



(12) 发明专利申请

(10) 申请公布号 CN 116185308 A

(43) 申请公布日 2023. 05. 30

(21) 申请号 202310451089.2

(22) 申请日 2023.04.25

(71) 申请人 山东英信计算机技术有限公司

地址 250000 山东省济南市高新区浪潮路
1036号浪潮科技园S05号楼北三层北
区

(72) 发明人 王继玉 郑玉会 赵潇瑶

(74) 专利代理机构 北京集佳知识产权代理有限
公司 11227

专利代理师 郭庆玲

(51) Int. Cl.

G06F 3/06 (2006.01)

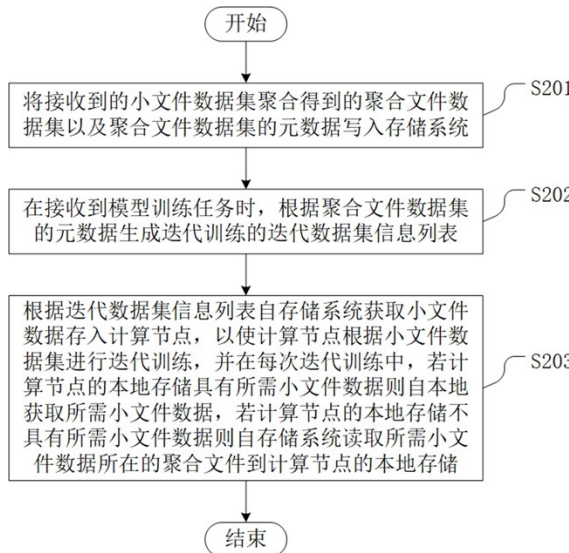
权利要求书4页 说明书20页 附图5页

(54) 发明名称

一种数据集处理方法、装置、设备、介质及模型训练系统

(57) 摘要

本发明涉及存储技术领域,具体公开了一种数据集处理方法、装置、设备、介质及模型训练系统,通过将小文件数据集以聚合文件数据集的形式写入存储系统,减少了将小文件数据写入存储系统的写次数;通过在模型训练中根据聚合文件数据集的元数据生成的迭代数据集信息列表自存储系统获取所需小文件时,对计算节点本地不具有所需小文件数据才在存储系统中获取该小文件数据所在的聚合文件到计算节点本地,减少了在模型训练过程中从存储系统读取小文件数据集的读次数,通过减少存储系统在模型训练过程中的小文件交互,提高了小文件处理效率以在提高存储性能的同时提高人工智能项目落地效率。



1. 一种数据集处理方法,其特征在于,包括:

将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入存储系统;

在接收到模型训练任务时,根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;

根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点,以使所述计算节点根据所述小文件数据集进行迭代训练,并在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储。

2. 根据权利要求1所述的数据集处理方法,其特征在于,所述将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入存储系统,具体包括:

将所述小文件数据集聚合为代码块,并生成各所述代码块的元数据;

将各所述代码块和各所述代码块的元数据写入所述存储系统。

3. 根据权利要求2所述的数据集处理方法,其特征在于,所述将所述小文件数据集聚合为代码块,具体为:

将所述小文件数据集聚合为无损压缩的所述代码块。

4. 根据权利要求2所述的数据集处理方法,其特征在于,所述将各所述代码块和各所述代码块的元数据写入所述存储系统,具体为:

将各所述代码块的元数据写入所述存储系统后,分批次将所述代码块写入所述存储系统。

5. 根据权利要求4所述的数据集处理方法,其特征在于,所述分批次将所述代码块写入所述存储系统,具体为:

将所述代码块放入代码块写队列,每累积预设大小的一批所述代码块,执行一次将所述代码块写入所述存储系统的操作。

6. 根据权利要求4所述的数据集处理方法,其特征在于,所述分批次将所述代码块写入所述存储系统,具体为:

将所述代码块分为多个批次后,并行执行将各批次所述代码块写入所述存储系统的操作。

7. 根据权利要求1所述的数据集处理方法,其特征在于,还包括:

在每次迭代训练中,根据所述迭代数据集信息列表加载当前所需小文件数据之后的小文件数据。

8. 根据权利要求1所述的数据集处理方法,其特征在于,所述根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表,具体包括:

根据所述聚合文件数据集的元数据生成所述聚合文件数据集的数据集信息列表;

对所述数据集信息列表进行乱序处理,得到所述迭代数据集信息列表。

9. 根据权利要求8所述的数据集处理方法,其特征在于,所述对所述数据集信息列表进行乱序处理,得到所述迭代数据集信息列表,具体包括:

对所述数据集信息列表中的各聚合文件进行乱序处理,生成聚合文件乱序列表;

对所述聚合文件乱序列表中的聚合文件进行分组,并以组为单位对组内的小文件数据进行乱序处理后,得到所述迭代数据集信息列表。

10. 根据权利要求9所述的数据集处理方法,其特征在于,所述若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储,具体为:

若所述计算节点的本地存储不具有所需小文件数据,则自所述存储系统读取所需小文件数据所在的聚合文件所在的组到所述计算节点的本地存储。

11. 根据权利要求9所述的数据集处理方法,其特征在于,还包括:

在每次迭代训练中,根据所述迭代数据集信息列表加载当前所需小文件数据所在组的下一组小文件数据。

12. 根据权利要求8所述的数据集处理方法,其特征在于,所述根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点,以使所述计算节点根据所述小文件数据集进行迭代训练,并在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储,具体包括:

在每次迭代训练前,重新执行乱序处理以生成当前迭代次数的所述迭代数据集信息列表;

根据当前迭代次数的所述迭代数据集信息列表依次获取小文件数据进行批量训练,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储,直至完成当前迭代次数的所述迭代数据集信息列表中所有小文件数据的训练;

当满足迭代结束条件后,完成迭代训练。

13. 根据权利要求12所述的数据集处理方法,其特征在于,还包括:

根据下一次迭代训练的所述迭代数据集信息列表和当前迭代次数的所述迭代数据集信息列表,对所述计算节点的本地存储中的小文件数据进行驱逐处理。

14. 根据权利要求13所述的数据集处理方法,其特征在于,所述根据下一次迭代训练的所述迭代数据集信息列表和当前迭代次数的所述迭代数据集信息列表,对所述计算节点的本地存储中的小文件数据进行驱逐处理,具体包括:

以在下次迭代训练的所述迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

以当前迭代次数的所述迭代数据集信息列表中对应的小文件数据均已完成训练为高优先级,调整所述驱逐优先级队列;

根据所述驱逐优先级队列,对所述计算节点的本地存储中的聚合文件进行驱逐处理。

15. 根据权利要求14所述的数据集处理方法,其特征在于,所述根据所述驱逐优先级队列,对所述计算节点的本地存储中的聚合文件进行驱逐处理,具体包括:

若所述计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对所述计算节点的缓存中的聚合文件的第一待驱逐数量;

根据所述第一待驱逐数量,将所述计算节点的缓存中在所述驱逐优先级队列靠前的聚合文件迁移至所述计算节点的本地硬盘中;

若所述计算节点的本地硬盘空间已满,则根据所述第一待驱逐数量,确定对所述计算节点的本地硬盘中的聚合文件的第二待驱逐数量;

根据所述第二待驱逐数量,将所述计算节点的本地硬盘中在所述驱逐优先级队列靠前的聚合文件进行删除处理。

16. 根据权利要求13所述的数据集处理方法,其特征在于,所述根据下一次迭代训练的所述迭代数据集信息列表和当前迭代次数的所述迭代数据集信息列表,对所述计算节点的本地存储中的小文件数据进行驱逐处理,具体包括:

