计算机科学
COMPUTER SCIENCE

基于擦除编码和副本复制的分布式混合存储研究

付雄, 宋朝阳, 王俊昌, 邓松

引用本文

付雄, 宋朝阳, 王俊昌, 邓松. 基于擦除编码和副本复制的分布式混合存储研究[J]. 计算机科学, 2025, 52(2): 42-47.

FU Xiong, SONG Zhaoyang, WANG Junchang, DENG Song. Study on Distributed Hybrid Storage Based on Erasure Coding and Replication [J]. Computer Science, 2025, 52(2): 42-47.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231200021

基于擦除编码和副本复制的分布式混合存储研究

付 雄 宋朝阳 王俊昌 邓 松
南京邮电大学计算机学院 南京 210023

摘 要 随着大数据技术、云计算、计算机技术和网络技术的迅猛发展,互联网数据呈爆炸性增长,海量数据的高效存储成为当前互联网技术亟待解决的问题。然而,传统的多副本冗余机制导致了巨大的存储成本,引起了研究者们对新型存储解决方案的关注。在这一背景下,提出了一种基于擦除编码和副本复制的分布式混合存储策略。该策略根据数据特性,对热数据采用副本复制以确保高可靠性和性能,而对冷数据则采用擦除编码以提高存储利用率。基于牛顿冷却定律将数据文件划分为热文件和冷文件,并引入一种自适应的数据温度识别及冷热数据自适应动态分配算法,使系统能够在运行时自动调整冷热数据的比例,然后根据实时数据冷热情况智能调整数据的存储策略,体现了系统在动态环境下的自适应性。其不仅增强了系统对动态工作负载的适应能力,也为提高分布式存储系统在实际应用中的效率和灵活性提供了新的范式。这一创新点在学术和实践层面都具有重要的推动意义。同时,通过仿真实验验证了该策略的有效性和可用性,其为分布式存储系统的优化提供了新的思路。

关键词:大数据;副本复制;擦除编码;冷热数据;存储利用率
中图分类号 TP393

Study on Distributed Hybrid Storage Based on Erasure Coding and Replication

FU Xiong,SONG Zhaoyang,WANG Junchang and DENG Song
School of Computer Science,Nanjing University of Posts and Telecommunications,Nanjing 210023,China

Abstract With the rapid development of big data technology,cloud computing,computer technology and network technology,Internet data has shown explosive growth,and efficient storage of massive data has become an urgent challenge for current Internet technology. However,traditional multi-copy redundancy mechanisms result in huge storage costs,thus drawing attention to new storage solutions. In this context,a distributed hybrid storage strategy based on erasure coding and replica replication is proposed. Based on data characteristics,this strategy uses replica replication for hot data to ensure high reliability and performance,while erasure coding is used for cold data to improve storage utilization. Based on Newton's cooling law,the data files is divided into hot files and cold files,and an adaptive data temperature identification and hot and cold data adaptive dynamic allocation algorithm are introduced,so that the system can automatically adjust the ratio of hot and cold data at runtime,and then intelligently adjust the data storage strategy according to the the hot and cold conditions of real-time data,which reflects the system's adaptability in a dynamic environment. It not only enhances the system's adaptability to dynamic workloads,but also provides a new paradigm for the efficiency and flexibility of distributed storage systems in practical applications. This innovation has important promotion significance at both the academic and practical levels. At the same time,the effectiveness and usability of the strategy have been verified through simulation experiments,which provides new ideas for the optimization of distributed storage systems.

Keywords Big data,Replica replication,Erasure coding,Hot and cold data,Storage utilization rate

1 引言

近年来云计算、大数据技术以及其他网络技术迅速发展,网络中的数据量呈现出指数级别的增长,海量数据的存储逐渐成为互联网技术发展亟需解决的问题[1]。19 世纪 80 年代,分布式存储系统应运而生。分布式存储系统,顾名思义就是将大量的普通服务器通过网络互联,对外作为一个整体提供存储服务。其具有可扩展性、可用性、可靠性、高性能、易维护、低成本等特性[2]。

分布式存储系统的数据存储策略一直在发展之中。为了保证数据的可靠性,大多数分布式存储系统采用了多副本冗余机制。例如,HDFS 文件系统、Lustre、PVFS 等分布式存储系统为每个数据块创建了 3 个副本[3-4],并确保每个数据块的所有副本分布在不同的故障域中,从而在两份数据块丢失的

