用于排序)为股东代码。除了key之外，跳跃表中的每个元素包括一个hash，用于将股票代码映射到特定信息。

优化措施，根据访问热度为每个股票代码设置一个hot值，用于分散访问，但是会带来比较麻烦的数据同步问题。

算法的时间复杂度为O(1)+log(2*M/n)+O(1)，其中M是特定持仓类型中股东代码的最大数量，n是线程或CPU数量。内存采用hash，文件采用B+树，所以采用hash是ok的，关键是采用何种数据结构降低操作的复杂度。只有完全采用数组，才会考虑稀疏性。这个是hash+数组+跳跃表+hash仅仅是针对于固定数量的持仓类型采用了数组。

需要额外的2*(持仓类型,股东代码,股票代码)数量的指针，每个指针64位计算，16*(持仓类型,股东代码,股票代码)个byte。

时间复杂度：O(1)+log(2*M/n)+O(1)，其中M是特定持仓类型中股东代码的最大数量，n是线程或CPU数量空间复杂度：需要额外的2*(持仓类型,股东代码,股票代码)数量的指针，每个指针按照64位计算，需要16*(持仓类型,股东代码,股票代码)个byte实现hash的方法，我个人感觉这个不是重点，关键是降低复杂度，因为数量太庞大了，即使通过hash，数量依然很多，需要一种数据结构降低操作复杂度。

O(logN)是比较优化的了，能够做的就是通过某种方式降低这个N，比如通过数组或再加hash等。下面是pseudocode longlong M=max((long long)持仓库类型.股东代码.股票代码);

```
1.  
*p=malloc(M*sizeof(U));  
memset(p,0,sizeof(U)*M);  
while(Proc ->lock(p))  
do  
read or write or change p  
done
```

添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动地分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使

软件提示：请不要咋反外挂上使月

相关课程

一站

腾讯云向量数据库-RAG七天...

[向量数据库](#) [腾讯云安灯](#)

Java

[java](#)

领券

数字化IT从业者知识体

[CODING ...](#)

[软件开发](#)

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

```
*p(N1)=
malloc(M/N*sizeof(U));
memset(p1,0,sizeof(U)*M/N);
memset(p2,0,sizeof(U)*M/N);
...
memset(p(n1),
0,sizeof(U)*M/N);
CPUi=>read or write or change p(((longlong)持仓库类型.股东代码.股票代码) mod M)
```

大型实时算法，就需要简单我估计他们的算法应该是hash+跳跃表，不同支出就是如何处理(持仓类型,股东代码,股票代码)这三个值作为hash和跳跃表的key

【书记员注：独孤虎用一篇超长微信，震撼登场，但杯具了...】

龙博（轻扫一眼）：@独孤虎，你要回忆一下白老师讲过的，hash表每天只初始化一次。你用的数据结构太复杂了，而且锁的力度太大。你这种设计，如何做到每个进程（线程）与其他几十个线程同步地对这一亿条记录进行操作？没可能的。所以，我给你零分！

帮主（插话）：用sharding保证无锁？

龙博（立刻否定）：不用任何sharding，无锁的核心在数据结构设计。你可以想象一下，两个线程如何同步地对一个10个元素大小的数组进行无锁操作（增加、删除、修改、查询）。想通这个问题，你就知道我怎么做这个无锁设计的了。

帮主：一个是任务queue，然后thread pool。【书记员注：其实帮主偷偷地上微博，向广大网友征集答案，如何设计无锁结构，用两个线程操作10个元素的数组？】

龙博：我再给你们说一遍，数据是所有进程完全共享的，不做划分。任何一条记录在任何时候都可能被该物理机上的一个进程访问（包括修改、删除、新增）

擅长高性能多处理器系统设计的唐博登场

【书记员注：下面，另一大牛，唐博，要登场了。唐博在高性能多处理器系统的设计上，独树一帜。】

唐博：我们DDoS的处理也有类似的问题，早就有答案了。与之不同的是，hash key 是可变长的URL。可以理解为16150长度的字符串。大家复习一下

1. atomic instruction

2. CAS primitive

然后再来讨论比较靠谱。

【书记员注：唐博一上来，就叫大家去复习功课。计算机系的小伙伴们，你们还记得原子指令和CAS原语吗？】

龙博：类似的内存 [数据管理](#) 技术，用到了交易系统的几乎所有的内存数据结构中。因此可以最大程度发挥CPU的运算能力，减小同步、进程切换的开销。全异步、无锁、用户态和核心态数据共享等技术。为了提高运算效率，我甚至连乘法都是自己实现。。。