以在下一次迭代训练的所述迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

若所述计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对所述计算节点的缓存中的聚合文件的第一待驱逐数量;

根据所述第一待驱逐数量,将所述驱逐优先级队列靠前且在当前迭代次数的所述迭代数据集信息列表中对应的小文件数据均已完成训练的聚合文件迁移至所述计算节点的本地硬盘中;

若所述计算节点的本地硬盘空间已满,则根据所述第一待驱逐数量,确定所述计算节点的本地硬盘中聚合文件的第二待驱逐数量;

根据所述第二待驱逐数量,将所述计算节点的本地硬盘中在所述驱逐优先级队列靠前且在当前迭代次数的所述迭代数据集信息列表中对应的小文件数据均已完成训练的聚合文件进行删除处理。

17. 根据权利要求1所述的数据集处理方法,其特征在于,还包括:

在接收到所述模型训练任务时,将所述聚合文件数据集的元数据分发至各所述计算节点以存储在所述计算节点的本地硬盘中,以在使用时将所述聚合文件数据集的元数据加载到所述计算节点的内存并根据时间戳维护更新。

18. 根据权利要求1所述的数据集处理方法,其特征在于,还包括:

在接收到所述模型训练任务时,将所述聚合文件数据集的元数据持久化保存到所述计算节点的缓存;

运行超时机制对所述计算节点的缓存中经过预设时间未使用的所述聚合文件数据集的元数据进行清空,或者,在所述模型训练任务完成后,对所述计算节点的缓存中的所述聚合文件数据集的元数据进行清空。

19. 根据权利要求1所述的数据集处理方法,其特征在于,所述聚合文件数据集的元数据包括聚合文件的标识、聚合文件中的小文件数据名列表、聚合文件中的小文件数据目录结构、聚合文件中的小文件数据总数、聚合文件的大小、聚合文件中的小文件数据的修改时间戳。

20. 一种数据集处理装置,其特征在于,包括:

接收单元,用于将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入存储系统;

生成单元,用于在接收到模型训练任务时,根据所述聚合文件数据集的元数据生成迭

代训练的迭代数据集信息列表；

读取单元,用于根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点,以使所述计算节点根据所述小文件数据集进行迭代训练,并在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储。

21. 一种数据集处理设备,其特征在于,包括:

存储器,用于存储计算机程序;

处理器,用于执行所述计算机程序,所述计算机程序被所述处理器执行时实现如权利要求1至19任意一项所述数据集处理方法的步骤。

22. 一种介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至19任意一项所述数据集处理方法的步骤。

23. 一种模型训练系统,其特征在于,包括:存储系统、计算节点,以及设于所述存储节点和所述计算节点之间的数据集处理节点;

其中,所述数据集处理节点用于将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入所述存储系统;在接收到模型训练任务时,根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点;

所述计算节点用于根据所述小文件数据集进行迭代训练;

其中,在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自所述计算节点的本地存储获取所需小文件数据;若所述计算节点的本地存储不具有所需小文件数据,则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储。

一种数据集处理方法、装置、设备、介质及模型训练系统

技术领域

[0001] 本发明涉及存储技术领域,特别是涉及一种数据集处理方法、装置、设备、介质及模型训练系统。

背景技术

[0002] 人工智能(Artificial Intelligence, AI)技术涉及机器人、语言识别、图像识别、自然语言处理和专家系统等领域的研究。基于深度学习(Deep Learning, DL)的人工智能方案能够通过学习样本数据来获得对文字、图像和声音等数据的解释,从而使机器能够获得分析学习能力,能够识别文字、图像和声音等数据,在搜索技术、数据挖掘、机器学习、机器翻译、自然语言处理、多媒体学习、语音、推荐和个性化技术,以及其他相关领域都取得了很多成果。人工智能发展的初始阶段,是将数据集本地化后再进行训练,这不仅需要本地有足够的磁盘空间,还需要花费大量时间等待数据集本地化完成。随着大数据和人工智能技术的发展,复杂场景的人工智能模型训练以及需要海量数据的人工智能模型训练开始出现,受限于本地存储空间大小,将海量数据集存储在本地的方案不再可行。

[0003] 为解决海量数据集的存储问题,相关领域开发出许多数据集的存储方法,通过分布式存储,聚合存储资源,来实现海量数据集的存储。然而,这种将数据集的存储与训练分离的场景,带来了数据访问延迟和远程获取数据集带宽开销大等问题。

[0004] 而相关技术提出的数据集处理框架,为了兼顾不同存储、不同数据类型文件、不同应用场景的需求,更侧重于通用性的研究,而导致出现千万级别规模的海量小文件数据集。同时,由于大规模计算集群并发执行训练任务时,每个训练任务使用的数据集可能不同,计算集群向底层存储系统发出大量的小文件读取请求,这种访问方式在给存储系统带来极大的工作压力的同时,也影响了训练任务的效率,进而导致人工智能项目落地效率低。

[0005] 如何提高小文件数据集处理效率,以在提高存储性能的同时提高人工智能项目落地效率,是本领域技术人员需要解决的技术问题。

发明内容

[0006] 本发明的目的是提供一种数据集处理方法、装置、设备、介质及模型训练系统,用于提高提高小文件数据集处理效率,以在提高存储性能的同时提高人工智能项目落地效率。

[0007] 为解决上述技术问题,本发明提供一种数据集处理方法,包括:

将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入存储系统;

在接收到模型训练任务时,根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;

根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点,以使所述计算节点根据所述小文件数据集进行迭代训练,并在每次迭代训练中,若所述计算

节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储。

[0008] 在一些实施例中,所述将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入存储系统,具体包括:

将所述小文件数据集聚合为代码块,并生成各所述代码块的元数据;

将各所述代码块和各所述代码块的元数据写入所述存储系统。

[0009] 在一些实施例中,所述将所述小文件数据集聚合为代码块,具体为:

将所述小文件数据集聚合为无损压缩的所述代码块。

[0010] 在一些实施例中,所述将各所述代码块和各所述代码块的元数据写入所述存储系统,具体为:

将各所述代码块的元数据写入所述存储系统后,分批次将所述代码块写入所述存储系统。

[0011] 在一些实施例中,所述分批次将所述代码块写入所述存储系统,具体为:

将所述代码块放入代码块写队列,每累积预设大小的一批所述代码块,执行一次将所述代码块写入所述存储系统的操作。

[0012] 在一些实施例中,所述分批次将所述代码块写入所述存储系统,具体为:

将所述代码块分为多个批次后,并行执行将各批次所述代码块写入所述存储系统的操作。

[0013] 在一些实施例中,还包括:

在每次迭代训练中,根据所述迭代数据集信息列表加载当前所需小文件数据之后的小文件数据。

[0014] 在一些实施例中,所述根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表,具体包括:

根据所述聚合文件数据集的元数据生成所述聚合文件数据集的数据集信息列表;

对所述数据集信息列表进行乱序处理,得到所述迭代数据集信息列表。

[0015] 在一些实施例中,所述对所述数据集信息列表进行乱序处理,得到所述迭代数据集信息列表,具体包括:

对所述数据集信息列表中的各聚合文件进行乱序处理,生成聚合文件乱序列表;

对所述聚合文件乱序列表中的聚合文件进行分组,并以组为单位对组内的小文件数据进行乱序处理后,得到所述迭代数据集信息列表。

[0016] 在一些实施例中,所述若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储,具体为:

