(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2023 年 12 月 28 日 (28.12.2023)

WIPO | PCT

(10) 国际公布号
WO 2023/246754 A1

(51) 国际专利分类号:
G06F 16/174 (2019.01)

(21) 国际申请号: PCT/CN2023/101303

(22) 国际申请日: 2023 年 6 月 20 日 (20.06.2023)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202210730080.0 2022年6月24日 (24.06.2022) CN
202211132110.4 2022年9月16日 (16.09.2022) CN

(71) 申请人: 华为技术有限公司 (**HUAWEI TECHNOLOGIES CO., LTD.**) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 朱洪德 (**ZHU, Hongde**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 董如良 (**DONG, Ruliang**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 陈泽晖 (**CHEN, Zehui**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 罗斯哲 (**LUO, Sizhe**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(74) 代理人: 深圳市深佳知识产权代理事务所（普通合伙） (**SHENPAT INTELLECTUAL PROPERTY AGENCY**); 中国广东省深圳市罗湖区南湖街道春风路庐山大厦B座18C2、18D、18E、18E2, Guangdong 518001 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

(54) **Title:** DATA DEDUPLICATION METHOD AND RELATED SYSTEM

(54) 发明名称: 一种数据重删方法及相关系统

图 5

0 Controller
100 Application server
134 Hard disk
S502 The controller 0 receives a write request from the application server 100
S504 Divide data in the write request into blocks, so as to obtain at least one data block
S506 Determine a fingerprint of the at least one data block
S508 According to a fingerprint of a first data block among the at least one data block, query a fingerprint table
S509 The controller 0 returns a write response
S510 The controller 0 writes the first data block into the hard disk 134
S511 Write a logical address and a physical address of the first data block into an address mapping table
S512 Insert the fingerprint and the logical address of the first data block in a first partition of an inverse mapping table
S514 The controller 0 deletes the first data block from the hard disk 134 according to address information of the first data block
S516 Delete the fingerprint and the logical address of the first data block from the first partition of the inverse mapping table
S518 Write the fingerprint and the logical address in the inverse mapping table, which is subjected to deduplication, into the fingerprint table
S520 When a partition in the mapping table meets an elimination condition, eliminate metadata from the partition
S522 Modify, into a fingerprint, the physical address of the first data block, which is subjected to deduplication, in the address mapping table
AA A fingerprint, which is the same as the fingerprint of the at least one first data block, is present
BB A fingerprint, which is the same as the fingerprint of the at least one first data block is not present
CC A fingerprint, which is the same as the fingerprint of the first data block, is present in the first partition

(57) **Abstract:** Provided in the present application is a data deduplication method. The method comprises: receiving a write request; writing a first data block in the write request into a storage device; writing metadata of the first data block, such as a fingerprint and a logical address, into a first partition, which is determined according to a feature of the first data block, among a plurality of partitions of a metadata management structure; when a fingerprint, which is the same as the fingerprint of the first data block, is present in the first partition, deleting the metadata of the first data block from the first partition; and according to address information of the first data block, deleting the first data block from the storage device. In the method, a metadata management structure is actively partitioned, and metadata of a data block is written into a partition corresponding to a feature of the data block, such that the problem of unfrequently

WO 2023/246754 A1

[见续页]

BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:
— 包括国际检索报告(条约第21条(3))。

updated data being eliminated because resources of the unfrequently updated data are occupied by frequently updated data, thereby increasing the deduplication rate.

(57) 摘要: 本申请提供了一种数据重删方法，包括：接收写请求，将写请求中的第一数据块写入存储设备，将第一数据块的元数据，如指纹和逻辑地址，写入元数据管理结构的多个分区中根据第一数据块的特征确定的第一分区，当第一分区中存在与第一数据块的指纹相同的指纹时，删除第一分区中第一数据块的元数据，并根据第一数据块的地址信息从存储设备中删除第一数据块。该方法通过对元数据管理结构主动分区，将数据块的元数据写入与该数据块的特征对应的分区，由此避免更新不频繁的数据被更新频繁的数据挤占资源，进而被淘汰，提高了重删率。

一种数据重删方法及相关系统

本申请要求于2022年06月24日提交中国国家知识产权局、申请号为202210730080.0、发明名称为“一种数据重删方法”的中国专利申请的优先权，以及要求于2022年09月16日提交中国国家知识产权局、申请号为202211132110.4、发明名称为“一种数据重删方法及相关系统”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及存储技术领域，尤其涉及一种数据重删方法、装置、存储系统、计算机可读存储介质、计算机程序产品。

背景技术

随着计算产业的发展，数据价值得到充分释放，数据中心的规模从拍字节（petabyte，PB）级向泽字节（zettabyte，ZB）级增长。数据中心存储的大量数据中存在大量冗余数据，统计表明在主存储系统和备份存储系统这两大主要应用场景中，分别存在约50%和85%的数据冗余。如何有效减少冗余数据，进而降低存储成本已成为研究的热点方向。

业界通常采用重复数据删除（Data Deduplication，DD）以减少数据冗余。数据重复删除也可以简称为重删，具体是通过对数据进行分块，并基于数据块的内容计算得到数据块的指纹，再通过比对不同数据块的指纹，识别并删除内容重复的数据块，进而达到消除数据冗余的目标。

其中，数据块的指纹通常是以追加写入方式写入日志文件。在进行重删时，通过手动或周期性触发的方式将日志文件中的指纹排序，并将排序后的指纹与指纹文件中的指纹合并，根据合并结果删除内容重复的数据块。

数据中心存储的数据可以根据更新频次分为更新频繁的数据和更新不频繁的数据。然而，更新频繁的数据的占比通常高于更新不频繁的数据的占比。更新频繁的数据通常难以被重删，也即难以被重删的数据占比反而高，此时就会出现资源挤占的情况，更新不频繁的数据所分配的空间占比就会较低，由此导致易被淘汰，从而损失重删率。

发明内容

本申请提供了一种数据重删方法，该方法通过对元数据管理结构进行主动分区，将数据块的指纹以及地址信息等元数据写入与数据块的特征对应的分区，由此避免更新不频繁的数据被更新频繁的数据挤占资源，进而被淘汰，提高了重删率。本申请还提供了上述方法对应的装置、存储系统、计算机可读存储介质以及计算机程序产品。

第一方面，本申请提供一种数据重删方法。该方法可以应用于存储系统，包括集中式存储系统或者分布式存储系统。其中，集中存储式系统还可以分为盘控一体或盘控分离的集中式存储系统，分布式存储系统还可以分为存算一体的分布式存储系统或存算分离的分布式存储系统。集中式存储系统具有引擎，该引擎包括控制器，控制器可以包括处理器和内存，处理器可以加载内存中的程序代码，从而执行本申请的数据重删方法。类似地，分布式存储系统包括计算节点和存储节点，计算节点包括处理器和内存，处理器可以加载内存中的程序代码，从而执行本申请的数据重删方法。

具体地，存储系统接收写请求，该写请求中包括第一数据块，然后存储系统将第一数据块写入存储设备（例如是硬盘），接着将第一数据块的元数据写入元数据管理结构的多个分区中的第一分区。该第一分区为根据第一数据块的特征确定的。第一数据块的元数据包括第一数据块的指纹及地址信息。当第一分区中存在与第一数据块的指纹相同的指纹时，存储系统删除第一分区中第一数据块的元数据，并根据第一数据块的地址信息从存储设备中删除第一数据块。

在该方法中，不同特征的数据块的元数据可以写入元数据管理结构的不同分区，例如更新频繁的数据块的元数据可以写入容量较小的分区，更新不频繁的数据块的元数据可以写入容量较大的分区，由此

避免更新不频繁的数据被更新频繁的数据挤占资源，进而被淘汰，提高了重删率。

在一些可能的实现方式中，第一数据块的特征为所述第一数据块的指纹。需要说明的是，不同数据块可以对应相同的指纹，例如在进行备份时，可以存在多个数据块对应同一指纹。存储系统在将所述第一数据块的元数据写入元数据管理结构时，可以先确定第一数据块对应的指纹的热度，根据第一数据块对应的指纹的热度确定该热度对应的元数据管理结构的多个分区中的第一分区，然后将第一数据块的元数据写入所述第一分区。

该方法中，存储系统通过确定第一数据块对应的指纹的热度，并根据该热度将第一数据块写入对应的第一分区，可以避免更新不频繁的数据被更新频繁的数据挤占资源，进而被淘汰，提高了重删率。

在一些可能的实现方式中，第一数据块的地址信息包括第一数据块的逻辑地址，相应地，存储系统还可以在将第一数据块的元数据写入第一分区之后，对第一数据块对应的指纹的热度进行更新。其中，存储系统可以确定第一数据块的逻辑地址的热度，将逻辑地址的热度累加至所述第一数据块对应的指纹的热度，以更新所述第一数据块对应的指纹的热度。

在该方法中，通过将逻辑地址的热度累加至数据块对应的指纹的热度，进行指纹的热度更新，可以为后续元数据写入提供参考。

在一些可能的实现方式中，写请求中还可以包括第二数据块，第二数据块对应的指纹的热度可以高于所述第一数据块对应的指纹的热度。相应地，存储系统还可以将第二数据块写入存储设备，将第二数据块的元数据写入元数据管理结构的多个分区中的第二分区。其中，第二分区的容量小于第一分区的容量。

该方法通过将指纹的热度不同的数据块写入元数据管理结构的不同分区，由此避免热度低的数据块的元数据被热度高的数据块的元数据挤占资源，进而被淘汰，提高了重删率。

在一些可能的实现方式中，写请求中还可以包括第三数据块，第三数据块的指纹与第一数据块的指纹相同。当写入第一数据块的元数据时，第一数据块对应的指纹的热度小于预设热度，当写入第三数据块的元数据时，第三数据块对应的指纹的热度大于预设热度，则存储系统可以将第三数据块的元数据写入第二分区，并将第一分区中与第三数据块具有相同指纹的数据块的元数据移动至第二分区。

如此，可以实现随着数据块的不断写入，对元数据的存储位置进行调整，例如将指纹的热度较高的数据块的元数据移动至第二分区，从而在第一分区中为指纹的热度较低的数据块的元数据留出存储空间，以及在第二分区中存储具有相同指纹的数据块的元数据，以支持第二分区触发重删，进一步提升重删率。

在一些可能的实现方式中，存储系统也可以将第三数据块的元数据写入第二分区，将第一分区中与第三数据块具有相同指纹的数据块的元数据淘汰，无需移动至第二分区，一方面可以减少移动开销，另一方面可以在第一分区中为指纹的热度较低的数据块的元数据留出存储空间，进一步提升重删率。

在一些可能的实现方式中，元数据管理结构的多个分区的容量根据分区决策模型确定。其中，分区决策模型用于预测预设的分区容量组合中每个分区容量组合应用于所述元数据管理结构后对应的分区收益，并确定分区收益最大的分区容量组合作为所述元数据管理结构的多个分区的容量，所述分区收益根据重删率和分区调整成本中的至少一个确定。

该方法通过构建分区决策模型，通过分区决策模型对元数据管理结构进行主动分区，由此避免了更新频繁的数据挤占更新不频繁的数据的资源，进而避免更新不频繁的数据被淘汰，导致损失重删率。

在一些可能的实现方式中，分区收益可以为重删率，预设的分区容量组合可以包括第一容量组合。分区决策模型可以通过预估数据的命中率，从而预测分区容量组合应用于数据管理结构后对应的重删率。

具体地，分区决策模型通过如下方式预测所述第一分区容量组合应用于所述元数据管理结构后对应的重删率：获取所述第一分区容量组合应用于所述元数据管理结构所形成的多个分区中各个分区对应的工作负载特征，根据所述各个分区对应的工作负载特征，获得所述各个分区对应的数据分布，根据所述各个分区对应的数据分布以及所述各个分区的容量，获得所述重删率。

该方法基于各个分区对应的工作负载特征，拟合各个分区对应的数据分布，基于各个分区对应的数据分布和分区容量可以预测命中率，进而预测出应用分区容量组合后对应的重删率，无需实际运行存储系统，即可通过较低成本预测出重删率最大的分区容量组合，能够满足业务的需求。

在一些可能的实现方式中，考虑到工作负载可能发生变化，分区的容量还支持调整。例如，存储系

统可以周期性调整分区的容量。在具体工程实施过程中，分区调整需要重新进行分区初始化等操作，产生分区调整成本。分区决策模型可以基于调整前后的分区容量占比预测分区调整成本。分区决策模型可以根据收益率和分区调整成本，预测分区收益。例如，分区决策模型可以将预测的收益率与预测的分区调整成本的差值，作为预测的分区收益。

该方法通过重构分区收益，可以使得分区收益的评估更精准、合理，以重构的分区收益最大化为目标，所确定的分区容量组合更具有参考价值，能够实现重删率和分区调整成本的均衡。

在一些可能的实现方式中，存储系统可以周期性地调整元数据管理结构的多个分区的容量，当到达调整时刻时，根据所述调整时刻前的周期对应的分区收益、分区容量组合或各个分区对应的工作负载特征，确定是否调整多个分区的容量。

该方法通过调整时刻前的周期对应的反馈信息，如分区收益、分区容量组合或各个分区对应的工作负载特征，决策是否对分区的容量进行调整，使得分区能够基于工作负载的变化灵活调整，尽可能保障在不同阶段均具有较好的分区收益。

在一些可能的实现方式中，存储系统可以在第一分区中存储与第一数据块的指纹相同的指纹，且第一分区中与第一数据块的指纹相同的指纹的数量达到预设阈值时，删除第一数据块的元数据，并根据第一数据块的地址信息从存储设备中删除第一数据块。

其中，预设阈值可以根据经验值设置。例如，预设阈值可以设置为1，则第一分区中存在与第一数据块相同的指纹，存储系统即删除第一数据块的元数据以及第一数据块。又例如，预设阈值可以设置为2，则第一分区中存在2个数据块的指纹与第一数据块的指纹相同时，存储系统删除第一数据块的元数据以及第一数据块，进一步地，存储系统保留与第一数据块的指纹相同的2个数据块中的一个数据块及其元数据，删除另一个数据块及其元数据。

当预设阈值设置为较小值时，可以及时删除冗余的数据块和元数据，当预设阈值设置为较大值时，可以减少重删次数，避免频繁重删占用大量资源，影响业务正常运行。

在一些可能的实现方式中，第一数据块的地址信息为第一数据块的逻辑地址。存储系统还可以将第一数据块的逻辑地址和物理地址写入地址映射表。相应地，存储系统在删除第一数据块时，可以根据所述第一数据块的逻辑地址，从所述地址映射表中获取所述第一数据块的物理地址，然后根据所述物理地址从所述存储设备中找到所述第一数据块，并删除所述第一数据块。

在该方法中，存储系统基于地址映射表中逻辑地址到物理地址的单跳映射直接定位第一数据块，缩短了查找时间，提高了重删效率。

在一些可能的实现方式中，在删除第一分区中的第一数据块的元数据之后，存储系统还可以将前向映射表中第一数据块的物理地址修改为所述第一数据块的指纹。

该方法可以实现对被重删的数据块的重定位，以便于后续可以基于指纹查找到具有相同指纹的数据块的物理地址，并通过访问该物理地址访问该数据块。

在一些可能的实现方式中，存储系统还可以在所述逆映射表中的至少一个分区满足淘汰条件时，对所述至少一个分区中的所述元数据进行淘汰。

该方法通过对逆映射表进行元数据淘汰，以降低元数据的规模，进而降低内存开销，保障系统性能。

第二方面，本申请提供一种数据重删装置。所述装置包括：

通信模块，用于接收写请求，所述写请求中包括第一数据块；

写数据模块，用于将所述第一数据块写入存储设备；

所述写数据模块，还用于将所述第一数据块的元数据写入元数据管理结构的多个分区中的第一分区，所述第一分区为根据所述第一数据块的特征确定的，所述第一数据块的元数据包括所述第一数据块的指纹及地址信息；

重删模块，用于在所述第一分区中存在与所述第一数据块的指纹相同的指纹时，删除所述第一分区中所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

在一些可能的实现方式中，所述第一数据块的特征为所述第一数据块对应的指纹，所述写数据模块具体用于：

确定所述第一数据块对应的指纹的热度；

根据所述第一数据块对应的指纹的热度确定所述热度对应的所述元数据管理结构的多个分区中的所述第一分区；

将所述第一数据块的元数据写入所述第一分区。

在一些可能的实现方式中，所述写数据模块还用于：

在将所述第一数据块的元数据写入所述第一分区之后，确定所述第一数据块的逻辑地址的热度；

将所述逻辑地址的热度累加至所述第一数据块对应的指纹的热度，以更新所述第一数据块对应的指纹的热度。

在一些可能的实现方式中，所述写请求中还包括第二数据块，所述第二数据块对应的指纹的热度高于所述第一数据块对应的指纹的热度，所述写数据模块还用于：

将所述第二数据块写入所述存储设备，将所述第二数据块的元数据写入所述元数据管理结构的多个分区中的第二分区，所述第二分区的容量小于所述第一分区的容量。

在一些可能的实现方式中，所述元数据管理结构的多个分区的容量根据分区决策模型确定，所述分区决策模型用于预测预设的分区容量组合中每个分区容量组合应用于所述元数据管理结构后对应的分区收益，并确定分区收益最大的分区容量组合作为所述元数据管理结构的多个分区的容量，所述分区收益根据重删率和分区调整成本中的至少一个确定。

在一些可能的实现方式中，所述分区收益为重删率，所述预设的分区容量组合包括第一分区容量组合，所述分区决策模型通过如下方式预测所述第一分区容量组合应用于所述元数据管理结构后对应的重删率：

获取所述第一分区容量组合应用于所述元数据管理结构所形成的多个分区中各个分区对应的工作负载特征；

根据所述各个分区对应的工作负载特征，获得所述各个分区对应的数据分布；

根据所述各个分区对应的数据分布以及所述各个分区的容量，获得所述重删率。

在一些可能的实现方式中，所述装置还包括分区模块，所述分区模块用于：

周期性地调整所述元数据管理结构的多个分区的容量；

当到达调整时刻时，根据所述调整时刻前的周期对应的分区收益、分区容量组合或各个分区对应的工作负载特征，确定是否调整所述多个分区的容量。

在一些可能的实现方式中，所述重删模块具体用于：

在所述第一分区中存在与所述第一数据块的指纹相同的指纹，且所述第一分区中的所述指纹的数量达到预设阈值时，删除所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