到稿日期:2023-12-04 返修日期:2024-04-14
基金项目:国家自然科学基金(61602264);江苏省重点研发计划(社会发展)(BE2017743)
This work was supported by the National Natural Science Foundation of China(61602264) and Key Research & Development Program (Social Development) of Jiangsu Province(BE2017743).
通信作者:付雄(fux@njupt.edu.cn)

情况下仍然能够保证集群数据的可靠性。但是，这种系统的实际存储利用率只有33%，并且副本存储的位置是随机散布在集群的服务器中的，这同时也影响了集群的性能。这种将不同副本放置于不同故障域中的多副本复制策略虽然保证了数据的可靠性，但它需要$(N+1)$倍存储空间来容忍N次故障，增加了系统的存储成本[5-6]。

虽然计算机硬件技术飞速发展，尤其是CPU计算能力显著增强，但是存储技术的进步却相对缓慢，这种不平衡的发展态势催生了对新型存储方案的需求。在这一背景下，擦除编码技术因在存储效率和可靠性方面具有潜在优势而受到学术界和工业界的广泛关注[7]。

擦除编码技术的核心在于将原始数据分割成多个数据块，并通过特定的算法生成一定数量的校验块。这种编码方式不仅能够在原始数据块发生丢失或损坏的情况下重建原始数据，而且相比于传统的副本复制策略，能够显著降低存储资源的冗余度。RS编码是一种常见的擦除编码方法，它通过线性代数原理在数据块中引入冗余，从而实现数据的容错和恢复。

然而，擦除编码技术在提升存储效率的同时，也带来了一系列挑战。编码和解码过程需要消耗大量的计算资源，这可能导致系统在处理数据请求时出现延迟，并且增加数据更新操作的复杂性，因为每次更新都涉及到对所有相关数据块的重新编码，这不仅增加了CPU的计算负担，还可能导致网络带宽和硬盘I/O资源的过度消耗。因此，擦除编码技术主要适用于对数据访问延迟要求不高的场景[8-9]。

此外，数据的温度模型是分布式存储系统研究的一个重要领域，它根据数据的访问频率和时间敏感性对数据进行分类。热数据通常需要极快的访问速度和低延迟的响应，而冷数据则可以容忍较慢的访问速度和较高的访问延迟。因此，对热数据和冷数据的识别及其存储策略的选择，对于优化存储资源的分配、降低存储成本以及提高系统整体性能具有重要意义[10-12]。

Zhang等[13]通过文件大小区分了热数据和冷数据，对大字节数据使用擦除编码，对小字节数据使用副本进行备份。Ying等[14]通过数据的访问频率来判断数据的冷热程度。Li等[15]基于数据的访问频率，并根据数据的冷热情况，对数据采用不同的存储策略，同时定期更新数据。在此基础上，本文提出一种基于擦除编码和副本复制的分布式混合存储策略，不再单纯地依靠副本复制来获得数据的可靠性，而是综合考虑数据特征，对冷热数据采取不同的存储策略。热数据承担了大部分的读写请求，如果采用擦除编码的方式进行数据存储，虽然可以节约空间，但是每次读写请求都需要消耗CPU资源，当并发量较高时，很容易引起系统的崩溃。因此本文对热数据采取副本复制的方式来保证数据的可靠性，因为热数据本身的数量并不大，且热数据大多为小文件，进行副本复制不会造成太多的空间浪费，并且多副本机制可以对并发的读写请求进行负载均衡，可以有效提高集群的性能。而冷数据虽然较多，但很少会有读写请求，所以对冷数据采取擦除编码的存储策略，可以有效地提高集群的存储利用率。

在系统运行阶段，随着数据访问频率和时间的推移，数据的冷热状态会发生实时变化。本文所引入的数据温度识别及冷热数据自适应动态分配算法具备系统自动调整冷热数据比例的特性，并且在变化的数据冷热情境下，智能地调整合适的存储策略以进行数据存储。

该算法在数据温度的感知上表现出极高的自适应性，能够动态识别不同时间段和访问频率下的冷热数据变化。通过自动调整冷热数据的比例，系统实现了对于动态环境的敏感性，从而能更有效地应对实际应用中的多变场景。本文提出的算法在学术研究中为分布式存储系统的自主优化提供了一种具有前瞻性的方法，其自适应性和智能性不仅增强了系统的灵活性，也为数据存储与访问效率的提升提供了理论支持。这一创新点为分布式存储系统的性能优化及适应性调整提供了有力的理论基础。本文的主要贡献如下：