书记员注：我硅谷寒搞了N年的芯片设计，觉得龙博这句“连乘法都是自己实现的”，有点儿吹牛了。他应该不知道怎么设计高速乘法器。估计他都不知道啥是booth multiplier？如何设计硬件pipeline？】

龙博：交易所交易系统，就是屠龙之技啊。。。呵呵。例如数组的查询，数据有key + payload。如果你把所有key放到一起，那么整个数组的key可能被装进cache中，这样查询效率就很高了。。。这个也经常要用

独孤虎回来了

独孤虎（憋了一下午，又和自己的团队有了新方案，再次发布超长微信）：

整个问题划分为两个部分1）、2）和三个阶段a）、b）、c）：

1) 网络操作部分，负责接收请求和返回应答；

2) 内存操作部分，负责内存持仓记录的操作；

a) 接收服务请求>

b) 操作持仓记录>

c) 返回服务应答

[添加站长](#) [进交流群](#)

领取 [10元无门槛券](#)，专享 [最新干货技术](#)

精选特惠 用云无忧

2核2G3M云服务器 [82元/年](#)

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动均衡分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

[精选特惠 用云无忧](#)

[加入讨论](#)

[金牌服务刘](#)

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使!

软件提示：请不要咋反外挂上使月

相关课程

一站

[腾讯云向量数据库-RAG七天...](#)

[向量数据库](#)

[腾讯云安灯](#)

[Java](#)

[java](#)

领券

[数字化IT从业者知识体](#)

[CODING ...](#)

[软件开发](#)

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

MPI_Irecv测试是台返操作应答，如果返回，则驱动执行SOCKET与入操作。

对于内存操作部分，前提条件为NUMA架构。【书记员注：请大家复习NUMA】

a) 每个Node为一个进程，每个进程中多个线程，线程数量与每个node内的CPU

数量*每颗CPU内的核数相关；

b) 线程按照线程池组织，采用一个无锁的队列实现生产者和消费者模式，即一个线程负责通过非阻塞的MPI将请求放入队列中，而其他线程则分别读取，并执行具体的持仓记录操作；

c) NUMA架构的数据存放策略采用firsttouch，即通过hash将持仓记录划分到每个节点，确保每个节点仅仅操作本地内存；

由于分配到每个节点的记录数量还是非常庞大，需要有效的数据结构组织这些数据，可选方案有三种：

a) hash

b) balanced trees

c) Skip List

上述三种方案都有很多种不同lockfree的实现方式，只要拿来用就ok了，但是balanced trees的操作比较复杂一点，首先排除。只剩下hash和Skip List：a)hash的问题需要每个节点分配一个非常非常大的数组，并且保证hash表的负载比较合理，可以采用将持仓类型+股东代码+股票代码形成一个字符串，在采用Murmurhash3这类算法进行hash；b)采用Hash+Skip List，当难以保证hash表的负载比较平衡的时候，可以采用小一点的hash表，但是每个表的元素为一个Skip List。

两种方式，关键要考虑持仓记录的变动情况，如果变动情况，不影响hash的负载均衡，采用方案a)。否则，采用方案b)

无锁操作，采用CAS方式实现hash和skip list的方式很多，拿过来用就可以。无锁方案很多，如果采用java实现，比如生产和消费者模式，就可以采用disruptor。

龙博（估计没看完独孤虎的长文，一直攻击其数据结构设计）：你说cas指令随便用，但前提是cas或者其他无锁的数据结构需要满足某些条件，你的数据结构能满足这些条件么？这些都是很重要的工程问题。不是说，有cas，有很多无锁设计和实现，直接拿来用就行。。。

独孤虎（赞许道）：龙博，你一招鲜，吃遍天。二十年后还靠无锁设计、CAS吃饭。

龙博：我现在已不靠这个吃饭，靠这个吹牛。。。

独孤虎（接着给方案）：1) 所有节点的操作的内存是本地的，无需操作其他节点内容；2) hash表是一个数组，数组中的不同元素可以并行操作；3) 数组内的每一个元素是一个链表，仅仅在头部插入和内部删除，有多种无锁操作方案。

