



(12)发明专利申请

(10)申请公布号 CN 112052196 A  
(43)申请公布日 2020.12.08

(21)申请号 202010427273.X

(22)申请日 2020.05.19

(30)优先权数据

16/435,238 2019.06.07 US

(71)申请人 斯泰拉斯科技股份有限公司

地址 美国加利福尼亚州圣荷西北一街3833号10室

(72)发明人 维杰亚·雅库拉 西瓦·拉米尼尼  
文卡塔·巴努·普拉卡什·葛拉普迪

(74)专利代理机构 北京汇知杰知识产权代理有限公司 11587

代理人 李洁 董江虹

(51)Int.Cl.

G06F 12/0877(2016.01)

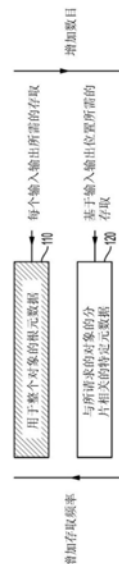
权利要求书3页 说明书8页 附图3页

(54)发明名称

高速缓存表项管理的方法、对象仓储和计算机可读介质

(57)摘要

本公开提供一种高速缓存表项管理的方法，所述方法包括：由高速缓存管理器确定多个高速缓存表项中的每一者的重要程度；由高速缓存管理器基于所确定的重要程度为高速缓存表项中的每一者指配元数据类型；由高速缓存管理器确定高速缓存表项中的每一者的存取频率；由高速缓存管理器基于所确定的高速缓存表项的存取频率产生热图；以及由高速缓存管理器基于高速缓存表项各自的元数据类型及各自的存取频率确定将高速缓存表项中的至少两者中的哪一者逐出。也提供一种分布式对象仓储和非暂时性计算机可读介质。



1. 一种高速缓存表项管理的方法,所述方法包括:
  - 由高速缓存管理器确定多个高速缓存表项中的每一者的重要程度;
  - 由所述高速缓存管理器基于所确定的所述重要程度为所述多个高速缓存表项中的每一者指配元数据类型;
  - 由所述高速缓存管理器确定所述多个高速缓存表项中的每一者的存取频率;
  - 由所述高速缓存管理器基于所确定的所述多个高速缓存表项的所述存取频率产生热图;以及
  - 由所述高速缓存管理器基于所述多个高速缓存表项各自的所述元数据类型及各自的所述存取频率确定将所述多个高速缓存表项中的至少两者中的哪一者逐出。
2. 根据权利要求1所述的方法,还包括:由所述高速缓存管理器使所述热图的与具有较低重要程度的高速缓存表项对应的部分比所述热图的与具有较高重要程度的高速缓存表项对应的其他部分衰变得快。
3. 根据权利要求1所述的方法,还包括:一旦所述热图的与所述多个高速缓存表项中的一者对应的部分达到零,便由所述高速缓存管理器逐出所述多个高速缓存表项中的所述一者。
4. 根据权利要求1所述的方法,还包括:由所述高速缓存管理器使热图的与对象的分片对应的部分比所述热图的与所述对象的整体对应的部分衰变得快。
5. 根据权利要求1所述的方法,还包括:由所述高速缓存管理器对高速缓存进行移行,且由所述高速缓存管理器将与低于参考重要程度的重要程度对应的所有高速缓存表项逐出。
6. 根据权利要求5所述的方法,还包括:一旦所使用的高速缓存资源的量达到参考程度,便由所述高速缓存管理器触发所述高速缓存的移行。
7. 根据权利要求1所述的方法,还包括:由所述高速缓存管理器将最高重要程度指配给代表对象的整体根元数据,且由所述高速缓存管理器将较低重要程度指配给与所述对象相关的用于分片元数据的其他元数据,所述分片元数据代表所述对象的分片。
8. 一种被配置成允许在与工作集的大小按比例缩放的同时进行高速缓存管理的分布式对象仓储,所述分布式对象仓储包括高速缓存,所述高速缓存被配置成由高速缓存管理器进行管理,其中所述高速缓存管理器被配置成:
  - 确定多个高速缓存表项中的每一者的重要程度;
  - 基于所确定的所述重要程度为所述多个高速缓存表项中的每一者指配元数据类型;
  - 确定所述多个高速缓存表项中的每一者的存取频率;
  - 基于所确定的所述多个高速缓存表项的所述存取频率产生热图;以及
  - 基于所述多个高速缓存表项各自的所述元数据类型及各自的所述存取频率确定将所述多个高速缓存表项中的至少两者中的哪一者逐出。
9. 根据权利要求8所述的分布式对象仓储,其中所述高速缓存管理器还被配置成使所述热图的与具有较低重要程度的高速缓存表项对应的部分比所述热图的与具有较高重要程度的高速缓存表项对应的其他部分衰变得快。
10. 根据权利要求8所述的分布式对象仓储,其中所述高速缓存管理器还被配置成一旦所述热图的与所述多个高速缓存表项中的一者对应的部分达到零,便逐出所述多个高速缓