在一些可能的实现方式中，所述第一数据块的地址信息为所述第一数据块的逻辑地址，所述写数据模块还用于：

将所述第一数据块的逻辑地址和物理地址写入地址映射表；

所述重删模块具体用于：

根据所述第一数据块的逻辑地址，从所述地址映射表中获取所述第一数据块的物理地址；

根据所述物理地址从所述存储设备中找到所述第一数据块，并删除所述第一数据块。

在一些可能的实现方式中，所述重删模块还用于：

在删除所述第一分区中的所述第一数据块的元数据之后，将所述地址映射表中所述第一数据块的物理地址修改为所述第一数据块的指纹。

在一些可能的实现方式中，所述装置还包括：

淘汰模块，用于当所述逆映射表中的至少一个分区满足淘汰条件，对所述至少一个分区中的所述元数据进行淘汰。

第三方面，本申请提供一种计算机集群。所述计算机集群包括至少一台计算机，所述至少一台计算机包括至少一个处理器和至少一个存储器。所述至少一个处理器、所述至少一个存储器进行相互的通信。所述至少一个处理器用于执行所述至少一个存储器中存储的指令，以使得计算机或计算机集群执行如第一方面或第一方面的任一种实现方式所述的数据重删方法。

第四方面，本申请提供一种计算机可读存储介质，所述计算机可读存储介质中存储有指令，所述指令指示计算机或计算机集群执行上述第一方面或第一方面的任一种实现方式所述的数据重删方法。

第五方面，本申请提供了一种包含指令的计算机程序产品，当其在计算机或计算机集群上运行时，使得计算机或计算机集群执行上述第一方面或第一方面的任一种实现方式所述的数据重删方法。

本申请在上述各方面提供的实现方式的基础上，还可以进行进一步组合以提供更多实现方式。

附图说明

为了更清楚地说明本申请实施例的技术方法，下面将对实施例中所需使用的附图作以简单地介绍。

图 1 为本申请实施例提供的一种集中式存储系统的系统架构图；

图 2 为本申请实施例提供的一种分布式存储系统的系统架构图；

图 3 为本申请实施例提供的一种追加写入日志文件进行元数据管理的示意图；

图 4 为本申请实施例提供的一种通过日志文件和逆映射表进行元数据管理的示意图；

图 5 为本申请实施例提供的一种数据重删方法的流程图；

图 6 为本申请实施例提供的一种数据重删方法的流程图；

图 7 为本申请实施例提供的一种系统资源特征提取的示意图；

图 8 为本申请实施例提供的一种特征归并的示意图；

图 9 为本申请实施例提供的一种获取结构化特征的示意图；

图 10 为本申请实施例提供的一种分区决策建模的流程示意图；

图 11 为本申请实施例提供的一种评估策略选择的流程示意图；

图 12 为本申请实施例提供的一种数据重删方法的应用场景示意图；

图 13 为本申请实施例提供的一种数据重删方法应用于全局缓存的流程示意图；

图 14 为本申请实施例提供的一种数据重删装置的结构示意图。

具体实施方式

本申请实施例中的术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。

首先对本申请实施例中所涉及到的一些技术术语进行介绍。

重复数据删除（Data Deduplication，DD），也可以简称为重删，是一种对重复数据进行删除，使得存储介质中对于相同数据仅存储一份，从而节约数据存储空间的数据缩减方案。重删具体可以通过对数据进行分块，并基于数据块的内容计算得到数据块的指纹（fingerprint，FP），再通过比对不同数据块的指纹，识别并删除内容重复的数据块，进而达到消除数据冗余的目标。

指纹是指基于数据块的内容确定的、用于标识数据块的身份信息。数据块的指纹可以是通过消息摘要算法对数据块的内容计算所得的消息摘要。其中，消息摘要算法通常基于散列函数，也即哈希（hash）函数实现，因此，数据块的指纹也可以是通过散列函数或哈希函数确定的散列值或哈希值。

重删还可以按照执行时间分类。例如，重删可以包括前重删或后重删。前重删是指数据在写入存储介质（简称为存储，例如可以是硬盘等设备）之前，进行重删。前重删也可以称作在线重删。后重删是指数据在写入存储介质（例如是硬盘等设备）之后，进行重删。后重删也称作后台重删、离线重删。

本申请实施例提供的数据重删方法可以应用于不同应用场景，例如可以应用于集中式存储系统或分布式存储系统。

所谓集中式存储系统就是指由一台或多台主设备组成中心节点，数据集中存储于这个中心节点中，并且整个系统的所有数据处理业务都集中部署在这个中心节点上。换言之，集中式存储系统中，终端或客户端仅负责数据的录入和输出，而数据的存储与控制处理完全交由中心节点来完成。集中式系统最大的特点就是部署结构简单，无需考虑如何对服务进行多个节点的部署，也就不用考虑多个节点之间的分布式协作问题。

其中，集中式存储系统可以包括盘控一体的集中式存储系统，或盘控分离的集中式存储系统。盘控

一体是指存储介质（如硬盘）与控制器是一体化的，盘控分离是指存储介质与控制器分离。

图 1 为本申请实施例所应用的一种集中式存储系统的系统架构图，在图 1 所示的应用场景中，用户通过应用程序来存取数据。运行这些应用程序的计算机被称为“应用服务器”。应用服务器 100 可以是物理机，也可以是对物理机进行虚拟化形成的虚拟机。物理机包括但不限于桌面电脑、服务器、笔记本电脑以及移动设备。

应用服务器通过光纤交换机 110 访问存储系统以存取数据。然而，交换机 110 只是一个可选设备，应用服务器 100 也可以直接通过网络与存储系统 120 通信。或者，光纤交换机 110 也可以替换成以太网交换机、无限带宽（InfiniBand，IB）交换机、基于融合以太网的远程直接内存访问 (RDMA over Converged Ethernet，RoCE)交换机等。

图 1 所示的存储系统 120 是一个集中式存储系统。集中式存储系统的特点是有一个统一的入口，所有从外部设备来的数据都要经过这个入口，这个入口就是集中式存储系统的引擎 121。引擎 121 是集中式存储系统中最为核心的部件，许多存储系统的高级功能都在其中实现。

如图 1 所示，引擎 121 中有一个或多个控制器，图 1 以引擎包含两个控制器为例予以说明。控制器 0 与控制器 1 之间具有镜像通道，那么当控制器 0 将一份数据写入其内存 124 后，可以通过所述镜像通道将所述数据的副本发送给控制器 1，控制器 1 将所述副本存储在自己本地的内存 124 中。由此，控制器 0 和控制器 1 互为备份，当控制器 0 发生故障时，控制器 1 可以接管控制器 0 的业务，当控制器 1 发生故障时，控制器 0 可以接管控制器 1 的业务，从而避免硬件故障导致整个存储系统 120 的不可用。当引擎 121 中部署有 4 个控制器时，任意两个控制器之间都具有镜像通道，因此任意两个控制器互为备份。

引擎 121 还包含前端接口 125 和后端接口 126，其中前端接口 125 用于与应用服务器 100 通信，从而为应用服务器 100 提供存储服务。而后端接口 126 用于与硬盘 134 通信，以扩充存储系统的容量。通过后端接口 126，引擎 121 可以连接更多的硬盘 134，从而形成一个非常大的存储资源池。

在硬件上，如图 1 所示，控制器 0 至少包括处理器 123、内存 124。处理器 123 是一个中央处理器（central processing unit，CPU），用于处理来自存储系统外部（服务器或者其他存储系统）的数据访问请求（如读请求或写请求），也用于处理存储系统内部生成的请求。示例性的，处理器 123 通过前端端口 125 接收应用服务器 100 发送的写请求时，会将这些写请求中的数据暂时保存在内存 124 中。当内存 124 中的数据总量达到一定阈值时，处理器 123 通过后端端口将内存 124 中存储的数据发送给硬盘 134 进行持久化存储。

内存 124 是指与处理器直接交换数据的内部存储器，它可以随时读写数据，而且速度很快，作为操作系统或其他正在运行中的程序的临时数据存储器。内存包括至少两种存储器，例如内存既可以是随机存取存储器，也可以是只读存储器（Read Only Memory, ROM）。举例来说，随机存取存储器是动态随机存取存储器（Dynamic Random Access Memory, DRAM），或者存储级存储器（Storage Class Memory, SCM）。DRAM 是一种半导体存储器，与大部分随机存取存储器（Random Access Memory，RAM）一样，属于一种易失性存储器（volatile memory）设备。SCM 是一种同时结合传统储存装置与存储器特性的复合型储存技术，存储级存储器能够提供比硬盘更快速的读写速度，但存取速度上比 DRAM 慢，在成本上也比 DRAM 更为便宜。然而，DRAM 和 SCM 在本实施例中只是示例性的说明，内存还可以包括其他随机存取存储器，例如静态随机存取存储器（Static Random Access Memory，SRAM）等。而对于只读存储器，举例来说，可以是可编程只读存储器（Programmable Read Only Memory, PROM）、可抹除可编程只读存储器（Erasable Programmable Read Only Memory, EPROM）等。另外，内存 124 还可以是双列直插式存储器模块或双线存储器模块（Dual In-line Memory Module，简称 DIMM），即由动态随机存取存储器（DRAM）组成的模块，还可以是固态硬盘（Solid State Disk, SSD）。实际应用中，控制器 0 中可配置多个内存 124，以及不同类型的内存 124。本实施例不对内存 113 的数量和类型进行限定。此外，可对内存 124 进行配置使其具有保电功能。保电功能是指系统发生掉电又重新上电时，内存 124 中存储的数据也不会丢失。具有保电功能的内存被称为非易失性存储器。

内存 124 中存储有软件程序，处理器 123 运行内存 124 中的软件程序可实现对硬盘的管理。例如将硬盘抽象化为存储资源池，然后划分为逻辑单元设备（logic unit number device，LUN）提供给服务器使用等。这里的 LUN 其实就是在服务器上看到的硬盘。当然，一些集中式存储系统本身也是文件服务器，

可以为服务器提供共享文件服务。

控制器 1（以及其他图 1 中未示出的控制器）的硬件组件和软件结构与控制器 0 类似，这里不再赘述。

图 1 所示的是一种盘控分离的集中式存储系统。在该系统中，引擎 121 可以不具有硬盘槽位，硬盘 134 可以放置在硬盘框 130 中。按照引擎 121 与硬盘框 130 之间通信协议的类型，硬盘框 130 可能是串行连接小型计算机系统接口（Serial Attached Small Computer System Interface，SAS）硬盘框，也可能是非易失性内存主机控制器接口规范（non-volatile memory express，NVMe）硬盘框，网际协议（Internet Protocol，IP）硬盘框以及其他类型的硬盘框。SAS 硬盘框，采用 SAS3.0 协议，每个框支持 25 块 SAS 硬盘。引擎 121 通过板载 SAS 接口或者 SAS 接口模块与硬盘框 130 连接。NVMe 硬盘框，更像一个完整的计算机系统，NVMe 硬盘插在 NVMe 硬盘框内。NVMe 硬盘框再通过 RDMA 端口与引擎 121 连接。

后端接口 126 与硬盘框 130 通信。后端接口 126 以适配卡的形态存在于引擎 121 中，一个引擎 121 上可以同时使用两个或两个以上后端接口 126 来连接多个硬盘框 130。或者，适配卡也可以集成在主板上，此时适配卡可通过 PCIE 总线与处理器 112 通信。

需要说明的是，图 1 中只示出了一个引擎 121，然而在实际应用中，存储系统中可包含两个或两个以上引擎 121，多个引擎 121 之间做冗余或者负载均衡。

图 1 所示的存储系统 120 为盘控分离的存储系统。在一些可能的实现方式中，集中式存储系统也可以是盘控一体的存储系统。在盘控一体的存储系统中，引擎 121 可以具有硬盘槽位，硬盘 134 可直接部署在引擎 121 中，后端接口 126 属于可选配置，当系统的存储空间不足时，可通过后端接口 126 连接更多的硬盘或硬盘框。

分布式存储系统是指将数据分散存储在多台独立的存储节点上的系统。分布式存储系统采用可扩展的系统结构，利用多台存储节点分担存储负荷，它不但提高了系统的可靠性、可用性和存取效率，还易于扩展。

如图 2 所示，本实施例提供的存储系统包括计算节点集群和存储节点集群。计算节点集群包括一个或多个计算节点 110（图 2 中示出了三个计算节点 110，但不限于三个计算节点 110），各个计算节点 110 之间可以相互通信。计算节点 110 是一种计算设备，如服务器、台式计算机或者存储阵列的控制器等。在硬件上，如图 2 所示，计算节点 110 至少包括处理器 112、内存 113 和网卡 114。其中，处理器 112 是一个中央处理器（central processing unit，CPU），用于处理来自计算节点 110 外部的数据访问请求，或者计算节点 110 内部生成的请求。示例性的，处理器 112 接收用户发送的写请求时，会将这些写请求中的数据暂时保存在内存 113 中。当内存 113 中的数据总量达到一定阈值时，处理器 112 将内存 113 中存储的数据发送给存储节点 100 进行持久化存储。除此之外，处理器 112 还用于对数据进行计算或处理，例如元数据管理、重删、数据压缩、虚拟化存储空间以及地址转换等。图 2 中仅示出了一个处理器 112，在实际应用中，处理器 112 的数量往往有多个，其中，一个处理器 112 又具有一个或多个处理器核。本实施例不对处理器的数量，以及处理器核的数量进行限定。

内存 113 是指与处理器直接交换数据的内部存储器，它可以随时读写数据，而且速度很快，作为操作系统或其他正在运行中的程序的临时数据存储器。内存包括至少两种存储器，例如内存既可以是随机存取存储器 RAM，也可以是只读存储器 ROM。实际应用中，计算节点 110 中可配置多个内存 113，以及不同类型的内存 113。本实施例不对内存 113 的数量和类型进行限定。此外，可对内存 113 进行配置使其具有保电功能。保电功能是指系统发生掉电又重新上电时，内存 113 中存储的数据也不会丢失。具有保电功能的内存被称为非易失性存储器。

网卡 114 用于与存储节点 100 通信。例如，当内存 113 中的数据总量达到一定阈值时，计算节点 110 可通过网卡 114 向存储节点 100 发送请求以对所述数据进行持久化存储。另外，计算节点 110 还可以包括总线，用于计算节点 110 内部各组件之间的通信。在功能上，由于图 1 中的计算节点 110 的主要功能是计算业务，在存储数据时可以利用远程存储器来实现持久化存储，因此它具有比常规服务器更少的本地存储器，从而实现了成本和空间的节省。但这并不代表计算节点 110 不能具有本地存储器，在实际实现中，计算节点 110 也可以内置少量的硬盘，或者外接少量硬盘。

任意一个计算节点 110 可通过网络访问存储节点集群中的任意一个存储节点 100。存储节点集群包

括多个存储节点100（图1中示出了三个存储节点100，但不限于三个存储节点100）。一个存储节点100包括一个或多个控制器101、网卡104与多个硬盘105。网卡104用于与计算节点110通信。硬盘105用于存储数据，可以是磁盘或者其他类型的存储介质，例如固态硬盘或者叠瓦式磁记录硬盘等。控制器101用于根据计算节点110发送的读/写数据请求，往硬盘105中写入数据或者从硬盘105中读取数据。在读写数据的过程中，控制器101需要将读/写数据请求中携带的地址转换为硬盘能够识别的地址。由此可见，控制器101也具有一些简单的计算功能。

需要说明的是，网卡114或网卡104为智能网卡时，处理器112的功能如重删等也可以卸载至智能网卡。智能网卡是指融合有计算资源的网卡，例如是具有数据处理单元（data processing unit，DPU）的网卡。DPU具有CPU的通用性和可编程性，但更具有专用性，可以在网络数据包，存储请求或分析请求上高效运行。通过将重删等功能卸载至DPU，一方面可以减少对CPU资源的占用，另一方面可以缩短访问路径。

图2所示的存储系统为存算分离的分布式存储系统，在一些可能的实现方式中，存储系统也可以为存算一体的分布式存储系统。存储一体的分布式存储系统包括存储集群（也称作存储节点集群），存储节点集群可以包括一个或多个服务器，服务器之间可以相互通信。服务器是一种既具有计算能力又具有存储能力的设备。在硬件上，服务器至少包括处理器、内存、网卡和硬盘。处理器用于处理来自服务器外部（应用服务器或者其他服务器）的数据访问请求，也用于处理服务器内部生成的请求。示例性的，处理器接收写请求时，会将这些写请求中的数据暂时保存在内存中。当内存中的数据总量达到一定阈值时，处理器将内存中存储的数据发送给硬盘进行持久化存储。处理器还用于对数据进行计算或处理，例如元数据管理、重复数据删除、数据压缩、数据校验、虚拟化存储空间以及地址转换等。

需要说明，以上仅仅是对存储系统的示例说明，在本申请实施例其他可能的实现方式中，存储系统还可以是全融合架构的分布式存储系统或者是memory fabric架构的分布式存储系统。在此不再赘述。

为了实现重删功能，存储系统通常需要引入“两级元数据映射”（Two-level Metadata Mapping）。

一般情况下，例如是在不支持重删功能的存储系统中，元数据映射可以为逻辑块地址（Logical Block Address，LBA）到物理块地址（Physical Block Address，PBA）的直接映射。其中，逻辑块地址也称作逻辑地址，物理块地址也称作物理地址。逻辑地址是存储介质呈现给主机的逻辑空间的地址。主机在向存储介质发送写请求或读请求时，会将逻辑地址携带在写请求或读请求中。存储介质接收到送写请求或读请求时，会获取写请求或读请求携带的逻辑地址，对逻辑地址经过一次或多次地址转换确定物理地址，向物理地址写入数据或者从物理地址读取数据。通过采用LBA作为数据的地址，将物理地址这种基于磁头、柱面和扇区的三维寻址方式转变为一维的线性寻址，可以提高寻址的效率。