龙博（还是说独孤虎的数据结构不对）：你想象一下，所有这些数据在共享内存里面。。。如果是链表，很难实现完全的无锁设计。~零分

独孤虎：一下午，还是零分，我哭！

独孤虎：如果不用链表：第一种方式是copy on write，hash表中的每个元素为指向一个数组的指针，当指向插入和删除操作时，采用内存copy这个数组操作，然后在这个副本上进行插入和删除操作，此种方式不是最优，但是最容易想到的。

第二种方式是采用无链表的hash，有两个数组，第一个数组H是一个Hash表，存放一个无符号整数，指向第二个数组的一个具体位置，第二个数组D是一个非常大的数组，数组中的每个元素包括(持仓类型,股东代码,股票代码,记录数据, next)，其中next指向D中的另一个位置，从而模拟一个链表：

a)添加操作，一旦通过(持仓类型,股东代码,股票代码)计算出在数组A中key所指D的位置已经存在值，则在数组D中查到next为0的元素tail，并变化一下(持仓类型,股东代码,股票代码)重新计算自己的hash，并插入到D中，然后采用CAS方式将其位置添加到tail的next；

b)删除操作：删除一个位置d，则采用CAS方式将父亲p中的next指向自己的next即可。

Penny：独孤虎的角度基本是做一个erp系统的sense，不是做高性能系统的sense啊，链表内存分布都是飘的，咋缓存感知啊！

【书记员注：Penny是新鲜出炉的清华计算机博士，其创办的公司，拥有世界上最多的人均IP数量。】

独孤虎：工作太努力容易精神分裂。你们知道吗？Jeff Hammerbacher在哈佛读书时就精神分裂，返校毕业后先是加入华尔街做量子码工，之后是Facebook的早期数据科学家，后来是Cloudera的创始人，因为工作太疯狂，精神分裂了，现在投身于基于数据科学的疾病治疗，主要是免疫和药物及基因的关系，为病人做个性化治疗。

一夜过去了...独孤虎精神抖擞的回来了！

独孤虎：昨天给出的第三个方案存在如下两个问题：

1) 没有考虑结构体的填充问题，struct结构内的数据要对齐，从而导致其数据占有更大的内存；

2) CAS的操作受限，目前gcc仅仅针对1,2,4或8字节长度的int类型提供了CAS以及原子性

的加减逻辑运算；

添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动地分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使!

软件提示：请不要咋反外挂上使月

相关课程

一站

腾讯云向量数据库-RAG七天...

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

b) 线程按照线程池组织，采用K个无锁的队列实现生产者和消费者模式，即一个线程负责通过非阻塞的MPI接收请求，并将请求放入特定的队列中，而其他每一个线程读取特定队列中的请求，并执行具体的持仓记录操作；

c) NUMA架构的数据存放策略采用firsttouch，即通过hash将持仓记录划分到每个节点，确保每个节点仅仅操作本地内存；

采用一个非常大的数组D作为Hash表，表中的每个元素值包括(持仓类型,股东代码,股票代码,记录数据, count),如果存在key值冲突，则相同key的值连续放置，其中count表示后继节点有多少与之具有相同key的元素。其中股东代码可以表示为一个字节字符串和四个字节无符号整数，共5个字节，股票代码表示为2个字节的无符号整数，股票持仓为8个字节整数，持仓类型和count公用一个字节（分别占用四位），因此采用如下的结构保存

```
struct record {
    unsigned long compoundKey;
    unsigned long data;
}
```

其中data表示股票持仓数据，而compoundKey为(持仓类型,股东代码,股票代码, count) 的复用，其中最低四位表示count。将数组D划分为K段，每段由一个线程负责操作，即根据hash值确定所在范围，然后放入对应的队列中由对应的线程处理，由于确保了每个元素仅有一个线程操作，从而整个操作过程无需CAS或者枷锁。

1) 插入操作，根据(持仓类型,股东代码,股票代码) 计算key：如果D[key]所在位置的compoundKey非零，则冲突，查看下一个位置，即key+1位置，直到不冲突为止，即compoundKey为0时，对所在位置的元素赋值。

2) 删除操作，如果要删除一个hash值为key的元素，则首先找到该元素所在位置key+m，m大于等于0。a) 如果key+m所在位置的count为0，则直接赋值为0，并跳转到步骤c); b)如果key+m所在位置的count大于0，则逐次向后操作，将此位置设置为后继的、并且hash值为key的元素，直到hash值为key并且count值为0时结束，并跳转到步骤c);c)从此位置向前一直到key，将所有具有相同key所在位置的count减1。