若所述计算节点的本地存储不具有所需小文件数据,则自所述存储系统读取所需小文件数据所在的聚合文件所在的组到所述计算节点的本地存储。

[0017] 在一些实施例中,还包括:

在每次迭代训练中,根据所述迭代数据集信息列表加载当前所需小文件数据所在组的下一组小文件数据。

[0018] 在一些实施例中,所述根据所述迭代数据集信息列表自所述存储系统获取小文件

数据存入计算节点,以使所述计算节点根据所述小文件数据集进行迭代训练,并在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储,具体包括:

在每次迭代训练前,重新执行乱序处理以生成当前迭代次数的所述迭代数据集信息列表;

根据当前迭代次数的所述迭代数据集信息列表依次获取小文件数据进行批量训练,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储,直至完成当前迭代次数的所述迭代数据集信息列表中所有小文件数据的训练;

当满足迭代结束条件后,完成迭代训练。

[0019] 在一些实施例中,还包括:

根据下一次迭代训练的所述迭代数据集信息列表和当前迭代次数的所述迭代数据集信息列表,对所述计算节点的本地存储中的小文件数据进行驱逐处理。

[0020] 在一些实施例中,所述根据下一次迭代训练的所述迭代数据集信息列表和当前迭代次数的所述迭代数据集信息列表,对所述计算节点的本地存储中的小文件数据进行驱逐处理,具体包括:

以在下一迭代训练的所述迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

以当前迭代次数的所述迭代数据集信息列表中对应的小文件数据均已完成训练为高优先级,调整所述驱逐优先级队列;

根据所述驱逐优先级队列,对所述计算节点的本地存储中的聚合文件进行驱逐处理。

[0021] 在一些实施例中,所述根据所述驱逐优先级队列,对所述计算节点的本地存储中的聚合文件进行驱逐处理,具体包括:

若所述计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对所述计算节点的缓存中的聚合文件的第一待驱逐数量;

根据所述第一待驱逐数量,将所述计算节点的缓存中在所述驱逐优先级队列靠前的聚合文件迁移至所述计算节点的本地硬盘中;

若所述计算节点的本地硬盘空间已满,则根据所述第一待驱逐数量,确定对所述计算节点的本地硬盘中的聚合文件的第二待驱逐数量;

根据所述第二待驱逐数量,将所述计算节点的本地硬盘中在所述驱逐优先级队列靠前的聚合文件进行删除处理。

[0022] 在一些实施例中,所述根据下一次迭代训练的所述迭代数据集信息列表和当前迭代次数的所述迭代数据集信息列表,对所述计算节点的本地存储中的小文件数据进行驱逐处理,具体包括:

以在下一迭代训练的所述迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

若所述计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对所述计算节点的缓存中的聚合文件的第一待驱逐数量;

根据所述第一待驱逐数量,将所述驱逐优先级队列靠前且在当前迭代次数的所述迭代数据集信息列表中对应该小文件数据均已完成训练的聚合文件迁移至所述计算节点的本地硬盘中;

若所述计算节点的本地硬盘空间已满,则根据所述第一待驱逐数量,确定所述计算节点的本地硬盘中聚合文件的第二待驱逐数量;

根据所述第二待驱逐数量,将所述计算节点的本地硬盘中在所述驱逐优先级队列靠前且在当前迭代次数的所述迭代数据集信息列表中对应该小文件数据均已完成训练的聚合文件进行删除处理。

[0023] 在一些实施例中,还包括:

在接收到所述模型训练任务时,将所述聚合文件数据集的元数据分发至各所述计算节点以存储在所述计算节点的本地硬盘中,以在使用时将所述聚合文件数据集的元数据加载到所述计算节点的内存并根据时间戳维护更新。

[0024] 在一些实施例中,还包括:

在接收到所述模型训练任务时,将所述聚合文件数据集的元数据持久化保存到所述计算节点的缓存;

运行超时机制对所述计算节点的缓存中经过预设时间未使用的所述聚合文件数据集的元数据进行清空,或者,在所述模型训练任务完成后,对所述计算节点的缓存中的所述聚合文件数据集的元数据进行清空。

[0025] 在一些实施例中,所述聚合文件数据集的元数据包括聚合文件的标识、聚合文件中的小文件数据名列表、聚合文件中的小文件数据目录结构、聚合文件中的小文件数据总数、聚合文件的大小、聚合文件中的小文件数据的修改时间戳。

[0026] 为解决上述技术问题,本发明还提供一种数据集处理装置,包括:

接收单元,用于将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入存储系统;

生成单元,用于在接收到模型训练任务时,根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;

读取单元,用于根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点,以使所述计算节点根据所述小文件数据集进行迭代训练,并在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若所述计算节点的本地存储不具有所需小文件数据则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储。

[0027] 为解决上述技术问题,本发明还提供一种数据集处理设备,包括:

存储器,用于存储计算机程序;

处理器,用于执行所述计算机程序,所述计算机程序被所述处理器执行时实现如上述任意一项所述数据集处理方法的步骤。

[0028] 为解决上述技术问题,本发明还提供一种介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如上述任意一项所述数据集处理方法的步骤。

[0029] 为解决上述技术问题,本发明还提供一种模型训练系统,包括:存储系统、计算节点,以及设于所述存储节点和所述计算节点之间的数据集处理节点;

其中,所述数据集处理节点用于将接收到的小文件数据集聚合得到的聚合文件数据集以及所述聚合文件数据集的元数据写入所述存储系统;在接收到模型训练任务时,根据所述聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;根据所述迭代数据集信息列表自所述存储系统获取小文件数据存入计算节点;

所述计算节点用于根据所述小文件数据集进行迭代训练;

其中,在每次迭代训练中,若所述计算节点的本地存储具有所需小文件数据则自所述计算节点的本地存储获取所需小文件数据;若所述计算节点的本地存储不具有所需小文件数据,则自所述存储系统读取所需小文件数据所在的聚合文件到所述计算节点的本地存储。

[0030] 本发明所提供的数据集处理方法,通过在接收到小文件数据集时,将小文件数据集聚合为聚合文件数据集,再将聚合文件数据集及其元数据写入存储系统,减少了小文件数据写入存储系统的写次数;进而在接收到模型训练任务时,根据聚合文件的元数据生成迭代训练的迭代数据集信息列表,根据迭代数据集信息列表自存储系统获取小文件数据存入计算节点,以使计算节点根据小文件数据集进行迭代训练,并在每次迭代训练中,若计算节点的本地存储具有所需小文件数据则自本地获取小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,减少了在模型训练过程中从存储系统读取小文件数据的读次数,从而有效解决了模型训练过程中与存储系统频繁交互带来的效率瓶颈,有效提高了人工智能模型输出效率,由此将待处理数据输入人工智能模型,得到处理结果,在提高存储性能的同时提高了人工智能项目落地效率。

[0031] 本发明还提供了一种数据集处理装置、设备、介质及模型训练系统,具有上述有益效果,在此不再赘述。

附图说明

[0032] 为了更清楚的说明本发明实施例或现有技术的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单的介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0033] 图1为本发明实施例提供的一种模型训练系统的架构图;

图2为本发明实施例提供的一种数据集处理方法的流程图;

图3为本发明实施例提供的一种小文件数据集写处理流程示意图;

图4为本发明实施例提供的一种小文件数据集读处理流程示意图;

图5为本发明实施例提供的一种迭代数据集信息列表生成方式示意图;

图6为本发明实施例提供的一种数据集处理装置的结构示意图进行说明;