逻辑地址到物理地址的映射为单跳映射，在引入重删功能后，由于逻辑地址和数据内容并不匹配，如果按照逻辑地址进行路由难以找到数据，因此需要增加一跳按指纹的路由，因此，单跳映射可以变更为逻辑地址到指纹，再由指纹到物理地址的两级映射。

在图1或图2所示的存储系统中，处理器（例如是图1中的处理器123或者图2中的处理器112）可以将数据重删方法对应的计算机可读指令加载进内存（例如是图1中的内存124或者图2中的内存113），然后处理器执行上述计算机可读指令，以执行数据重删方法，从而节省存储空间、提升存储性能。

为了便于描述，以存储系统对应用产生的数据进行后重删示例说明。

具体地，应用写入数据时，存储系统（例如是存储系统中的处理器）可以将数据分块，该示例中假设以4千字节（kilo byte，KB）的粒度将数据分为多个数据块，然后存储系统中的重删模块可以对每个4KB的数据块计算指纹，将指纹写入日志文件，接着将数据块下盘（写入硬盘等存储设备）。用户可以手动触发重删操作，或者设置重删周期，如此，应用可以响应于用户的重删操作下发重删命令，或者是周期性地向重删模块下发重删命令，重删模块响应于重删命令，对日志文件中的指纹进行排序，将排序后的指纹与指纹文件中的指纹合并，根据合并结果删除内容重复的数据块。

如图3所示，对于新写入硬盘等存储介质（也可以为存储设备）的数据块，该数据块的指纹等元数据可以采用追加写入日志文件（Write-ahead Log，WAL）的方式进行管理。当触发重删时，重删模块将日志文件中的指纹进行排序，并将排序后的指纹与指纹文件中的指纹合并，基于合并结果可以保留具有相同指纹的多个数据块中的一个数据块，删除内容重复的数据块，并更新指纹文件。需要说明，图3中

不同图案的矩形块代表不同指纹。

然而，很多应用的工作负载（workload）通常是非均匀的，也即应用写入硬盘等存储设备进行持久化存储的数据可以具有不同的更新频次。其中，更新频繁的数据的占比通常高于更新不频繁的数据的占比。更新频繁的数据通常难以被重删，也即难以被重删的数据占比反而高，此时就会出现资源挤占的情况，更新不频繁的数据所分配的空间占比就会较低，由此导致易被淘汰，从而损失重删率。

有鉴于此，本申请提供的数据重删方法可以对元数据管理结构进行主动分区，在将写请求中的数据块写入存储设备（如硬盘）后，将数据块的指纹以及地址信息等元数据写入元数据管理结构的多个分区中与该数据块的特征对应的分区，并在该分区中存在与该数据块的指纹相同的指纹时，删除该分区中该数据块的元数据，以及根据该数据块的地址信息从存储设备删除该数据块。

如此，不同特征的数据块的元数据可以写入元数据管理结构的不同分区，例如更新频繁的数据块的元数据可以写入容量较小的分区，更新不频繁的数据块的元数据可以写入容量较大的分区，由此避免更新不频繁的数据被更新频繁的数据挤占资源，进而被淘汰，提高了重删率。

进一步地，本申请引入了一种新的元数据管理结构，即用于存储数据块的指纹到逻辑地址的映射关系的逆映射表（Inverse Mapping Table）。如图 4 所示，区别于基于日志文件的元数据管理，本申请实施例通过逆映射表对指纹、逻辑地址等元数据进行管理，当逆映射表的分区中与当前写入的数据块的指纹相同的指纹的数量达到预设阈值，即触发重删，无需等待用户手动触发或者周期性地触发，能够及时地删除内存中重复的指纹等元数据，减少了元数据内存开销，保障了系统性能。此外，当前写入数据块的逻辑地址和物理地址可以写入地址映射表，被重删的数据块的物理地址可以修改为指纹。如此，被重删的数据块可以通过两级映射进行寻址，未被重删的数据块可以通过单级映射进行寻址，缩短了未被重删的数据块的响应时间。

需要说明，重删后的逆映射表中的指纹和逻辑地址还可以写入指纹表，如此指纹表中可以存储重删后的数据块的指纹和逻辑地址。存储系统也可以通过指纹表、前向映射表进行寻址。基于此，逆映射表的各个分区还可以在满足淘汰条件时，对分区中的元数据进行淘汰，从而降低元数据的规模，减小内存开销。

为了使得本申请的技术方案更加清楚、易于理解，下面将以图 1 所示的存储系统 120 为例，对本申请提供的数据重删方法进行介绍。

参见图 5 所示的数据重删方法的流程图，存储系统 120 包括引擎 121 和硬盘框 130，引擎 121 中包括互为备份的控制器 0 和控制器 1，为了便于描述，图 5 从控制器 0 的角度进行示例说明，硬盘框 130 中包括多个硬盘 134，该方法包括如下步骤：

S502：控制器 0 接收来自于应用服务器 100 的写请求。

写请求是指用于写数据的请求。写请求中包括数据，该写请求即用于将该数据写入硬盘 134，进行持久化存储。其中，写请求可以由部署于应用服务器 100 上的应用基于业务需求生成。例如，应用服务器 100 上可以部署视频应用，视频应用可以为短视频应用或长视频应用，该视频应用可以生成写请求，其中，写请求中包括用户上传的视频流。又例如，应用服务器 100 上可以部署文件管理应用，该文件管理应用可以是文件管理器，文件管理器可以生成写请求，写请求中包括待归档的图像。

在图 1 的示例中，控制器 0 包括处理器 123 和前端接口 125，控制器 0 的处理器 123 可以通过前端接口 125，接收应用服务器 100 通过交换机 110 转发的写请求。

S504：控制器 0 将写请求中数据分块，获得至少一个数据块。

在本实施例中，控制器 0 可以采用定长分块或变长分块，对写请求中数据进行分块，从而获得至少一个数据块。其中，定长分块是指按照设置好的分块粒度对数据流进行分块。变长分块是将数据流分为大小不固定的数据块，变长分块可以包括基于滑动窗口的变长分块和基于内容的变长分块（content-defined chunking，CDC）。

为了便于理解，下面以定长分块进行示例说明。具体地，数据流的大小为分块粒度的整数倍时，控制器 0 可以将数据均匀地切分为一个或多个数据块。数据大小并非分块粒度的整数倍时，控制器 0 可以将数据进行填充，例如是在数据的末端填零，使得填充后的数据为分块粒度的整数倍，接着控制器 0 按

照该分块粒度将数据均匀地切分为一个或多个数据块。例如，数据的大小为 19KB 时，控制器 0 可以在该数据的末端填零，使得填充后的数据的大小为 20KB，然后控制器 0 按照 4KB 的分块粒度进行分块，可以获得 5 个大小为 4KB 的数据块。

考虑到不同存储场景的输入输出（input output，IO）模式、IO 大小、特性要求不同，控制器 0 可以根据存储场景选择合适的分块策略进行分块。例如，主存储场景中，IO 通常较小，并且 IO 模式以随机读写为主，控制器 0 可以选择采用定长分块；备份存储场景中，IO 通常较大，IO 模式以顺序读写为主，控制器 0 可以选择变长分块，以获得较好的重删率。

需要说明的是，对数据进行分块是本申请实施例中数据重删方法的可选步骤，执行本申请实施例的数据重删方法也可以不执行上述步骤。例如，数据的大小等于分块粒度，或者小于分块粒度时，可以直接将数据作为一个数据块。又例如，数据的大小是固定大小时，也可以直接将数据作为一个数据块。

S506：控制器 0 确定至少一个数据块的指纹。

针对至少一个数据块中的任意数据块，控制器 0 可以根据该数据块的内容，通过消息摘要算法进行计算，获得该数据块的指纹。数据块的指纹可以是该数据块的消息摘要，例如是数据块的哈希值。

S508：控制器 0 根据至少一个数据块中第一数据块的指纹查询指纹表。当第一数据块的指纹在指纹表中存在时，执行 S509，当第一数据块的指纹在指纹表中不存在时执行 S510、S511、S512。

S509：控制器 0 向应用服务器 100 返回写响应。

S510：控制器 0 将第一数据块写入硬盘 134。

指纹表用于记录硬盘 134 存储的数据块的指纹和地址信息。其中，地址信息可以包括逻辑地址。进一步地，地址信息还可以包括物理地址。指纹表可以采用键值对（key value，kv）方式存储指纹和地址信息。具体地，指纹表可以以指纹为 key，以逻辑地址等地址信息为 value 进行存储。

指纹表中记录的指纹和地址信息可以来自逆映射表。逆映射表是一种存储已下盘的数据块的指纹和逻辑地址的元数据管理结构。具体地，逆映射表触发重删后，控制器 0 可以将重删后的逆映射表同步至指纹表，具体是将重删后的逆映射表中的元数据（例如是指纹和逻辑地址）存储至指纹表。

该指纹表中存储有已下盘的数据块的指纹，因此可以支持前重删，从而减少硬盘 134 的存储压力。具体地，控制器 0 可以根据第一数据块的指纹查询指纹表。例如，控制器 0 可以将第一数据块的指纹与指纹表中的指纹进行比对，或者控制器 0 可以根据第一数据块的指纹以及指纹表的索引快速查找指纹。

当第一数据块的指纹在指纹表中存在时，表明具有相同内容的数据块已写入硬盘 134，控制器 0 可以执行 S509，以直接返回写响应，该写响应用于表征写成功。当第一数据块的指纹在指纹表中不存在时，表明磁盘 134 中并未存储相同内容的数据块，控制器 0 可以执行 S510，将第一数据块写入磁盘 134。进一步地，控制器 0 也可以在第一数据块写入硬盘 134 成功后，向应用服务器 100 返回写响应。

需要说明的是，在业务的初始阶段，指纹表可以为空。随着应用服务器 100 不断向硬盘 134 存储数据，逆映射表中可以记录已下盘的数据块的元数据，当逆映射表中的分区触发重删，重删后的逆映射表中的元数据可以同步至指纹表。在该阶段，控制器 0 可以查询指纹表，从而实现前重删。与后重删相比，前重删通过在数据落盘之前删除重复数据，无需将重复数据写入硬盘 134 等存储介质，避免了对资源的占用。

基于此，执行本申请实施例的方法也可以不执行上述 S508、S509。例如，在业务的初始节点，指纹表为空，控制器 0 可以直接将数据块下盘进行后重删，并在指纹表中记录的元数据达到预设条数，支持前重删。又例如，控制器 0 可以不对数据块进行前重删，直接将数据块下盘进行后重删。

S511：控制器 0 向地址映射表中写入第一数据块的逻辑地址和物理地址。

地址映射表用于存储写入硬盘 134 的数据块的地址信息。例如，地址映射表可以存储数据块的逻辑地址和物理地址。具体实现时，地址映射表可以采用 kv 方式存储逻辑地址和物理地址。为了便于查找或定位数据块，地址映射表可以以逻辑地址为 key，以物理地址为 value，存储逻辑地址到物理地址的映射关系。一方面可以便于后续访问该数据块时快速寻址，另一方面可以记录该操作，便于后续追溯或故障恢复。

区别于逆映射表以逻辑地址为 value，地址映射表以逻辑地址为 key，因此也可以称之为前向映射表。需要说明的是，如果第一数据块被前重删，也就表明第一数据块并未被写入硬盘 134，也就不存在相应

的物理地址，控制器 0 可以在前向映射表中存储第一数据块的逻辑地址和指纹。当需要查找第一数据块时，可以根据前向映射表查找该第一数据块的指纹，然后查找指纹表，获得与该第一数据块具有相同指纹的数据块，获得具有相同指纹的数据块的逻辑地址，基于具有相同指纹的数据块的逻辑地址可以通过前向映射表，获得具有相同指纹的数据块的物理地址，基于具有相同指纹的数据块的物理地址可以访问具有相同指纹的数据块，由此可以实现访问第一数据块。

需要说明的是，上述 S510、S511 的执行顺序并不限定。在一些可能的实现方式中，控制器 0 也可以先在前向映射表中写入逻辑地址和物理地址，然后再向硬盘 134 等存储设备中写入第一数据块。

S512：控制器 0 将第一数据块的指纹和逻辑地址插入逆映射表的第一分区。在第一分区中存在与第一数据块的指纹相同的指纹时，执行 S514、S516。

逆映射表用于存储指纹和逻辑地址。其中，逆映射表可以是以键值对形式组织的表结构。逆映射表中的键值对用于表征逆映射关系。区别于地址映射表中自逻辑地址映射到物理地址的映射形式，逆映射表中用于存储指纹到逻辑地址的逆映射关系。其中，键值对中的 key 为指纹，value 为逻辑地址。控制器 0 可以将第一数据块的指纹和逻辑地址有序插入逆映射表。

其中，控制器 0 可以将第一数据块的指纹和逆映射表中第一分区的指纹进行多路归并（N-Way Merge）排序，然后根据排序结果，将第一数据块的指纹和逻辑地址插入逆映射表的第一分区。

多路归并排序是指待排序的对象（例如是数据块的指纹）分为多路分别进行排序，然后将各路排序结果进行合并，从而实现归并排序。假设每路的排序结果可以记作具有 n 个元素的有序小集合 $S=\{x|x_i \le x_j, i,j \in [0,n)\}$，假设待排序的指纹被分为 m 路，则可以对 m 个小集合S_1、S_2、…S_m进行合并以实现归并排序。其中，控制器 0 可以按照如下公式确定所有小集合中的最小值，写入大集合：

$$\min = \min(\min(S_1), \min(S_2), \ldots \min(S_m)) \quad (1)$$

假设所有小集合中的最小值为集合S_i中最小值，控制器 0 可以在将该最小值写入大集合后，将该最小值从集合S_i去除，然后继续按照上述公式（1）计算，确定更新后的所有小集合的最小值，并将最小值写入大集合。控制器 0 重复上述过程，直至所有小集合中的元素均写入大集合，完成归并排序。

在一些可能的实现方式中，控制器 0 可以采用日志结构合并树（log structured merge tree，LSM tree）将第一数据块的指纹和逆映射表中的指纹进行排序，进而将第一数据块的指纹和逻辑地址有序插入逆映射表的第一分区。其中，逆映射表的各个分区均可以维护一个 LSM tree，从而将待写入该分区的指纹和逻辑地址等元数据有序插入。

在本实施例中，逆映射表包括多个分区，第一分区可以根据第一数据块的特征确定。其中，第一数据块的特征可以是第一数据块对应的指纹。在一些情况下，多个数据块可以对应相同指纹。例如，在备份场景中，多个数据块可以对应相同指纹。控制器 0 可以根据第一数据块对应的指纹，确定该指纹的热度，然后根据第一数据块对应的指纹的热度确定该热度对应的逆映射表的多个分区中的第一分区。例如，控制器 0 可以将指纹的热度与预设热度进行比较，当指纹的热度小于预设热度，则可以确定第一分区为逆映射表中用于存储冷数据的元数据的分区，也称作冷分区，当指纹的热度大于或等于预设热度，则可以确定第一分区为用于存储热数据的分区，也称作热分区。其中，冷分区可以是容量较大的分区，热分区可以是容量较小的分区。冷分区的容量大于热分区的容量。接着控制器 0 将第一数据块的指纹和逻辑地址写入第一分区。

进一步地，在将第一数据块的指纹和逻辑地址等元数据写入第一分区后，控制器 0 还可以更新第一数据块对应的指纹的热度，以用于确定后续写入的具有相同指纹的数据块所对应的分区。具体地，控制器 0 可以确定第一数据块的逻辑地址的热度，将逻辑地址的热度累加至第一数据块对应的指纹的热度，从而更新第一数据块对应的指纹的热度。

为了便于理解，下面结合一示例进行说明。该示例中，控制器 0 可以在每次写入数据块的元数据后，更新数据块对应的指纹的热度。例如，控制器 0 写入硬盘 134 的第一数据块为数据块 10，数据块 10 的指纹记作 FP3，由于之前曾写入具有相同指纹的数据块，并且最近一次写入指纹为 FP3 的数据块为数据块 8，控制器 0 可以获取写入数据块 8 的元数据后所更新的 FP3 的热度。该示例中假设 FP3 的热度为 5。控制器 0 可以基于该热度确定逆映射表的多个分区中的第一分区，例如第一分区可以是用于存储冷数据

的元数据的分区，也称作冷分区。控制器 0 还可以确定数据块 10 的逻辑地址的热度，假设逻辑地址的热度为 2，则可以将逻辑地址的热度累计至指纹的热度，从而更新指纹的热度。在该示例中，更新后的指纹的热度可以为 7。

随着数据块的不断写入，数据块对应的指纹的热度可以发生变化。为此，控制器 0 还可以根据数据块对应的指纹的热度，将数据块的元数据在不同分区移动。例如，在初始阶段，各数据块对应的指纹的热度通常较低，可以将这些数据块的元数据写入冷分区。随着数据块的不断写入，部分指纹的热度不断增加，当在写入某个数据块后，该数据块对应的指纹的热度大于预设热度时，控制器 0 可以将该数据块的元数据写入热分区，以及将具有相同指纹的数据块的元数据移动至热分区。

在一些可能的实现方式中，考虑到在不同分区移动元数据的开销，控制器 0 也可以不移动元数据，而是将元数据淘汰。例如，当在写入某个数据块后，该数据块对应的指纹的热度大于预设热度时，控制器 0 可以将该数据块的元数据写入热分区，以及将具有相同指纹的数据块的元数据淘汰出冷分区。

进一步地，控制器可以在第一分区存在与第一数据块的指纹相同的指纹时，即触发重删，也可以在第一分区中存在与第一数据块的指纹相同的指纹，且第一分区中与第一数据块的指纹相同的指纹的数量达到预设阈值时，触发重删，执行 S514、S516。