3) 更改操作/读取操作：比较简单。

这个方法，仅仅需要多个无锁队列，其他的操作即无需锁，也无需CAS。对数据和请求进行了两次划分，第一个是根据hash将数据和请求划分到不同的节点（每个节点具有多个CPU，共享本地内存），第二次是根据hash将数据和请求划分到数组的不同位置段，并由不同线程负责操作

龙博：@独孤虎在接近正确答案。“部分”思路正确了。不过你不用考虑生产者消费者这种复杂的模型。要考虑如果让多个进程无锁同步地访问同一个大数组（哈希表），不要分区。

独孤虎（相当开心）：我跟团队又讨论了一个下午。

龙博解密

龙博：如果你能设计出完全的无锁结构，就没必要做这个划分了。所以能够设计出完全的无锁数据结构，是关键。作为compound key的三部分，我已经告诉你每个部分的特征了，你试试看。

龙博：其实这个题目在白老师的发言里面已经给了一个很大的提示，哈希表。别看这个提示很小，绝大部分人连这一步都跨不过去。无锁数据结构的关键是什么？就是key不要超出字长...在我们现在的机器上，字长就是8字节，64bit...提示到这里，该做出来了吧...

龙博：因为只要你的key不出超字长，哈希表的新增、删除、修改操作都可以是原子操作。就是机器指令直接支持的原子操作，也就是“无锁”操作。哈希表的新增，删除和修改操作就是对这个8字节进行赋值而已。。。明白了么？

【书记员注：我其实蛮失望的，原来最关键的地方就是拼一个64bit以内的Key，感觉像是郭靖的亢龙有悔，最厉害的一招就是最平淡无奇的一招！】

龙博：这是最核心的，说出来其实也很简单。我在总结一下，最关键的几部分：

1. 开放地址探测的哈希（非链式哈希！）
2. 哈希表的key值压缩在一个8字以内
3. 稍微特别考虑一下哈希表的删除（你得维护key冲突的时候的链条）

龙博：实际测试结果，一亿条记录，在哈希表的装载率（load factor）为70%的情况下，平均查询次数为1.1次，最大查询次数不超过4次。。这应该是最好的结果了。key+value共16字节。这应该是最高效的存储结构了。

帮主：这么来说，这题的难点在哪？地址探测，最坏情况可能是O(n)。所以关键是将key编码成64bits？根据那三种信息，不是显而易见的吗？我倒觉得，线性探测是核心，但那玩意有worstcase。独孤虎居然花了那么多时间，那么严肃的研究，感觉龙博在逗我们玩。

【书记员注：呵呵，帮主跟我一样，心有不甘，觉得自己败在了最平凡的一招上。】

添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动地分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使

软件提示：请不要咋反外挂上使月

相关课程

一站

腾讯云向量数据库-RAG七天...

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

执行，但交易系统一切正常。包括内存被分配光也是一样，你要保证在极端情况下系统可以正常反应。

帮主：你说的一天可能上亿，整个都可能是新增的，对吗？

龙博：可以，因为这个哈希表增加一个元素的操作开销跟查询没什么分别。

【独孤虎团队，经过一番修整，终于给出了最终的第五个完美方案】

独孤虎：现设计数据结构如下：

```
struct record {
    unsigned long compoundKey;
    unsigned long data;
}
```

其中data表示股票持仓数据，而compoundKey为(股东代码,股票代码,持仓类型,flag)的复合体，股东代码占用最63~27位（5位字符和四个字节无符号整数），股票代码占26~11位（两个字节），持仓类型占用10~6位（四个字节），isTail占用第5位，表示是否为尾部，即后继元素中没有相同hash值的元素。doWrite占用第4位，表示该key正在更改/写入，readerCount占用3~0为表示该key正在读的线程数量。