1)基于牛顿冷却定律，将数据文件分为热数据文件和冷数据文件；

2)提出一种数据温度识别及冷热数据自适应动态分配算法(DTP-DAA)，对不同温度下的数据采取不同的存储策略，极大地提高了系统的性能；

3)基于该存储策略进行仿真实验，验证了该存储策略的效率和可用性。

2　模型与问题定义

本文基于Ceph系统结构，改变原来的数据存储策略，使用数据温度来区分数据文件。所提模型包含4个核心组件：Monitor Daemon(MON)，Object Storage Daemon(OSD)，Metadata Server(MDS)和RADOS。MON服务器负责维护集群的状态和拓扑信息，提供集群的管理接口，处理客户端请求，完成配置变更和故障恢复等操作。OSD服务器是存储集群中的工作节点，负责存储数据块或对象，并提供数据的读写和复制服务。MDS是元数据服务器，负责管理文件系统的元数据信息，包括目录结构、文件属性、权限信息等。RADOS是Ceph的底层对象存储系统，提供对象级别的存储和访问服务[16]。本文所使用的符号定义如表1所列。

表1　重要符号及其含义

Table 1　Important symbols and their meanings

符号	含义
Tid	文件p_i的ID
SP	存储策略(0表示副本复制，1表示擦除编码)
t	温度值
TS	温度状态(0表示热数据，1表示冷数据)
Tidsum	数据文件的个数
DTCT	数据温度转化表
Flag	1表示系统存储效率处于快速上升阶段，2表示系统存储效率处于缓慢上升阶段
$Td_{\text{hot}\,T}$	热数据所占据的比例
$Td_{\text{cold}\,T}$	冷数据所占据的比例
T_{size}	DTCT表的规模
F_o	系统上一个周期内的存储效率
F_c	系统当前周期内的存储效率
g_r	冷热数据文件增加或减少的比例
U_s	存储利用率
t_z	访问时延

2.1　牛顿冷却定律

基于温度模型的数据冷热程度的标识方法，是借鉴牛顿

冷却定律，通过指数衰减来模拟温度变化的过程[17]。将数据看作实际物体，随着时间的推移，物理环境中温度高的物体会逐渐冷却，数据存储中数据的温度也会逐渐降低；当访问数据时，类似于赋予物体新的能量，物体的温度会升高，访问操作也给数据带来了能量，数据的温度会升高，从而实现数据的加温。这样可以得到任意时刻所有数据的温度值，然后根据温度值大小进行排序，从而实现冷热数据的识别。

2.2 基于牛顿冷却定律的时间衰减模型

根据牛顿冷却定律中物体温度冷却速度与物体温度和环境温度的温差成正比，可以得到以下微分方程：

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t}=-k(T(t)-H) \tag{1}$$

其中，$T(t)$表示物体当前的温度，H 表示周围环境的温度，k 表示物体温度变化速度与周围环境温度差的比例系数。通过对该微分方程求解，可以得到：

$$T(t)=(T_0-H)\mathrm{e}^{-kt}+H \tag{2}$$

2.3 温度模型的定义

牛顿冷却定律描述的是物理环境中物体的温度受环境温度的影响而变化的规律，而数据存储中数据冷热程度的变化规律稍有不同。在数据存储中，可以将全部数据的平均温度看作环境温度，其中每一个数据都是独立的，数据的温度不会受到其他数据或存储介质的影响，只与数据本身的访问次数和访问时间有关。因此一个数据如果长时间不访问，其温度最后会无限接近于 0，也就是说对于一个数据来讲，其所处的环境温度对其本身的温度并没有影响，所以在计算数据温度随时间变化时可以忽略环境温度[18-19]。忽略环境温度 H 的影响，增加变量 T_{heat}，即每次访问后数据的增温幅度，得到：

$$T(t_n)=T(t_{n-1})e^{-\alpha(t_n-t_{n-1})}+T_{\mathrm{heat}}*F(t_{n-1}) \tag{3}$$

其中，$F(t_{n-1})$为数据文件在$(n-1)$时期的访问频率，T_{heat}表示每次数据被访问后提高的温度比例系数，α 为冷却系数，$T(t_n)$为数据在 n 时期的数据温度。