图7为本发明实施例提供的一种数据集处理设备的结构示意图。

具体实施方式

[0034] 本发明的核心是提供一种数据集处理方法、装置、设备、介质及模型训练系统,用于提高提高小文件数据集处理效率,以在提高存储性能的同时提高人工智能项目落地效率。

[0035] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0036] 下面对本发明实施例一进行说明。

[0037] 图1为本发明实施例提供的一种模型训练系统的架构图。

[0038] 为便于理解,首先对本发明实施例提供的模型训练系统架构进行介绍。如图1所示,本发明实施例提供的模型训练系统包括:存储系统101、计算节点102,以及设于存储节点101和计算节点102之间的数据集处理节点103;

其中,数据集处理节点103用于将接收到的小文件数据集聚合得到的聚合文件数据集以及聚合文件数据集的元数据写入存储系统101;在接收到模型训练任务时,根据聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;根据迭代数据集信息列表自存储系统101获取小文件数据存入计算节点102;

计算节点102用于根据小文件数据集进行迭代训练;

其中,在每次迭代训练中,若计算节点102的本地存储具有所需小文件数据则自计算节点102的本地存储获取所需小文件数据;若计算节点102的本地存储不具有所需小文件数据,则自存储系统101读取所需小文件数据所在的聚合文件到计算节点102的本地存储。

[0039] 在本发明实施例提供的模型训练系统中,存储系统101可以为任意类型的存储系统101,如对象存储系统101、文件存储系统101、BeeGFS存储系统101等。计算节点102可以为单个计算节点102,也可以为分布式训练系统中的计算节点102,各计算节点102之间通过共享内存存储临时训练数据。

[0040] 模型训练过程中的数据集处理环节主要涉及到将数据集写入存储系统101以及从存储系统101读取数据集到计算节点102。在相关技术提供的为实现大规模模型训练任务的方案中,采用计算与存储分离的方式,将小文件数据集整理分类整理好存储系统101,涉及到海量小数据的一次性写入问题;而在模型训练任务中,一般需要经过多次迭代计算(epoch),每次迭代计算都需要读取全部小文件数据集,即计算节点102在每次迭代计算中都需要将每个小文件数据读一遍。针对这种写一次、读频繁的场景,本发明实施例提供的模型训练系统中,为缓解海量小文件读写给存储系统101带来的压力以及对模型训练任务效率的影响,在传统的计算节点102与存储系统101之间增加数据集处理节点103来执行本发明所提出的数据集处理方法,缓解存储系统101的数据集处理压力,提高模型训练任务的效率。

[0041] 从模块化应用的角度来说,本发明实施例涉及到的数据集处理方案可以由数据集处理服务器模块(zzi-server)和数据集处理客户端模块(zzi-client)组成。数据集处理服务器模块用于处理原始数据集(即小文件数据集),将小文件数据集聚合后写入存储系统101,并维护聚合文件数据集的元数据信息。数据集处理客户端模块用于在模型训练任务中

解析聚合文件数据集的元数据信息,生成迭代训练的迭代数据集信息列表,根据计算节点102所需小文件数据提供小文件数据所在的聚合文件。

[0042] 需要说明的是,图1中将数据集处理服务器模块部署于数据集处理节点103,将数据集处理客户端模块部署于计算节点102,仅是本发明实施例提供的模型训练系统的一种实现方式。在实际应用中,数据集处理服务器模块和数据集处理客户端模块可以分别部署于存储系统101和计算节点102,也可以均部署于数据集处理节点103,也可以其中一个部署于存储系统101或计算节点102、另一个部署于数据集处理节点103。数据集处理客户端模块可以以应用程序接口(Application Programming Interface,API)的形式提供给计算节点102的训练任务使用。

[0043] 如图1所示,在模型训练的准备阶段,需要准备数据集。数据集处理节点103运行数据集处理服务器模块从网络文件系统(Network File System,NFS)等存储设备中获取原始海量小文件数据集信息,对小文件数据集中的小文件进行聚合处理为聚合文件,得到聚合文件数据集,将聚合文件数据集和聚合文件数据集的元数据写入存储系统101,使得将数据集写入存储系统101的写次数显著减少。存储系统101将聚合文件数据集和聚合文件数据集的元数据存储在本地的存储模块。

[0044] 可以将数据集处理客户端模块部署于各计算节点102。在模型训练前,将聚合文件数据集的元数据通过各数据集处理客户端提供给各计算节点102,以供各计算节点102运行数据集处理客户端模块从存储系统101中读取聚合文件。各计算节点102存储的聚合文件可以互相传输,或采用共享内存的方式,则当分布式训练系统中一个计算节点102存储有聚合文件,则认为其他计算节点102的本地存储具有该聚合文件。数据集处理客户端模块用于根据聚合文件数据集的元数据确定小文件数据的数据集信息列表,并进行乱序(shuffle)处理生成每次迭代训练中的迭代数据集信息列表,将迭代数据集信息列表提供给计算节点102的计算模块,以供计算模块根据迭代数据集信息列表依次读取小文件数据。

[0045] 计算模块在读取小文件数据时,向数据集处理客户端模块发送小文件数据读命令,数据集处理客户端模块自计算节点102的本地查询,若查询到该所需小文件数据则从计算节点102的本地存储中读取小文件数据作为应答结果给计算模块,若计算节点102的本地存储不具有所需小文件数据,则数据集处理客户端模块根据聚合文件数据集的元数据,通过与存储系统101客户端交互以获取存储在存储系统101的存储模块中的所需小文件数据所在的聚合文件到计算节点102的本地存储。

[0046] 下面对本发明实施例二进行说明。

[0047] 在上述架构的基础上,下面结合附图对本发明实施例提供的数据集处理方法进行说明。

[0048] 图2为本发明实施例提供的一种数据集处理方法的流程图;图3为本发明实施例提供的一种小文件数据集写处理流程示意图;图4为本发明实施例提供的一种小文件数据集读处理流程示意图。

[0049] 如图2所示,本发明实施例提供的数据集处理方法包括:

S201:将接收到的小文件数据集聚合得到的聚合文件数据集以及聚合文件数据集的元数据写入存储系统。

[0050] S202:在接收到模型训练任务时,根据聚合文件数据集的元数据生成迭代训练的

迭代数据集信息列表。

[0051] S203:根据迭代数据集信息列表自存储系统获取小文件数据存入计算节点,以使计算节点根据小文件数据集进行迭代训练,并在每次迭代训练中,若计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储。

[0052] 在具体实施中,小文件数据集具体可以包括但不限于图片集、声音文件集、视频文件集,对应的模型训练任务包括但不限于图像处理模型训练任务(如图像识别)、声音处理模型训练任务、视频处理模型训练任务。

[0053] 对于S201来说,在准备数据集阶段,将数据集写入存储系统时,按照所选存储系统读取文件特性,将小小文件数据集中的小文件数据进行聚合处理为聚合文件,得到聚合文件数据集,生成各聚合文件的元数据(meta)作为聚合文件数据集的元数据。小文件数据集可以来自网络文件系统(Network File System,NFS)等存储设备,也可以由用户通过模型训练客户端上传。

[0054] 聚合文件的聚合形式可以为代码块,即将多个小文件数据聚合为一个代码块。则S201具体可以包括:

将小文件数据集聚合为代码块,并生成各代码块的元数据;

