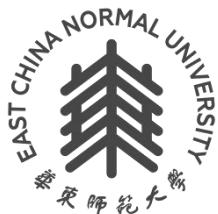


2022 届硕士专业学位研究生学位论文

分类号: _____ 学校代码: _____ 10269

密 级: _____ 学 号: _____



華東師範大學

硕士专业学位论文

Master's Degree Thesis (Professional)

论文题目: _____

院 系 名 称: _____ 计算机科学与技术学院

专业学位类别: _____ 工程硕士

专业学位领域: _____ 计算机技术

研 究 方 向: _____ 数据库

指 导 教 师: _____ Xxx 教授

学 位 申 请 人: _____ xxx

Thesis (Professional) for Master's Degree in 2022

University Code: 10269

Student ID: _____

East China Normal University

Title:Design xxx Load

Department: School of Computer Science and Technology

Category: Master of Engineering

Domain: Computer Technology

Research Direction: Database storage

Supervisor: Prof. X xx

Candidate: xx X

May, 2022

华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《_____》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的科研成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：

日期： 年 月 日

华东师范大学学位论文著作权使用声明

《_____》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的著作权归本人所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和学校指定的相关机构送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

- ☐ 1. 经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文*，于 年 月 日解密，解密后适用上述授权。
- ☐ 2. 不保密，适用上述授权。

导师签名：_____

本人签名：_____

年 月 日

* “涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

硕士学位论文答辩委员会成员名单

姓名	职称	单位	备注
			主席

摘要

[illegible][illegible]

关键词：xxx架构，持xxx存，高性能缓存

ABSTRACT

Content Content Content Content Content Content Content Content Content
Content Content Content Content Content Content Content Content Content Content
Content Content Content Content Content Content Content Content Content Content
Content Content Content Content Content Content Content Content

Content Content Content Content Content Content Content Content Content
Content Content Content Content Content Content Content Content Content Content
Content Content Content Content Content Content Content Content Content Content
Content Content Content Content Content Content Content Content

Keywords: *xxxre, nonxxxory, cache*

目录

第一章 绪论.....	1
1.1 研究背景.....	1
1.2 研究内容与挑战.....	1
1.3 本文工作.....	1
1.4 本文结构.....	1
第二章 相关工作.....	1
2.1 架构图.....	1
2.2 持架构中的算法.....	2
2.3 缓存性能.....	3
2.3.1 命中率对比.....	3
第七章 总结与展望.....	4
7.1 本文总结.....	4
7.2 未来展望.....	4
参考文献.....	5
致谢.....	6
攻读硕士学位期间科研情况.....	7