其中，预设阈值可以根据经验值进行设置。例如，预设阈值可以设置为 2，以图 4 示例说明，该示例中，逆映射表的分区 1（如第一分区）中与第一数据块的指纹相同的指纹的数量达到 2，则可以触发重删。又例如，预设阈值也可以设置为 1，也即存在与第一数据块的指纹相同的指纹，即可触发重删。也就是说，当第一分区中存在与第一数据块的指纹相同的指纹时，控制器 0 可以执行 S514、S516 实现重删，但本申请实施例对重删时机不作限定，例如，控制器 0 可以在第一分区中相同的指纹的数量达到预设阈值时触发重删，也可以在存在相同的指纹时即触发重删。

在本实施例中，控制器 0 通过直接将第一数据块写入硬盘 134，然后执行后重删。如此可以避免消耗计算资源影响业务正常运行，以及出现计算瓶颈时导致存储性能下降。

需要说明的是，逆映射表为本实施例引入的一种新型元数据管理结构，在本申请实施例其他可能的实现方式中，元数据管理结构也可以采用其他组织形式。

S514：控制器 0 根据第一数据块的地址信息从硬盘 134 中删除第一数据块。

对于具有相同指纹的数据块，控制器 0 可以保留一个数据块，从硬盘 134 中删除其他具有相同指纹的数据块。具体实现时，控制器 0 可以根据第一数据块的逻辑地址，从地址映射表中获取第一数据块的物理地址，然后根据该物理地址从硬盘 134 等存储设备中找到该第一数据块，并删除第一数据块。

进一步地，预设阈值大于 1 时，控制器 0 还可以保留一个与第一数据块指纹相同的数据块，删除其他指纹相同的数据块。具体地，针对指纹相同的数据块，控制器 0 可以保留最先写入的数据块，删除后写入的数据块。

S516：控制器 0 从逆映射表的第一分区中删除第一数据块的指纹以及逻辑地址。

具体地，控制器 0 可以从逆映射表中查找第一数据块的指纹，然后删除第一数据块的指纹以及逻辑地址。需要说明的是，当逆映射表中指纹与逻辑地址的键值对采用 LSM tree 的方式存储时，控制器 0 可以采用表格合并的方式删除第一数据块的指纹以及对应的逻辑地址。

进一步地，预设阈值大于 1 时，控制器 0 还可以保留一个与第一数据块指纹相同的数据块的元数据，删除其他指纹相同的数据块的元数据。具体地，针对指纹相同的数据块，控制器 0 可以保留最先写入的数据块的元数据，删除后写入的数据块的元数据。

此外，控制器 0 在执行上述 S514、516 时可以并行执行，也可以按照设定的顺序先后执行。例如，控制器 0 也可以先执行 S516，然后执行 S514。本实施例对 S514 和 S516 的执行顺序不作限定。

需要说明的是，至少一个数据块还可以包括第二数据块。其中，第二数据块对应的指纹的热度可以高于第一数据块对应的指纹的热度，例如第二数据块可以为热数据，第一数据块可以为冷数据。相应地，控制器 0 可以将第二数据块写入硬盘 134，将第二数据块的元数据，如第二数据块的指纹和逻辑地址，写入逆映射表的多个分区中的第二分区。第二分区具体是用于存储热数据的元数据的热分区。第二分区的容量小于第一分区的容量。

进一步地，数据块对应的指纹的热度还可以分为更多的类型或级别，例如指纹的热度也可以分为热、

温、冷三个级别。相应地，逆映射表可以包括更多的分区，例如逆映射表可以包括用于存储热数据的元数据的分区、用于存储温数据的元数据的分区和用于存储冷数据的元数据的分区。

上述 S516 为本申请实施例中删除第一分区中第一数据块的元数据的一种实现方式，当元数据管理结构用于存储指纹、逻辑地址和物理地址时，控制器 0 可以删除元数据管理结构的第一分区中第一数据块的指纹、逻辑地址和物理地址。

S518：控制器 0 将重删后的逆映射表中的指纹和逻辑地址写入指纹表。

具体地，控制器 0 还可以将重删后的逆映射表中的指纹和逻辑地址等元数据，以同步方式写入至指纹表。其中，控制器 0 可以以分区为粒度，将重删后的逆映射表中的元数据同步写入指纹表。例如，第一分区触发重删后，控制器 0 可以将第一分区重删后的元数据同步写入指纹表，第二分区触发重删后，控制器 0 再将第二分区重删后的元数据同步写入指纹表。考虑到分区可以触发多次重删，为了减少资源占用，控制器 0 可以采用增量同步机制，将重删后的元数据同步写入指纹表。

S520:当逆映射表中的至少一个分区满足淘汰条件,控制器0对至少一个分区中的元数据进行淘汰。

具体地，逆映射表中的分区可以设置用于淘汰的水位，当该分区中的元数据占用的资源量到达该水位，则控制器 0 可以对该分区中的元数据进行淘汰，从而避免元数据溢出。其中，由于不同分区存储有不同热度的指纹，分区的水位可以是不同的。例如，第一分区的容量为逆映射表的总容量的 80%，第二分区的容量为逆映射表的总容量的 20%时，第一分区的水位可以是逆映射表的总容量的 70%，第一分区的水位可以是逆映射表的总容量的 10%。

在一些可能的实现方式中，逆映射表中的分区的水位可以包括高水位和低水位。当分区中的元数据占用的资源量到达高水位时，可以对分区中的元数据进行淘汰，使得淘汰后的分区中元数据占用的资源量，不低于上述低水位，且不高于高水位。

控制器 0 对逆映射表的分区中的元数据进行淘汰，尤其是对更新频繁的热数据对应的分区中的元数据进行淘汰，可以大幅降低元数据规模，减少内存开销。

S522：控制器 0 将地址映射表中被重删的第一数据块的物理地址修改为第一数据块的指纹。

由于第一数据块已经从硬盘 134 中删除，基于地址映射表中的物理地址难以定位该第一数据块，因此，控制器 0 可以将地址映射表中被重删的第一数据块的物理地址修改为第一数据块的指纹，表征该第一数据块为被重删的数据块，可以通过指纹表查找到与该数据块具有相同指纹的数据块，进而确定具有相同指纹的数据块的物理地址，如此可以实现被重删的第一数据块的寻址。

基于此，未被重删的数据块仍能基于前向映射表中逻辑地址与物理地址之间的映射关系实现寻址，而不需要通过两级元数据映射，大幅缩短了寻址时间，进而缩短了响应时间，提高了响应效率。

需要说明的是，上述 S518 至 S522 为本申请实施例的可选步骤，例如内存为大容量内存时，控制器 0 也可以不执行将逆映射表中的数据写入指纹表，并对逆映射表中的元数据进行淘汰的步骤。

基于上述内容描述，本申请提供的数据重删方法通过对元数据管理结构进行主动分区，以差异化处理不同更新频次的数据的元数据。其中，更新不频繁的数据能够分配足够的配额资源以支持重删，而更新频繁的数据因为被频繁无效掉，因此分配的配额资源就相对较少；这样使得最终可以被重删的数据基本都可以被重删掉而不是被挤占淘汰，获得重删率的提升；同时难以重删的数据能够被淘汰，进而降低整体元数据映射规模，获得系统性能提升。

进一步地，该方法还引入逆映射表这一新型元数据管理结构，当逆映射表的第一分区中存在与第一数据块的指纹相同的指纹可以触发重删，无需用户手动重删，或者周期性地触发重删，如此可以实现及时进行重删，减少元数据规模，进而减少了元数据内存开销，保障了系统性能。而且，该方法在逆映射表中记录下盘的数据块的逻辑地址和物理地址，并将被重删的数据块的物理地址修改为指纹，如此可以使得未被重删的数据块仍能通过单跳映射寻址，缩短了寻址时间，提高了响应效率。

图 5 所示实施例的关键在于对逆映射表等元数据管理结构进行分区。其中，元数据管理结构中各个分区的容量可以根据分区决策模型确定。分区决策模型用于预测预设的分区容量组合中每个分区容量组合应用于元数据管理结构后对应的分区收益，并确定分区收益最大的分区容量组合作为所述元数据管理结构的多个分区的容量。

其中，分区容量组合表示一组分区中各个分区的容量。分区的容量是指分区所分配到的资源量。一组分区中各个分区的容量之和等于元数据管理结构的总容量。基于此，分区容量组合可以通过各分区的实际容量表征，也可以通过各分区的容量占比表征。

为了便于描述，下文均以分区容量组合通过分区容量占比示例说明。例如，一个分区容量组合可以表示为 80%：20%，用于表征该元数据管理结构包括两个分区，容量分别为总容量的 80%和 20%。又例如，一个分区容量组合可以表示为 60%：30%:10%，用于表征该元数据管理结构包括三个分区，容量分别为总容量的 60%、30%和 10%。

分区收益是指对元数据管理结构实施分区后所获得的收益。例如，在对元数据管理结构实施分区后可以获得重删率的提升，因此，分区收益可以为重删率。

分区决策模型可以通过预估数据的命中率，从而预测重删率。为了便于描述，以预设的分区容量组合中的第一分区容量组合示例说明。分区决策模型可以通过以下方式预测第一分区容量组合应用于所述元数据管理结构后对应的重删率：

获取第一分区容量组合应用于所述元数据管理结构所形成的多个分区中各个分区对应的工作负载特征，然后根据各个分区对应的工作负载特征，获得各个分区对应的数据分布，接着根据各个分区对应的数据分布以及各个分区的容量，获得重删率。

进一步地，考虑到工作负载可能发生变化，分区的容量还支持调整。工作负载可以为一段时间内正在使用或等待使用 CPU 等计算资源的任务。在业务高峰期，正在使用或等待使用 CPU 等计算资源的任务较多，工作负载较大，在业务低谷期，正在使用或等待使用 CPU 等计算资源的任务较少，工作负载较小。基于此，控制器 0 可以对分区进行调整。在具体工程实施过程中，分区调整需要重新进行分区初始化等操作，产生分区调整成本。基于此，分区收益可以根据收益率和分区调整成本中的至少一个确定。

在一些可能的实现方式中，控制器 0 可以周期性地调整元数据管理结构的多个分区的容量。其中，每个周期也可以称作分区调整周期。当到达调整时刻时，控制器 0 可以根据调整时刻前的周期（例如是上一分区调整周期）对应的分区收益、分区容量组合或各个分区对应的工作负载特征，确定是否调整多个分区的容量。其中，工作负载特征是指从工作负载信息中提取的特征，例如工作负载特征可以包括重用距离、重用周期、重用频次中的一种或多种。

为了便于理解，下面结合附图对本申请实施例的数据重删方法中分区过程进行详细介绍。为了便于描述，下面仍以对逆映射表分区进行示例说明。

参见图 6 所示的数据重删方法的流程图，在图 5 所示实施例基础上，控制器 0 在将第一数据块写入硬盘 134 之后还可以执行如下步骤：

S602：控制器 0 获取上一分区调整周期的系统资源使用信息和各个分区对应的工作负载信息。

系统资源包括计算资源（如 CPU 等处理器资源）、内存资源、磁盘资源、网络资源中的一种或多种。基于此，系统资源使用信息包括 CPU 占用比例、内存占用比例、磁盘 IO 量、以及带宽占用量。

工作负载（workload）是指一段时间内正在使用或等待使用 CPU 等计算资源的任务。该任务可以是写数据。基于此，工作负载信息可以包括数据的重用距离（Reuse Distance）、重用周期或重用频次中的一种或多种。重用距离可以为对同一数据的相邻两次访问之间所间隔的访问次数。在统计重用距离时，可以以数据块为粒度进行统计。重用周期可以为不同写请求中对同一数据的相邻两次访问之间所间隔的写请求个数。重用频次可以是重用周期的倒数。需要说明的是，控制器 0 在获取重用距离、重用周期或重用频次等工作负载信息时，可以针对各个分区分别提取重用距离、重用周期或重用频次，从而获得各个分区对应的工作负载信息。

S604：控制器 0 从系统资源使用信息中提取系统资源特征，从各个分区对应的工作负载信息中提取各个分区对应的工作负载特征。

具体地，控制器 0 可以对系统资源使用信息进行向量化，获得系统资源特征，以及对工作负载信息进行向量化，获得工作负载特征。为了便于理解，下文以对 CPU 占用比例、重用距离的向量化进行示例说明。

在对 CPU 占用比例进行向量化时，控制器 0 可以将 CPU 占用比例与 CPU 占用阈值进行比较。其中，CPU 占用阈值可以根据历史业务经验设置。例如，CPU 占用阈值可以设置为 70%。当 CPU 占用比

例高于 CPU 占用阈值，则可以输出“1”，当 CPU 占用比例不高于 CPU 占用阈值，则可以输出“0”，具体可以参见如下公式：

$$\mathrm{F}(\text{CPU 占用})=\begin{cases}1，CPU占用\geq 70\%\\0。CPU占用<70\%\end{cases} \tag{2}$$

其中，F 表示特征，F 可以采用向量表示。

类似地，参见图 7 所示的系统资源特征提取的示意图，控制器 0 可以将内存占用比例、磁盘 IO 量、带宽占用量等系统资源使用信息与该资源对应的阈值进行比较，根据比较结果输出上述系统资源使用信息对应的系统资源特征，例如为 F（内存占用）、F（磁盘 IO 量）、F（带宽占用量）。

在对重用距离进行向量化时，可以累计本批次数据（如本次写请求中数据包括的数据块）的重用距离，然后由此计算出重用距离的均值和方差，具体如下所示：

$$\begin{aligned}mean_{\text{重用距离}}&=\frac{1}{N}\sum_{i=0}^{N}d_{\text{重用距离}}\\variance_{\text{重用距离}}&=\frac{1}{N}\sum_{i=0}^{N}(d_{\text{重用距离}}-mean_{\text{重用距离}})^2\end{aligned} \tag{3}$$

类似地，控制器 0 可以采用和处理重用距离相同的统计方法，对重用周期、重用周期进行统计，以充分挖掘其中的时间关联性和空间关联性，从而实现从工作负载信息中提取工作负载特征。

需要说明的是，上述 S602、S604 为本申请实施例的可选实施方式，执行本申请实施例的方法也可以不执行上述 S602、S604。例如，控制器 0 也可以不获取系统资源使用信息，提取系统资源特征。

S606：控制器 0 根据系统资源特征和工作负载特征，获得结构化特征。

结构化特征包括系统资源特征和工作负载特征。该系统资源特征从系统资源使用信息提取得到，该工作负载特征从工作负载信息提取得到。控制器 0 可以对系统资源特征和工作负载特征进行融合，从而得到结构化特征。例如，控制器 0 可以将系统资源特征和工作负责特征进行拼接，从而实现融合得到结构化特征。

其中，控制器 0 还可以对系统资源特征进行归并。从业务角度分析，某些时刻某一系统资源出现占用过高，就会造成系统性能瓶颈，此时相同类型的系统资源特征（也可以称作关联特征、共性影响特征）已不再必要。对于此类特征，可执行特征归并。如图 8 所示，控制器 0 可以采用“或”运算即可实现特征归并。例如，F（CPU 占用）、F（内存占用）、F（磁盘 IO 量）、F（带宽占用量）属于共性影响特征，控制器 0 可以将上述共性影响特征归并。

如图 9 所示，控制器 0 通过多源信息处理，获得工作负载特征以及具有共性影响的系统资源特征。在归并共性影响特征后，控制器 0 可以对共性影响特征进行标准化、归一化，类似地，控制器 0 可以在提取到工作负载特征后，对工作负载特征进行标准化、归一化。其中，标准化、归一化后的系统资源特征可以为$a_0a_1\cdots a_k$，标准化、归一化后的工作负载特征可以为$b_0b_1\cdots b_k$。控制器 0 可以将上述系统资源特征$a_0a_1\cdots a_k$以及工作负载特征$b_0b_1\cdots b_k$进行拼接，从而获得结构化特征。

控制器 0 通过特征归并与归一化等通用特征处理手段，对特征进行清洗处理，可实现以较低的计算开销，生成对当前情况准确刻画的特征模型，为分区决策提供可靠依据。

上述 S602 至 S606 为控制器 0 获取结构化特征的一种实现方式，在本申请实施例其他可能的实现方式中，控制器 0 也可以通过其他方式获取结构化特征。进一步地，当控制器 0 不获取系统资源使用信息，并从中提取系统资源特征时，控制器 0 也可以不执行上述 S606。

S608：控制器 0 获取上一分区调整周期的反馈信息，根据上一分区调整周期的反馈信息确定是否触发分区调整。若是，则执行 S610；若否，则执行 S622。

控制器 0 可以设置分区调整的触发条件。控制器 0 可以根据上一分区调整周期的反馈信息，例如是上一分区调整周期的分区收益（如重删率）、各个分区对应的工作负载特征，判断分区调整的触发条件是否被满足，从而确定是否触发分区调整。

其中，针对当前分区调整周期，分区调整的触发条件可以设置为：上一分区调整周期的重删率小于预设值或者下降幅度达到预设幅度，或者上一分区调整周期的工作负载特征相对于上上分区调整周期的工作负载特征的变化满足预设条件。其中，预设值、预设幅度或预设条件可以根据历史的业务经验设置。

例如，上上分区调整周期的工作负载特征表示工作负载以大 IO 为主，上一分区调整周期的工作负载特征表征工作负载以小 IO 为主，也即上一分区调整周期的工作负载特征相对于上上分区调整周期的工作负载特征的变化较为显著，可以触发分区调整。