采用非常大的数组，数组中的每个元素为struct record

具体操作如下：

1) 插入操作，根据(持仓类型,股东代码,股票代码)计算hashkey：

a) 如果D[key]所在位置doWrite为1，则循环判断，直到其值为0为止（在循环中采用cache操作，直接从内存读取doWrite），继续执行；

b) 采用CAS操作，将doWrite设置为1，如果操作失败，则跳转到步骤a)，如果成功，则继续执行；

c) 如果readerCount≠0，则循环等待，直到readerCount为0（在循环中采用cache操作，直接从内存读取）；

d) 如果readerCount为0，则从key开始连续查找到key相同并且isTail为1的元素，然后将其设置为0，并从后继元素找到第一个compoundKey为0的位置，并写入；

e) 将doWrite为0，并采用cache操作，将其写回内存。

2) 删除操作：如果要删除一个hash值为key的元素，其操作类似与插入操作，不同的地方是在获取doWrite和readerCount之后：

a) 如果被删除元素的isTail为1，则从该位置开始到key进行查找，找到第一个

hash值为key的元素，将其的isTail设置为1；

b) 如果被删除的元素的isTail为0，则需要继续查找，将其isTail为1并且hash值为

key的元素拷贝到这里，然后将原来尾部元素的上一个元素的isTail设置为1；

独孤虎收获满满，一掷千金

独孤虎（兴奋道）：我们团队上可证NP，下可写哈希。

独孤虎：感谢龙博慷慨地贡献出一个如此优秀系统的核心算法！比如做大规模流量交易平台，算法相当重要。如果一天要处理百亿千亿次交易，就需要考虑系统核心算法的性能，而不仅仅是拼机器的数量和硬件性能。

独孤虎：我发个红包给群里所有人！！

【书记员注：独孤虎乃真土豪也，群里平均每个人收到了¥50...独孤虎在此次论战中表现出了超级的持久作战力，强大的团队协作精神，堪称“中国互联网之铁血军魂”！】

【书记员注：最后，本群创始人，院长，出来总结陈词，论功行赏。院长是网络系统安全界的超级高手，尤其精通PowerPC、MIPS、ARM这些RISC计算机系统！】

院长总结陈词

院长：

1. 龙博在巨大项目压力下，靠straightsmart能想出算法，并在白老师的鼓励和指导下，实践完成，为8年后的股波波的到来立下了卓越贡献。建议中央军委给予龙博记一等功，花翅一枚。

2. 帮主基础扎实，算法雄厚。由于长期在search领域，对hash算法，retrivial系统了若指掌。非常的脚工赞。但由于他打酱油太多。窃以为龙博略微牛一些。

3. 独孤虎，从理论界，再次横空跨界。身体力行，团队作战。令人泣血。方案里有链表，有cas，还考虑了cache。深感算法设计需要重新设计。手工赞！

添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动地分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使

软件提示：请不要咋反外挂上使月

相关课程

一站

腾讯云向量数据库-RAG七天...

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

一些特殊的指令可以监视其他逻辑对一个memoryspace是否有touch，通过监控bus上的transaction。

简单说来，就是，如果我想own，我reserve这个mem。一旦reserve失败或者被abort，来回try。

史上最强之技术聊天，持续了两天半，以独孤虎团队的胜利告终。台上选手高声论战，台下群友默默潜水，虽然台下也有许多“高手”，但都怕一出声，不着调，毁了自己半生清誉。这也算是一道有趣的风景吧。

本次讨论，难得众高手相聚于“美丽互联”，正如古人所云：

“今番良晤，豪兴不浅，若得山水重逢，再当把酒言欢。”

是夜，暖风轻，圆月明，朋友聚还散，路人停复行。

本文参与 [腾讯云自媒体同步曝光计划](#)，分享自微信公众号。

原始发表：2015-06-28，如有侵权请联系 cloudcommunity@tencent.com 删除

编程算法

评论

[登录](#) 后参与评论

推荐阅读

编辑精选文章

[换一批](#)

眼看他搭中台，眼看他又拆了	19864	QQ 25年技术巡礼 技术探索下的清...	2482
【万字长文】论如何构建一个资金账...	5511	因为一部遮天，我用三种语言实现了...	5383
得物 Redis 设计与实践	4519	【万字长文】K8s部署前后端分离we...	6619

《史上最强算法论战》是怎样炼成的？

编程算法 大数据

转自 | 知象科技 作者 | Emma、喵小裴 导读：上周大数据文摘发了一篇火爆的干货文章：《史上最强算法论战：请不要嘻哈，这是哈希》（可点击查看），今天，我们在[知象科技]的官方微信发现了这篇介绍龙白滔博士

大数据文摘 2018-05-21

672

0

个人对哈希数据结构学习总结 -- 实践篇 -- 上

数据结构 迁移 链表 实践 数组

哈希表这个数据结构相信各位都不陌生，无论是高级语言，还是各大数据库底层实现都不离开它，所以本文我想来聊聊我个人对哈希表的一些看法，同时也是对哈希表这个知识点做一次系统性的梳理...