将各代码块和各代码块的元数据写入存储系统。

[0055] 具体来说,可以在保持小文件数据集的根目录的基础上,抽取二级目录信息,构建出二级目录队列,可以按照目录名、日期、数据类型,各目录下文件总数、文件总量等规则,排序目录队列,使用线程池,以二级目录为单位,处理二级目录下的所有小文件数据。继而按照目录队列顺序,抽取各二级目录所有的小文件,并放入小文件队列中,与二级目录映射,文件队列可以按照修改日期、大小、文件类型、文件名等规则排序。为了降低海量小文件数据与存储系统的元数据服务交互的次数,需要聚合小文件。如图3所示,可以按照小文件队列的顺序,抽取满足能聚合为一定大小(如4MB左右)的代码块的文件子列表,同时记录该文件子列表的元数据信息,将代码块以唯一识别码(UUID)的形式命名,以代码块的形式进行小文件数据的存储。为方便后续对小文件数据进行检索和读取,可以保持小文件数据在聚合文件中的相对路径不变。

[0056] 进一步的,将小文件数据集聚合为代码块,具体可以为:将小文件数据集聚合为无损压缩的代码块(chunk)。以聚合无损压缩的形式存储处理小文件数据,使无损压缩的代码块(chunk)中的小文件数据的总大小远远小于原始小文件数据集大小。基于此,本发明实施例提供的数据集处理方法,利用数据集写少读多的特性,干预小文件原始数据集写入存储系统的过程,通过将小文件数据聚合无损压缩划分块的形式,将千万小文件数据与元数据服务密集型千万次交互级别,降低为百万或几十万次级别,极大提升小文件数据集的写入效率。同时,以聚合文件的形式将小文件数据写入存储系统,也从千万次级别降低至百万或几十万次级别,极大提高了存储系统的写数据效率,因为无损聚合压缩小文件,小文件数据集远远小于原始数据集,也显著降低了数据集所需的存储空间,节约存储成本,缩短了海量数据集的总体写入时间。同时由于将小文件数据聚合为聚合文件,也减少了元数据管理小文件目录的工作量,以及与元数据的交互量,提高了元数据管理效率。

[0057] 为了提高写入效率,数据集处理服务器模块可以先将聚合文件的元数据信息保存

到关键字取值(Key-Value)数据库中,如远程字典服务(Remote Dictionary Server, Redis)数据库,或者关系型数据库MySQL中。聚合文件数据集的元数据可以包括但不限于聚合文件的标识、聚合文件中的小文件数据名列表、聚合文件中的小文件数据目录结构、聚合文件中的小文件数据总数、聚合文件的大小、聚合文件中的小文件数据的修改时间戳。

[0058] 将处理好的聚合文件存储到存储系统的指定目录,具体可以将聚合文件写入存储系统后,按照原始小文件数据集的目录结构将其中的小文件数据存储在存储系统中,并通过聚合文件数据集的元数据中的小文件数据目录结构等元数据进行存储信息维护。将全部的聚合文件数据集处理完毕后,数据集处理服务器模块生成聚合文件数据集的元数据信息。

[0059] 对聚合文件数据集的元数据,本发明实施例提供的数据集处理方法还可以包括:在接收到模型训练任务时,将聚合文件数据集的元数据分发至各计算节点以存储在计算节点的本地硬盘中,以在使用时将聚合文件数据集的元数据加载到计算节点的内存并根据时间戳维护更新。由模型训练任务触发将元数据写到计算节点的本地硬盘,以供计算节点加载使用。可以由计算节点部署的数据集处理客户端模块来维护聚合文件数据集的元数据的时间戳,以根据计算模块的取用情况以及聚合文件数据集的元数据的保存时间来执行对聚合文件数据集的元数据的驱逐操作。

[0060] 进一步的,本发明实施例提供的数据集处理方法还可以包括:

在接收到模型训练任务时,将聚合文件数据集的元数据持久化保存到计算节点的缓存;

运行超时机制对计算节点的缓存中经过预设时间未使用的聚合文件数据集的元数据进行清空,或者,在模型训练任务完成后,对计算节点的缓存中的聚合文件数据集的元数据进行清空。

[0061] 在初次使用时,可以由模型任务触发将元数据持久化保存到计算节点的缓存,从而在使用时可以直接自缓存读取,提高读取效率;设置超时机制或跟随任务机制对缓存中的元数据进行清理,如经过1小时(可配置更改)则清空缓存中的元数据信息,或跟随模型训练任务,在完成一次模型训练任务后清理计算节点本地的聚合文件数据集的元数据。

[0062] 通过将聚合文件数据集的元数据存储到计算节点,在使用时,计算节点无需再与存储系统的元数据交互,可以直接加载元数据信息,获取聚合文件列表信息以及抽取聚合文件内的文件列表信息,降低了并发训练时对存储系统中聚合文件数据集的元数据的密集型访问负载。

[0063] 对于S202和S203来说,数据集处理客户端模块部署在各计算节点,可以以应用程序接口(Application Programming Interface,API)的形式提供给计算节点的训练任务使用。在接收到模型训练任务中的数据集访问请求时,数据集处理客户端模块首先与数据集处理服务器模块交互,从数据集处理服务器模块预先用于存储数据集元数据的数据库中检索该数据集的信息,如果未命中,则触发从网络文件系统等原始数据集存储系统缓存和处理原始的小文件数据集;如果命中,则数据集处理服务器模块将抽取聚合文件数据集的元数据,发送给各计算节点,可以以关键字取值(Key-Value)的形式保存到计算节点的内存中。在模型训练任务的第一次迭代训练中触发了元数据缓存后,后续迭代训练可以直接使用,可以由数据集处理客户端模块定时任务,自动触发、定时更新,维持计算节点本地的聚

合文件数据集的元数据与存储系统中聚合文件数据集的元数据的一致性。数据集处理服务器模块也可以在计算节点所在的分布式训练系统的集群中通过广播同步更新聚合文件数据集的元数据。

[0064] 为减少存储空间的占用,聚合文件数据集的元数据在数据集处理客户端模块对应存储空间中的存储形式可以为快照形式。在受到模型训练任务的触发后,数据集处理客户端模块根据本地存储的聚合文件数据集的元数据的快照信息,解析出小文件数据集的全局目录结构,提供给模型训练任务使用。则各计算节点在查看数据集目录视图时,无需再与存储系统交互,可以直接使用数据集处理服务器模块的关键字取值数据库存储的聚合文件数据集的元数据,解析出原始的小文件数据集的目录结构、小文件数据的文件列表以及各聚合文件的名称(具体可以以唯一识别码形式表示)列表等。为方便用户查看,可以设置数据集处理客户端模块生成解析得到的聚合文件数据集对应的文件列表(如聚合文件的唯一识别码列表)的查看接口提供给用户查看。

[0065] 根据聚合文件数据集对应的小文件列表生成迭代训练所需的迭代数据集信息列表,指明在迭代训练中获取小文件数据的顺序。

[0066] 对于S203来说,如果计算节点所在的分布式训练集群提供的存储空间能够缓存全部的小文件数据集,则在第一次迭代训练时缓存所有的聚合文件,后续迭代训练可以直接读取缓存中的聚合文件,无需再与存储系统交互读取,提高了读取效率和命中效率,降低了对存储系统的并发访问压力。但通常情况下,计算节点的本地存储不可能存储到全部的数据集。因此第一次迭代训练处理小文件数据的缓存方式直接影响了后续迭代训练的训练效率。