需要说明的是，执行本申请实施例的数据重删方法也可以不执行上述 S608。例如，控制器 0 可以直接触发分区调整，进而根据分区决策建模的结果进行分区更新。

S610：控制器 0 根据所述结构化特征从建模策略集合中确定目标建模策略。

S612：控制器 0 根据结构化特征和反馈信息从分区收益评估策略集合中选择目标评估策略。

S614：控制器 0 根据所述目标评估策略确定所述分区决策模型的目标函数。

S616：控制器 0 根据所述结构化特征，通过所述目标建模策略和所述目标函数进行分区决策建模，获得分区决策模型。

S618：控制器 0 根据分区决策模型获得分区收益最大的分区容量组合。

控制器 0 可以根据多源信息处理所得的结构化特征，通过分区决策建模，并在前期的反馈信息，例如是上一分区调整周期的工作负载特征、重删率等先验知识的辅助下完成分区决策。如图 10 所示，控制器 0 可以根据结构化特征，从建模策略集合中进行建模策略选择，以确定目标建模策略，根据上一分区调整周期的工作负载特征、重删率等反馈信息，从分区收益评估策略集合中进行评估策略选择，以确定目标评估策略，根据该目标评估策略可以确定分区决策模型的目标函数，接着控制器 0 可以基于结构化特征，通过目标建模策略、目标函数进行建模，进而根据建模得到的分区决策模型获得分区收益最大的分区容量组合。其中，分区收益最大的分区容量组合也可以称作分区决策。

为了使得本申请的技术方案更加清楚、易于理解，下面结合示例对建模策略选择、评估策略选择、分区决策建模等过程进行示例说明。

以工作负载较大、系统资源占用比例大的情况为例。

在该示例中，建模策略集合包括：①基于打点分区的建模策略；②基于高斯过程回归的建模策略。基于打点分区的建模策略面向一般场景，也即简单场景，基于高斯过程回归的建模策略面向复杂场景。其中，打点分区是指提供多种预设的分区容量组合，从中选择选择分区收益最大的分区容量组合。

在该示例中，结构化特征向量中工作负载特征反映出该业务场景为简单场景，此外，与 CPU 占用特征相关的标志位为 1，表征 CPU 等系统资源占用较高，为了避免分区决策建模占用较多的系统资源，控制器 0 可以选择基于打点分区的建模策略进行建模。

该示例中，假设工作负载的重用距离分布服从正态分布，如下所示：

$$f(d_{\text{重用距离}}) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(d_{\text{重用距离}}-\mu)^2}{2\sigma^2}) \quad (4)$$

其中，σ表示方差，μ表示均值。控制器 0 可以采用多源信息处理所得的结构化特征中重用距离的均值$\mu_{\text{重用距离均值}}$和重用距离的方差$\sigma_{\text{重用距离方差}}$拟合重用距离的概率密度函数，具体参见如下公式：

$$f(d_{\text{重用距离}}) = \frac{1}{\sqrt{2\pi}\sigma_{\text{重用距离方差}}}\exp(-\frac{(d_{\text{重用距离}}-\mu_{\text{重用距离均值}})^2}{2\sigma^2}) \quad (5)$$

该示例以提供两个分区进行示例说明。两个分区具体为更新频繁的数据对应的热分区和更新不频繁的数据对应的冷分区，控制器 0 可以根据基于打点分区的建模策略，设置如下分区组合：

表 1 分区组合

热分区占比	10%	20%	30%	40%	50%	60%	70%	80%	90%
冷分区占比	90%	80%	70%	60%	50%	40%	30%	20%	10%

结合上面公式（5）的重用距离概率密度函数，控制器 0 可通过积分的方式获得两种类型数据分布的命中率，再通过命中率和数据占比的乘积，获得该种分区方案所获得的重删率，如下所示：

$$f(P) = P * \int F1 + (1-P) * \int F2 \quad (6)$$

其中，$F1$和$F2$分别表示两个分区对应的数据分布，具体可以通过两个分区对应的数据的重用距离的概率密度函数表示，P为一个分区的容量占比。

在该示例中，控制器 0 可以根据结构化特征，从分区收益评估策略集合中选择“重删率最大化”的

评估策略，作为目标评估策略。也就是，控制器 0 可以直接以上述公式（6）的函数为目标函数，进行分区决策建模。具体地，控制器 0 将不同分区组合的参数代入上述目标函数，获得不同分区组合的重删率，控制器 0 从中选择重删率最大的分区容量组合。

上述示例是直接基于结构化特征进行评估策略选择示例说明。在一些可能的实现方式中，控制器 0 还可以结合其他约束条件对分区收益的评估策略进行重构。例如，在具体工程实施过程中，前后两次分区大小变化幅度不应过大，否则将会影响分区部署性能，造成不必要的开销。基于此，其他约束条件可以包括最小化分区调整成本。重构的分区收益的评估策略可以是基于重删率和分区调整成本的评估策略。

如图 11 所示，控制器 0 可以获取上一分区调整周期的反馈信息，该反馈信息可以包括工作负载特征、重删率中的一种或多种，控制器 0 可以根据该反馈信息调整分区收益的评估策略。具体地，上一分区调整周期的重删率小于预设值或者下降幅度达到预设幅度，控制器 0 可以将分区收益的评估策略调整为基于重删率和分区调整成本的评估策略。类似地，上一分区调整周期的工作负载特征相对于上上分区调整周期的工作负载特征的变化满足预设条件，控制器 0 可以将分区收益的评估策略调整为基于重删率和分区调整成本的评估策略。

与基于重删率的评估策略相比，基于重删率和分区调整成本的评估策略考虑了多种因素，更具有针对性，由此选出的分区容量组合也具有更高的分区收益。

分区收益的评估策略调整为基于重删率和分区调整成本的评估策略时，目标函数可以由上述公式（6）调整为：

$$f(P) = P * \int F1 + (1-P) * \int F2 - \left\| P - P_{\text{当前分区比例}} \right\|^2 \quad (7)$$

其中，上述 S610 至 S616 为控制器 0 根据上一分区调整周期的工作负载特征，构建所述分区决策模型的一种实现方式。其中，S618 为确定分区决策（具体为目标分区组合）的一种实现方式。需要说明的是，控制器 0 在建模分区决策模型时，也可以不执行上述 S612、S614，例如，控制器 0 可以基于目标建模策略以及默认的目标函数，进行分区决策建模，获得分区决策模型。

S620：控制器 0 根据分区收益最大的分区容量组合对逆映射表进行分区。

分区容量组合包括不同分区的容量占比，也即不同分区所分配到的资源占比。例如，分区容量组合可以包括热分区所分配到的资源占比和冷分区所分配到的资源占比。控制器 0 可以根据热分区所分配到的资源占比和冷分区所分配到的资源占比，对逆映射表的系统资源进行分区。

例如，分区容量组合为热分区所分配到的资源占比为 20%，冷分区所分配到的资源占比为 80%，控制器 0 可以将逆映射表中 20%的存储空间分配至热分区，逆映射表中 80%的存储空间分配至冷分区。

S620：控制器 0 将第一数据块的指纹和逻辑地址写入第一分区。

其中，第一分区是根据第一数据块的特征从多个分区中确定的分区，例如第一数据块对应的指纹具有较高热度时，第一分区可以是热分区，第一数据块对应的指纹具有较低热度时，第一分区可以是冷分区。

控制器 0 将第一数据块的指纹和逻辑地址写入第一分区的具体实现可以参见图 5 所示实施例相关内容描述，在此不再赘述。

S622：控制器 0 确定是否触发重删。若是，则执行 S624，若否，则执行 S626。

参见图 5 所示实施例相关内容描述，控制器 0 可以比较第一数据块的指纹与逆映射表的第一分区中的指纹，当第一分区中存在与第一数据块的指纹相同的指纹，可以触发重删。例如，第一分区中与第一数据块的指纹相同的指纹的数量达到预设阈值时，可以执行 S624，以进行重删。

S624：控制器 0 基于 LSM tree 进行重删。

控制器 0 可以通过合并 LSM tree 的方式对第一分区中的指纹和逻辑地址进行重删，并根据逻辑地址对应的物理地址从硬盘 134 中重删相应的数据块，例如控制器 0 可以通过合并 LSM tree 删除第一分区中第一数据块的指纹和逻辑地址，并根据第一数据块的逻辑地址对应的物理地址，从硬盘 134 删除第一数据块。基于 LSM tree 进行重删的具体实现可以参见图 5 所示实施例中 S514 至 S516 相关内容描述，在此不再赘述。

S626：控制器 0 对逆映射表的至少一个分区中的指纹和逻辑地址进行淘汰。

逆映射表的各个分区可以设置相应的淘汰条件，当逆映射表中的分区满足对应的淘汰条件，则可以对该分区中的指纹和逻辑地址等元数据进行淘汰。具体地，控制器 0 可以在各分区内部，基于分区容量以及数据块对应的指纹的热度，决策需要淘汰的元数据，例如是非重删元数据的指纹和逻辑地址，部分重删元数据的指纹和逻辑地址，然后将其淘汰出逆映射表，如此可以降低元数据规模，保障系统性能。

S628：控制器 0 将重删后的指纹和逻辑地址存放于指纹表。

S630：控制器 0 获取当前分区调整周期的反馈信息，以用于控制器 0 判断下一分区调整周期是否触发分区调整。

在本实施例中，控制器 0 可以获取上述反馈信息，从而协助进行分区决策调整，提高分区精度。

需要说明的是，上述 S628 至 S630 为本申请实施例的可选步骤，执行本申请实施例的数据重删方法也可以不执行上述步骤。

基于上述内容描述，本申请实施例的数据重删方法，提供一种主动分区机制，将隐式分区变为主动分区，例如将逆映射表分为热分区、冷分区，热分区以及冷分区被分配相应配额的系统资源，其中，更新不频繁的数据能够分配足够的存储空间以支持重删，而更新频繁的数据因为被频繁无效掉，因此分能够配的配额资源就相对较少；这样使得最终可以被重删的数据基本都可以被重删掉而不是被挤占淘汰，获得重删率的提升；同时难以重删的数据能够被淘汰，进而降低整体元数据映射规模，获得系统性能提升。

图 6 主要以简单场景中，控制器 0 采用基于打点分区的建模策略进行建模示例说明。在复杂场景中，控制器 0 也可以采用基于高斯过程回归的建模策略进行建模。下面对复杂场景下的建模过程进行说明。

在机器学习中，机器学习算法通常情况下是根据输入值x预测出一个最佳输出值y，用于分类或回归任务。这种情况将y看作普通的变量。某些情况下，任务并不需要预测出一个函数值，而是给出这个函数值的后验概率分布，记作$p_{(y|x)}$。此时，函数值y可以视作随机变量。高斯过程回归（Gaussian Process Regression，GPR）即是对表达式未知的函数（也称黑盒函数）的一组函数值进行贝叶斯建模，给出函数值的概率分布。

在利用高斯过程回归建模分区策略模型时，可以将问题描述为在分区配置资源有限的前提下，通过合理分配各分区配置的资源占比，进而获得最大重删率。为了更好的描述问题，本实施例采用分区容量组合 S 来描述各个分区的资源占比，S 为一个 n 维数组，第 i 个元素 Si 表示第 i 个分区的资源占比，在其他因素不变的前提下，重删率可以被视为由 S 决定，此时可以定义重删率与分区容量组合之间的关系为 f(S)。由于 f(S)不可以显式获得，本实施例采用高斯过程回归的建模策略来刻画 f(S)。

基于高斯过程回归的建模策略来建模 f(S)，可以包括如下阶段：

初始化阶段：控制器 0 随机生成若干个分区容量组合并添加进集合 Set。然后控制器 0 分别将上述分区容量组合应用于存储系统，通过存储系统运行得到各分区容量组合配置下的重删率。通过上述分区容量组合与重删率间的对应关系，初步建立分区容量组合与重删率之间的高斯模型 G 来刻画 f(S)。

迭代更新阶段：高斯模型 G 推荐出一个分区容量组合，将该分区容量组合添加进集合 Set。控制器 0 将该分区容量组合应用到存储系统，通过存储系统运行得到该配置下的重删率。控制器 0 将该分区容量组合及其对应的重删率反馈给高斯模型进行模型更新，重复执行迭代更新步骤 L 次（L 为预先设定迭代次数）。

输出阶段：输出集合 Set 中对应重删率最高的一组分区资源配置。

如此，可以实现复杂场景下基于高斯过程回归的建模策略进行建模，并基于上述高斯模型提供重删率最高的一组分区资源配置，即目标分区组合。

为了使得本申请的技术方案更加清楚、易于理解，下面以本申请实施例的数据重删方法应用于具有全局缓存（global cache）的存储系统进行示例说明。

参见图 12 所示的数据重删方法的应用场景示意图，针对存算分离的分布式存储系统，至少一台主机上运行的客户端连接多个存储节点。存储节点如 Node1、Node2 可以启动重删服务进程，以执行数据重删方法。其中，存储节点还可以拉起 LUN 服务进程，以协助重删服务进程执行数据重删方法。

具体地，客户端可以发送写请求，写请求中包括数据，该数据可以被分为至少一个数据。对于数据块，存储节点可以先在前向映射表中记录数据位置信息后即进行数据落盘操作。然后，存储节点如Node1上的重删服务进程可以执行数据重删方法对磁盘等存储介质中的冗余数据进行识别和重删，从而显著提高用户可用容量，降低用户成本。

在数据落盘的基础上，存储节点采用异步写入的方式，将数据块的指纹和逻辑地址有序插入逆映射表。该逆映射表为本申请引入的指纹元数据管理结构。对于不同更新频度的数据，存储节点采用分别构建LSM Tree的数据结构进行有效管理，该结构能够以表格合并的方式触发重删。在完成重删后，存储节点的重删服务进程将逆映射表中重删后的元数据发送至指纹表中，并相应更新前向映射表（也即LUN地址映射表）中重删数据的物理地址信息为指纹。

接着参见图13所示的数据重删方法应用于具有全局缓存的存储系统的流程示意图，全局缓存的上层具有写缓存和读缓存。其中，写缓存可以用于LBA的热度统计，无需额外设计统计模块进行热度统计。

如图13所示，客户端可以通过网络发送写请求，该写请求经过全局缓存的服务端适配层到达写缓存，重删服务进程可以将写请求中数据进行分块，然后计算各个数据块的指纹。接着，重删服务进程查询指纹表。若数据块的指纹在指纹表中命中，则表明存储设备中存储具有相同指纹的数据块，重删服务进程可以直接返回写响应。若数据块的指纹在指纹表中未命中，则表明存储设备中未存储具有相同指纹的数据块，重删服务进程可以将数据块写入硬盘等存储设备，然后返回写响应。

重删服务进程还可以将数据块的指纹与逻辑地址有序插入逆映射表。此外，重删服务进程还可以通过批量获取的方式，从写缓存获取LBA的热度，根据LBA的热度更新数据块对应的指纹的热度。重删服务进程还可以获取重用距离、重用周期等工作负载特征，以及获取系统资源使用信息，并根据系统资源使用信息获取系统资源特征，重删服务进程基于上述特征进行通用特征处理，获得结构化特征。

重删服务进程根据上述结构化特征，进行分区决策建模，进而根据建模得到的分区决策模型确定分区收益最大的分区容量组合。重删服务进程基于该分区容量组合对逆映射表的系统资源进行分区，具体是按照热分区所分配到的资源占比和冷分区所分配到的资源占比，对逆映射表的系统资源进行分区。重删服务进程在不同分区分别构建LSM Tree的数据结构，以实现对指纹、逻辑地址等元数据进行有效管理。例如，重删服务进程可以根据LSM tree通过表格合并的方式触发重删。重删服务进程还可以根据分区容量和数据块对应的指纹的热度，决策需要淘汰的元数据，并将其从逆映射表的LSM tree中淘汰，由此缩减元数据规模。

在该示例中，重删服务进程还可以在完成重删后，将逆映射表中重删后的元数据发送至指纹表中，并通过LUN服务进程相应更新地址映射表中重删数据对应的物理地址为指纹。进一步地，重删服务进程还可以获取反馈信息，以便基于该反馈信息，在后续阶段如下一分区调整周期确定是否触发分区调整，由此可以实现更精确地分区，进而实现重删率的提升。

基于本申请实施例提供的数据重删方法，本申请实施例还提供了一种数据重删装置。接下来，从功能模块化的角度，结合附图对本申请实施例的数据重删装置进行介绍。

参见图14所示的数据重删装置的结构示意图，该装置1400包括：

通信模块1402，用于接收写请求，所述写请求中包括第一数据块；

写数据模块1404，用于将所述第一数据块写入存储设备；

所述写数据模块1404，还用于将所述第一数据块的元数据写入元数据管理结构的多个分区中的第一分区，所述第一分区为根据所述第一数据块的特征确定的，所述第一数据块的元数据包括所述第一数据块的指纹及地址信息；

重删模块1406，用于在所述第一分区中存在与所述第一数据块的指纹相同的指纹时，删除所述第一分区中所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

其中，通信模块1402可以用于实现图5所示实施例中S502相关内容描述。写数据模块1404用于实现图5所示实施例中S510相关内容描述，写数据模块1404还用于实现图5所示实施例中S512相关

内容描述。重删模块 1406 用于实现图 5 所示实施例中 S514、S516 相关内容描述。

在一些可能的实现方式中，所述第一数据块的特征为所述第一数据块对应的指纹，所述写数据模块 1404 具体用于：

确定所述第一数据块对应的指纹的热度；

根据所述第一数据块对应的指纹的热度确定所述热度对应的所述元数据管理结构的多个分区中的所述第一分区；

将所述第一数据块的元数据写入所述第一分区。

其中，写数据模块 1404 确定第一数据块对应的指纹的热度，基于该热度确定第一分区，并将元数据写入第一分区的实现可以参见图 5 所示实施例中 S506、S512 相关内容描述，在此不再赘述。

在一些可能的实现方式中，所述写数据模块 1404 还用于：

在将所述第一数据块的元数据写入所述第一分区之后，确定所述第一数据块的逻辑地址的热度；

将所述逻辑地址的热度累加至所述第一数据块对应的指纹的热度，以更新所述第一数据块对应的指纹的热度。

其中，写数据模块 1404 更新热度的实现可以参见图 5 所示实施例中相关内容描述，在此不再赘述。

在一些可能的实现方式中，所述写请求中还包括第二数据块，所述第二数据块对应的指纹的热度高于所述第一数据块对应的指纹的热度，所述写数据模块 1404 还用于：

将所述第二数据块写入所述存储设备，将所述第二数据块的元数据写入所述元数据管理结构的多个分区中的第二分区，所述第二分区的容量小于所述第一分区的容量。

其中，写数据模块 1404 写入第二数据块以及第二数据块的元数据的具体实现可以参考写入第一数据块及其元数据的相关内容描述，在此不再赘述。

在一些可能的实现方式中，所述元数据管理结构的多个分区的容量根据分区决策模型确定，所述分区决策模型用于预测预设的分区容量组合中每个分区容量组合应用于所述元数据管理结构后对应的分区收益，并确定分区收益最大的分区容量组合作为所述元数据管理结构的多个分区的容量，所述分区收益根据重删率和分区调整成本中的至少一个确定。