2.4 问题定义

假设数据文件组为 $P=[p_1\ \ p_2\ \ \cdots\ \ p_n]$。在当前时刻，一个请求 i 的总时延为t_i，发送请求的时延为t_{send}，接收回复的时延为t_{recv}，服务端获取一个文件块数据的时延为$t_{srv_{\mathrm{process}}}$。若请求的数据为热数据，则有：

$$t_i=t_{\mathrm{send}}+t_{\mathrm{recv}}+t_{srv_{\mathrm{process}}} \tag{4}$$

若请求 i 请求的数据为冷数据，进行冗余数据恢复的时间为t_{reco}，因为擦除编码采取 $p+m$ 的冗余策略，则有：

$$t_i=t_{\mathrm{send}}+t_{\mathrm{recv}}+(p+m)*t_{srv_{\mathrm{process}}}+t_{\mathrm{reco}} \tag{5}$$

对数据文件组 $P=[p_1\ \ p_2\ \ \cdots\ \ p_n]$中的文件进行冷热数据的区分，如果有 k 个文件为热文件，则有 $n-k$ 个文件为冷文件。文件组 $P=[p_1\ \ p_2\ \ \cdots\ \ p_n]$的平均时延则为：

$$t_z=\left(\frac{k}{3p}*t_{\mathrm{hot}}+\frac{(n-k)}{(p+m)}t_{\mathrm{cold}}\right)\Big/\frac{n}{3p} \tag{6}$$

假定全部数据文件都是以 r 副本的形式进行，数据存储时的存储利用率为U_{rep}，数据访问平均时延为t_a，忽略 m 个校验块的影响，则文件组 $P=[p_1\ \ p_2\ \ \cdots\ \ p_n]$的实际存储利用率为：

$$U_s=\frac{kp+km+rpn-rpk}{(p+m)*n}*U_{\mathrm{rep}} \tag{7}$$

所求目标为最大的存储利用率U_s，最小的平均时延t_z，为存储利用率设置权重系数 w，为平均时延设置权重系数$(1-w)$，则可转化为多目标优化问题：

$$f(U_s,t_z)=w*U_s+(1-w)*t_z \tag{8}$$

因此，数据文件组 $P=[p_1\ \ p_2\ \ \cdots\ \ p_n]$的最大存储效率问题可以描述为：

$$\mathrm{Max}\ f(U_s,t_z) \tag{9}$$

$$\mathrm{s.t.}\ U_r>U_{\mathrm{rep}} \tag{10}$$

$$t_z<\frac{3}{2}t_a \tag{11}$$

3 基于擦除编码和副本复制的混合存储策略

本文改变了原来的三副本复制存储策略，使用数据温度来区分数据文件。热数据采用传统的三副本复制策略，冷数据采用擦除编码($K=4$，$M=2$)技术来提高系统的存储利用率[20]。数据文件温度的精确测量直接影响数据文件的存储策略，进而影响整个系统的存储效率和性能。在每个周期 T 开始时，MDS 节点计算存储的数据文件的访问频率。在新的时期$(T+1)$开始之前，更新每个数据文件的温度 t。当获得所有数据文件的新温度后，对其进行排序，并运用系统当前的温度门限重新确定每个数据文件的温度类型。这里的温度门限指的是热数据和冷数据各自所占比例。这个比例并非静态不变，而是会根据分布式存储系统对冷热数据访问率的动态变化而自适应调整。MDS 可以通过对比每个数据文件的新温度类型 TS 和数据文件原始 SP 来获得数据温度转换表(DTCT)，RADOS 根据 DTCT 信息对数据进行重新分布后，修改相应数据文件的冗余存储信息[21-22]。

本文所提出的数据温度识别及冷热数据自适应动态分配算法，允许系统在实时数据的冷热状态发生变化时，智能地调整冷热数据比例。通过此算法，系统能够灵活应对分布式存储系统中冷热数据访问率的波动，使得温度门限随之调整。这一创新性的算法设计在学术上不仅深化了对分布式存储系统动态性能优化的理解，也为实际应用提供了有前瞻性的方法。数据温度识别及冷热数据自适应动态分配算法如算法 1 所示。

算法 1 数据温度识别及冷热数据自适应动态分配算法(DTP-DAA)

输入：$T(t_{n-1})$，$F(t_{n-1})$ of each data file，$T_{\mathrm{heat}}=1$，α；Thresholds of each temperature-type Tablet set by the system，$Td_{\mathrm{hotT}}=10\%$，$Td_{\mathrm{coldT}}=90\%$；$F_c=0$；Temperature increase range，$g_r=1\%$；Initialize a flag representing the rising or falling phase，Flag=1