存表项中的所述一者。

11. 根据权利要求8所述的分布式对象仓储,其中所述高速缓存管理器还被配置成使热图的与对象的分片对应的部分比所述热图的与所述对象的整体对应的部分衰变得快。

12. 根据权利要求8所述的分布式对象仓储,其中所述高速缓存管理器还被配置成对所述高速缓存进行移行,且将与低于参考重要程度的重要程度对应的所有高速缓存表项逐出。

13. 根据权利要求12所述的分布式对象仓储,其中所述高速缓存管理器还被配置成一旦所使用的高速缓存资源的量达到参考程度,便触发所述高速缓存管理器对所述高速缓存进行移行。

14. 根据权利要求8所述的分布式对象仓储,其中所述高速缓存管理器还被配置成将最高重要程度指配给代表对象的整体根元数据,且将较低重要程度指配给与所述对象相关的用于分片元数据的其他元数据,所述分片元数据代表所述对象的分片。

15. 一种实施在分布式对象仓储上的非暂时性计算机可读介质,所述非暂时性计算机可读介质具有计算机代码,所述计算机代码在处理器上执行时通过所述分布式对象仓储的高速缓存的高速缓存管理器实施一种高速缓存管理方法,所述方法包括:

由所述高速缓存管理器确定多个高速缓存表项中的每一者的重要程度;

由所述高速缓存管理器基于所确定的所述重要程度为所述多个高速缓存表项中的每一者指配元数据类型;

由所述高速缓存管理器确定所述多个高速缓存表项中的每一者的存取频率;

由所述高速缓存管理器基于所确定的所述多个高速缓存表项的所述存取频率产生热图;以及

由所述高速缓存管理器基于所述多个高速缓存表项各自的所述元数据类型及各自的所述存取频率确定将所述多个高速缓存表项中的至少两者中的哪一者逐出。

16. 根据权利要求15所述的非暂时性计算机可读介质,其中所述计算机代码在由所述处理器执行时还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法:由所述高速缓存管理器使所述热图的与具有较低重要程度的高速缓存表项对应的部分比所述热图的与具有较高重要程度的高速缓存表项对应的其他部分衰变得快。

17. 根据权利要求15所述的非暂时性计算机可读介质,其中所述计算机代码在由所述处理器执行时还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法:一旦所述热图的与所述多个高速缓存表项中的一者对应的部分达到零,便由所述高速缓存管理器逐出所述多个高速缓存表项中的所述一者。

18. 根据权利要求15所述的非暂时性计算机可读介质,其中所述计算机代码在由所述处理器执行时还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法:由所述高速缓存管理器使热图的与对象的分片对应的部分比所述热图的与所述对象的整体对应的部分衰变得快。

19. 根据权利要求15所述的非暂时性计算机可读介质,其中所述计算机代码在由所述处理器执行时还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法:由所述高速缓存管理器对所述高速缓存进行移行,且由所述高速缓存管理器将与低于参考重要程度的重要程度对应的所有高速缓存表项逐出。

20. 根据权利要求19所述的非暂时性计算机可读介质,其中所述计算机代码在由所述处理器执行时还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法:一旦所使用的高速缓存资源的量达到参考程度,便由所述高速缓存管理器触发所述高速缓存的移行。

## 高速缓存表项管理的方法、对象仓储和计算机可读介质

### 技术领域

[0001] 本公开的实施例的一个或多个方面总体上涉及高速缓存表项管理的方法,以及被配置成实行所述方法的一种分布式对象仓储(distributed objectstore)。

### 背景技术

[0002] 为了实现操作,包括存储器驱动器的网络的“分布式对象仓储”一般来说使用复杂的元数据来管理所存储的对象数据的不同方面。由元数据管理的几个方面可包括对象的增长或收缩、数据位置、与对象对应的数据管理策略(例如,复制副本是否被授权)、与分布式对象仓储的不可变性质对应的对象版本化(object-versioning)、数据仓储上给定对象标识(identity, ID)的对象数据的位置跟踪等。此外,对象可能不得不在分布式对象仓储内四处移动(例如,在分布式系统中发生局部故障的情况下)。

[0003] 分布式对象仓储通常会存储对应于同一数据对象的多个元数据对象,以有助于管理数据对象的不同方面。此外,分布式对象仓储可采用元数据的高速缓存来随着分布式对象仓储的大小按比例缩放而保持高性能,因为由于存取高速缓存一般来说比存取其中驻留有对象的媒体快得多的事实,因此通过简单地对频繁存取的元数据进行高速缓存可改善性能。

[0004] 然而,当按比例缩放时,高速缓存的有效性随着工作集(例如,数据对象及元数据对象的数目及大小)的增加而降低。也就是说,当“热”(例如,频繁存取的)项重复地相互逐出时,高速缓存可能变得不太有效,从而导致被称为“高速缓存抖动(cache thrashing)”的现象。随着工作集增加,高速缓存抖动可能会影响输入/输出(input/output, I/O)路径的性能,且因此对于大型分布式对象仓储(large scale distributed object store)来说,这可能是更为重要的问题。

### 发明内容

[0005] 本文阐述的实施例通过基于多种因素确定标记哪些高速缓存表项用于逐出以提供对高速缓存表项管理的改善。

[0006] 根据本公开的一个实施例,提供一种高速缓存表项管理的方法,所述方法包括:由高速缓存管理器确定多个高速缓存表项中的每一者的重要程度;由所述高速缓存管理器基于所确定的所述重要程度为所述高速缓存表项中的每一者指配元数据类型;由所述高速缓存管理器确定所述高速缓存表项中的每一者的存取频率;由所述高速缓存管理器基于所确定的所述高速缓存表项的所述存取频率产生热图;以及由所述高速缓存管理器基于所述高速缓存表项各自的所述元数据类型及各自的所述存取频率确定将所述高速缓存表项中的至少两者中的哪一者逐出。