在一些可能的实现方式中，分区收益为重删率，预设的分区容量组合包括第一分区容量组合。所述装置 1400 还包括分区模块 1408，所述分区模块 1408 用于通过如下方式预测所述第一分区容量组合应用于所述元数据管理结构后对应的重删率：

获取所述第一分区容量组合应用于所述元数据管理结构所形成的多个分区中各个分区对应的工作负载特征；

根据所述各个分区对应的工作负载特征，获得所述各个分区对应的数据分布；

根据所述各个分区对应的数据分布以及所述各个分区的容量，获得所述重删率。

其中，分区模块 1408 构建分区决策模型的具体实现可以参见图 6 所示实施例中 S610 至 S616 相关内容描述，在此不再赘述。

在一些可能的实现方式中，所述装置 1400 还包括分区模块 1408，所述分区模块 1408 用于：

周期性地调整所述元数据管理结构的多个分区的容量；

当到达调整时刻时，根据所述调整时刻前的周期对应的分区收益、分区容量组合或各个分区对应的工作负载特征，确定是否调整所述多个分区的容量。

在一些可能的实现方式中，所述重删模块 1406 具体用于：

在所述第一分区中存在与所述第一数据块的指纹相同的指纹，且所述第一分区中的所述指纹的数量达到预设阈值时，删除所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

其中，重删模块 1406 用于实现图 5 所示实施例中 S514、S516 相关内容描述。在此不再赘述。

在一些可能的实现方式中，所述第一数据块的地址信息为所述第一数据块的逻辑地址，所述写数据模块 1404 还用于：

将所述第一数据块的逻辑地址和物理地址写入地址映射表；

所述重删模块 1406 具体用于：

根据所述第一数据块的逻辑地址，从所述地址映射表中获取所述第一数据块的物理地址；

根据所述物理地址从所述存储设备中找到所述第一数据块，并删除所述第一数据块。

其中，写数据模块 1404 还用于实现图 5 所示实施例中 S518 相关内容描述。在此不再赘述。

在一些可能的实现方式中，所述重删模块 1406 还用于：

在删除所述第一分区中的所述第一数据块的元数据之后，将所述地址映射表中所述第一数据块的物理地址修改为所述第一数据块的指纹。

其中，重删模块 1406 还用于实现图 5 所示实施例中 S518 相关内容描述。在此不再赘述。

在一些可能的实现方式中，所述装置 1400 还包括：

淘汰模块 1409，用于当所述逆映射表中的至少一个分区满足淘汰条件，对所述至少一个分区中的所述元数据进行淘汰。

其中，淘汰模块 1409 还用于实现图 5 所示实施例中 S520 相关内容描述。在此不再赘述。

根据本申请实施例的数据重删装置 1400 可对应于执行本申请实施例中描述的方法，并且数据重删装置 1400 的各个模块/单元的上述和其它操作和/或功能分别为了实现图 5、图 6 所示实施例中的各个方法的相应流程，为了简洁，在此不再赘述。

本申请实施例还提供了一种计算机可读存储介质。所述计算机可读存储介质可以是计算机能够存储的任何可用介质或者是包含一个或多个可用介质的数据中心等数据存储设备。所述可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如固态硬盘）等。该计算机可读存储介质包括指令，所述指令指示计算设备或计算设备集群（例如是存储系统）执行上述数据重删方法。

本申请实施例还提供了一种计算机程序产品。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机或数据中心通过有线（例如同轴电缆、光纤、数字用户线（DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机或数据中心进行传输。所述计算机程序产品可以为一个软件安装包，在需要使用前述数据重删方法的任一方法的情况下，可以下载该计算机程序产品并在计算设备或计算设备集群上执行该计算机程序产品。

上述各个附图对应的流程或结构的描述各有侧重，某个流程或结构中没有详述的部分，可以参见其他流程或结构的相关描述。

权 利 要 求

1. 一种数据重删方法，其特征在于，所述方法包括：

接收写请求，所述写请求中包括第一数据块；

将所述第一数据块写入存储设备；

将所述第一数据块的元数据写入元数据管理结构的多个分区中的第一分区，所述第一分区为根据所述第一数据块的特征确定的，所述第一数据块的元数据包括所述第一数据块的指纹及地址信息；

在所述第一分区中存在与所述第一数据块的指纹相同的指纹时，删除所述第一分区中所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

2. 根据权利要求 1 所述的方法，其特征在于，所述第一数据块的特征为所述第一数据块对应的指纹，所述将所述第一数据块的元数据写入元数据管理结构的多个分区中的第一分区，包括：

确定所述第一数据块对应的指纹的热度；

根据所述第一数据块对应的指纹的热度确定所述热度对应的所述元数据管理结构的多个分区中的所述第一分区；

将所述第一数据块的元数据写入所述第一分区。

3. 根据权利要求 2 所述的方法，其特征在于，在将所述第一数据块的元数据写入所述第一分区之后，所述方法还包括：

确定所述第一数据块的逻辑地址的热度；

将所述逻辑地址的热度累加至所述第一数据块对应的指纹的热度，以更新所述第一数据块对应的指纹的热度。

4. 根据权利要求 1 至 3 任一项所述的方法，其特征在于，所述写请求中还包括第二数据块，所述第二数据块对应的指纹的热度高于所述第一数据块对应的指纹的热度，所述方法还包括：

将所述第二数据块写入所述存储设备，将所述第二数据块的元数据写入所述元数据管理结构的多个分区中的第二分区，所述第二分区的容量小于所述第一分区的容量。

5. 根据权利要求 1 至 4 任一项所述的方法，其特征在于，所述元数据管理结构的多个分区的容量根据分区决策模型确定，所述分区决策模型用于预测预设的分区容量组合中每个分区容量组合应用于所述元数据管理结构后对应的分区收益，并确定分区收益最大的分区容量组合作为所述元数据管理结构的多个分区的容量，所述分区收益根据重删率和分区调整成本中的至少一个确定。

6. 根据权利要求 5 所述的方法，其特征在于，所述分区收益为重删率，所述预设的分区容量组合包括第一分区容量组合，所述分区决策模型通过如下方式预测所述第一分区容量组合应用于所述元数据管理结构后对应的重删率：

获取所述第一分区容量组合应用于所述元数据管理结构所形成的多个分区中各个分区对应的工作负载特征；

根据所述各个分区对应的工作负载特征，获得所述各个分区对应的数据分布；

根据所述各个分区对应的数据分布以及所述各个分区的容量，获得所述重删率。

7. 根据权利要求 5 或 6 所述的方法，其特征在于，所述方法还包括：

周期性地调整所述元数据管理结构的多个分区的容量；

当到达调整时刻时，根据所述调整时刻前的周期对应的分区收益、分区容量组合或各个分区对应的工作负载特征，确定是否调整所述多个分区的容量。

8. 根据权利要求 1 至 7 任一项所述的方法，其特征在于，所述在所述第一分区中存在与所述第一数据块的指纹相同的指纹时，删除所述第一分区中所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块，包括：

在所述第一分区中存在与所述第一数据块的指纹相同的指纹，且所述第一分区中的所述指纹的数量达到预设阈值时，删除所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

9. 根据权利要求 1 至 8 任一项所述的方法，其特征在于，所述第一数据块的地址信息为所述第一数据块的逻辑地址，所述方法还包括：

将所述第一数据块的逻辑地址和物理地址写入地址映射表；

所述根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块，包括：

根据所述第一数据块的逻辑地址，从所述地址映射表中获取所述第一数据块的物理地址；

根据所述物理地址从所述存储设备中找到所述第一数据块，并删除所述第一数据块。

10. 根据权利要求 9 所述的方法，其特征在于，在删除所述第一分区中的所述第一数据块的元数据之后，所述方法还包括：

将所述地址映射表中所述第一数据块的物理地址修改为所述第一数据块的指纹。

11. 根据权利要求 9 所述的方法，其特征在于，所述方法还包括：

当所述元数据管理结构中的至少一个分区满足淘汰条件，对所述至少一个分区中的所述元数据进行淘汰。

12. 一种数据重删装置，其特征在于，所述装置包括：

通信模块，用于接收写请求，所述写请求中包括第一数据块；

写数据模块，用于将所述第一数据块写入存储设备；

所述写数据模块，还用于将所述第一数据块的元数据写入元数据管理结构的多个分区中的第一分区，所述第一分区为根据所述第一数据块的特征确定的，所述第一数据块的元数据包括所述第一数据块的指纹及地址信息；

重删模块，用于在所述第一分区中存在与所述第一数据块的指纹相同的指纹时，删除所述第一分区中所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

13. 根据权利要求 12 所述的装置，其特征在于，所述第一数据块的特征为所述第一数据块对应的指纹，所述写数据模块具体用于：

确定所述第一数据块对应的指纹的热度；

根据所述第一数据块对应的指纹的热度确定所述热度对应的所述元数据管理结构的多个分区中的所述第一分区；

将所述第一数据块的元数据写入所述第一分区。

14. 根据权利要求 13 所述的装置，其特征在于，所述写数据模块还用于：

在将所述第一数据块的元数据写入所述第一分区之后，确定所述第一数据块的逻辑地址的热度；

将所述逻辑地址的热度累加至所述第一数据块对应的指纹的热度，以更新所述第一数据块对应的指纹的热度。

15. 根据权利要求 12 至 14 任一项所述的装置，其特征在于，所述写请求中还包括第二数据块，所述第二数据块对应的指纹的热度高于所述第一数据块对应的指纹的热度，所述写数据模块还用于：

将所述第二数据块写入所述存储设备，将所述第二数据块的元数据写入所述元数据管理结构的多个分区中的第二分区，所述第二分区的容量小于所述第一分区的容量。

16. 根据权利要求 12 至 15 任一项所述的装置，其特征在于，所述元数据管理结构的多个分区的容量根据分区决策模型确定，所述分区决策模型用于预测预设的分区容量组合中每个分区容量组合应用于所述元数据管理结构后对应的分区收益，并确定分区收益最大的分区容量组合作为所述元数据管理结构的多个分区的容量，所述分区收益根据重删率和分区调整成本中的至少一个确定。

17. 根据权利要求 16 所述的装置，其特征在于，所述分区收益为重删率，所述预设的分区容量组合包括第一分区容量组合，所述分区决策模型通过如下方式预测所述第一分区容量组合应用于所述元数据管理结构后对应的重删率：

获取所述第一分区容量组合应用于所述元数据管理结构所形成的多个分区中各个分区对应的工作负载特征；

根据所述各个分区对应的工作负载特征，获得所述各个分区对应的数据分布；

根据所述各个分区对应的数据分布以及所述各个分区的容量，获得所述重删率。

18. 根据权利要求 16 或 17 所述的装置，其特征在于，所述装置还包括分区模块，所述分区模块用于：

周期性地调整所述元数据管理结构的多个分区的容量；

当到达调整时刻时，根据所述调整时刻前的周期对应的分区收益、分区容量组合或各个分区对应的工作负载特征，确定是否调整所述多个分区的容量。

19. 根据权利要求 12 至 18 任一项所述的装置，其特征在于，所述重删模块具体用于：

在所述第一分区中存在与所述第一数据块的指纹相同的指纹，且所述第一分区中的所述指纹的数量达到预设阈值时，删除所述第一数据块的元数据，并根据所述第一数据块的地址信息从所述存储设备中删除所述第一数据块。

20. 根据权利要求 12 至 19 任一项所述的装置，其特征在于，所述第一数据块的地址信息为所述第一数据块的逻辑地址，所述写数据模块还用于：

将所述第一数据块的逻辑地址和物理地址写入地址映射表；

所述重删模块具体用于：

根据所述第一数据块的逻辑地址，从所述地址映射表中获取所述第一数据块的物理地址；

根据所述物理地址从所述存储设备中找到所述第一数据块，并删除所述第一数据块。

21. 根据权利要求 20 所述的装置，其特征在于，所述重删模块还用于：

在删除所述第一分区中的所述第一数据块的元数据之后，将所述地址映射表中所述第一数据块的物理地址修改为所述第一数据块的指纹。

22. 根据权利要求 20 所述的装置，其特征在于，所述装置还包括：

淘汰模块，用于当所述逆映射表中的至少一个分区满足淘汰条件，对所述至少一个分区中的所述元数据进行淘汰。

23. 一种存储系统，其特征在于，所述存储系统包括至少一个处理器和至少一个存储器，所述至少一个存储器中存储有计算机可读指令；所述至少一个处理器执行所述计算机可读指令，以使得所述存储系统执行如权利要求 1 至 11 中任一项所述的方法。

24. 一种计算机可读存储介质，其特征在于，包括计算机可读指令；所述计算机可读指令用于实现权利要求 1 至 11 中任一项所述的方法。

25. 一种计算机程序产品，其特征在于，包括计算机可读指令；所述计算机可读指令用于实现权利要求 1 至 11 中任一项所述的方法。

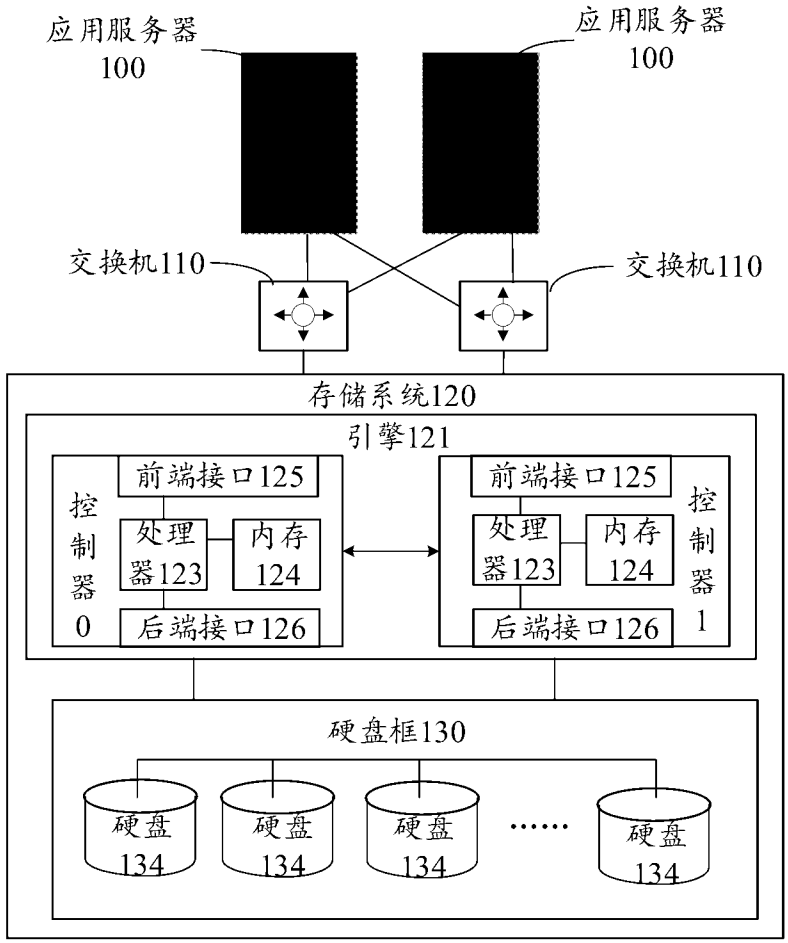
应用服务器
100
应用服务器
100
交换机110
交换机110
存储系统120
引擎121
前端接口125
处理器123
内存124
后端接口126
控制器0
前端接口125
处理器123
内存124
后端接口126
控制器1
硬盘框130
硬盘134
硬盘134
硬盘134
硬盘134

图 1

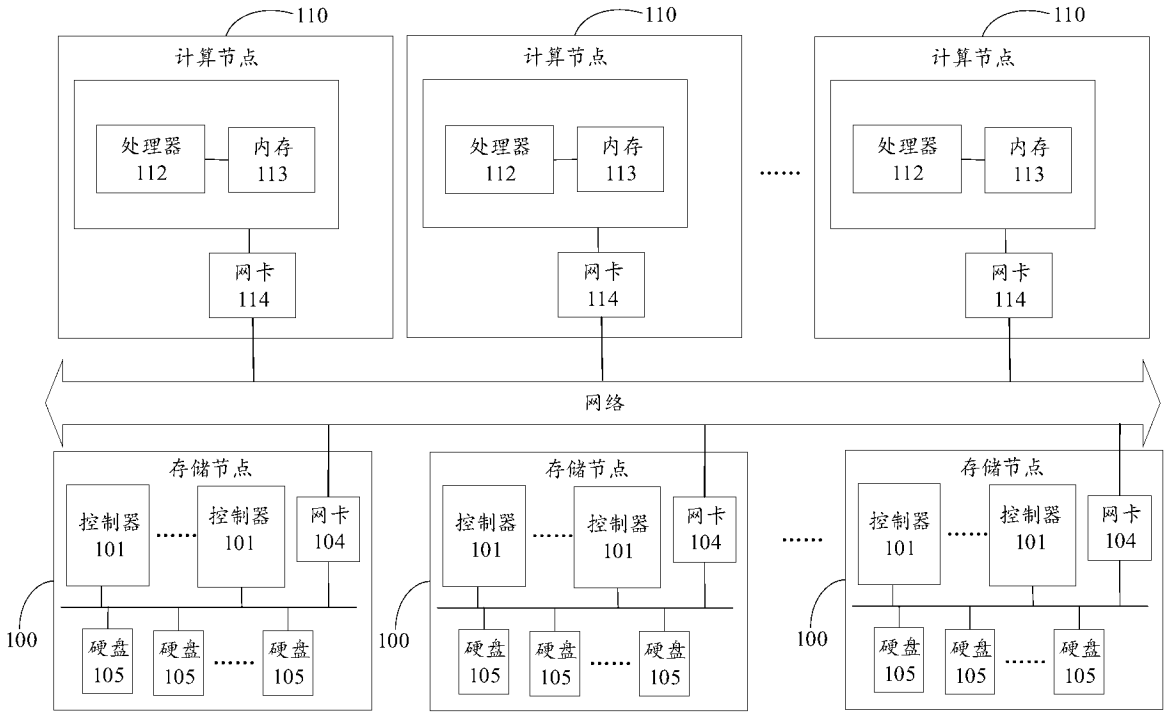
110
计算节点
处理器112
内存113
网卡114
网络
存储节点
控制器101
网卡104
硬盘105
100

