

# 基于可擦除布隆过滤器的分布式档案查阅技术



罗纯琦<sup>1</sup>, 刘旭<sup>2</sup>, 肖鹏程<sup>1</sup>, 朱洲森<sup>2</sup>, 廖雪花<sup>1,\*</sup>

<sup>1</sup>四川师范大学计算机科学学院, 四川成都 610101

<sup>2</sup>四川师范大学物理与电子工程学院, 四川成都 610101

**摘要:** 在当前大数据环境下, 电子档案数据规模呈日益增长的趋势, 由于其散布于不同地域和系统, 其分散性导致了跨域查阅时间成本急剧上升, 在高并发条件下误判率持续攀升, 这直接影响了电子档案信息检索的准确与高效性。针对以上问题提出一种基于计数型布隆过滤器的跨域分布式存储电子档案查阅技术模型, 以支撑对电子档案跨域查阅的准确和高效检索。该技术模型选用计数型布隆过滤器作为检索中间件, 相较于传统布隆过滤器, 计数型布隆过滤器在保证时间复杂度的前提下, 显著减少内存空间消耗, 其关键特性在于不仅能够进行元素存在性检查, 同时记录了元素的访问次数, 且支持动态增删检索元素, 降低误判率, 提升查阅准确度。为解决大规模数据处理需求, 采用分布式架构设计, 达到分散服务器压力的目标, 提高了系统的整体响应速度。同时, 引入Zookeeper确保了系统的高可用性, 确保各节点间的协同工作。经测试及实践应用, 该模型能够提高档案系统查阅效率, 保证系统的高可用性, 更为用户提供了可靠、高效的电子档案管理服务。

**关键词:** 可擦除布隆过滤器; 跨域分布式存储; 电子档案管理

**DOI:** [10.57237/j.cst.2023.04.008](https://doi.org/10.57237/j.cst.2023.04.008)

## Distributed Archival Access Technology Based on Erasable Bloom Filters

Luo Chunqi<sup>1</sup>, Liu Xu<sup>2</sup>, Xiao Pengcheng<sup>1</sup>, Zhu Zhousen<sup>2</sup>, Liao Xuehua<sup>1,\*</sup>

<sup>1</sup>School of Computer Science, Sichuan Normal University, Chengdu 610101, China

<sup>2</sup>School of Physics and Electronic Engineering, Sichuan Normal University, Chengdu 610101, China

**Abstract:** In the current big data environment, the scale of electronic archive data is increasing, due to its dispersion in different regions and systems, its dispersion has led to a sharp increase in the time cost of cross-domain access, and the misjudgment rate continues to rise under high concurrency conditions, which directly affects the accuracy and efficiency of electronic archive information retrieval. In order to solve the above problems, a cross-domain distributed storage electronic archives access technology model based on counting Bloom filter was proposed to support the accurate and efficient retrieval of cross-domain access to electronic archives. Compared with the traditional Bloom filter, the countable bloom filter significantly reduces the memory space consumption under the premise of ensuring the time complexity, and its key feature is that it can not only check the existence of elements, but also record the number of visits to the elements, and support dynamic addition and deletion of retrieved elements Reduce the false positive rate and improve the accuracy of the

\*通信作者: 廖雪花, liaoxuehua@163.com

query. In order to solve the needs of large-scale data processing, the distributed architecture design is adopted to achieve the goal of decentralized server pressure and improve the overall response speed of the system. At the same time, the introduction of Zookeeper ensures the high availability of the system and ensures the cooperation between all nodes. After testing and practical application, the model can improve the efficiency of the file system, ensure the high availability of the system, and provide users with reliable and efficient electronic file management services.