[0067] 在本发明实施例提供的数据集处理方法中,基于部署在各计算节点的数据集处理客户端模块,配合计算节点的计算模块(如中央处理器)进行模型训练任务。在第一次迭代训练时,每次小批量(mini-batch)训练可以以聚合文件中任意一个小文件数据为触发单位与存储系统交互一次,触发该聚合文件的缓存,后续该聚合文件中其他小文件数据再次触发时,直接读取缓存中的聚合文件的内容,不再与存储系统交互。在模型训练任务中,按照迭代数据集信息列表中小文件数据的排列顺序依次获取小文件数据。若当前需要获取的小文件数据在计算节点的本地存储中,则直接从计算节点的本地存储中加载所需小文件数据。需要说明的是,如果当前计算机节点的本地存储中没有所需小文件数据,但位于同一分布式训练系统中的其他计算节点的本地存储中具有所需小文件数据,也可以认为是计算节点的本地存储中具有所需小文件数据,即通过与分布式训练系统的其他计算节点进行交互,获取所需小文件数据即可。或者如果采用分布式训练系统采用共享内存的方式,则各计算节点的本地存储中具有所需小文件数据,即指共享内存中具有所需小文件数据。

[0068] 若当前需要获取的小文件数据不在计算节点的本地存储中,则需要访问存储系统来获取所需小文件数据。此时并非仅仅向存储系统请求所需小文件数据,而是根据小文件数据所在的聚合文件,向存储系统获取该聚合文件。这样一次读取的不仅仅是一个小文件数据,还有其他的小文件数据,以便在迭代训练的以便在后续迭代训练中使用。

[0069] 如图4所示,根据聚合文件数据集的元数据获取所需小文件数据所在的聚合文件在存储系统中的存储目录,由存储系统客户端读取聚合文件,经过数据集处理客户端模块将聚合文件读到计算节点本地存储为多个小文件数据,其中包括当前所需小文件数据以

及后续所需小文件数据。

[0070] 如此,在模型训练过程中,可以不必每次需要获取一个小文件数据时都访问一次存储系统,而是有几率可以在计算节点的稳定存储中获取所需小文件数据,从而显著降低了在模型训练过程中计算节点与存储系统的交互次数。

[0071] 为进一步提高训练效率,本发明实施例提供的数据集处理方法还可以包括:在每次迭代训练中,根据迭代数据集信息列表加载当前所需小文件数据之后的小文件数据。由于计算节点的本地存储的空间限制,通常无法一次性将小文件数据集的整体加载到计算节点的本地存储,故需要根据迭代数据集信息列表依次加载所需小文件数据进行批量训练。而在每次迭代训练中,在根据迭代数据集信息列表依次加载所需小文件数据到计算节点的本地存储时,可以在当前所需小文件数据投入批量训练的同时加载之后所需小文件数据到计算节点的本地存储。加载方式同S203中所记载的,若计算节点的本地存储具有所需小文件数据则自计算节点本地获取小所需小文件数据,若计算节点的本地存储中不具有所需小文件数据则到存储系统中读取所需小小文件数据所在的聚合文件到计算节点的本地存储。可以在训练第一批小文件数据的同时触发第二批小文件数据的后台缓存,从而等批量训练访问到第二批小文件数据时可以直接缓存命中,不需要再与存储系统进行交互以跨网络读取小文件数据。

[0072] 本发明实施例提供的数据集处理方法,通过在接收到小文件数据集时,将小文件数据集聚合为聚合文件数据集,再将聚合文件数据集及其元数据写入存储系统,减少了小文件数据写入存储系统的写次数;进而在接收到模型训练任务时,根据聚合文件的元数据生成迭代训练的迭代数据集信息列表,根据迭代数据集信息列表自存储系统获取小文件数据存入计算节点,以使计算节点根据小文件数据集进行迭代训练,并在每次迭代训练中,若计算节点的本地存储具有所需小文件数据则自本地获取小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,减少了在模型训练过程中从存储系统读取小文件数据的读次数,从而有效解决了模型训练过程中与存储系统频繁交互带来的效率瓶颈,有效提高了人工智能模型输出效率,由此将待处理数据输入人工智能模型,得到处理结果,在提高存储性能的同时提高了人工智能项目落地效率。

[0073] 本发明实施例提供的数据集处理方法若应用于BeeGFS存储系统,由于BeeGFS存储系统原本的读写方案均是以小文件数据为单位,则提升存储性能的优势更为明显。

[0074] 下面对本发明实施例三进行说明。

[0075] 在上述实施例的基础上,本发明实施例进一步对数据集处理方法的写过程进行介绍。

[0076] 在本发明实施例提供的数据集处理方法中,将各代码块和各代码块的元数据写入存储系统,具体可以为:将各代码块的元数据写入存储系统后,分批次将代码块写入存储系统。

[0077] 将小文件数据聚合(例如聚合为代码块)的基础上,在将代码块的元数据写入存储系统后分批次执行将代码块写入存储系统的任务。由此可以累计多个代码块再执行一次写入,从而进一步减少将数据集写入存储系统的写入次数。例如可以累计128MB或512MB(可配置)大小的代码块后,执行一次写入存储系统的操作。为此,分批次将代码块写入存储系统,

具体可以为:将代码块放入代码块写队列,每累积预设大小的一批代码块,执行一次将代码块写入存储系统的操作。

[0078] 为充分利用网络带宽和I/O,提高存储系统的小文件写入效率,可以在分批写入代码块的同时,并行执行写入操作。则分批次将代码块写入存储系统,具体可以为:将代码块分为多个批次后,并行执行将各批次代码块写入存储系统的操作。

[0079] 下面对本发明实施例四进行说明。

[0080] 图5为本发明实施例提供的一种迭代数据集信息列表生成方式示意图。

[0081] 在上述实施例的基础上,本发明实施例进一步对数据集处理方法的读过程进行介绍。

[0082] 在本发明实施例提供的数据集处理方法中,S202中根据聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表,具体可以包括:

根据聚合文件数据集的元数据生成聚合文件数据集的数据集信息列表;
对数据集信息列表进行乱序处理,得到迭代数据集信息列表。

[0083] 在进行模型时需要执行将数据集从存储系统读到计算节点的本地存储的读操作。在本发明上述实施例中已介绍在迭代训练中,根据当前迭代训练对应的迭代数据集信息列表依次获取小文件数据到计算节点的本地存储来执行批量训练过程。为避免模型训练过拟合,需要在迭代训练中将小文件数据的训练顺序进行乱序(shuffle)处理。

[0084] 具体来说,在模型训练任务的第一次迭代训练时,加载计算节点的本地存储的聚合文件数据集的元数据,根据该元数据信息构建原始小文件数据集的元数据信息,其中可以包括聚合文件的唯一识别码、目录层级结构、文件路径列表等。再根据聚合文件数据集的元数据对应的数据集信息列表进行乱序处理,得到第一次迭代训练对应的迭代数据集信息列表。

[0085] 在此基础上,本发明实施例提供一种执行乱序处理的方法,即对数据集信息列表进行乱序处理,得到迭代数据集信息列表,具体可以包括:

对数据集信息列表中的各聚合文件进行乱序处理,生成聚合文件乱序列表;
对聚合文件乱序列表中的聚合文件进行分组,并以组为单位对组内的小文件数据进行乱序处理后,得到迭代数据集信息列表。