图 2

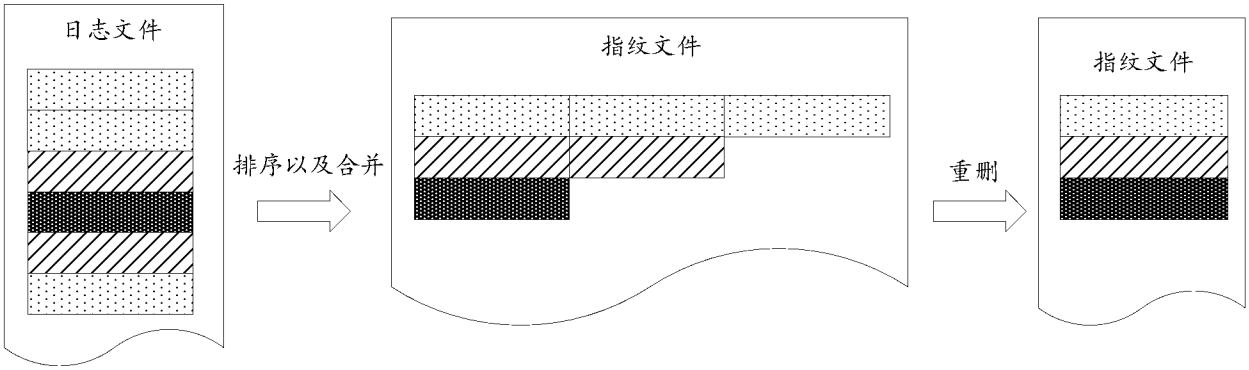
日志文件
排序以及合并
指纹文件
重删
指纹文件

图 3

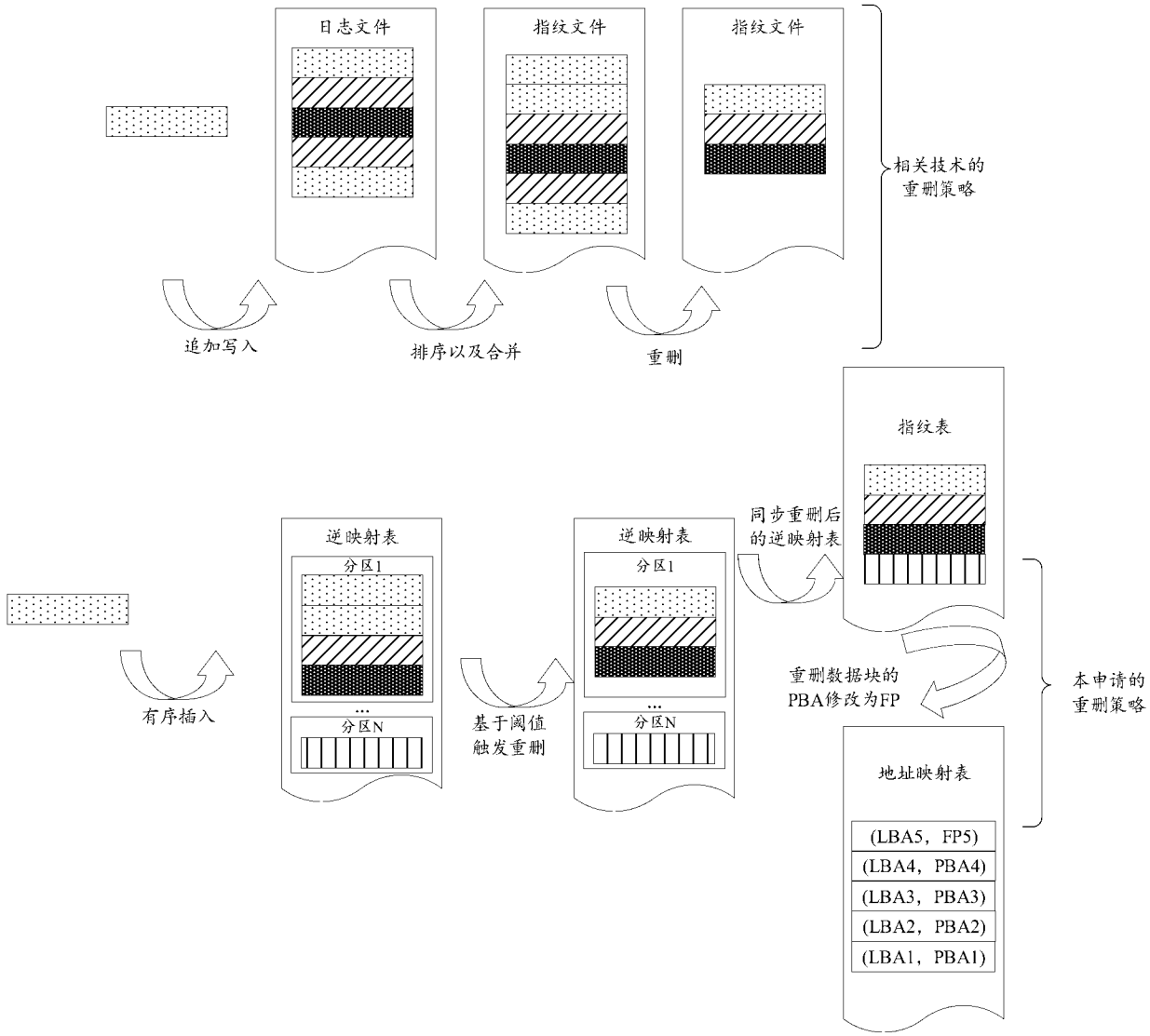
日志文件
指纹文件
指纹文件
相关技术的
重删策略
追加写入
排序以及合并
重删
指纹表
同步重删后
的逆映射表
逆映射表
分区1
...
分区N
逆映射表
分区1
...
分区N
有序插入
基于阈值
触发重删
重删数据块的
PBA修改为FP
本申请的
重删策略
地址映射表
(LBA5，FP5)
(LBA4，PBA4)
(LBA3，PBA3)
(LBA2，PBA2)
(LBA1，PBA1)

图 4

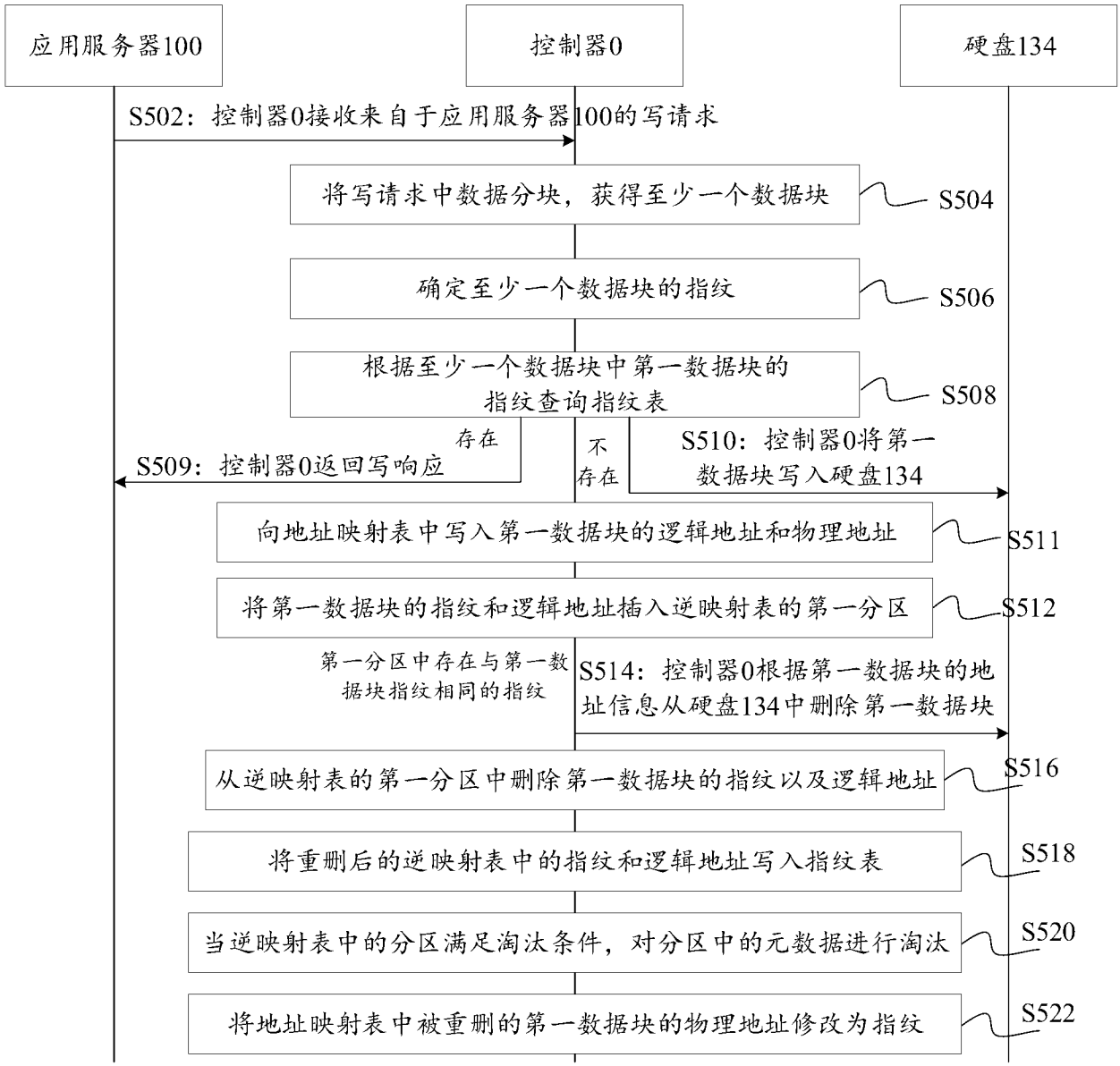
应用服务器100
控制器0
硬盘134
S502：控制器0接收来自于应用服务器100的写请求
将写请求中数据分块，获得至少一个数据块
S504
确定至少一个数据块的指纹
S506
根据至少一个数据块中第一数据块的指纹查询指纹表
S508
存在
S509：控制器0返回写响应
不存在
S510：控制器0将第一数据块写入硬盘134
向地址映射表中写入第一数据块的逻辑地址和物理地址
S511
将第一数据块的指纹和逻辑地址插入逆映射表的第一分区
S512
第一分区中存在与第一数据块指纹相同的指纹
S514：控制器0根据第一数据块的地址信息从硬盘134中删除第一数据块
从逆映射表的第一分区中删除第一数据块的指纹以及逻辑地址
S516
将重删后的逆映射表中的指纹和逻辑地址写入指纹表
S518
当逆映射表中的分区满足淘汰条件，对分区中的元数据进行淘汰
S520
将地址映射表中被重删的第一数据块的物理地址修改为指纹
S522

图 5

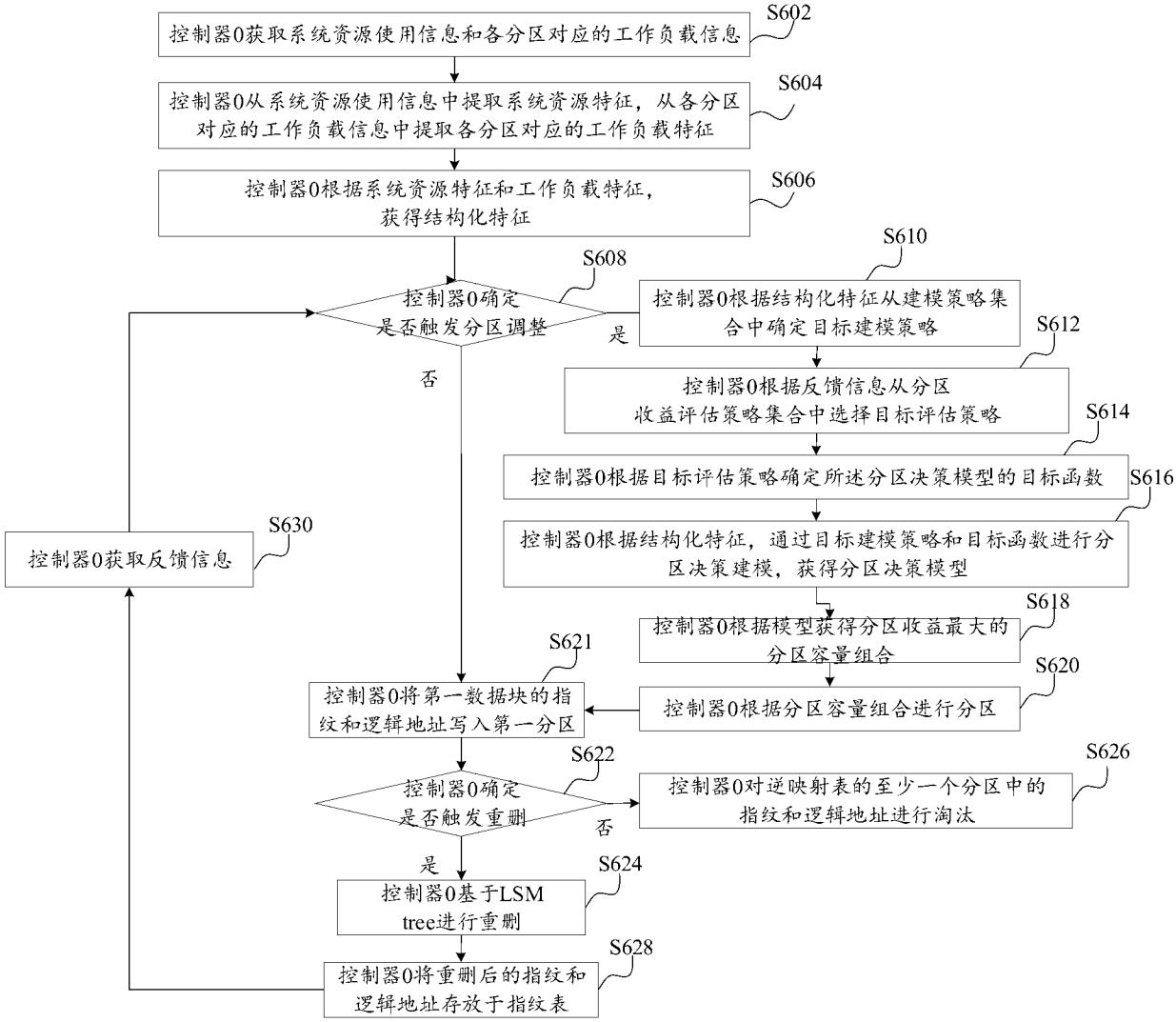
控制器0获取系统资源使用信息和各分区对应的工作负载信息
S602
控制器0从系统资源使用信息中提取系统资源特征，从各分区对应的工作负载信息中提取各分区对应的工作负载特征
S604
控制器0根据系统资源特征和工作负载特征，获得结构化特征
S606
控制器0确定是否触发分区调整
S608
是
否
控制器0根据结构化特征从建模策略集合中确定目标建模策略
S610
控制器0根据反馈信息从分区收益评估策略集合中选择目标评估策略
S612
控制器0根据目标评估策略确定所述分区决策模型的目标函数
S614
控制器0根据结构化特征，通过目标建模策略和目标函数进行分区决策建模，获得分区决策模型
S616
控制器0根据模型获得分区收益最大的分区容量组合
S618
控制器0根据分区容量组合进行分区
S620
控制器0将第一数据块的指纹和逻辑地址写入第一分区
S621
控制器0确定是否触发重删
S622
否
是
控制器0对逆映射表的至少一个分区中的指纹和逻辑地址进行淘汰
S626
控制器0基于LSM tree进行重删
S624
控制器0将重删后的指纹和逻辑地址存放于指纹表
S628
控制器0获取反馈信息
S630