大忽悠爱学习 2023-10-11

230

0

高并发编程系列：ConcurrentHashMap的实现原理(JDK1.7和JDK1.8)

jdk 编程算法 hashmap 数据结构

添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动地分发到后端多台云服务器上。

[产品介绍](#) [产品文档](#)

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使

软件提示：请不要咋反外挂上使

相关课程

一站

腾讯云向量数据库-RAG七天...

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发



大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

Java集合篇：HashMap 与 ConcurrentHashMap 原理总结

jdk hashmap 编程算法 数据结构

(1) HashMap 是基于 Map 接口的非同步实现，线程不安全，是为了快速存取而设计；它采用 key-value 键值对的形式存放元素（并封装成 Node 对象），允许使用 null 键和 null 值，但只允...
...
...
...

全栈程序员站长 2022-09-12

4K

0



添加站长 进交流群

领取 10 元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

HashMap常见面试题_java面试题大汇总

hashmap jdk 编程算法

大家好，又见面了，我是你们的朋友全栈君。 目录 1.HashMap的数据结构? 2.HashMap的工作原理?
3.当两个对象的hashCode相同会发生什么? 4.你知道hash的实现吗?为什么要这

全栈程序员站长 2022-09-22

358

0



HashMap和Hashtable的联系与区别

hashmap java 编程算法

HashMap继承自AbstractMap类，而HashTable继承自Dictionary类。它们都同时实现了Map（图）、Cloneable（可克隆）、Serializable（可序列化）这三个接口。Dictionary类现已被弃用，父类已被弃用，自然没有人使用它的子类Hashtable。

VIBE 2022-12-02

674

0

面试系列之-ConcurrentHashMap实现原理（JAVA基础）

java 基础 面试 线程 原理

concurrentHashMap用 transient volatile Node<K,V>[] table修饰，使用volatile来保证某个变量内存的改变对其他线程即时可见，在配合CAS可以实现不加锁对并发操作的支持。get操作可以无锁是由于...

用户4283147 2023-08-21

523

0



请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使...

软件提示：请不要咋反外挂上使...

导师计划--数据结构和算法系列（上）

数据结构 集合 数据 数组 算法

导师计划已经开始一个月了，自己的讲解的课程选择了数据结构和算法。这个系列的讲解分为上下两章，javascript语言辅助。本篇文章为上章，涉及的内容是基本的数据结构。在日本，晚上没事安排...

Jimmy_is_jimmy 2023-09-01

138

0



相关课程

一站

腾讯云向量数据库-RAG七天...

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

【HashMap我可以讲半小时】

hashmap 数据结构 容器 编程算法

底层工作原理及数据结构 工作中用到最多的是hashmap，它支持key-value这种键值对存储。当往hashmap中添加一个键值对时，会将key-value的对应关系封装成一个Entry，就是键值对对象，它会拿着key做hash算法，把hash的值映射到内存地址，找到内...

Java廖志伟 2022-03-07

478

0

Java 集合（List、Set、Map 等）相关问答归纳再整理

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

数据结构基础温故-6.查找（下）：哈希表

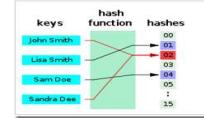
其他

哈希（散列）技术既是一种存储方法，也是一种查找方法。然而它与线性表、树、图等结构不同的是，前面几种结构，数据元素之间都存在某种逻辑关系，可以用连线图示表示出来，而哈希技术的...

Edison Zhou 2018-08-20

601

0



添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

精选特惠 用云无忧

2核2G3M云服务器 82元/年

一文读懂JDK7,8,JD9的hashmap, hashtable, concurrenthashmap及他们的区别

hashmap

图中，紫色部分即代表哈希表，也称为哈希数组（默认数组大小是16，每对key-value键值对其实是存在map的内部类entry里的），数组的每个元素都是一个单链表的头节点，跟着的绿色链表是用来解...

java进阶架构师 2018-12-05

847

0



PHP的哈希实现

php 编程算法 数据结构

PHP内核中的哈希表是十分重要的数据结构，PHP的大部分语言特性都是基于哈希表实现的，例如：变量的作用域，寒暑表，类的属性，方法等，zend引擎内部的很多数据都是保存在哈希表中的。

