

数据缓存及读取方法、数据存取系统

5 本申请要求于 2022 年 01 月 13 日提交中国专利局、申请号为 202210037805.8、申请名称为“数据缓存及读取方法、数据存取系统”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

本公开涉及数据缓存技术，特别涉及数据的缓存及读取。

10

背景技术

近年来，在大数据分析场景中，云上对象存储服务被不断地使用。当大规模分布式分析服务读取云上对象存储服务中的数据时，大量热点数据占用了对象存储服务提供的大部分带宽，导致集群整体利用率不高。

15 因此，利用集群空闲存储空间建立分布式缓存系统来提高数据读取的效率，以加速对象存储服务上的大数据分析，从成本和效率上看都是一种理想的方案。

然而，大数据分析的强一致性、高性能和高可用性的要求对数据缓存系统的整体设计、性能指标等各方面都带来了巨大的挑战。

图 1 示意性地示出了现有一种缓存系统 Alluxio 的架构。

20 Alluxio 是计算服务和底层文件(对象)存储之间的一个抽象文件缓存系统，主要提供对底层文件存储的缓存能力。

当计算服务要使用 Alluxio 读取对象存储服务中的文件时，需要先通过 Alluxio 服务获取文件元数据。

25 在步骤 S1，计算服务的客户端基于数据缓存服务维护的文件名，例如“alluxio:///mount_path/directory/file1”，向数据缓存服务的主机发送数据请求。

Alluxio 内部主要通过底层文件的<文件名+更新时间>来建立对象存储服务系统的文件和数据缓存服务系统的文件之间的索引关系。即，如图 1 所示，对应存储数据缓存服务维护的文件名和其所对应的对象存储服务系统的文件的文件名及文件更新时间。

30 在步骤 S2，数据缓存服务系统的主机向对象存储服务系统请求对应底层文件(“oss://bucket/directory/file1”)的更新时间，并在步骤 S3 接收对象存储服务系统返回的底层文件更新时间信息。

然后，主机对从对象存储服务系统接收到的更新时间和数据缓存服务系统记录的文件更新时间进行对比，判断更新时间是否一致，从而判断数据缓存服务系统缓存的文件与对象存储服务系统存储的数据是否一致。

35 无论是否存在与所请求的底层文件一致的缓存文件，都需要再次连接指定的工作机

(Alluxio Worker) 节点。工作机将数据缓存为以默认 64M 的尺寸进行存储, 然后再响应计算服务的读取请求。

主机在步骤 S4 向计算服务的客户端通知所指定的工作机。

在步骤 S5, 客户端向工作机请求获取数据。

- 5 在存在与所请求的底层文件一致的缓存文件的情况下, 工作机在步骤 S8 直接向客户端发送缓存数据 (或缓存文件)。

在不存在与所请求的底层文件一致的缓存文件的情况下, 工作机在步骤 S6 向对象存储服务系统请求该底层文件的数据, 并在步骤 S7 接收来自对象存储服务系统的数据, 将其缓存。然后在步骤 S8, 向客户端发送缓存数据 (或缓存文件)。

- 10 然而, 在该现有技术方案中, 缓存的有效性依赖于底层存储记录的所请求文件的“更新时间”。文件系统的更新时间通常为秒级。而对于 put、append 等毫秒级的更新操作, 极有可能发生在 1 秒以内。在发生了这样的毫秒级更新操作, 而文件系统尚未更新时, 如果读取数据就会存在一致性问题, 造成所读取数据不正确。

- 而且, 这种方案强依赖于 Alluxio 服务节点, 降低了系统的高可用指标。例如, 当 Alluxio 15 服务出现异常时, 整个存储系统也将无法访问。

另外, 底层对象存储服务系统需要通过 Alluxio 数据缓存服务系统来进行中间元数据映射, 服务无法使用原生的访问路径, 需要使用 Alluxio 修改的特殊路径访问, 不能支持透明缓存。