输出：The data temperature conversion table in n period，DTCT

1. for Tid=1 to Tidsum do
2. $T(t_n)_{\mathrm{Tid}}=T(t_{n-1})_{\mathrm{Tid}}\mathrm{e}^{-\alpha(t_n-t_{n-1})}+T_{\mathrm{heat}}*F(t_{n-1})_{\mathrm{Tid}}$；
3. end for；
4. $F_o=F_c$；
5. Calculate the average access latency of data files，t_z；Calculate the storage utilization of the data file according to the storage policy of the data file，U_s；Calculate the current storage efficiency according to the formula，F_c。
6. if $F_c\geqslant F_o$

```
7.      if Flag == 1 then
8.        TdhotT = TdhotT + gr;
          TdcoldT = TdcoldT − gr, gr = 2 * gr;
9.      else if Flag == 2 then
10.       TdcoldT = TdcoldT − gr; TdhotT = TdhotT + gr;
11.     end if
12.   else if Fc < Fo
13.     if Flag == 1 then
        TdhotT = TdhotT − gr/2 + 1%;
        TdcoldT = TdcoldT + gr/2 − 1%;
        gr = 1%, Flag = 2
14.     else if Flag = 2 then
15.       if Math.abs(Fc − Fo) >= 2
16.         TdhotT = TdhotT/2;
            TdcoldT = TdcoldT + TdhotT;
            Flag = 1, gr = 1%;
17.       end if
18.     end if
19. end if
20. Use any sorting algorithm to rank the tablet temperature in the
    descending order; sTid is the sorted position of this tablet; Tsize is
    the size of DTCT;
1. for sTid = 1 to Tsize do
2.   if sTid ≤ TdhotT * Tidsum then
3.     T STid = 0;
4.     if T STid ≠ S PTid then
5.       DTCT.insert(Tid, T STid);
6.     end if
7.   else
8.     T STid = 1;
9.     if T STid ≠ S PTid then
10.      DTCT.insert(Tid, T STid);
11.    end if
12. end for
13. return DTCT
```

系统每隔一段时间执行一遍该算法。该算法的时间复杂度为 $O(Tn)$，其中 T 为系统运行时间段内该算法的迭代次数，n 为数据文件数量。通过式(3)迭代更新数据文件的温度，对数据文件的温度重新排序，并根据当前周期系统的存储效率与前一个周期系统存储效率之间的变化情况，动态地调整系统的冷热数据温度门限，最终追求系统在稳定运行时获取最大的存储效率。

4　实验结果及分析

本章采用真实的数据集对 DTP-DAA 算法的性能进行了全面评估，并将其与其他经典分布式存储策略进行了对比。实验结果清晰地展示了 DTP-DAA 算法在多个关键方面的卓越表现。相较于一致性哈希算法，DTP-DAA 算法显著提高了存储利用率。而与纠错码算法相比，DTP-DAA 算法能够提供更加稳定的访问时延。DTP-DAA 算法在实际场景中更为灵活，能够动态调整存储利用率与访问时延之间的平衡，以追求最大的存储效率。

具体而言，将 DTP-DAA 算法与以下几种分布式存储策略进行了比较。

1)一致性哈希算法[23]：通过将数据和节点映射到环形空间，利用哈希函数在环上确定数据的存储位置，以提高系统的扩展性和容错性。

2)纠错码算法[24]：一种用于在传输或存储数据时，通过在数据中添加冗余信息，以便在数据受损或丢失时能够检测并纠正错误的编码技术。

3)副本复制：将数据复制到多个节点或服务器，以提高数据的冗余性和可靠性，确保即使某个节点发生故障，数据仍然可用。

本文选择 Gutenberg 数据库的文本文件[25]，进行了数据访问的仿真实验。所有的实验都是在同一台计算机上进行的，其使用的是 Inter Core(TM)i7-7700HQ CPU 2.80GHz 处理器和 16GB 的 RAM，操作系统为 64 位 Windows 10 Professional。

4.1　存储利用率

第一个实验将 DTP-DAA 算法与各存储方法进行了对比，分析了它们在相同数据集下的存储利用率。图 1 给出了不同文件数量下 DTP-DAA 算法与其他存储方式在冷热数据所占比例不同时的磁盘消耗情况。