[0007] 所述方法可还包括:由所述高速缓存管理器使所述热图的与所述高速缓存表项中具有较低重要程度的高速缓存表项对应的部分比所述热图的与所述高速缓存表项中具有较高重要程度的高速缓存表项对应的其他部分衰变得快。

[0008] 所述方法可还包括：一旦所述热图的与所述高速缓存表项中的一者对应的部分达到零，便由所述高速缓存管理器逐出所述高速缓存表项中的所述一者。

[0009] 所述方法可还包括：由所述高速缓存管理器使热图的与对象的分片对应的部分比所述热图的与所述对象的整体对应的部分衰变得快。

[0010] 所述方法可还包括：由所述高速缓存管理器对所述高速缓存进行移行，且由所述高速缓存管理器将所述高速缓存表项中与低于参考重要程度的重要程度对应的所有高速缓存表项逐出。

[0011] 所述方法可还包括：一旦所使用的高速缓存资源的量达到参考程度，便由所述高速缓存管理器触发所述高速缓存的所述移行。

[0012] 所述方法可还包括：由所述高速缓存管理器将最高重要程度指配给代表对象的整体根元数据 (root metadata)，且由所述高速缓存管理器将较低重要程度指配给与所述对象相关的用于分片元数据的其他元数据，所述分片元数据代表所述对象的分片。

[0013] 根据本公开的另一实施例，提供一种被配置成允许在与工作集的大小一起按比例缩放的同时进行高速缓存管理的分布式对象仓储，所述分布式对象仓储包括高速缓存，所述高速缓存被配置成由高速缓存管理器进行管理，其中所述高速缓存管理器被配置成：确定多个高速缓存表项中的每一者的重要程度；基于所确定的所述重要程度为所述高速缓存表项中的每一者指配元数据类型；确定所述高速缓存表项中的每一者的存取频率；基于所确定的所述高速缓存表项的所述存取频率产生热图；以及基于所述高速缓存表项各自的所述元数据类型及各自的所述存取频率确定将所述高速缓存表项中的至少两者中的哪一者逐出。

[0014] 所述高速缓存管理器可还被配置成使所述热图的与所述高速缓存表项中具有较低重要程度的高速缓存表项对应的部分比所述热图的与所述高速缓存表项中具有较高重要程度的高速缓存表项对应的其他部分衰变得快。

[0015] 所述高速缓存管理器可还被配置成一旦所述热图的与所述高速缓存表项中的一者对应的部分达到零，便逐出所述高速缓存表项中的所述一者。

[0016] 所述高速缓存管理器可还被配置成使热图的与对象的分片对应的部分比所述热图的与所述对象的整体对应的部分衰变得快。

[0017] 所述高速缓存管理器可还被配置成对所述高速缓存进行移行，且将所述高速缓存表项中与低于参考重要程度的重要程度对应的所有高速缓存表项逐出。

[0018] 所述高速缓存管理器可还被配置成一旦所使用的高速缓存资源的量达到参考程度，便触发所述高速缓存管理器对所述高速缓存进行移行。

[0019] 所述高速缓存管理器可还被配置成将最高重要程度指配给代表对象的整体根元数据，且将较低重要程度指配给与所述对象相关的用于分片元数据的其他元数据，所述分片元数据代表所述对象的分片。

[0020] 根据本公开的又一实施例，提供一种实施在用于显示器件的显示编解码器上的非暂时性计算机可读介质，所述非暂时性计算机可读介质实施在分布式对象仓储上，所述非暂时性计算机可读介质具有计算机代码，所述计算机代码在处理器上执行时通过所述分布式对象仓储的高速缓存的高速缓存管理器实施一种高速缓存管理方法，所述方法包括：由所述高速缓存管理器确定多个高速缓存表项中的每一者的重要程度；由所述高速缓存管理

器基于所确定的所述重要程度为所述高速缓存表项中的每一者指配元数据类型；由所述高速缓存管理器确定所述高速缓存表项中的每一者的存取频率；由所述高速缓存管理器基于所确定的所述高速缓存表项的所述存取频率产生热图；以及由所述高速缓存管理器基于所述高速缓存表项各自的所述元数据类型及各自的所述存取频率确定将所述高速缓存表项中的至少两者中的哪一者逐出。

[0021] 所述计算机代码在由所述处理器执行时可还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法：由所述高速缓存管理器使所述热图的与所述高速缓存表项中具有较低重要程度的高速缓存表项对应的部分比所述热图的与所述高速缓存表项中具有较高重要程度的高速缓存表项对应的其他部分衰变得快。

[0022] 所述计算机代码在由所述处理器执行时可还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法：一旦所述热图的与所述高速缓存表项中的一者对应的部分达到零，便由所述高速缓存管理器逐出所述高速缓存表项中的所述一者。