插图

2.1	Lxx构图	6
2.2	d	

表格

3.1	不同比例下索引查询的命中率	17
3.2	三种负载下主键查询与索引查询的命中率	19
5.1	精确淘汰和近似淘汰的命中率	46
6.1	服务器配置	51

第一章 绪论

1.1 研究背景

随着互xxx的思路。

1.2 研究内容与挑战

本文主要的研究内容有两个，一是xxx的挑战之一。

1.3 本文工作

本文通过大量的实验，分析了xxx。本文贡献如下：

1.4 本文结构

下面介绍了后续章节的安排：

第二章介绍了xxx研究成果。

第三章分析了xxx存策略提供思路。

第四章介绍了xxx池。

第五章阐述了xx现。

第六章对本文xx进行了实验对比。

第七章总结全文以及对未来工作的展望。

第二章 相关工作

2.1 架构图

从读写流程可以看出，架构拥有良好的写性能，但读性能随着文件数和磁盘数据层的增多逐渐变差。出了几种提高读性能的方法内部有序，不同固态表之间的

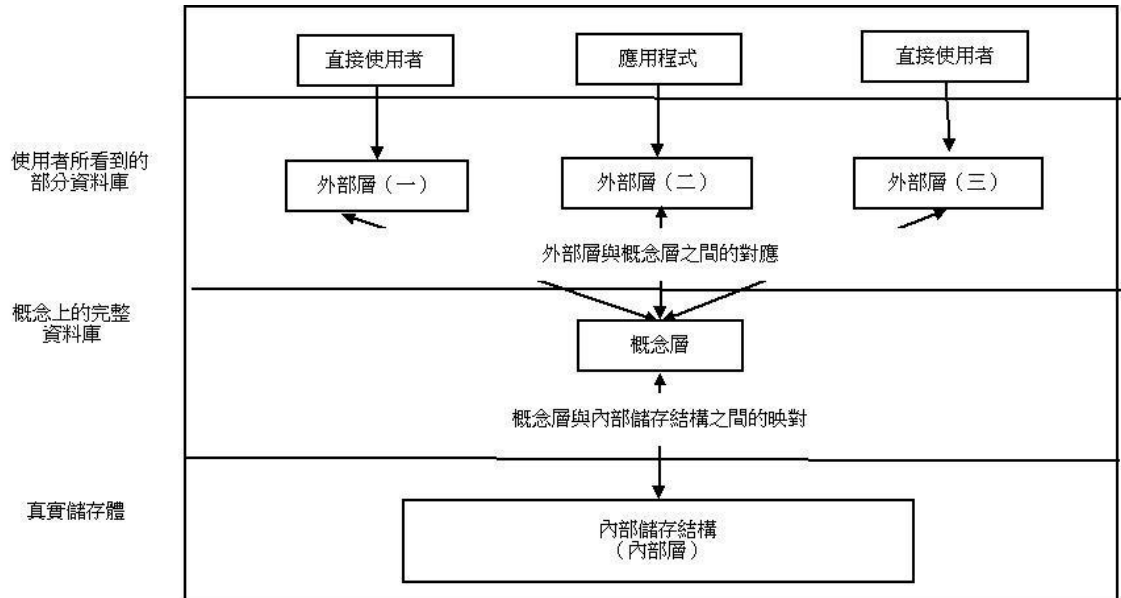


图 2.1: xxx流程及架构图

内部有序，不同固态表之间的键值存在交集；另一种是大重新生成多个固态表，写入较深的一层。每层的数据量按照固定倍数增长，大将数据直接刷到磁盘，可以快速腾出内存空间，以便后续数据的写入，大合并将不同层的数据进行合并，可以剔除已删除数据、清除旧数据，减少存储冗余，释放磁盘空间，也可以提高查询效率。

读取数据时，首先找到一个包含键值范围的 Table，就不需要在本层继续查找了。如果该固态表中找到了查询的数据，则返回查询结果；如果没有，则继续往更深层查找。

$$\text{Priority} = \frac{\text{Frequency} * \text{DataDensity}}{\text{LifeTime}} \quad (5.1)$$

从读写流程可以看出，架构拥有良好的写性能，但读性能随着数据层的增多逐渐变差。Table 中提出了几种提高读性能的方法：

1. **缓存。**由于写入磁盘的xxx可保证数据的正确性。
2. **布隆过滤器**^{[14][15]}。按照上诉查询流程，用布隆过滤器判断数据是否存在，就能在数据不存在的情况下节省许多不必要的磁盘 I/O。

3. **合并**。在第同一个数据可能存在多个版本分布在不同层的固态 减少查询次数。但是合并操作会占用大量的 CPU 资源和磁盘 I/O，在业务高峰期，可能导致写停顿，严重影响数据库的写性能。许多工业界的数据库都会选择白天用大内存支持一天的数据写入，晚上在业务低谷时进行合并，例如 OceanBase，不过 2020 年 ob 提出可以将合并操作转移到 FPGA^[16] 上，实验评估可以使 CPU 性能提升 25% 左右。

综上所述，通过合并操作能清除失效数据，但也占用了大量 CPU 资源，造成了严重的写放大。

2.2 持架构中的算法

算法3 无锁空闲队列数据出队算法

输出：空闲页指针

```

1: oldv  $\leftarrow$  consumer_;
2: cmpv  $\leftarrow$  oldv;
3: bret  $\leftarrow$  False;
4: while oldv < producer_ do
5:     index  $\leftarrow$  (oldv mod size_)
6:     if NULL == (oldv  $\leftarrow$  ATOMIC_LOAD(&data_[index]))
then 7:         PAUSE();
8:         oldv  $\leftarrow$  ATOMIC_LOAD(consumer_);
9:         cmpv  $\leftarrow$  oldv;
10:    else if cmpv  $\neq$  (oldv  $\leftarrow$  VCAS(&consumer_, cmpv, oldv + 1)) then
11:        cmpv  $\leftarrow$  oldv
12:    else
13:        bret  $\leftarrow$  True;
14:        break;
15:    end if
16: end while
17: if bret is True then

```

```
18:   while NULL== (ret ← ATOMIC_SET(&data_[index], NULL) ) do
19:       PAUSE( );
20:   end while
21: end if
22: return ret;
```

算法外针对 PM 不同于 DRAM 和 SSD 的特性，减少内存表刷新到磁盘的次数，而且大容量的 PM 内存表有助于缓解小合并带来的写停顿。

2.3 缓存性能

缓存是用于提高访问速度的主要手段之一，在计算机系统中很多地方都会用到缓存，只是不同的位置有着不同目的的缓存设计。例如 CPU 中的高速缓存，是用来缓解 CPU 处理速度和内存访问速度不匹配的问题；网络中的缓存，是用来解决网络拥堵、服务器超载等问题。

2.3.1 命中率对比

表 3.1: 不同比例下索引查询的命中率

比例	总命中率	索引块命中率	记录块命中率
1:1	62.9%	85.9%	40.5%
1:5	55.4%	66.7%	48.1%
1:10	51.0%	51.1%	49.6%
1:50	49.0%	17.3%	49.9%