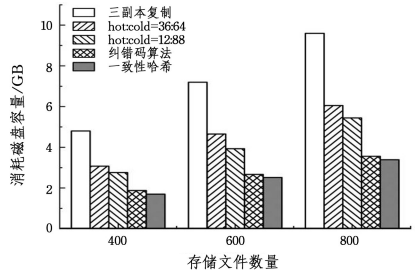

图 1　不同策略在不同文件数量下的总磁盘消耗

Fig. 1　Total disk consumption of different strategies in different the number of stored files

从图 1 可以看出，采用传统的三副本存储策略时，系统存储效率仅为 33.3%；而一致性哈希算法，虽然保证了数据的高可用性，但需要进行多次哈希，且在 3 次哈希情况下，存储效率仍与三副本存储策略相当，均为 33.3%。然而，引入 DTP-DAA 算法后，当冷数据占比达到 64%时，存储利用率显著提高至 54.12%。

相比三副本和一致性哈希策略，DTP-DAA 算法在冷数据占比为 64%时，每存储 1 PB的数据文件，可节省高达 1.17 PB 的磁盘空间。这说明 DTP-DAA 算法相较于传统存储方法，在存储效率上具有显著优势，特别是在处理大规模数据时，为系统提供了更为经济高效的存储解决方案。这一实验验证了 DTP-DAA 算法在存储利用率方面的卓越性能，其为分布式存储领域提供了一种具有实质优势的创新解决方案。

此外，如前所述，DTP-DAA 算法使用擦除编码的方式来存储冷数据，并实现容错功能。因此，如果系统对容错有更高的要求[26]，那么 DTP-DAA 算法可以更显著地提高存储容量的利用率。

本文对不同的容错率进行了验证，如图 2 所示。当容错率

提高到4时，传统的多副本和一致性哈希存储方式每存储1PB的数据文件需要额外消耗4PB的存储空间；而DTP-DAA算法，当冷数据文件占比为88%时，系统仅额外消耗了1.36PB的存储空间。

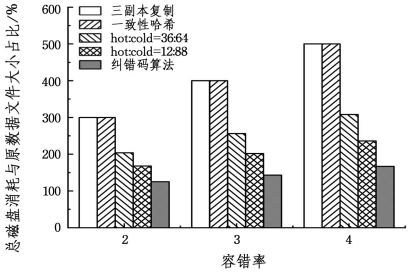

图2 不同策略在不同容错率下的总磁盘消耗与原文件大小占比

Fig. 2 Total disk consumption vs original file size ratio of different strategies at different fault tolerance rates

4.2 访问时延

第二个实验测试了DTP-DAA算法与其他存储策略在不同数量并发线程、不同数据访问场景下的性能差异。本文设定了一个固定数量的访问操作，并模拟了2种不同的数据访问场景：均匀分布的访问序列和非均匀分布的访问序列。在均匀分布场景中，所有访问操作均被允许均匀地访问每个数据片。

实验结果如图3所示。在访问序列均匀分布场景中，DTP-DAA算法相比于传统的三副本机制和一致性哈希算法，在访问延迟方面并未显著改善。这或许源于系统操作对每个数据片的均匀访问，而擦除编码采用RS纠删码机制处理冷数据，导致冷数据片的副本数量较三副本机制减少了2，但计算时延增加。在冷数据所占比例为32%时，副本数量减少22%；而在冷数据所占比例为64%时，副本数量减少43%。由于系统中响应访问操作的副本数量有所减少，因此性能略微下降。然而，相较于纠错码算法，DTP-DAA算法在冷热数据不同占比情况下的访问时延都显著降低。

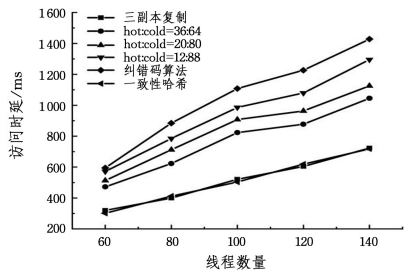

图3 均匀访问下访问时延随并发数量的变化

Fig. 3 Access delay changing with the number of concurrency under uniform access

在真实应用场景中，80%的访问请求发生在20%的数据中，不可能让数据访问操作均匀地访问每个数据文件。因此，本文模拟了非均匀分布的访问序列场景，让大多数访问操作集中在少数热门数据上。用平均访问时延作为衡量性能的指标之一，进行了多次实验，取平均值作为最终结果。