[0086] 在具体实施中,先将数据集信息列表中的各聚合文件(代码块)进行乱序处理,得到聚合文件乱序列表。继而对打乱顺序的聚合文件进行分组,以组为单位对组内的小文件数据进行乱序处理后,生成所有小文件数据的迭代数据集信息列表。如图5所示,以数据集信息列表包括顺序排列的代码块1~代码块6为例,经过打乱顺序后分为两组,第一组包括代码块4、代码块1、代码块5,第二组包括代码块2、代码块5、代码块3,定义此时对应的小文件数据分别为小文件1~小文件24,再分别对第一组的小文件数据和第二组的小文件数据进行乱序处理,生成小文件数据的迭代数据集信息列表,作为当前迭代训练取用小文件数据的顺序,例如参照迭代数据集信息列表的顺序,每四个小文件数据作为一个小批量数据。

[0087] 可以理解的是,除了该乱序处理方法,还可以采用其他对数据集信息列表进行乱序处理得到迭代数据集信息列表的方法。

[0088] 此时若采用无损压缩的代码块(chunk),则读取时缓存的数据集远小于原始数据集。由于计算节点的内存空间有限,同时也为了不影响计算节点的服务,可以配置数据集处

理客户端模块利用计算节点内存的缓存空间,结合计算节点的内存空间,数据集客户端模块可以配置代码块分组的数量保障本组的代码块能够缓存到该计算节点的内存中。则在将乱序后的代码块列表按照规则分组后,再以组为单位抽取每组的小文件列表,乱序处理该组的小文件列表汇总成总的小文件列表,即迭代数据集信息列表。在模型训练中使用迭代数据集信息列表执行训练。

[0089] 而基于上述乱序处理方法,为进一步减少模型训练中需要到存储系统的读次数,S203中若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,具体为:若计算节点的本地存储不具有所需小文件数据,则自存储系统读取所需小文件数据所在的聚合文件所在的组到计算节点的本地存储。即是说,若计算节点的本地存储不具有所需小文件数据,则自存储系统中将所需小文件数据所在的聚合文件所在的组均读取到计算节点的本地存储,由此对应乱序处理中在每次迭代训练的每次大batch批量训练读取的文件可能来源于几个组内的聚合文件的设置,即在每次到存储系统进行一次读操作时,均在计算节点的本地存储中存储了本batch批次的文件数据。则在小批量(mini-batch)训练时,如果某聚合文件中的一个小文件数据被读取,则触发该聚合文件乃至该聚合文件所在的组的存储,在后续小批量(mini-batch)训练时无需再与存储系统交互,可以直接从计算节点的本地存储中读取所需数据集,提高数据命中效率,减少与存储系统的交互次数的同时降低存储系统的并发访问压力。

[0090] 而组为单位加载数据集,在分组后的聚合文件乱序列表中仍然具备组的分布顺序,可以配置策略,在进行模型训练时,以组为单位加载聚合文件。在训练当前组聚合文件的时候,数据集处理客户端模块触发下一组聚合文件的后台缓存。等到小批量(mini-batch)训练访问到下一组文件时可以直接缓存命中,不需要再与存储系统交互以跨网络读取数据。则本发明实施例提供的数据集处理方法还可以包括:在每次迭代训练中,根据所述迭代数据集信息列表加载当前所需小文件数据所在组的下一组小文件数据。

[0091] 为防止模型过拟合,在每次迭代训练前均重新进行乱序处理,生成当前迭代次数的迭代数据集信息列表。具体可以在最初的数据集信息列表的基础上重新进行乱序处理,生成新的迭代数据集信息列表,也可以在前次迭代训练的迭代数据集信息列表的基础上进行乱序处理,生成当前迭代次数的迭代数据集信息列表。乱序处理的方式可以参考上文所记载的处理方式。

[0092] 则在本发明实施例提供的数据集处理方法中,S203:根据迭代数据集信息列表自存储系统获取小文件数据存入计算节点,以使计算节点根据小文件数据集进行迭代训练,并在每次迭代训练中,若计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,具体可以包括:

在每次迭代训练前,重新执行乱序处理以生成当前迭代次数的迭代数据集信息列表;

根据当前迭代次数的迭代数据集信息列表依次获取小文件数据进行批量训练,若计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,直至完成当前迭代次数的迭代数据集信息列表中所有小文件数据的训

练；

当满足迭代结束条件后，完成迭代训练。

[0093] 下面对本发明实施例五进行说明。

[0094] 由于计算节点的本地存储通常不可能存储全部的小文件数据集，故在根据迭代数据集信息列表加载小文件数据到计算节点的本地存储以完成批量训练时，需要执行对此前存储的小文件数据的驱逐处理。在上述实施例的基础上，本发明实施例进一步对计算节点的本地存储中的小文件数据的驱逐处理方式进行说明。

[0095] 在上述实施例的基础上，本发明实施例提供的数据集处理方法还可以包括：还包括：

根据下一次迭代训练的迭代数据集信息列表和当前迭代次数的迭代数据集信息列表，对计算节点的本地存储中的聚合文件进行驱逐处理。

[0096] 在具体实施中，对计算节点的本地存储的小文件数据进行驱逐处理的原则在于在下一迭代训练的迭代数据集信息列表中靠前，而在当前迭代训练中已参加批量训练的小文件数据为优先。则需要执行驱逐策略前生成下一次迭代训练的迭代数据集信息列表以配合在当前迭代训练中的小文件数据驱逐处理。执行驱逐处理的单位可以是小文件数据，也可以是小文件数据所在的聚合文件或小文件数据所在的组。

[0097] 以驱逐处理的单位为小文件数据所在的聚合文件为例，根据下一次迭代训练的迭代数据集信息列表和当前迭代次数的迭代数据集信息列表，对计算节点的本地存储中的聚合文件进行驱逐处理，具体可以包括：

以在下一迭代训练的迭代数据集信息列表中索引位置靠后为高优先级，生成聚合文件的驱逐优先级队列；

以当前迭代次数的迭代数据集信息列表中对应的小文件数据均已完成训练为高优先级，调整驱逐优先级队列；

根据驱逐优先级队列，对计算节点的本地存储中的聚合文件进行驱逐处理。

[0098] 在具体实施中，维护驱逐优先级队列，以在下一迭代训练的迭代数据集信息列表中索引位置靠后为高优先级，同时以在当前迭代次数的迭代数据集信息列表中对应的小文件数据均已完成训练为高优先级，调整驱逐优先级队列中聚合文件的排序，从而对计算节点的本地存储中在驱逐优先级队列排序靠前的聚合文件进行驱逐处理。驱逐处理的顺序可以为自计算节点的内存到计算节点的本地硬盘再到删除的综合方式。

[0099] 则根据驱逐优先级队列，对计算节点的本地存储中的聚合文件进行驱逐处理，具体可以包括：

若计算节点的缓存空间已满，则根据当前迭代次数中剩余批量训练所需的缓存空间，确定对计算节点的缓存中的聚合文件的第一待驱逐数量；

根据第一待驱逐数量，将计算节点的缓存中在驱逐优先级队列靠前的聚合文件迁移至计算节点的本地(SSD)硬盘中；

若计算节点的本地硬盘空间已满，则根据第一待驱逐数量，确定对计算节点的本地硬盘中的聚合文件的第二待驱逐数量；

根据第二待驱逐数量，将计算节点的本地硬盘中在驱逐优先级队列靠前的聚合文件进行删除处理。

[0100] 在具体实施中,在按照当前迭代次数对应的迭代数据集信息列表加载小文件数据时,若所有计算机里的内存已达到配置的缓存上限,则触发缓存转移,将内存中的缓存转移到本地硬盘中。硬盘存储空间支持可配置,防止存储爆满和影响计算节点服务器正常使用。根据当前迭代次数对应的迭代数据集信息列表与下一次迭代训练对应的迭代数据集信息列表执行转移策略,针对每个计算节点内存缓存的聚合文件与驱逐优先级队列中的聚合文件的信息比对,比较其在驱逐优先级队列中的索引位置,首先转移索引靠前的聚合文件,存储到本地硬盘。如果本地硬盘的存储也达到配置上限,则触发驱逐策略,也是利用索引位置规则从本地硬盘中驱逐在驱逐优先级队列索引位置靠前的聚合文件,同时从计算节点的内存中转移在驱逐优先级队列的索引位置靠前的聚合文件到本地硬盘中。