仇诺伊 2020-04-24

1.1K

0



Python 算法基础篇：哈希表与散列函数

python 存储 函数 基础 算法

哈希表是一种高效的数据结构，常用于存储键值对并支持快速的插入、查找和删除操作。散列函数是哈希表的关键组成部分，用于将键映射到哈希表的索引位置。本篇博客将介绍哈希表和散列函数的基本概念，并通过实例代码演示它们的应用。

小蓝枣 2023-07-24

327

0

Java进阶（六）从ConcurrentHashMap的演进看Java多线程核心技术

java hashmap 安全 人工智能

原创文章，转载请务必下面这段话置于文章开头处（保留超链接）。本文转发自技术世界，原文链接

<http://www.jasongj.com/java/concurrenthashmap/> 线程不安全的HashMap 众所周知，HashMap是非线程安全的。而HashMap...

Jason Guo 2018-06-14

704

0

详解ConCurrentHashMap源码（jdk1.8）

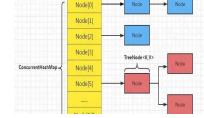
缓存 hashmap 数据结构 编程算法 node.js

ConCurrentHashMap是一个支持高并发集合，常用的集合之一，在jdk1.8中ConCurrentHashMap的结构和操作和HashMap都很类似：

用户10384376 2023-02-26

268

0



构建高性能队列，你不得不知道的底层知识！

缓存 数据分析 java 编程算法 数据结构

栈和队列，可以说是除了数组和链表之外最基础的数据结构了，在很多场景中都有用到，后面我们也会陆陆续续的看到。

相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动...

产品介绍 产品文档

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使...

软件提示：请不要咋反外挂上使...

相关课程

腾讯云向量数据库-RAG七天...

向量数据库

腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

大数据文摘

作者相关精选

史上最强算法论战：请不要嘻哈，这是哈希

面试Java基础问题汇总 part1

hashmap jdk java 自动驾驶 打包

c++要更复杂，Java相对而言更容易回答。多态按执行过程分为两种情况，编译时多态和运行时多态。

Steve Wang 2022-05-10

294

0

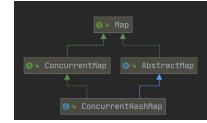
添加站长 进交流群

领取 10元无门槛券，专享 最新干货技术

3秒搞定ConcurrentHashMap

java

1、 ConcurrentHashMap 是 Java 并发包中自 JDK1.5 后提供的一个线程安全且高效的 HashMap 实现，可以用来替代 HashTable。直接实现了 ConcurrentHashMap 接口，同时继承了 AbstractMap 抽象类。



老兵程序员 2021-07-01

575

0

ConcurrentHashMap 锁机制进化的考量

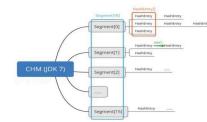
jdk hashmap 数据分析 编程算法

我们知道，HashMap 是无法保证线程安全性的，如果在并发环境下插入一个HashMap，哈希桶数组扩容时，有可能会造成链表出现环（美团技术的文章有详解）。若要保证线程安全性，就得使用...

王知无不import_bigdata 2020-05-09

1.6K

0



相关产品与服务

负载均衡

负载均衡（Cloud Load Balancer，CLB）提供流量分发服务，访问流量经由 CLB 可以自动地分发到后端的云服务器上。

产品介绍 产品文档

精选特惠 用云无忧

加入讨论

金牌服务刘

请不要在虚拟机中运行,怎么解决?

软件提示：“请不要在反外挂上使...

软件提示：请不要咋反外挂上使...

社区

活动

资源

关于

腾讯云

技术文章

自媒体同步曝光计划

技术周刊

社区规范

一站

技术问答

邀请作者入驻

社区标签

免责声明

腾讯云向量数据库-RAG七天...

技术沙龙

自荐上首页

开发者手册

联系我们

向量数据库 腾讯云安灯

技术视频

技术竞赛

开发者实验室

友情链接

学习中心

技术百科

技术专区

相关课程

一站

腾讯云向量数据库-RAG七天...

向量数据库 腾讯云安灯

Java

java

领券

数字化IT从业者知识体

CODING ...

软件开发

热门产品

域名注册
云存储云服务器
视频直播

区块链服务

消息队列

网络加速

热门推荐

人脸识别
SSL 证书腾讯会议
语音识别

企业云

CDN 加速

视频通话

图像分析

MySQL 数据库

更多推荐

数据安全
网站监控负载均衡
数据迁移

短信

文字识别

云点播

商标注册

小程序开发