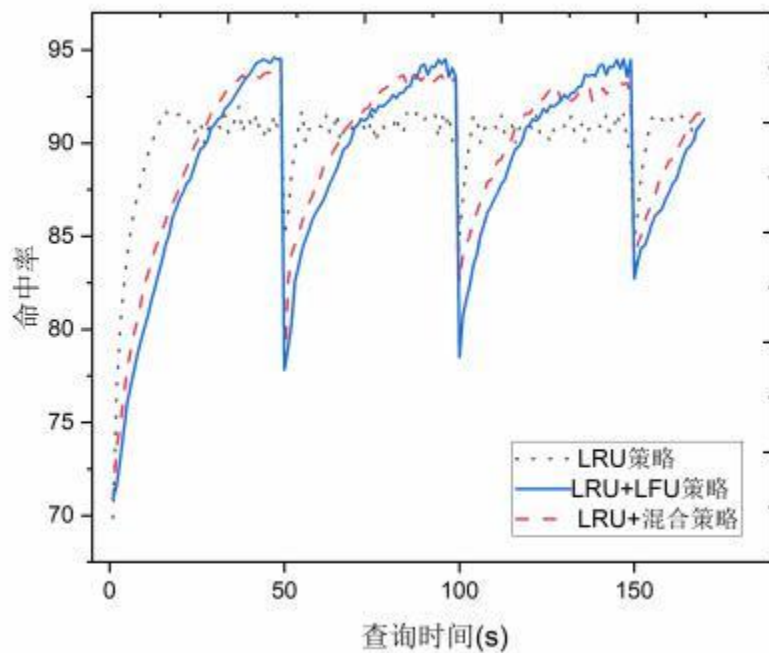


图 6.9: 热点变化的负载下对进行主键查询的命中率对比图

第七章 总结与展望

随着数据量的爆发式增长，基于为了支持一天的业务，工业级数据库中的内存表往往设置得很大，缓存急需寻找另外的存储空间。而近年来出现的持久化内存填补了内存和 SSD 之间的性能空缺，为缓存设计提供了新的思路。

7.1 本文总结

本文通过分析数据库中查询xxx特征，深入研究了xxx的关系，并提出了一种针对xxx算法。此外为了xxx，本文还提出了一种xxx结构。本文工作可总结为以下三点：

1. **对数据库xxx进行了实验分析。** 本文通过对数据库中xxx率越高等。
2. **设计了一种xxx结构。** 针对贡献一中的xxx二级缓存, 最终提出了一种面向xxx缓存结构。
3. **设计了一种xxx算法。** 针对贡献一中xxx载更加友好。

7.2 未来展望

本文针对数据库xxx的性能:

1. **自xxx大小。** 本文中实xxx的调整。
2. **过滤xxx数据。** 在本文的xxx访问频率。
3. **xxx淘汰。** 本文xxx算法。

参考文献

- [1] HUANG D, LIU Q, CUI Q, et al. Tidb: A raft-based HTAP database[J/OL]. Proc. VLDB Endow., 2020, 13(12):3072-3084.
<http://www.vldb.org/pvldb/vol13/p3072-huang.pdf>. DOI:
10.14778/3415478.3415535.
- [2] ALI W, SHAMSUDDIN S M. Intelligent dynamic aging approaches in web proxy cache replacement[J]. Journal of Intelligent Learning Systems Applications, 2015, 07(4):117-127.

致谢

时间飞逝，转眼间，硕士生活即将结束，在这里我收获很多，我真心的感谢大家。

首先，我要感谢我的导师，谢谢您对我孜孜不倦的教诲，对我论文从选题到构思再到定稿中的每个环节给予的细心的指导，您给出的建议对论文的完成起到了关键性的作用。

在我的学业和论文写作的点点滴滴中，无不倾注着导师辛勤的汗水和心血。老师兢兢业业的工作作风，高尚的人格、宽广的胸怀、渊博的知识、严谨的治学态度、对教育工作的严格要求使我终生受益。在此论文完成之际，谨向我们的老师致以最崇高的敬意和最衷心的感谢！

另外，在校图书馆查找资料的时候，图书馆的老师也给我提供了很多方面的支持与帮助。在此向帮助和指导过我的各位老师表示最中心的感谢。感谢这篇论文所涉及到的各位学者。本文引用了数位学者的研究文献，如果没有各位学者的研究成果的帮助和启发，我将很难完成本篇论文的写作。

最后由衷感谢在生活中一直支持、鼓励和照顾我的家人以及我的舍友，感谢你们三年来的一路相伴以及照顾。

姓名

二零二二年二月

攻读硕士学位期间科研情况

■ 已申请的软著

[1] 一种基于xxx管理系统v1.0 登记号: 2022Rs1010xxxxx.

■ 参与的科研课题

[1] 基于xxx实现, 2020 – 2021, 参与人