因此, 仍然需要一种数据缓存及读取方案, 以解决上述至少一方面的问题。

20

发明内容

本公开要解决的一个技术问题是提供一种数据缓存和读取方案, 其能够方便快捷而又实时准确地验证数据缓存系统所缓存的数据与数据存储系统中存储的相应数据之间的一致性。

- 25 根据本公开的第一个方面, 提供了一种数据缓存方法, 用于数据缓存系统, 包括: 接收客户端发来的针对要获取的目标数据的请求, 请求包括数据存储系统所存储的目标数据的元数据; 以及基于目标数据的元数据, 判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存目标数据。

- 30 可选地, 该方法还可以包括: 在判定缓存有与数据存储系统的目标数据一致的缓存目标数据的情况下, 向客户端发送缓存目标数据的位置信息; 以及/或者在判定未缓存与数据存储系统的目标数据一致的缓存目标数据的情况下, 向客户端发送命中失败消息。

可选地, 在判定未缓存与数据存储系统的目标数据一致的缓存目标数据的情况下, 该方法还可以包括: 从客户端接收客户端从数据存储系统获取的目标数据; 以及缓存目标数据。

- 35 可选地, 判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存

目标数据的步骤可以包括：基于目标数据的识别信息，判断是否缓存有对应于目标数据的缓存目标数据；以及在判定缓存有对应于目标数据的缓存目标数据的情况下，基于缓存目标数据的元数据和目标数据的元数据，判断缓存目标数据与目标数据是否一致。

可选地，目标数据的元数据是客户端从数据存储系统获取的。

- 5 可选地，目标数据的元数据包括目标数据的更新时间、循环冗余校验值、数据尺寸中的至少一项。

根据本公开的第二个方面，提供了一种数据读取方法，包括：向数据存储系统发送针对目标数据的第一请求；从数据存储系统接收目标数据的元数据；以及向数据缓存系统发送针对目标数据的第二请求，第二请求包括元数据。

- 10 可选地，该方法还可以包括：在数据缓存系统基于元数据判定缓存有与数据存储系统的目标数据一致的数据的情况下，从数据缓存系统获取目标数据；以及/或者在数据缓存系统基于元数据判定未缓存与数据存储系统的目标数据一致的数据的情况下，从数据存储系统获取目标数据。

- 15 可选地，该方法还可以包括：在从数据存储系统获取目标数据之后，向数据缓存系统发送所获取的目标数据。

根据本公开的第三个方面，提供了一种数据缓存装置，包括：请求接收装置，用于接收客户端发来的针对要获取的目标数据的请求，请求包括数据存储系统所存储的目标数据的元数据；以及一致性判断装置，基于目标数据的元数据，判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存目标数据。

- 20 根据本公开的第四个方面，提供了一种数据读取装置，包括：第一请求装置，用于向数据存储系统发送针对目标数据的第一请求；元数据接收装置，用于从数据存储系统接收目标数据的元数据；以及第二请求装置，用于向数据缓存系统发送针对目标数据的第二请求，第二请求包括元数据。

- 25 根据本公开的第五个方面，提供了一种数据存取系统，包括：数据存储系统、数据缓存系统和客户端，其中，客户端向数据存储系统发送针对目标数据的第一请求，数据存储系统响应于第一请求，向客户端返回目标数据的元数据，客户端向数据缓存系统发送针对目标数据的第二请求，第二请求包括元数据，数据缓存系统基于目标数据的元数据，判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存目标数据，在判定缓存有与数据存储系统的目标数据一致的缓存目标数据的情况下，数据缓存系统向客户端发送缓存目标数据的位置信息，客户端从数据缓存系统获取目标数据，在判定未缓存与数据存储系统的目标数据一致的缓存目标数据的情况下，数据缓存系统向客户端发送命中失败消息，客户端从数据存储系统获取目标数据，并将所获取的目标数据发送给数据缓存系统，数据缓存系统缓存目标数据。
- 30

- 35 根据本公开的第六个方面，提供了一种计算设备，包括：处理器；以及存储器，其上存储有可执行代码，当可执行代码被处理器执行时，使处理器执行如上述第一方面所述的