在模拟的真实数据访问场景中，DTP-DAA算法展现出卓越的系统性能，其对数据片进行温度分区，使得大多数访问操作集中在具有3个副本的热数据上，因而不会受到显著影响。如图4所示，只有在热数据占比为12%的情况下，DTP-DAA的访问时延有轻微增加。相比于其他算法，在符合80/20规则的真实数据访问场景中，DTP-DAA算法显著降低了系统的访问时延。

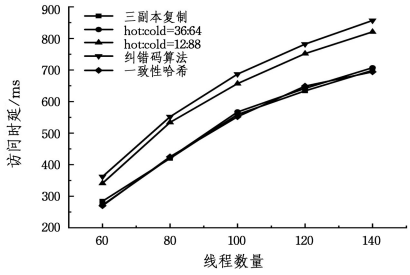

图4 真实场景下访问时延随并发数量的变化

Fig. 4 Access delay changes with the number of concurrencies in real scenarios

这一实验证实了DTP-DAA算法在真实数据访问场景中的优越性，尤其是符合80/20规则的数据分布情况，为其在实际应用中的性能表现提供了有力支持。相较于其他算法，DTP-DAA算法在保证系统性能的同时，更加高效地处理了真实场景中的数据访问需求，为分布式存储系统的性能提升提供了重要参考。

4.3 存储效率

在系统运行过程中，尽管实验证明数据的访问序列符合80/20的规则，但由于数据访问具有随机性，数据温度仍然在不断变化，部分数据可能在冷热数据之间转换。为了应对这种动态变化，DTP-DAA算法会动态调整冷热数据的温度阈值。在实验中，我们观察了当将存储利用率权重系数 w 设置为0.7，访问请求数量设为固定值时，系统整体存储效率的变化。根据式(9)，综合考虑存储效率和访问性能两个指标，动态调整数据的存储策略，以求在权重系数 w 下寻找存储利用率和访问性能的平衡，实现存储效率的最大化。

图5所示结果展示了DTP-DAA算法在动态环境下的自适应性和动态调整实时数据冷热比例方面的优越性，为其在面对随机性数据访问和温度变化的实际应用中的灵活性提供了强有力的支持。相较于传统的静态存储策略，DTP-DAA算法通过动态调整温度阈值，实现了对系统整体性能的更好平衡，为分布式存储系统提供了有效的性能优化方案。

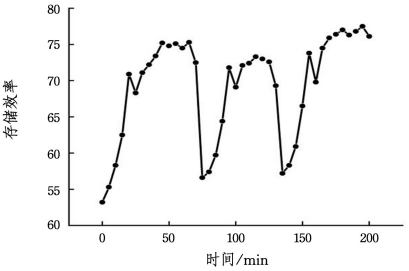

图5 存储效率随时间的变化

Fig. 5 Diagram of storage efficiency changes over time

结束语 存储利用率和访问时延是分布式数据存储的两个关键特征。文章讨论了基于擦除编码和副本复制的分布式混合存储策略的存储利用率和访问时延，并综合两者进行了

性能评估。根据数据之间访问频率的差异，使用数据温度来区分数据。通过数据温度的划分，本文提出了 DTP-DAA 算法，它结合了副本复制和擦除编码的特点来平衡性能和存储利用率。

仿真实验结果表明，该策略可以在不影响访问时延的前提下，有效提高系统的存储利用率；并且随着数据访问频率的变化，自适应地动态调整冷热数据的比例，保证系统存储效率的最大化。

下一步将对数据的分布进行进一步研究，即对多副本或冗余块的数据存储位置进行讨论，不仅要从存储节点的容量进行考虑，更要考虑存储节点的各方面负载，以保证整个系统的负载均衡和稳定运行。

参 考 文 献

[1] CHOU R A, KLIEWER J. Secure distributed storage: Optimal trade-off between storage rate and privacy leakage[C]// 2023 IEEE International Symposium on Information Theory(ISIT). IEEE, 2023: 1324-1329.

[2] NAEEM M, JAMAL T, DIAZ-MARTINEZ J, et al. Trends and future perspective challenges in big data[C]// Advances in Intelligent Data Analysis and Applications: Proceeding of the Sixth Euro-China Conference on Intelligent Data Analysis and Applications, 15-18 October 2019, Arad, Romania, Springer Singapore, 2022: 309-325.

[3] GHAZI M R, GANGODKAR D. Hadoop, MapReduce and HDFS: a developers perspective[J]. Procedia Computer Science, 2015, 48: 45-50.