**Keywords:** Erasable Bloom Filter; Cross-domain Distributed Storage; Electronic Records Management

## 1 引言

在信息化不断发展的时代背景下,电子档案作为重要的信息资源,已成为各类组织机构日常运营的核心要素。电子档案的数量和规模呈现出爆炸性增长趋势,尤其在涉及不同领域、不同系统之间的跨域电子档案查阅,传统电子档案管理和检索方法逐渐显现出明显局限性,包括查阅速度缓慢等现象。

本研究的目标是解决大数据环境下电子档案跨域分布式存储问题,同时提高其检索效率。跨域分布式存储在实际应用中可能涉及到数据同步一致性、访问性能等方面的重要考虑因素。本研究旨在针对这些关键问题进行深入研究,以实现高效的电子档案管理和查阅。针对以上问题提出一种基于可擦除布隆过滤器的技术模型,旨在提高电子档案查阅效率和准确性。<sup>[1]</sup>

布隆过滤器是一种高效数据结构,用于快速确定元素是否存在于集合中<sup>[2]</sup>。由于传统布隆过滤器在元素删除方面存在不足,在元素加入后无法进行实时修改及删除。

为克服这一限制,本文采用了Zookeeper与计数型布隆过滤器相结合,以满足电子档案数据实时更新与查询,且支持动态增删操作。此外,为应对大数据环境下高并发查询和分布式存储,设计了一套分布式架构模型,支持电子档案的跨域分布式存储和查阅。通过大量测试和实践应用,对该技术模型进行了全面仿真评估,实证结果证明了其实用性和可行性。

综上所述,本文旨在构建高效可靠的电子档案跨域分布式存储与查阅技术模型,引入可擦除布隆过滤器和分布式架构来解决电子档案管理中的问题。在接下来的章节中,本文将深入探讨技术模型的创新之处以及在电子档案管理领域的实际应用。

## 2 相关工作

在当前档案系统的研究中,跨域数据访问和分布式存储是备受关注的领域。研究人员正在致力于开发能够

在不同地域和组织之间实现高效数据共享的跨域档案系统。在分布式系统中频繁出现的信息尤其是譬如电子档案等敏感信息在网络中的传输,因而确保其传输途中的安全性将尤为重要<sup>[3]</sup>。此外,随着大数据和人工智能技术AI的崛起,档案系统与数据挖掘、智能搜索等方向的整合已成为研究的焦点。安全性和隐私保护方面的工作也在不断深化,以应对不断升级的网络威胁。

本文在此基础上引入了布隆过滤器,布隆过滤器自巴顿布隆于1970年提出以来<sup>[4, 5]</sup>,被广泛应用于各种计算机系统来表示庞大数据以及提高查找效率<sup>[6]</sup>。为解决传统布隆过滤器的缺陷,本研究提出一种基于可擦写布隆过滤器的创新型改进方案。

标准布隆过滤器是一种概率型数据结构,具有节省空间和高安全性等优点,并且能够快速响应系统的查询<sup>[7]</sup>。例如,卢建元<sup>[8]</sup>提出了将多个哈希函数改为单一哈希函数的方法,以提高散列速度,并通过分组连续素数的方式降低哈希碰撞率以提高查询准确度;王乾<sup>[9]</sup>等人提出将布隆过滤器改为双向量的结构使其适合硬件,实现可以并行过滤,降低了查找延迟,提高了端口吞吐率;孟慧君<sup>[10]</sup>采用布隆过滤器为URL数据集进行分块并为每一个数据块分配一个线程来实现多个数据块的并行去重操作。耿宏<sup>[11]</sup>等人提出动态布隆计数型数据结构来存储元素,对每个字节增加计数器记录对应相应字节被置为不同的频数实现删除操作。华文镝等人<sup>[12]</sup>在保证每个分块的最大元素负载因子依然处于安全的情况下,尽可能减小分块的粒度,从而提升布谷鸟过滤器的空间效率。

尽管上述改进在某些方面提高了布隆过滤器性能,但仍存在待深入研究的关键问题。这些问题包括以下几点:

- (1) 哈希函数的选择:布隆过滤器的哈希时间占据总体时间耗时大。选择正确的哈希函数以及在哈希函数上的改进十分重要。

- (2) 并发操作的异步执行：计数型布隆过滤器在删除元素时面临并发问题，在删除不同元素对应的相同比特位映射出现哈希碰撞。克服该问题是一个不小的挑战。

### 3 基本原理

#### 3.1 可擦除布隆过滤器的基本原理

布隆过滤器是一种用于解决集合存在问题的概率数据结构。布隆过滤器由一个  $m$  位的位数组和  $k$  个相互独立的哈希函数构成。布隆过滤器空间占用低，查询速度快，实现简单。用  $b(0), b(1), \dots, b(m-1)$  表示位数组的所有位。初始状态下，所有位都初始化为 0。 $k$  个哈希函数的输出  $h_0(x), h_1(x), \dots, h_k(x)$ ，范围是 0 到  $m-1$ 。用  $S$  表示要存储的键的集合。为了将元素  $x'$  插入到布隆过滤器中，将  $k$  个位  $b(h_0(x')), b(h_1(x')), \dots, b(h_{k-1}(x'))$  置为 1。为了检查元素  $x'$  是否在集合  $S$  中，检查是否所有  $k$  位输出等于 1。如果结果为真，布隆过滤器给出

元素  $x'$  可能存在于集合  $S$  中的结论，假阳性率为  $e$ 。否则，布隆过滤器给出结论元素  $x'$  一定不在集合  $S$  中 [13]。其中以下两个指标在布隆过滤器中十分重要：

误判率问题：布隆过滤器的误判率是一个关键性能指标，通常用以下两个公式描述：

误判率计算：

$$FPR = (1 - (1 - (1/n))^{(k * m)})^k$$

其中， $n$  表示位数组的长度（位数）。 $k$  表示使用的哈希函数的数量。 $m$  表示插入到过滤器中元素数量。

最佳哈希函数数量的估算：

$$k \approx (n / m) * \ln(2)$$

一般哈希函数数量以 3 至 5 个为宜。

可擦除布隆过滤器是对传统布隆过滤器的拓展。传统布隆过滤器通过多个哈希函数将数据映射到位数组中不同位置并判断某个元素是否存在集合中，传统布隆过滤器结构示例如图 1 所示。

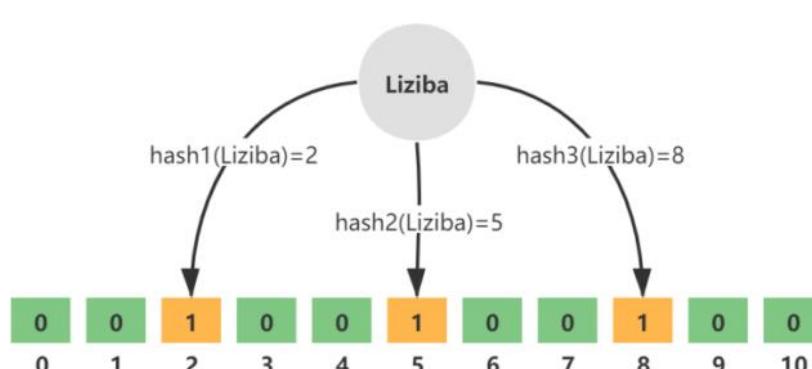


图 1 传统布隆过滤器结构示例

传统布隆过滤器无法有效地删除已添加的元素，因为存在哈希碰撞导致在删除一个元素的时候可能影响其它元素判断结果。可擦除布隆过滤器引入计数器，实现对元素的精准删除，计数型布隆过滤器与传统布