方法。

根据本公开的第七个方面，提供了一种计算机程序产品，包括可执行代码，当所述可执行代码被电子设备的处理器执行时，使所述处理器执行如上述第一方面所述的方法。

根据本公开的第八个方面，提供了一种非暂时性机器可读存储介质，其上存储有可执行代码，当可执行代码被电子设备的处理器执行时，使处理器执行如上述第一方面所述的方法。

由此，通过由客户端从数据存储系统获取目标数据的元数据，作为缓存目标数据一致性的判断依据，能够方便快捷而又实时准确地判断数据缓存系统所缓存的数据与数据存储系统中存储的相应数据是否一致，从而最终确保客户端所获取的数据的准确性。

10

附图说明

通过结合附图对本公开示例性实施方式进行更详细的描述，本公开的上述以及其它目的、特征和优势将变得更加明显，其中，在本公开示例性实施方式中，相同的参考标号通常代表相同部件。

15

图 1 示意性地示出了现有一种缓存系统 Alluxio 的架构。

图 2 示意性地示出了根据本公开的数据存取系统的架构。

图 3 是根据本公开可用于计算服务客户端的数据读取装置的示意性框图。

图 4 是根据本公开可用于数据缓存系统的数据缓存装置的示意性框图。

图 5 是根据本公开的数据缓存和读取方法的示意性流程图。

20

图 6 示出了根据本发明一实施例可用于实现上述数据读取及缓存方法的计算设备的结构示意图。

具体实施方式

下面将参照附图更详细地描述本公开的优选实施方式。虽然附图中显示了本公开的优选实施方式，然而应该理解，可以以各种形式实现本公开而不应被这里阐述的实施方式所限制。相反，提供这些实施方式是为了使本公开更加透彻和完整，并且能够将本公开的范围完整地传达给本领域的技术人员。

本公开针对例如大规模分布式系统中大数据分析的场景，提供了一种基于计算服务客户端的按需构建数据缓存流程，基本保持了原有的访问方式和数据高可用性，从而进行透明加速。同时，解决了分布式缓存的一致性判断难题和缓存数据更新问题。

30

图 2 示意性地示出了根据本公开的数据存取系统的架构。

如图 2 所示，根据本公开的数据存取系统包括数据存储系统、数据缓存系统以及计算服务中的客户端。

图 3 是根据本公开可用于计算服务客户端的数据读取装置的示意性框图。

35

如图 3 所示，该数据读取装置 100 可以包括第一请求装置 110、元数据接收装置 120

以及第二请求装置 130。在一些实施例中，数据读取装置 100 还可以进一步包括第一数据获取装置 155、第二数据获取装置 180 以及目标数据发送装置 190。

图 4 是根据本公开可用于数据缓存系统的数据缓存装置的示意性框图。

如图 4 所示，该数据缓存装置 200 可以包括请求接收装置 210、一致性判断装置 220、
5 第一信息发送装置 230 以及第二信息发送装置 240。

下文中将进一步参考图 2 和图 5 所示出的数据存取方法流程图详细描述这些装置的功能。

图 5 是根据本公开的数据缓存和读取方法的示意性流程图。

如图 5 所示，在步骤 S110，计算服务的客户端，例如通过第一请求装置 110，向数据
10 存储系统发送针对目标数据的第一请求。

这里，计算服务通过客户端发起的第一请求向数据存储系统请求获取目标数据的元数据。目标数据的元数据例如可以包括目标数据（数据文件）的更新时间（update_time）、文件 CRC 校验值（checksum）、文件尺寸（length）等。

第一请求可以直接基于数据存储系统所维护的文件名称，例如
15 “oss://bucket/directory/file1”，来请求获取目标数据的元数据。

在步骤 S310，数据存储系统接收第一请求，并在步骤 S320 向客户端发送所请求的目标数据（数据文件，例如“oss://bucket/directory/file1”）的元数据。

在步骤 S120，客户端，例如通过元数据接收装置 120，从数据存储系统接收目标数据的元数据。

20 在步骤 S130，客户端，例如通过第二请求装置 130，向数据缓存系统发送针对目标数据的第二请求，第二请求包括元数据。由此，通过第二请求，客户端向数据缓存系统请求获取目标数据的缓存位置。

在步骤 S210，数据缓存系统的例如名称空间，例如通过请求接收装置 210，接收客户端发来的针对要获取的目标数据的第一请求。如上所述，第一请求包括数据存储系统所存
25 储的目标数据的元数据。