[0023] 所述计算机代码在由所述处理器执行时可还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法：由所述高速缓存管理器使热图的与对象的分片对应的部分比所述热图的与所述对象的整体对应的部分衰变得快。

[0024] 所述计算机代码在由所述处理器执行时可还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法：由所述高速缓存管理器对所述高速缓存进行移行，且由所述高速缓存管理器将所述高速缓存表项中与低于参考重要程度的重要程度对应的所有高速缓存表项逐出。

[0025] 所述计算机代码在由所述处理器执行时可还通过以下方式实施所述分布式对象仓储的所述高速缓存的所述高速缓存管理方法：一旦所使用的高速缓存资源的量达到参考程度，便由所述高速缓存管理器触发所述高速缓存的所述移行。

[0026] 因此，本公开实施例的自调整高速缓存(self-tuning cache)允许存储装置随着存储装置大小的增加而在性能方面按比例缩放。

## 附图说明

[0027] 结合附图，根据以下对实施例的说明，上述和/或其他方面将变得显而易见且更容易理解，在附图中：

[0028] 图1是绘示根据本公开的一个或多个实施例的分布式对象仓储的元数据的一般性质的方块图。

[0029] 图2A是绘示根据本公开的一个或多个实施例的用于对相应高速缓存表项进行高速缓存的高速缓存的两个不同分区的方块图。

[0030] 图2B是绘示根据本公开的一个或多个实施例，在从分区中的一个分区逐出高速缓存表项中的一些高速缓存表项之后，图2A的高速缓存的所述两个不同分区的方块图。

[0031] [符号的说明]

[0032] 110:根元数据；

[0033] 120:特定元数据/分片元数据；

[0034] 200:高速缓存；

[0035] 210:根对象元数据分区/分区；

[0036] 220:对象分片元数据分区/分区。

### 具体实施方式

[0037] 通过参照实施例的详细说明及附图,可更容易理解本发明概念的特征及其实现方法。在下文中,将参照附图更详细地阐述实施例。然而,所阐述的实施例可被实施成各种不同的形式,而不应被视为仅限于本文中所例示的实施例。确切来说,提供这些实施例是作为实例来使本公开将透彻及完整,并将向所属领域中的技术人员充分传达本发明概念的各个方面及特征。因此,可不再阐述对于所属领域中的普通技术人员完整地理解本发明概念的各个方面及特征而言并非必需的工艺、元件及技术。除非另外注明,否则在所有附图及本书面说明通篇中相同的参考编号表示相同的元件,且因此,可不再对其予以重复说明。此外,可能未示出与实施例的说明无关的部件,以使说明清楚。在图式中,为了清晰起见,可放大元件、层及区的相对大小。

[0038] 在本文中参照剖视图阐述各种实施例,所述剖视图为实施例和/或中间结构的示意性例示。因此,预期会因例如制造技术和/或容差而导致相对于例示形状的变化。此外,本文所公开的具体结构说明或功能说明仅是例示性的,目的在于阐述根据本公开概念的实施例。因此,本文所公开的实施例不应被视为仅限于各个区的特定例示形状,而是应包含由例如制造引起的形状偏差。举例来说,被例示为矩形的植入区通常应具有圆形特征或曲线特征和/或在其边缘处存在植入浓度的梯度而非从植入区到非植入区为二元改变。同样地,通过植入而形成的掩埋区可在所述掩埋区与在进行植入时所经过的表面之间的区中引起一些植入。因此,图式中所例示的区本质上为示意性的且其形状并非旨在例示器件的区的实际形状且并非旨在进行限制。另外,如所属领域中的技术人员将认识到的,所阐述的实施例可以各种不同的方式修改,所有这些都背离本公开的精神或范围。

[0039] 如上所述,随着工作集增加,由重复相互逐出的热数据项(hot data item)导致的高速缓存抖动会对高速缓存的性能产生负面影响。

[0040] 传统上,可按比例缩放的分布式对象仓储(scalable distributed object store)的性能取决于分布式对象仓储在存取对应于对象的元数据时的性能有多高。有效地存取元数据通常需要对对象的频繁存取的元数据进行高速缓存,以便可相对快速地存取元数据。然而,用于在按比例缩放的分布式对象仓储中对更频繁存取的元数据进行高速缓存的高速缓存通常面临挑战。

[0041] 举例来说,可按比例缩放的分布式对象仓储可期望地针对数十亿个对象进行按比例缩放,且还可期望地允许成千上万个使用不同应用的并发客户端,各种客户端中的每一者潜在地针对客户端的不同相应工作负载存取不同的数据集。然而,由于从具有不同要求的并发存取中产生的大的工作集,因此对应于存取要求的按比例缩放性(scalability)需求可能会使高速缓存的效率变低。

[0042] 此外,简单地增加高速缓存的大小仅仅会局部地缓解由于可被存取的大量对象以及由于对象中的每一者潜在地具有可被高速缓存的多个相关联的元数据对象而导致的问题。举例来说,I/O可能无法成功地实行查找操作以在高速缓存中找到元数据,且由于高速缓存的大的大小,因此必须对存储器存储装置进行存取来找到元数据。