[4] RYBINTSEV V O. Optimizing the parameters of the Lustre-file-system-based HPC system for reverse time migration[J]. The Journal of Supercomputing, 2020, 76: 536-548.

[5] WANG Y, YE M, HE Q, et al. Ceph storage system node selection method based on software-defined network and multi-attribute decision-making [J]. Journal of Computer Science, 2019, 42(2): 93-108.

[6] XIA Y, WANG Y. Fault-tolerant selection algorithm of nodes in Ceph storage system [J]. Journal of Guilin University of Electronic Science and Technology, 2022, 42(5): 384-390.

[7] BALAJI S B, KRISHNAN M N, VAJHA M, et al. Erasure coding for distributed storage: An overview[J]. Science China Information Sciences, 2018, 61: 1-45.

[8] CADAMBE V R, LYU S. Brief Announcement: CausalEC: A Causally Consistent Data Storage Algorithm based on Cross-Object Erasure Coding[C]// Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing. 2023: 374-377.

[9] SHIN D J, KIM J J. Cache-Based Matrix Technology for Efficient Write and Recovery in Erasure Coding Distributed File Systems[J]. Symmetry, 2023, 15(4): 872.

[10] DING Y, NIU C, WU F, et al. Federated submodel optimization for hot and cold data features[J]. Advances in Neural Information Processing Systems, 2022, 35: 1-13.

[11] LIU J, FAN X, WU Y, et al. HoaKV: High-Performance KV Store Based on the Hot-Awareness in Mixed Workloads [J]. Electronics, 2023, 12(15): 3227.

[12] YE X, ZHAI Z, LI X. Off-line Deduplication Method for Solid-State Disk Based on Hot and Cold Data[J]. Tehnički Vjesnik, 2020, 27(2): 368-373.

[13] CHEN H, ZHANG H, DONG M, et al. Efficient and available in-memory KV-store with hybrid erasure coding and replication [J]. ACM Transactions on Storage(TOS), 2017, 13(3): 1-30.

[14] HSU Y F, IRIE R, MURATA S, et al. A novel automated cloud storage tiering system through hot-cold data classification[C]// 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). IEEE, 2018: 492-499.

[15] LI Z, XIAO C. ER-Store: A Hybrid Storage Mechanism with Erasure Coding and Replication in Distributed Database Systems [J]. Scientific Programming, 2021, 2021: 1-13.

[16] CHANG C H, WENG J Y, YEN N Y, et al. Using the Ceph File System and RADOS Gateway to Construct an Integrated Shared Storage [J]. Human-centric Computing and Information Sciences, 2024, 14.

[17] MARUYAMA S, MORIYA S. Newton's Law of Cooling: Follow up and exploration [J]. International Journal of Heat and Mass Transfer, 2021, 164: 120544.

[18] PATIL D P, PATIL S A, PATIL K J. Newton's law of cooling by Emad-Falih transform[J]. International Journal of Advances in Engineering and Management, 2022, 4(6): 1515-1519.

[19] DA SILVA S L E F. Newton's cooling law in generalised statistical mechanics[J]. Physica A: Statistical Mechanics and its Applications, 2021, 565: 125539.

[20] LIN Y, SHEN H. Eafr: An energy-efficient adaptive file replication system in data-intensive clusters[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 28(4): 1017-1030.

[21] HE Q, ZHANG F, BIAN G, et al. File block multi-replica management technology in cloud storage [J]. Cluster Computing, 2023: 1-20.

[22] LLOPIS P, BLAS J G, ISAILA F, et al. Survey of energy-efficient and power-proportional storage systems[J]. The Computer Journal, 2014, 57(7): 1017-1032.

[23] QIU N, HU X, WANG P, et al. Research on data cluster storage optimization strategy of consistent hashing [J]. Information and Control, 2016, 45(6): 747-752.

[24] ZHANG H, LIU S, TANG D, et al. Low repair cost erasure coding in distributed storage systems [J]. Computer Applications, 2020, 40(10): 2942.

[25] ADAMOU A, EGLOFF M, PICCD D. Enabling Ontology-Based Data Access to Project Gutenberg[C]// Workshop on Humanities in the Semantic Web. 2020: 21-32.

[26] REHMAN A U, AGUIAR R L, BARRACA J P. Fault-tolerance in the scope of cloud computing[J]. IEEE Access, 2022, 10: 63422-63441.

FU Xiong, born in 1979, Ph.D, professor. His main research interests include cloud computing and distributed computing.

（责任编辑：柯颖）