在步骤 S220，数据缓存系统的例如名称空间，例如通过一致性判断装置 220，基于目标数据的元数据，判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存目标数据。

30 这里，可以首先基于目标数据的识别信息，例如文件名称，判断是否缓存有对应于目标数据的缓存目标数据。

然后，在判定缓存有对应于目标数据的缓存目标数据的情况下，基于缓存目标数据的元数据和目标数据的元数据，判断缓存目标数据与目标数据是否一致。在缓存目标数据的元数据和目标数据的元数据一致的情况下，判定缓存目标数据与目标数据一致，否则判定缓存目标数据与目标数据不一致。

35 如图 2 所示，数据缓存系统可以维护目标数据（文件）与先前缓存该目标数据时所获

取的元数据之间的对应关系，例如可以对应地记录每个所缓存文件例如“oss://bucket/directory/file1”等的文件名称和元数据如循环冗余校验值、文件尺寸、更新时间等，以用于进行上述一致性判断。

5 这里，数据缓存系统可以通过文件名称（file_name）判断是否存在期望的目标数据对应的缓存目标数据，并根据更新时间（update_time）、文件 CRC 校验值（checksum）、文件尺寸（length）等元数据来精确地判断当前数据存储系统存储的目标数据和数据缓存系统内的缓存目标数据是否一致。

10 在步骤 S220 判定缓存有与数据存储系统的目标数据一致的缓存目标数据（命中）的情况下，在步骤 S230，数据缓存系统的例如名称空间，例如通过第一信息发送装置 230，向客户端发送缓存目标数据的位置信息。例如，这里可以向客户端返回缓存该缓存目标数据的缓存节点的信息，作为缓存目标数据的位置信息。

这种情况下，客户端可以从数据缓存系统获取目标数据。

具体说来，例如可以在步骤 S140，客户端从数据缓存系统的例如名称空间接收缓存目标数据的位置信息，例如缓存该缓存目标数据的缓存节点的信息。

15 在步骤 S150，客户端向缓存目标数据所在的位置（缓存节点）请求缓存目标数据。

在步骤 S250，数据缓存系统的缓存节点接收客户端发来的缓存目标数据请求，并在步骤 S260，缓存节点向计算服务的客户端发送缓存目标数据。

在步骤 S155，客户端，例如通过第一数据获取装置 155，从数据缓存系统的缓存节点接收缓存目标数据。由此，所获取的缓存目标数据可以直接用于计算服务。

20 在步骤 S220 判定未缓存与数据存储系统的目标数据一致的缓存目标数据（未命中）的情况下，在步骤 S240，数据缓存系统的例如名称空间，例如通过第二信息发送装置 240，向客户端发送命中失败消息。

在步骤 S160，客户端从数据缓存系统的例如名称空间接收命中失败消息。

这种情况下，客户端可以直接从数据存储系统获取目标数据。

25 具体说来，例如可以在步骤 S170 向数据存储系统发送目标数据请求，请求获取目标数据。

在步骤 S330，数据存储系统接收来自客户端的目标数据请求，并在步骤 S340 向客户端发送所请求的目标数据。

30 在步骤 S180，客户端，例如通过第二数据获取装置 180，接收数据存储系统发来的目标数据。

然后，在从数据存储系统获取目标数据之后，客户端可以在步骤 S190，例如通过目标数据发送装置 190，进一步向数据缓存系统发送所获取的目标数据。由此，所获取的目标数据可以直接用于计算服务。

在步骤 S290，数据缓存系统接收客户端发来的目标数据，并将其缓存到缓存节点。

35 另外，数据缓存系统的名称空间还可以与所缓存的目标数据的文件名称关联地记录客

户端在步骤 S130 发来的目标数据的元数据，作为之后判断目标数据一致性的依据。

这样，可以直接基于客户端从数据存储系统获取的目标数据的元数据来判断数据缓存系统所缓存的目标数据是否与数据存储系统所存储的目标数据一致。存在一致的目标数据的情况下，客户端从数据缓存系统的缓存节点获取缓存目标数据。不存在一致的目标数据的情况下，客户端直接从数据存储系统获取目标数据，并将所获取的目标数据发送给数据缓存系统进行缓存，以备之后读取使用。