[0043] 因此,尽管传统的高速缓存可能有助于解决分布式对象仓储的I/O路径的性能,但



是随着分布式对象仓储按比例缩放,从对更频繁存取的元数据进行高速缓存实现的好处可能会由于“高速缓存抖动”被抵消,“高速缓存抖动”可能在并发存取对高速缓存资源相互进行竞争时发生。本公开的实施例通过减少高速缓存抖动的不希望的影响来解决这个挑战。

[0044] 本公开的实施例提供一种自学习及自调整高速缓存,所述自学习及自调整高速缓存能够做出关于决定何时高速缓存元数据、高速缓存哪个元数据、逐出哪个元数据以及已被高速缓存的元数据应该保留多长时间的智能选择。

[0045] 本公开的实施例还提供一种高速缓存,所述高速缓存知道高速缓存表项的类型,且能够基于高速缓存表项的类型来判断重要程度。

[0046] 因此,通过向相应的高速缓存表项指派不同的重要性价值,高速缓存能够朝随着工作集增加而指配有高重要程度的高价值高速缓存表项偏置或使所述高价值高速缓存表项优先。

[0047] 图1是绘示根据本公开的一个或多个实施例的分布式对象仓储的元数据的一般性质的方块图。

[0048] 参照图1,各种类型的元数据可作为元数据对象存储在分布式对象仓储的高速缓存中。一种类型的元数据是可代表整个数据对象的根元数据110。根元数据110一般来说将被试图存取对象的某个方面的任何I/O线程存取。举例来说,即使I/O线程试图仅存取对象的分片,但I/O线程仍将存取根元数据110。

[0049] 相反,其他特定元数据(例如,分片元数据)120仅特定地与所请求的对象的分片相关,所述对象的整体由根元数据110代表。也就是说,如果I/O线程不试图存取整个数据对象,而是仅试图存取对象的分片,则I/O线程可存取对应的分片元数据120,以检索仅对应于所寻找的对象分片的信息。因此,分片元数据120代表对象的各个分片,且可用于分别管理特定于对应的对象分片的属性。

[0050] I/O线程对于对象的存取利用了对与作为整体的对象对应的根元数据110的存取,接着是根据所请求的I/O存取的位置及性质进行的对与对象分片相关联的分片元数据120的存取。因此,对应于分片元数据120的元数据对象通常会相对多,因为一般来说将会存在与由分片元数据120管理的单个数据对象对应的许多相应方面。相反,仅根元数据110的一个例子被用于管理作为整体的数据对象。

[0051] 然而,分片元数据120的每一单独例子一般来说远不如根元数据110被存取取得频繁,因为对数据对象的某个方面关注的各种I/O线程一般来说将与对象的一些数目的单独分片无关。也就是说,由于针对数据对象实行查找操作的所有I/O线程一般来说将存取根元数据110,因此分片元数据120的更多例子将不如根元数据110被单独存取取得频繁。

[0052] 本公开的实施例利用对象的元数据的固有性质来对高速缓存进行自调整以确保即使当工作集随着客户端连接及并发存取的增加而按比例缩放时,高速缓存也保持有效。为了实现这一点,高速缓存可使用高速缓存管理器来利用元数据的类型进行自调整。也就是说,本公开的实施例可通过利用由对象的各个方面产生的固有元数据特性来实现大型分布式对象仓储中的有效高速缓存。

[0053] 图2A是绘示根据本公开的一个或多个实施例的用于对相应的高速缓存表项进行高速缓存的高速缓存的两个不同分区的方块图,且图2B是绘示根据本公开的一个或多个实施例,在从分区中的一个分区逐出高速缓存表项中的一些高速缓存表项之后,图2A的高速

缓存的所述两个不同分区的方块图。

[0054] 参照图2A,基于与要存储在其中的高速缓存表项的元数据对应的元数据类型来对高速缓存200进行分区。应注意,图2A及图2B中例示的两个分区仅仅是用于例示本公开的方面的实例。在其他实施例中,可存在许多这样的分区(即,多于两个)。在本公开的实施方式中,可存在几个分区用于采用所公开的实施例的方面来实现有效的高速缓存。在本实施例中,高速缓存200包括:根对象元数据分区210,用于处理指配有与根元数据110对应的元数据类型的高速缓存表项;以及对象分片元数据分区220,用于处理指配有与仅对应于对象的分片的分片元数据120对应的元数据类型的高速缓存表项。

[0055] 与对象分片对应的分片元数据120的存取模式使得只有当I/O存取的位置落于相应的分片内时才需要进行存取。因此,随着I/O活动在关于对象的范围内从一个分片移动到另一分片,I/O存取不再需要前一个对象分片的分片元数据120。然而,由于对对象进行的每个I/O存取都需要独立于对象中的I/O位置来存取根元数据110,因此代表整个对象的根元数据110的存取模式是不同的。因此,如前所述,根元数据110一般来说将比分片元数据120被存取得频繁。

[0056] 根据本实施例,作为实例,高速缓存管理器将两种不同的“元数据类型”中的一者指配给高速缓存表项。然而,应注意,在本公开的其他实施例中,可使用更多数目的相应类型的元数据(例如,3个或更多个)。因此,在本实例中,对应于根元数据110的高速缓存表项指配有第一元数据类型(例如,“类型0”),且对应于分片元数据120的高速缓存表项指配有第二元数据类型(例如,“类型1”)。