图 6

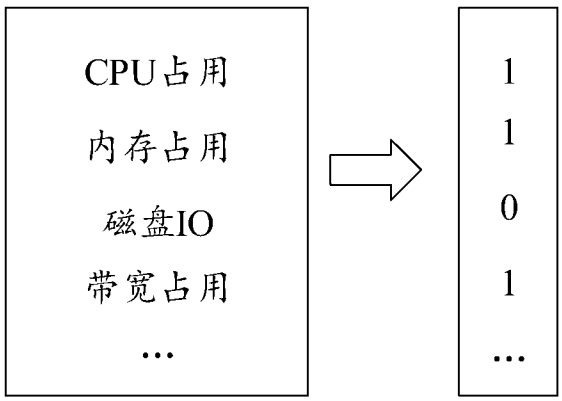
CPU占用
内存占用
磁盘IO
带宽占用
…
1
1
0
1
…

图 7

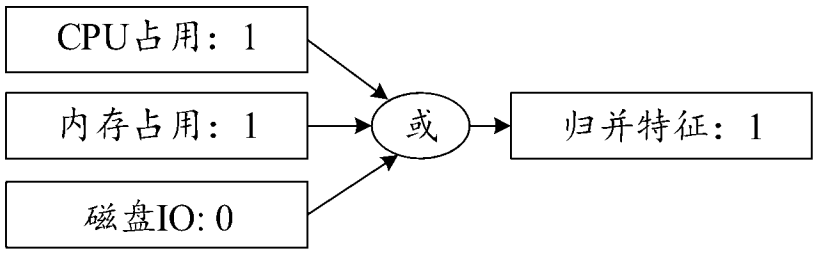
CPU占用：1
内存占用：1
磁盘IO: 0
或
归并特征：1

图 8

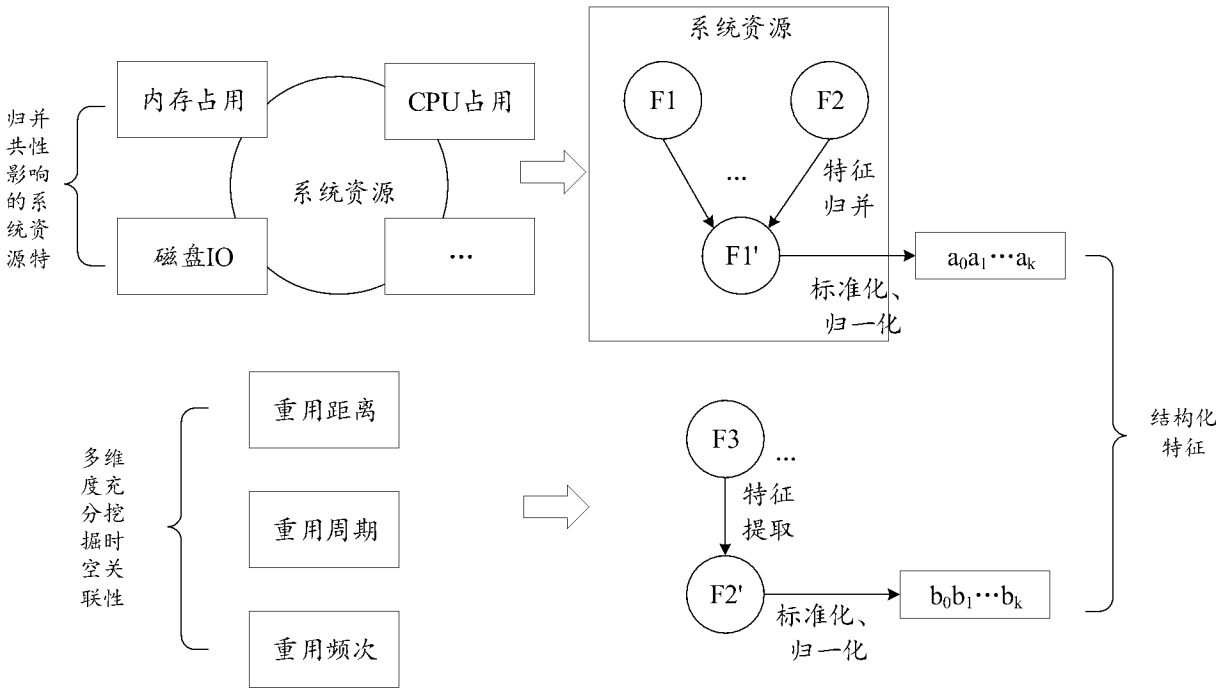
归并共性影响的系统资源特
内存占用
CPU占用
系统资源
磁盘IO
…
系统资源
F1
F2
…
特征归并
F1'
标准化、归一化
$a_0a_1\cdots a_k$
结构化特征
多维度充分挖掘时空关联性
重用距离
重用周期
重用频次
F3
…
特征提取
F2'
标准化、归一化
$b_0b_1\cdots b_k$

图 9

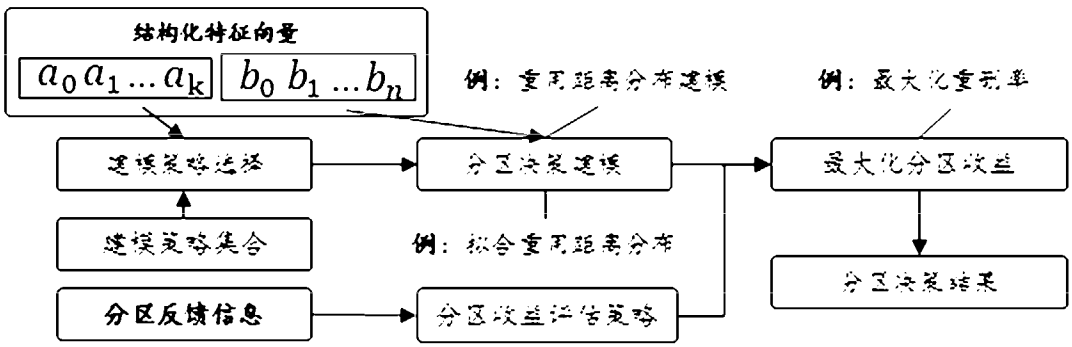
结构化特征向量
$a_0\,a_1\ldots a_k$
$b_0\,b_1\ldots b_n$
例：重用距离分布建模
例：最大化重用率
建模策略选择
分区决策建模
最大化分区收益
建模策略集合
例：拟合重用距离分布
分区反馈信息
分区收益评估策略
分区决策结果

图 10

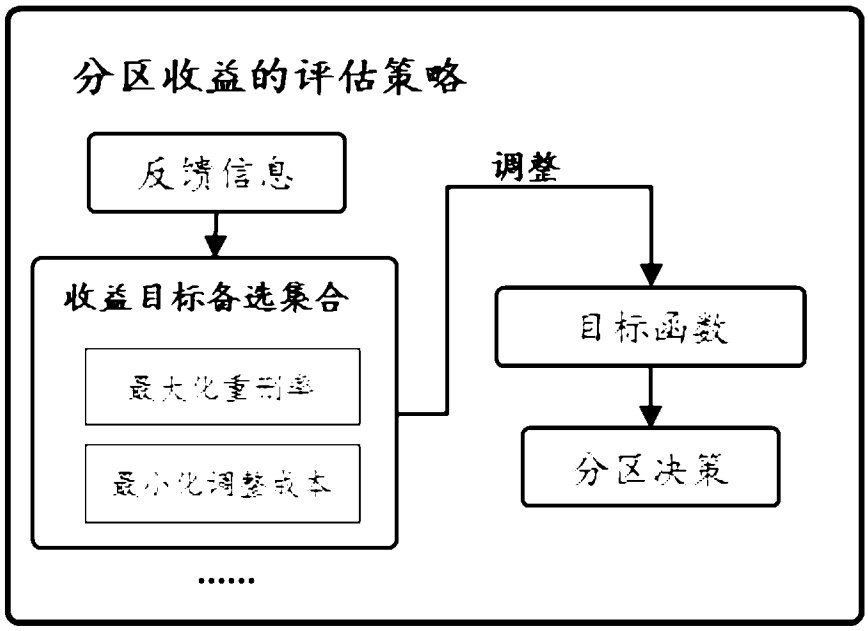
分区收益的评估策略
反馈信息
调整
收益目标备选集合
最大化重删率
最小化调整代本
.....
目标函数
分区决策

图 11

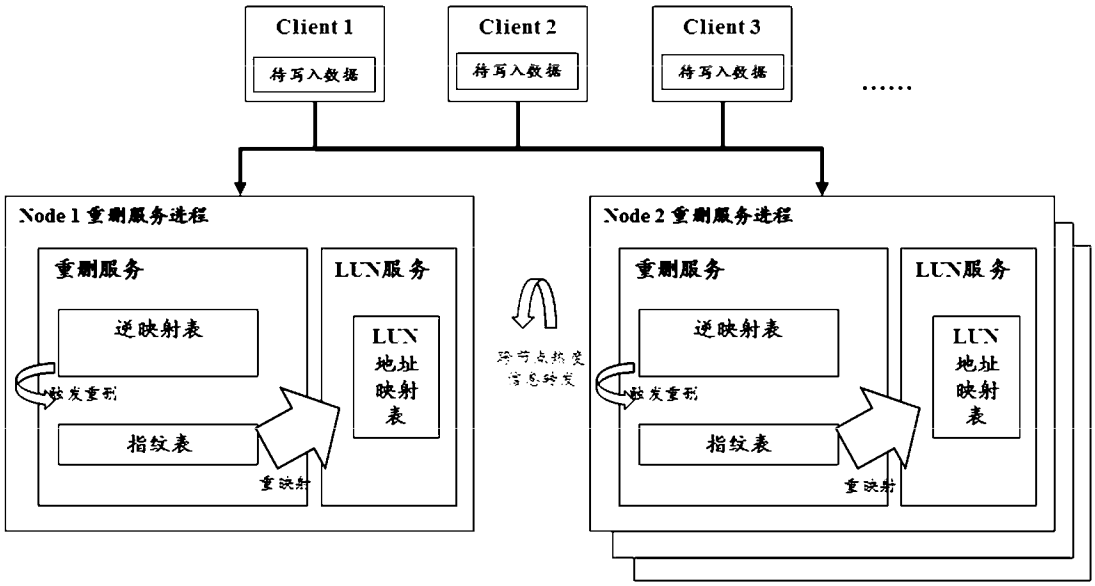
Client 1
待写入数据
Client 2
待写入数据
Client 3
待写入数据
.....
Node 1 重删服务进程
重删服务
逆映射表
触发重删
指纹表
重映射
LUN服务
LUN 地址 映射 表
Node 2 重删服务进程
重删服务
逆映射表
触发重删
指纹表
重映射
LUN服务
LUN 地址 映射 表

图 12

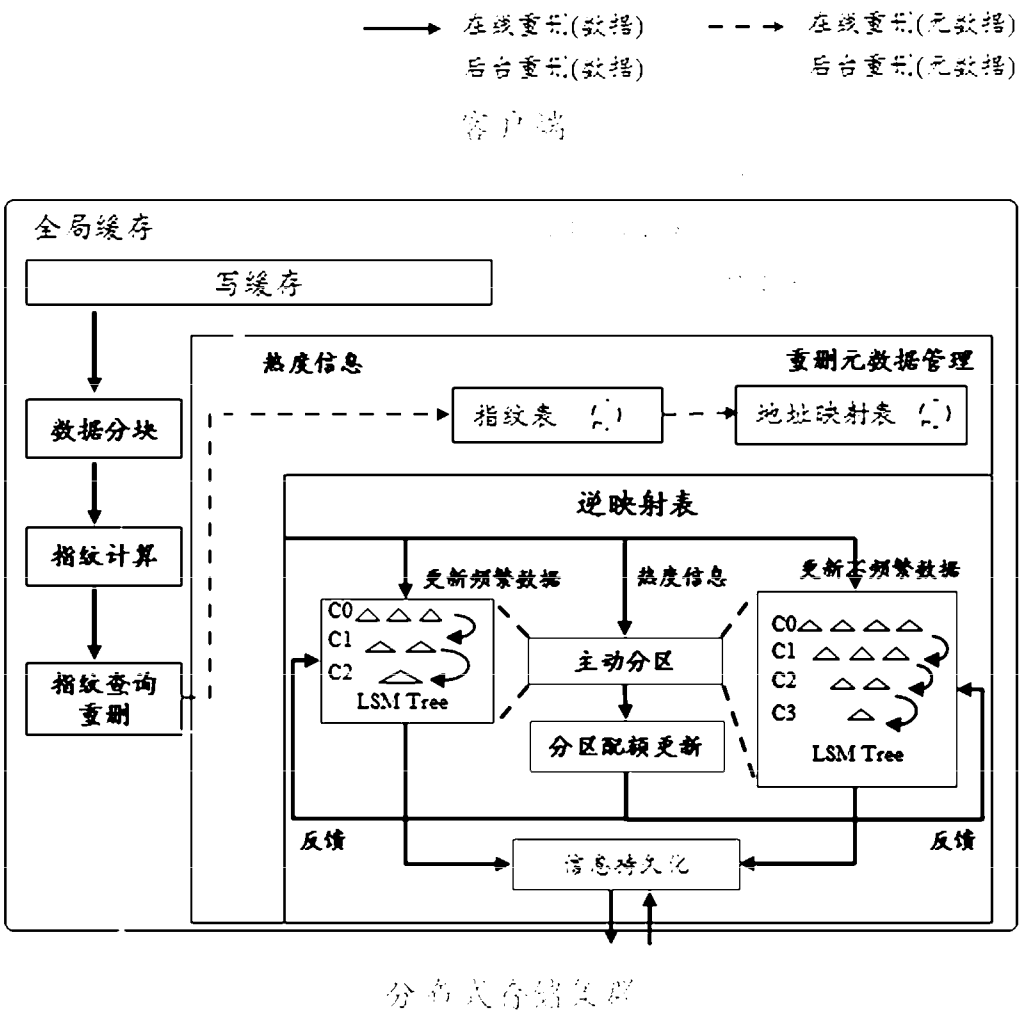
在线重删(数据)
后台重删(数据)
在线重删(元数据)
后台重删(元数据)
客户端
全局缓存
写缓存
数据分块
指纹计算
指纹查询
重删
热度信息
重删元数据管理
指纹表
地址映射表
逆映射表
更新频繁数据
热度信息
更新不频繁数据
C0
C1
C2
LSM Tree
主动分区
分区配额更新
C0
C1
C2
C3
LSM Tree
反馈
反馈
信息持久化
分布式存储集群

图 13

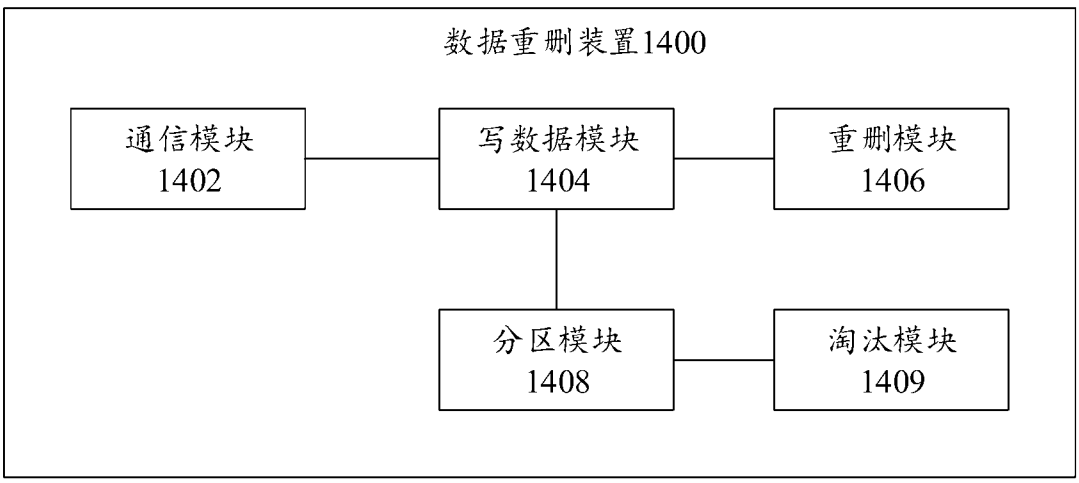
数据重删装置1400
通信模块
1402
写数据模块
1404
重删模块
1406
分区模块
1408
淘汰模块
1409

图 14

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2023/101303

A. CLASSIFICATION OF SUBJECT MATTER

G06F16/174(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT, ENTXT, DWPI, CNKI: 重删, 去重, 指纹, 摘要, 哈希, 散列, 元数据, 地址, 分区, deduplication, fingerprint, hash, meta, address, partition

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 110618789 A (HUAWEI TECHNOLOGIES CO., LTD.) 27 December 2019 (2019-12-27) description, paragraphs [0062]-[0083] and [0102]-[0103]	1-4, 8-15, 19-25
Y	CN 107329692 A (MACROSAN TECHNOLOGIES CO., LTD.) 07 November 2017 (2017-11-07) description, paragraphs [0090] and [0107]-[0109]	1-4, 8-15, 19-25
A	CN 103514250 A (YI LETIAN et al.) 15 January 2014 (2014-01-15) entire document	1-25
A	CN 111381779 A (SANGFOR TECHNOLOGIES INC.) 07 July 2020 (2020-07-07) entire document	1-25
A	US 2020133547 A1 (EMC IP HOLDING CO. LLC.) 30 April 2020 (2020-04-30) entire document	1-25

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
01 September 2023	**12 September 2023**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **China No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2023/101303

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 110618789 A	27 December 2019	WO 2021027541 A1	18 February 2021
		US 2022164316 A1	26 May 2022
		EP 4016276 A1	22 June 2022
CN 107329692 A	07 November 2017	None	
CN 103514250 A	15 January 2014	None	
CN 111381779 A	07 July 2020	None	
US 2020133547 A1	30 April 2020	None	

国际检索报告

国际申请号
PCT/CN2023/101303

A. 主题的分类

G06F16/174(2019.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC: G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CNTXT,ENTXT,DWPI,CNKI: 重删,去重,指纹,摘要,哈希,散列,元数据,地址,分区, deduplication, fingerprint, hash, meta, address, partition

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
Y	CN 110618789 A (华为技术有限公司) 2019年12月27日 (2019-12-27) 说明书第[0062]-[0083],[0102]-[0103]段	1-4, 8-15,19-25
Y	CN 107329692 A (杭州宏杉科技股份有限公司) 2017年11月7日 (2017-11-07) 说明书第[0090],[0107]-[0109]段	1-4, 8-15, 19-25
A	CN 103514250 A (易乐天 等) 2014年1月15日 (2014-01-15) 全文	1-25
A	CN 111381779 A (深信服科技股份有限公司) 2020年7月7日 (2020-07-07) 全文	1-25
A	US 2020133547 A1 (EMC IP HOLDING COMPANY LLC) 2020年4月30日 (2020-04-30) 全文	1-25

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"D" 申请人在国际申请中引证的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2023年9月1日	2023年9月12日

ISA/CN的名称和邮寄地址	受权官员
中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088	庄湧 电话号码 (+86) 010-53961296

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2023/101303

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	110618789	A	2019年12月27日	WO	2021027541	A1	2021年2月18日
				US	2022164316	A1	2022年5月26日
				EP	4016276	A1	2022年6月22日
CN	107329692	A	2017年11月7日		无		
CN	103514250	A	2014年1月15日		无		
CN	111381779	A	2020年7月7日		无		
US	2020133547	A1	2020年4月30日		无		