另外，在步骤 S220 判定未缓存与数据存储系统的目标数据一致的缓存目标数据（未命中）的情况下，客户端在步骤 S170 向数据存储系统请求读取目标数据时，可以根据步骤 S120 获取的元数据中的文件尺寸（length）对目标数据（例如文件）进行逻辑分片（LogicBlock），即按照预设的逻辑数据块尺寸（Block size）将目标数据（文件）逻辑切分为多个逻辑块，例如如图 2 所示的逻辑块 1、逻辑块 2、逻辑块 3、……、逻辑块 n，n 为正整数。

这样，客户端可以按照逻辑分块在数据存储系统的指定位置直接读取相应数据。

具体说来，客户端在数据存储系统中读取数据时，可以按照读取偏移量（position），首先确定要读取目标数据（文件）的哪一个逻辑分片（逻辑块）。然后，可以根据步骤 S120 返回的元数据，直接去指定位置（例如文件起始位置+偏移量对应的位置）读取数据，而无需再次查询元数据服务。

另一方面，数据缓存系统从客户端接收到目标数据的一个个逻辑块后，也可以分别按照逻辑块来进行缓存，将一个逻辑块缓存在同一个缓存节点上，而同一个文件（目标数据）的不同逻辑块则可以缓存在不同缓存节点上。数据缓存系统可以维护该文件（目标数据）的逻辑块及缓存节点的对应关系表，例如如图 2 所示，对于文件“oss://bucket/directory/file1”，记录多个逻辑块及缓存节点对应关系，<逻辑块 1，节点 1>、<逻辑块 2，节点 1>……，由此，可以建立文件的各个逻辑块与缓存节点之间的索引关系。

这样，后续客户端再次请求该目标数据时，在基于元数据判定数据缓存系统的缓存目标数据与数据存储系统存储的目标数据一致的情况下，也可以按照客户端当前请求的数据在目标数据中的偏移量确定要读取目标数据（文件）的哪一个逻辑块，并直接到对应的缓存节点进行读取。

至此，本公开提出了一种高效的数据缓存方案，能够在读取数据时自动检查数据存储系统中的原始数据是否发生了变更，高效地对数据进行缓存更新，必要时还可以进行高效地分块缓存。

另外，针对云上数据存储的特点，还可以做到数据缓存的全生命周期管理，在发现所缓存数据已与原始数据不一致时能够及时更新。

由此，本公开的数据读取和缓存方案，不仅可以高效地解决数据存储系统所存储的底层数据发生变更时所产生的缓存一致性问题，而且还可以在不降低系统可用性的情况下引入缓存机制，解决了大量热点数据造成的带宽瓶颈，提升集群的整体资源利用率。

本公开的数据缓存和读取方案能够实现缓存数据与原始数据的强一致性。通过客户端预先请求的元数据信息，能够实时精确地判断当前目标数据（文件）和缓存目标数据是否一致。

而且，本公开提出的方案具有良好的兼容性。在客户端引入数据缓存机制，兼容原有的直接面向数据存储系统的路径访问。客户端发起针对目标数据的请求时，可以直接使用数据存储系统所维护的文件名称（和路径）。这样，可以避免引入缓存造成路径的改变，因而无需对现有计算框架进行改造。

另外，由于本公开的方案基于客户端对数据存储系统的直连访问能力，因此本公开的实施例还可以具有高可用性。当数据缓存系统失效时，上层客户端可以根据策略自行判断，是等待数据缓存系统恢复，还是直连远端的数据存储系统等，从而避免了由于引入缓存系统降低系统的可用性，更符合大规模生产使用的场景。

应当理解，本公开的数据缓存和数据读取方案，不但适用于上文中提到的对象存储服务（OSS）的数据存储系统，而且还可以适用于其它各种数据存储系统，例如简单存储服务（S3）、Hadoop 分布式文件系统（HDFS）、网络附属存储（NAS）等。