[0057] 通过将不同的元数据类型指配给不同种类的元数据,能够形成层次(hierarchy)。因此,高速缓存200可进行自调整以通过分配额外的资源来实现与指配有高重要程度的高速缓存表项对应的高价值数据的高速缓存,从而在工作集增加时保持有效。由于用于自调整的计算是基于恒定大小输入(例如,基于元数据的类型),而不是基于可能变得非常大的工作集的大小,因此自调整机制随着对象的工作集中对象的数目而按比例缩放。

[0058] 此外,如以下进一步论述,可将对于每一分区210、220的逐出策略调整成与分区相关联的特定元数据类型固有的或一般来说对应的预期热度。这些及其他逐出操作可通过以低优先级运行的专用后台线程来实现,以确保随着工作集增加,高速缓存保持有效。

[0059] 参照图2B,随着高速缓存200的资源被消耗,高速缓存200可确定应该逐出的高速缓存表项中的一些高速缓存表项。高速缓存200可接着开始对各种高速缓存表项进行移行,以便它可确定要将哪些高速缓存表项逐出。

[0060] 举例来说,高速缓存200可使用参考容量程度(例如,90%满)来确定何时开始将其中高速缓存的高速缓存表项中的一些高速缓存表项逐出。然而,应注意,根据本公开的实施例,可使用不同的容量程度及不同的逐出触发条件(例如,高速缓存可在管理员的指导下开始逐出)。

[0061] 因此,高速缓存200可基于指配给高速缓存表项的元数据类型来确定将哪些高速缓存表项逐出来释放额外的资源。举例来说,高速缓存200可决定逐出特定元数据类型、或者特定元数据类型或“较低”(例如,特定重要程度或较低)的高速缓存表项中的每一者。另外,高速缓存200可决定将特定元数据类型中的每一高速缓存表项逐出,直到发生触发(例如,直到达到高速缓存200的给定容量程度,例如80%满,或者直到给定百分比的容量被逐

出,例如10%)。

[0062] 在本实例中,高速缓存200可确定指配有“类型0”元数据类型的高速缓存表项一般来说比指配有“类型1”元数据类型的高速缓存表项重要。即使分别指配有“类型0”元数据类型及“类型1”元数据类型的两个特定高速缓存表项以近似相等的频率被存取,这也可能是真的。因此,高速缓存200可有效地将其自身朝更高价值元数据偏置。

[0063] 因此,通过比较图2A与图2B可看出,高速缓存200可从对象分片元数据分区220逐出包括分片元数据120的“类型1”高速缓存表项中的一些“类型1”高速缓存表项,从而释放可用于在根元数据分区210中产生包括根元数据110的附加高速缓存表项的资源。

[0064] 此外,如上所述,包括高速缓存200的分布式对象仓储可通过在适当时对低价值高速缓存表项进行高速缓存来有效地处理I/O热点,而不管指配给高速缓存表项的元数据类型如何。这可通过考虑附加参数来实现。

[0065] 举例来说,高速缓存200还可采用高速缓存表项的热图来保持跟踪对高速缓存表项的存取频率。这使得本公开的实施例能够学习如何基于元数据对象被存取的频率来在相同元数据类型的不同元数据对象中进行区分。

[0066] 高速缓存管理器可维持被高速缓存的每一元数据对象的热图。高速缓存管理器可接着针对对于对应的元数据对象的每次存取增加热图的一部分。然而,热图的衰变速率可基于指配给元数据的元数据类型来决定。

[0067] 举例来说,热图的与指配有低价值的分片元数据对应的部分可能衰变得更快,而热图的与更重要的根元数据对应的部分可能衰变得更慢。也就是说,尽管可能发生热图衰变,但是可基于高速缓存表项的元数据类型来调节这种衰变。举例来说,根元数据可能更热,因为对对象的任何分片进行存取的所有I/O线程都将存取根元数据。如热图上所呈现,一旦对应于元数据对象的对象温度达到零,便可从高速缓存逐出元数据对象。

[0068] 应注意,低价值的元数据(例如,指配有与较低重要程度相关联的元数据类型的元数据)可被视为具有高价值。低价值的元数据被视为具有高价值的实例是突然被频繁存取的新闻故事。即使当与新闻故事相关联的元数据与低价值相关联时,其存取频率也会使高速缓存管理器将元数据视为具有更高的价值。

[0069] 举例来说,通过考虑热图,如果元数据对象非常热,则尽管与低价值相关联,但高速缓存管理器可决定在元数据否则将被逐出时将元数据保持在高速缓存中。这确保了当工作集的大小增加时,高速缓存会自调整以区分更有价值的元数据与较低价值的元数据,从而通过仅高速缓存高价值的元数据来保持有效性。同时,高速缓存对I/O热点保持有效,因为它将对用于作为此类热点的目标的对象分片的较低价值的元数据进行高速缓存。

[0070] 因此,本公开的实施例既使用元数据的元数据类型以具有元数据类别的层次,且也使用热图以保持对本地热点的敏感。

[0071] 如上所述,本公开的实施例为高速缓存提供可按比例缩放的自调整机制,提供随着工作集的大小而按比例缩放的可按比例缩放的高速缓存性能,且通过随着工作集的大小的增加而管理I/O热点的性能来保持有效。因此,本公开的实施例通过确保I/O路径性能不会随着工作集的大小按比例缩放而劣化,且通过允许I/O路径性能即使在工作集的大小增加的情况下也能处理I/O热点,从而形成高速缓存的高效自调整机制,来提供对分布式对象仓储技术的改善。