隆过滤器不同的是前者记录了该比特位被添加的次数，即使发生哈希碰撞不影响查询结果的准确。计数型布隆过滤器结构示例如图 2 所示。

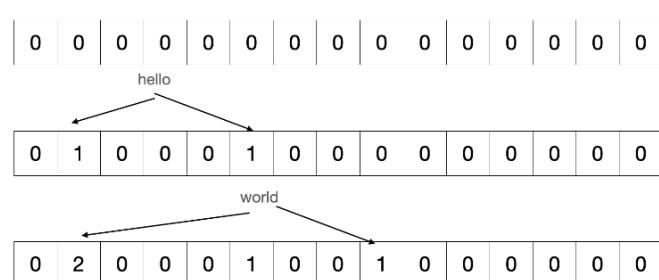


图 2 计数型布隆过滤器结构示例

当删除操作发生时, 相应计数器值递减。只有当计数器值为零时, 表示该元素不存在于过滤器中。它允许数据的添加、查询和删除。

### 3.2 Zookeeper 基本原理

Zookeeper 命名空间内部拥有一个树状的内存模型, 其中各节点被称为 Znode。每个 Znode 包含一个路径和与之相关的元数据[14]。

Zookeeper 对客户端发出对应的事件通知, 又称基于通知的事件消息机制[15]。其中 Watch 机制是一种事件通知系统, 为分布式环境中的实时数据变化提供了有效的通知机制。在这个机制中, 客户端可以通过注册 Watch 来监视特定节点的状态变化, 以便在数据被创建、更新或删除时及时收到通知。一次注册的 Watch 是一次性触发的, 即在 Watch 被触发后, 客户端需要重新注册以继续监视后续的变化。

Zookeeper 的 Watch 分为两种类型, 分别用于监视数据节点和子节点。数据节点的 Watch 用于检测节点数据的变化, 而子节点的 Watch 则用于监视子节点的创建、删除等变化。

### 3.3 各参数的抉择

在实现高效的跨域分布式存储电子档案查阅系统中, 性能优化是确保系统顺畅运行和提高用户体验的关键因素。

首先, 哈希函数的选择至关重要。本研究使用高效且分布良好的散列算法——MurmurHash, 该算法通常在散列短文本例如数字和字母的组合相比于其它哈希函数具有速度快, 散列系数大的优势。

其次, 哈希函数的数量是平衡性能和误判率的关键。经计算, 将每个位数组的大小控制成连续的质数避免发生哈希碰撞降低误判率。哈希函数的数量过多同时也会带来更多的计算和存储开销。涉及到使用布隆过滤器的设计公式以及根据实际域的数据量来选择哈希函数的数量从而实现性能和存储效率的平衡。

在跨域分布式存储系统中, 数据访问策略是性能优化中的一个关键因素。本研究采用 Redis 缓存策略, 将频繁访问的数据缓存至 Redis 服务器中, 以减少网络延迟提升数据访问速度。

此外, 为保持数据的实时性, 设置适当的过期时间是必要的, 特别是对于那些频繁变动的数据。如热门的

档案、常用的配置信息等。对于计算密集型的数据, 通过缓存计算结果能够避免不必要的重复计算, 极大提升系统响应速度。静态且不经常变化的数据, 比如静态页面内容或全局设置, 同样是合适的缓存对象。

总体而言, 实现高效的跨域分布式存储电子档案查阅系统需要综合考虑误判率、数据访问策略等方面。通过优化布隆过滤器、采用有效的 Redis 缓存机制、确保系统在复杂的分布式环境中具备高性能和稳定的运行能力, 为用户提供流畅的电子档案查阅体验。

## 4 实现过程

### 1 数据添加

在系统中选择 MurmurHash 哈希函数, 为了确保对电子档案编号的哈希输出具有良好的均匀分布特性。MurmurHash 因其高效的计算速度和对输入数据的均匀分散而成为本研究选择的哈希函数。数据添加步骤如下:

- 1) 在将新电子档案编号插入系统时, 首先应用 MurmurHash 哈希函数, 生成多个不同的哈希值, 减少冲突可能。确保对电子档案编号的全面覆盖和均匀分布。
- 2) 针对每个生成的哈希值, 对相应的计数器值进行增加。计数器值用于记录每个电子档案编号的插入次数, 有助于系统更精确地判断数据的存在或查询频数, 为进一步性能优化提供基础。
- 3) 通过取模运算将哈希值映射到布隆过滤器的比特位范围内。
- 4) 将相应的比特位上的计数器值进行更新, 维护了每个档案编号的插入计数。

### 2 数据查询

在查询处理阶段, 其实现步骤如下:

- 1) 将这个哈希值集合中的每个哈希值映射到布隆过滤器的相应位置
- 2) 检查这些位置上的比特位状态。
- 3) 当所有位置上的比特位都被置为 1 时, 系统认为查询数据可能存在于布隆过滤器中。
- 4) 反之, 任何一个位置的比特位为 0, 系统即判定查询数据不存在。

### 3 数据的擦写

数据的擦写具体的实现步骤如下:

- 1) 输入所需删除元素。
- 2) 将所需删除元素通过哈希映射到对应位上。

### 3) 实现对应哈希位置上计数器的递减。

计数型布隆过滤器的计数器删除操作经历一系列逐步减小的步骤，有助于防止误删其他数据。该策略有效地降低了误删的概率，保障了数据删除的准确性。这样的设计在保持布隆过滤器高效查询性能的同时，实现了对数据的安全、渐进式擦写。

### 4 跨域数据同步

在基于 Zookeeper 的分布式环境中进行跨域数据同步，确保数据一致性的步骤涉及到多个方面。

数据同步的过程需要从当前域中提取变更信息。首先，进行节点的创建、更新和删除，以及相关的计数器信息、版本号。同时让客户端监听节点的变更，一旦节点发生变更，客户端将收到通知。

其次，数据同步通过 Zookeeper 的 Watch 机制实现。当在一个域内的节点发生变化时，Zookeeper 会触发 Watch 事件，通知注册了 Watch 的客户端。通过注册 Watch，可以实现对数据变更的即时感知。其它域订阅这些变更通知，及时更新其本地的可擦除布隆过滤器和数据状态。

跨域数据同步流程如图 3 所示，对于不同域中电子档案，分别构建基于位数组不同的可擦除布隆过滤器。每个域维护所在域的过滤器，快速判断特定数据是否存在于本域中。当用户需要访问特定数据时，首先查询所在域的布隆过滤器，若判断结果为存在，则从数据库中获取数据。若不存在则用 Zookeeper 向所有域广播，发送查询请求，每个请求在域中使用可擦除布隆过滤器快速返回查询结果是否在该域中，将请求返回给发送请求的一方，利用布隆过滤器高效查询优势，提高系统响应速度。该技术在跨域分布式存储电子档案查阅中具有广泛应用潜力。

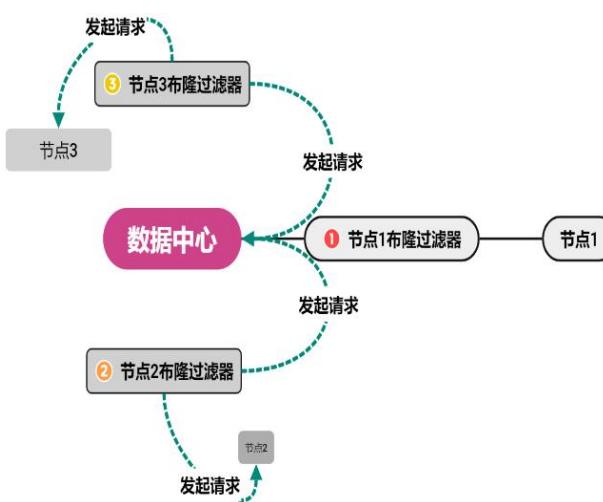


图 3 系统流程图

当某个域需要删除特定数据时，只需递减相应计数器值。保证了在删除数据的同时不影响其它数据判断结果，有效释放存储空间。

## 5 实验对比与分析

### 5.1 环境配置说明

具体配置如表 1 所示：

表 1 测试环境配置

名称	配置
处理器	Intel(R)Core(TM)i7-10870U CPU @ 1.10GHz 1.61 GHz
内存	16GB DDR4
系统类型	64 位操作系统

### 5.2 时间性能测试分析

在时间性能测试中，对该模型进行了时间性能测试，并分析实验结果。

实验设计：模拟不同并发用户数的查阅请求，以评估系统的时间性能。

测试设置如表 2 所示：

表 2 两种不同查询方式消耗时间对比

数据集规模	传统查询方式 (ms)	可擦写布隆过滤器 (ms)
10	167	35
20	389	110
40	689	280
60	1028	940
80	1273	1081
100	2093	1198

根据实验数据，我们对系统性能进行了深入分析，全面地理解其性能特征。

响应时间与并发用户数关系：实验结果展示了系统性能与并发用户数量之间的关系。在低并发情况下，系统表现出色，响应时间较短。并发用户数量的增加，响应时间显著上升，表明系统在高负载下面临性能瓶颈。