事实上，目前的各种数据存储系统，往往都会记录有文件或数据的元数据。因此，可以由请求数据的客户端从数据存储系统请求到数据对应的元数据，发送给缓存系统来进行比对以判断所缓存数据的一致性。因此，各种数据存储系统都可以用于本公开的数据缓存和数据读取方案。

图 6 示出了根据本发明一实施例可用于实现上述数据读取及缓存方法的计算设备的结构示意图。

参见图 6，计算设备 600 包括存储器 610 和处理器 620。

处理器 620 可以是一个多核的处理器，也可以包含多个处理器。在一些实施例中，处理器 620 可以包含一个通用的主处理器以及一个或多个特殊的协处理器，例如图形处理器（GPU）、数字信号处理器（DSP）等等。在一些实施例中，处理器 620 可以使用定制的电
路实现，例如特定用途集成电路（ASIC，Application Specific Integrated Circuit）或者现场可编程逻辑门阵列（FPGA，Field Programmable Gate Arrays）。

存储器 610 可以包括各种类型的存储单元，例如系统内存、只读存储器（ROM），和永久存储装置。其中，ROM 可以存储处理器 620 或者计算机的其他模块需要的静态数据或者指令。永久存储装置可以是可读写的存储装置。永久存储装置可以是即使计算机断电后也不会失去存储的指令和数据的非易失性存储设备。在一些实施方式中，永久性存储装置采用大容量存储装置（例如磁或光盘、闪存）作为永久存储装置。另外一些实施方式中，永久性存储装置可以是可移除的存储设备（例如软盘、光驱）。系统内存可以是可读写存储设备或者易失性可读写存储设备，例如动态随机访问内存。系统内存可以存储一些或者所有处理器在运行时需要的指令和数据。此外，存储器 610 可以包括任意计算机可读存储媒介的组合，包括各种类型的半导体存储芯片（DRAM，SRAM，SDRAM，闪存，可编程

只读存储器), 磁盘和/或光盘也可以采用。在一些实施方式中, 存储器 610 可以包括可读和/或写的可移除的存储设备, 例如激光唱片(CD)、只读数字多功能光盘(例如 DVD-ROM, 双层 DVD-ROM)、只读蓝光光盘、超密度光盘、闪存卡(例如 SD 卡、min SD 卡、Micro-SD 卡等等)、磁性软盘等等。计算机可读存储媒介不包含载波和通过无线或有线传输的瞬间电子信号。

存储器 610 上存储有可执行代码, 当可执行代码被处理器 620 处理时, 可以使处理器 620 执行上文述及的数据读取及缓存方法。

上文中已经参考附图详细描述了根据本发明的数据读取及缓存方案。

此外, 根据本发明的方法还可以实现为一种计算机程序或计算机程序产品, 该计算机程序或计算机程序产品包括用于执行本发明的上述方法中限定的上述各步骤的计算机程序代码指令。

或者, 本发明还可以实施为一种非暂时性机器可读存储介质(或计算机可读存储介质、或机器可读存储介质), 其上存储有可执行代码(或计算机程序、或计算机指令代码), 当所述可执行代码(或计算机程序、或计算机指令代码)被电子设备(或计算设备、服务器等)的处理器执行时, 使所述处理器执行根据本发明的上述方法的各个步骤。

本领域技术人员还将明白的是, 结合这里的公开所描述的各种示例性逻辑块、模块、电路和算法步骤可以被实现为电子硬件、计算机软件或两者的组合。

附图中的流程图和框图显示了根据本发明的多个实施例的系统和方法的可能实现的体系架构、功能和操作。在这点上, 流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分, 所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意, 在有些作为替换的实现中, 方框中所标记的功能也可以以不同于附图中所标记的顺序发生。例如, 两个连续的方框实际上可以基本并行地执行, 它们有时也可以按相反的顺序执行, 这依所涉及的功能而定。也要注意, 框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合, 可以用执行规定的功能或操作的专用的基于硬件的系统来实现, 或者可以用专用硬件与计算机指令的组合来实现。

以上已经描述了本发明的各实施例, 上述说明是示例性的, 并非穷尽性的, 并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下, 对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择, 旨在最好地解释各实施例的原理、实际应用或对市场中的技术的改进, 或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