[0072] 在说明中,出于阐释目的,阐述各种具体细节来提供对各种实施例的透彻理解。然而,显而易见的是,可不使用这些具体细节或者可使用一种或多种等效配置来实践各种实施例。在其他情况下,以方块图形式示出众所周知的结构及器件以避免不必要地使各种实施例模糊不清。

[0073] 本文所用术语仅是出于阐述特定实施例的目的而并非旨在对本发明进行限制。除非上下文清楚地另外指明,否则本文所用单数形式“一(a及an)”旨在也包括复数形式。还应理解,当在本说明书中使用用语“包括(comprises、comprising)”、“具有(have、having)”及“包含(includes、including)”时,是指明所陈述特征、整数、步骤、操作、元件和/或组件的存在,但不排除一个或多个其他特征、整数、步骤、操作、元件、组件和/或其群组的存在或添加。本文所用用语“和/或”包括相关联的列出项中的一个或多个项的任意及所有组合。

[0074] 当某一实施例可被以不同方式实施时,特定工艺次序可与所阐述的次序不同地实行。举例来说,两个连续阐述的工艺可实质上同时实行或以与所阐述的次序相反的次序实行。

[0075] 根据本文所述本公开的实施例的电子器件或电气器件和/或任何其他相关器件或组件可利用任何适当的硬件、固件(例如,专用集成电路(application-specific integrated circuit))、软件、或软件、固件、及硬件的组合来实施。举例来说,可将这些器件的各种组件形成在一个集成电路(integrated circuit, IC)芯片上或单独的集成电路芯片上。此外,可将这些器件的各种组件实施在柔性印刷电路膜(flexible printed circuit film)、载带封装(tape carrier package, TCP)、印刷电路板(printed circuit board, PCB)上、或形成在一个衬底上。此外,这些器件的各种组件可为在一个或多个计算器件中由一个或多个处理器运行、执行计算机程序指令并与用于实行本文所述各种功能性的其他系统组件进行交互的过程或线程。计算机程序指令存储在可在使用例如(举例来说)随机存取存储器(random access memory, RAM)等标准存储器器件的计算器件中实施的存储器中。计算机程序指令也可存储在例如(举例来说)压缩盘只读存储器(compact disc read only memory, CD-ROM)、闪存驱动器(flash drive)或类似元件等其他非暂时性计算机可读媒体中。另外,所属领域中的技术人员应知,在不背离本公开实施例的精神及范围的条件下,可将各种计算器件的功能性组合或整合成单一的计算器件,或者可使一特定计算器件的功能性跨越一个或多个其他计算器件分布。

[0076] 除非另外定义,否则本文所用所有用语(包括技术及科学用语)的含义均与本发明概念所属领域中的普通技术人员所通常理解的含义相同。还应理解,用语(例如在常用词典中所定义用语)应被解释为具有与其在相关技术的上下文和/或本说明书中的含义一致的含义,且除非在本文中明确定义,否则不应将其解释为具有理想化或过于正式的意义。

[0077] 本文中已公开了实施例,且尽管采用了特定用语,但是它们仅用于一般意义及阐述性意义或将以一般意义及阐述性意义解释,而并非用于限制目的。在一些情况下,除非例如另外指明,否则对于所属领域中的普通技术人员来说,在提交本申请时,结合特定实施例阐述的特征、特性和/或元件可单独使用,或者与结合其他实施例阐述的特征、特性和/或元件结合使用。因此,所属领域中的技术人员应理解,在不背离如以上权利要求书中所提出的本公开的精神及范围的条件下可作出形式及细节上的各种改变,其中权利要求的功能等效形式包含在本文中。