随着并发用户数量的增加，系统面临着资源竞争的问题。在高并发量前多个用户同时竞争有限的系统资源，CPU 资源、内存和网络带宽，导致资源争夺阻碍了系统及时满足所有请求的能力，延长了整体的响应时间。

实验结果可知，传统查询方式在高并发场景下表现较为疲弱，主要原因之一是跨域访问耗时久且查询

速度慢。在高并发查询的情况下,一旦查询时间升高传统系统方式会遇到 IO 瓶颈系统阻塞,甚至面临着崩溃的风险,无法再容纳正常请求,从而导致系统拥挤甚至不可用。与传统查询方式相比,本文提出的可擦写布隆过滤器性能优越,系统开销低,表现良好,更加适合用户进行跨域方式进行电子档案的查询与传递。

### 5.3 空间性能测试分析

**实验设计:** 测试不同大小的电子档案数据集,以评估系统的空间性能。

测试设置如下:

(1) 数据集规模分别是: 10、15、20、30、40、50、60、70、80、90、100GB。

#### (2) 数据类型: 短文本

测试结果: 以下是不同数据集规模下的存储需求(以 GB 为单位),如表 3、图 4 所示。

表 3 不同规模数据集下的存储空间消耗

数据集规模 (GB)	传统查询方式 (MB)
10	27.34
15	41.02
20	54.72
30	82.10
40	109.48
50	136.85
60	164.23
70	191.61
80	219.00
90	246.38
100	273.76

消耗空间大小 (单位MB)

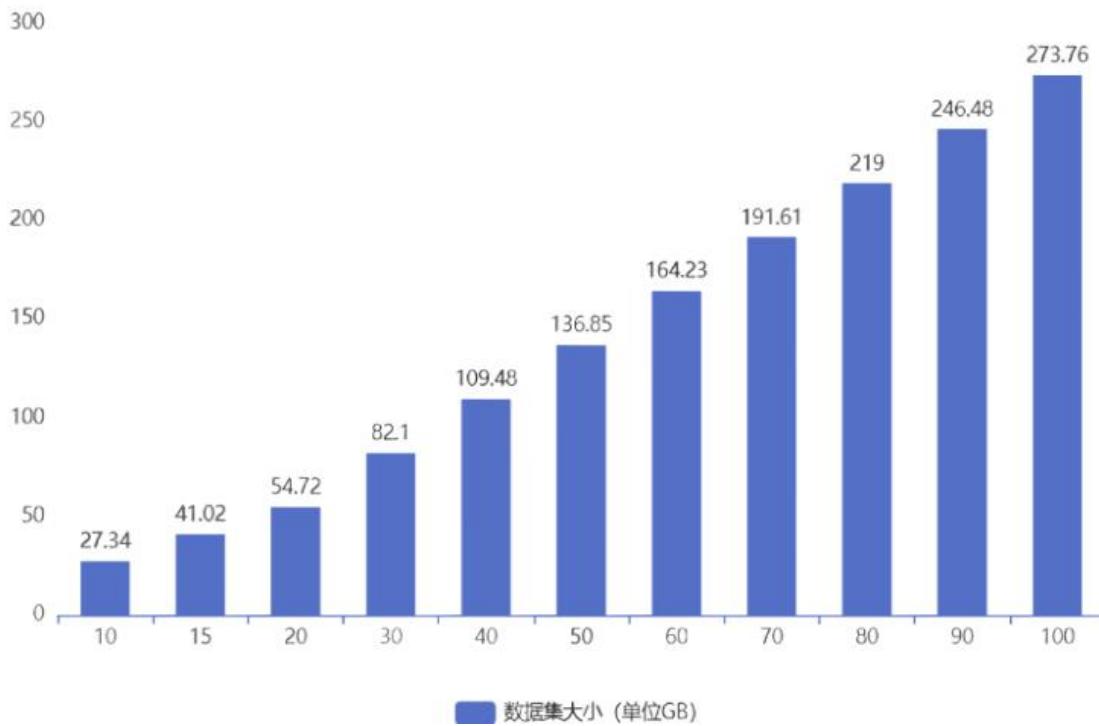


图 4 不同存储消耗空间大小

存储数据集从 10 万到 100 万时可擦除布隆过滤器的内存空间使用情况如表 3、图 4 所示,本文提出的可擦除高效过滤器在位数组大小、哈希函数的数量、哈希算法等方面进行优化,在内存空间消耗方面占据优势。在不断增长的数据规模下,新型可擦写过滤器持续展现其内存空间利用的卓越性能。这对于高并发场