权 利 要 求 书

1. 一种数据缓存方法，用于数据缓存系统，包括：
接收客户端发来的针对要获取的目标数据的请求，所述请求包括数据存储系统所存储的目标数据的元数据；以及
- 5 基于所述目标数据的元数据，判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存目标数据。
2. 根据权利要求 1 所述的数据缓存方法，还包括：
在判定缓存有与数据存储系统的所述目标数据一致的缓存目标数据的情况下，向客户
- 10 端发送所述缓存目标数据的位置信息；以及/或者
在判定未缓存与数据存储系统的所述目标数据一致的缓存目标数据的情况下，向客户
- 端发送命中失败消息。
3. 根据权利要求 2 所述的数据缓存方法，在判定未缓存与数据存储系统的所述目标
- 15 数据一致的缓存目标数据的情况下，该方法还包括：
从客户端接收客户端从数据存储系统获取的目标数据；以及
缓存所述目标数据。
4. 根据权利要求 1 所述的数据缓存方法，所述判断数据缓存系统是否缓存有与数据
- 20 存储系统所存储的目标数据一致的缓存目标数据的步骤包括：
基于所述目标数据的识别信息，判断是否缓存有对应于所述目标数据的缓存目标数据；
以及
在判定缓存有对应于所述目标数据的缓存目标数据的情况下，基于缓存目标数据的元
- 数据和所述目标数据的元数据，判断所述缓存目标数据与所述目标数据是否一致。
- 25
5. 根据权利要求 1 所述的数据缓存方法，其中，
所述目标数据的元数据是客户端从数据存储系统获取的；并且/或者
所述目标数据的元数据包括目标数据的更新时间、循环冗余校验值、数据尺寸中的至
- 少一项。
- 30
6. 一种数据读取方法，包括：
向数据存储系统发送针对目标数据的第一请求；
从数据存储系统接收所述目标数据的元数据；以及
向数据缓存系统发送针对所述目标数据的第二请求，所述第二请求包括所述元数据。
- 35

7. 根据权利要求 6 所述的数据读取方法, 还包括:

在数据缓存系统基于所述元数据判定缓存有与数据存储系统的所述目标数据一致的数据的情况下, 从所述数据缓存系统获取所述目标数据; 以及/或者

在数据缓存系统基于所述元数据判定未缓存与数据存储系统的所述目标数据一致的数据的情况下, 从所述数据存储系统获取所述目标数据。

8. 根据权利要求 7 所述的数据读取方法, 还包括:

在从所述数据存储系统获取所述目标数据之后, 向所述数据缓存系统发送所获取的目标数据。

10

9. 一种数据存取系统, 包括: 数据存储系统、数据缓存系统和客户端, 其中,

客户端向数据存储系统发送针对目标数据的第一请求,

数据存储系统响应于所述第一请求, 向客户端返回所述目标数据的元数据,

客户端向数据缓存系统发送针对所述目标数据的第二请求, 所述第二请求包括所述元数据,

数据缓存系统基于所述目标数据的元数据, 判断数据缓存系统是否缓存有与数据存储系统所存储的目标数据一致的缓存目标数据,

在判定缓存有与数据存储系统的所述目标数据一致的缓存目标数据的情况下, 数据缓存系统向客户端发送所述缓存目标数据的位置信息, 客户端从所述数据缓存系统获取所述目标数据,

在判定未缓存与数据存储系统的所述目标数据一致的缓存目标数据的情况下, 数据缓存系统向客户端发送命中失败消息, 客户端从所述数据存储系统获取所述目标数据, 并将所获取的目标数据发送给数据缓存系统, 所述数据缓存系统缓存所述目标数据。

25 10. 一种计算设备, 包括:

处理器; 以及

存储器, 其上存储有可执行代码, 当所述可执行代码被所述处理器执行时, 使所述处理器执行如权利要求 1 至 8 中任何一项所述的方法。

30 11. 一种非暂时性机器可读存储介质, 其上存储有可执行代码, 当所述可执行代码被电子设备的处理器执行时, 使所述处理器执行如权利要求 1 至 8 中任何一项所述的方法。