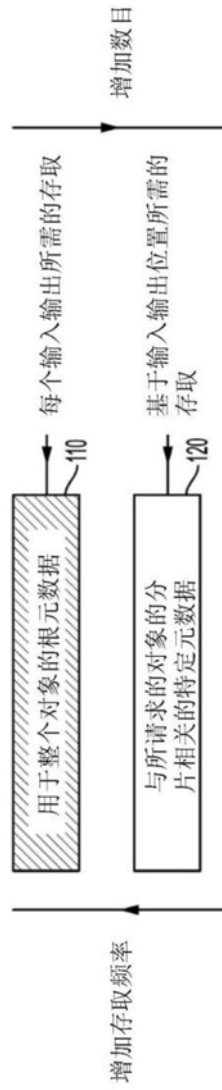


图1

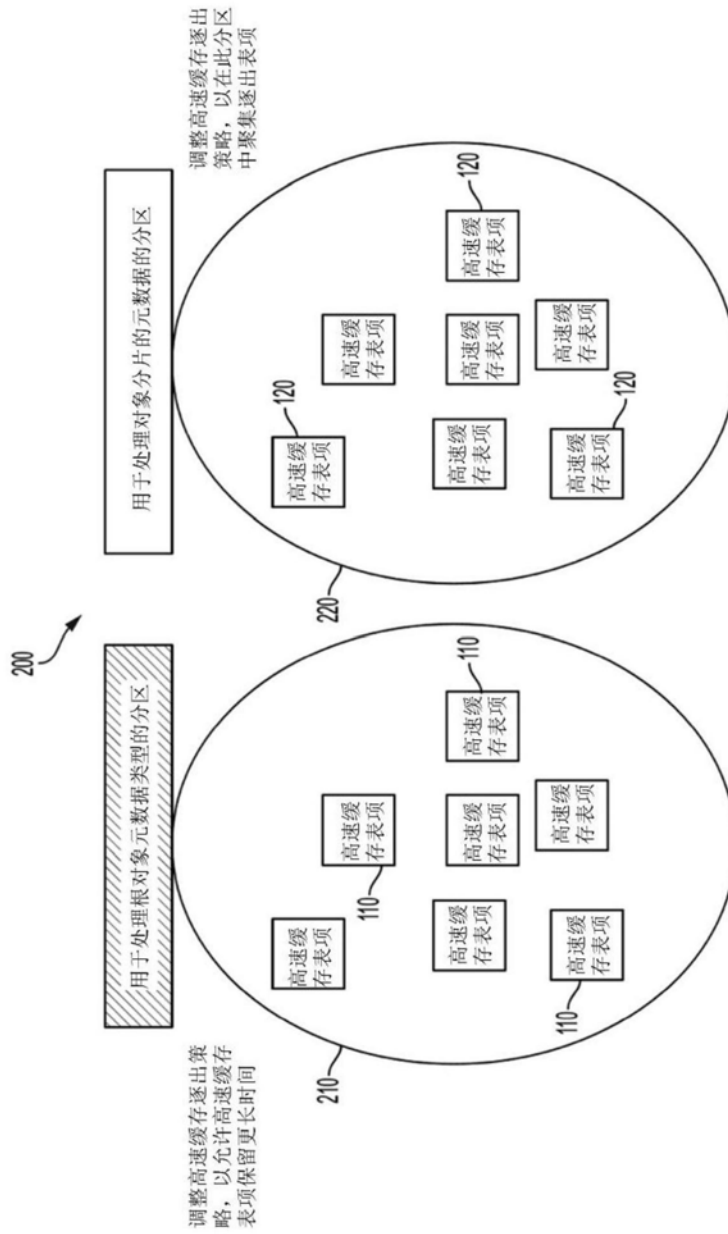


图2A

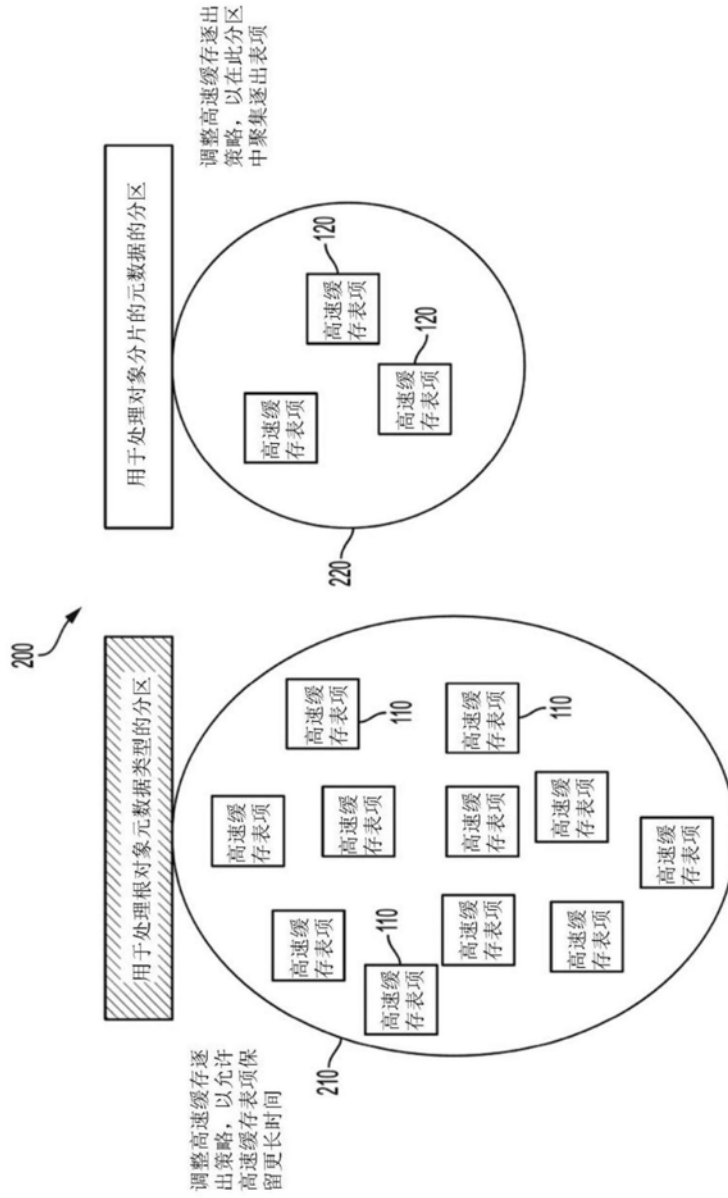


图2B