景下系统的应用具有重大意义,更为紧凑的内存占用既能够提升系统的整体资源利用率,又有助于降低系统维护和运营的成本,为系统在大规模、高并发环境中的长期稳定运行提供了更为可靠的基础。

通过实验结果的分析,本文提出的基于可擦除布隆过滤器的跨域分布式存储电子档案查阅技术,解决

了在分布式环境下跨域查询问题，将可擦写布隆过滤器和 Zookeeper 的结合，更快地进行跨域查询。为了提高系统的可扩展性[16]，在高并发情况下，该系统依然保持出色的性能。提升了系统的查询性能和并发处理能力。

## 6 结语

本文引入了一种创新性的解决方案，即新型可擦除布隆过滤器与跨域分布式存储电子档案查阅技术结合以应对所面临的多重挑战。该解决方案经过设计和实现，在功能、操作和跨平台运行上具有显著的实用价值，解决了传统查询方法的限制，构筑高效可靠的分布式跨域电子档案查阅系统；降低了系统内存空间消耗。在提升查询效率的同时极大地减少了系统资源占用，为高并发下的系统运行提供了更为灵活和高效的解决途径。通过实际实验验证了该系统在跨域分布式环境中的可行性和有效性。

## 参考文献

- [1] 雷蒙; 肖文超; 高佳宁; 廖雪花 基于位标识的可擦写高效过滤器算法与实现 [J]. 软件导刊, 2022(8): 120-125.
- [2] 张镱; 丁帅; 乔庐峰; 陈庆华; 刘熹; 邹仕祥 一种基于分段式路由查找的布隆过滤方案 [J]. 通信技术, 2022(10): 1300-1306.
- [3] 梁静 微服务框架下敏感信息的交叉跨域安全通信技术研究 [D]. 成都: 四川师范大学, 2020.
- [4] 安振林; 林琼正; 杨磊; 娄伟; 谢磊 物理层中跨商用RFID 的布隆过滤器获取 [J]. 计算机网络与通信, 2020, 28(4): 1804-1817.
- [5] 王飞越. 基于负载均衡的高效布谷鸟过滤器研究 [D]. 华中科技大学, 2019.
- [6] 李云川. 基于深度学习的多键布隆过滤器 [D]. 成都: 电子科技大学, 2023.
- [7] 南军虎; 高欢; 汪超群; 一种新型旋流排沙渠道的试验研究和数值模拟 [J]. 2022, 54(6): 116-123.
- [8] 卢建元. 高性能哈希技术及其应用的研究 [D]. 北京: 清华大学.
- [9] 王乾, 乔庐峰, 陈庆华. 可加速最长前缀匹配的布隆过滤查找方案 [J]. 通信技术, 2020, 53(07): 1674-1679.
- [10] 孟慧君. 基于 Bloom Filter 算法的 URL 去重算法研究及其应用 [D]. 河南大学, 2019.
- [11] 耿宏, 李勇猛. 基于分层布隆过滤器的发布订阅自动发现算法 [J]. 计算机工程与设计, 2019, 40(12): 3494-3499.
- [12] 华文镝. 基于布谷鸟过滤器的高效近似成员查询结构研究 [D]. 西宁: 青海师范大学, 2023.
- [13] 杨斐. 学习型布隆过滤器优化方法研究与实现 [D]. 合肥: 中国科学技术大学, 2023.
- [14] 陈乐. 基于分布式集群的高可用日志分析系统的设计 [J]. 中国电子科学研究院学报, 2020(5): 420-426.
- [15] 郭晓东. 基于 ZooKeeper 的计费系统多级故障隔离方案 [J]. 长江信息通信, 2021(10): 186-189.
- [16] 程庚. 基于 Zookeeper 的大数据处理调度系统的设计与实现 [D]. 武汉: 华中科技大学, 2023.