[0101] 本发明实施例提供的数据集处理方法,可以使海量小文件数据集在下次迭代训练时尽量减少与存储系统的交互以跨网络读取数据,在下次迭代训练时可以继续上述策略来乱序处理生成下下次的迭代数据集信息列表,同时在本次迭代训练中的小文件数据,有些可以在缓存中命中,因为在第一次迭代训练时是从后往前驱逐数据,则靠前的小文件数据大部分已在计算节点的内存和本地硬盘中,可以直接缓存和本地化读取,从而提高了数据加载效率和训练效率。基于此,在读取小文件数据时,如果本地缓存命中,则直接从本地缓存中加载;如果远程命中,则从远端加载;如果未命中则与存储系统交互加载该聚合文件的内容。待任务结束时使用超时配置清理缓存或任务周期清理缓存。

[0102] 下面对本发明实施例六进行说明。

[0103] 除了上述实施例提供的维护驱逐优先级队列以执行对计算节点本地的本地存储中的小文件数据的驱逐策略外,本发明实施例提供另一种维护驱逐优先级队列以驱逐计算节点本地的小文件数据的方法。

[0104] 在本发明实施例提供的数据集处理方法中,根据下一次迭代训练的迭代数据集信息列表和当前迭代次数的迭代数据集信息列表,对计算节点的本地存储中的小文件数据进行驱逐处理,具体可以包括:

以在下次迭代训练的迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

若计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对计算节点的缓存中的聚合文件的第一待驱逐数量;

根据第一待驱逐数量,将驱逐优先级队列靠前且在当前迭代次数的迭代数据集信息列表中对的小文件数据均已完成训练的聚合文件迁移至计算节点的本地硬盘中;

若计算节点的本地硬盘空间已满,则根据第一待驱逐数量,确定计算节点的本地硬盘中聚合文件的第二待驱逐数量;

根据第二待驱逐数量,将计算节点的本地硬盘中在驱逐优先级队列靠前且在当前迭代次数的迭代数据集信息列表中对的小文件数据均已完成训练的聚合文件进行删除处理。

[0105] 在本发明实施例中,驱逐优先级队列的排序方式仅以在下次迭代训练的迭代数据集信息列表中索引位置靠后为高优先级进行聚合文件的信息的排列。即在下次迭代训练的迭代数据集信息列表中,索引位置越靠后则优先级越高,在驱逐优先级队列中索引位置越靠前。在本发明实施例提供到驱逐优先级队列的维护方式的基础上,以在当前迭代训

练中对应的小文件数据均已完成训练为验证方式,来执行对计算节点的本地存储中的小文件数据的驱逐处理。

[0106] 与上述实施例相同的是,也可以配置计算节点的内存中提供给数据集的存储空间以及本地硬盘中提供给数据集的存储空间。在内存中配置的存储空间达到上限后,先转移在驱逐优先级队列中索引靠前的且在当前迭代训练中小文件数据均已完成训练的聚合文件到本地硬盘中;而在本地硬盘中配置的存储空间达到上限后,对在驱逐优先级队列中索引靠前的且在当前迭代训练中小文件数据均已完成训练的聚合文件进行删除处理,再将内存中在驱逐优先级队列中索引靠前的且在当前迭代训练中小文件数据均已完成训练的聚合文件转移到本地硬盘中。

[0107] 上文详述了数据集处理方法对应的各个实施例,在此基础上,本发明还公开了与上述方法对应的数据集处理装置、设备及计算机可读存储介质。

[0108] 下面对本发明实施例七进行说明。

[0109] 图6为本发明实施例提供的一种数据集处理装置的结构示意图进行说明。

[0110] 如图6所示,本发明实施例提供的数据集处理装置包括:

接收单元601,用于将接收到的小文件数据集聚合得到的聚合文件数据集以及聚合文件数据集的元数据写入存储系统;

生成单元602,用于在接收到模型训练任务时,根据聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表;

读取单元603,用于根据迭代数据集信息列表自存储系统获取小文件数据存入计算节点,以使计算节点根据小文件数据集进行迭代训练,并在每次迭代训练中,若计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储。

[0111] 其中,接收单元601将接收到的小文件数据集聚合得到的聚合文件数据集以及聚合文件数据集的元数据写入存储系统,具体可以包括:

将小文件数据集聚合为代码块,并生成各代码块的元数据;

将各代码块和各代码块的元数据写入存储系统。

[0112] 其中,接收单元601将小文件数据集聚合为代码块,具体可以为:

将小文件数据集聚合为无损压缩的代码块。

[0113] 进一步的,接收单元601将各代码块和各代码块的元数据写入存储系统,具体可以为:

将各代码块的元数据写入存储系统后,分批次将代码块写入存储系统。

[0114] 其中,接收单元601分批次将代码块写入存储系统,具体可以为:

将代码块放入代码块写队列,每累积预设大小的一批代码块,执行一次将代码块写入存储系统的操作。

[0115] 其中,接收单元601分批次将代码块写入存储系统,具体可以为:

将代码块分为多个批次后,并行执行将各批次代码块写入存储系统的操作。

[0116] 本发明实施例提供的数据集处理装置还可以包括:

第一预加载单元,用于在每次迭代训练中,根据迭代数据集信息列表加载当前所

需小文件数据之后的小文件数据。

[0117] 生成单元602根据聚合文件数据集的元数据生成迭代训练的迭代数据集信息列表,具体可以包括:

根据聚合文件数据集的元数据生成聚合文件数据集的数据集信息列表;
对数据集信息列表进行乱序处理,得到迭代数据集信息列表。

[0118] 生成单元602对数据集信息列表进行乱序处理,得到迭代数据集信息列表,具体可以包括:

对数据集信息列表中的各聚合文件进行乱序处理,生成聚合文件乱序列表;
对聚合文件乱序列表中的聚合文件进行分组,并以组为单位对组内的小文件数据进行乱序处理后,得到迭代数据集信息列表。

[0119] 读取单元603若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,具体可以为:

若计算节点的本地存储不具有所需小文件数据,则自存储系统读取所需小文件数据所在的聚合文件所在的组到计算节点的本地存储。

[0120] 本发明实施例提供的数据集处理装置还可以包括:

第二预加载单元,用于在每次迭代训练中,根据迭代数据集信息列表加载当前所需小文件数据所在组的下一组小文件数据。

[0121] 读取单元603根据迭代数据集信息列表自存储系统获取小文件数据存入计算节点,以使计算节点根据小文件数据集进行迭代训练,并在每次迭代训练中,若计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,具体可以包括:

在每次迭代训练前,重新执行乱序处理以生成当前迭代次数的迭代数据集信息列表;

根据当前迭代次数的迭代数据集信息列表依次获取小文件数据进行批量训练,若计算节点的本地存储具有所需小文件数据则自本地获取所需小文件数据,若计算节点的本地存储不具有所需小文件数据则自存储系统读取所需小文件数据所在的聚合文件到计算节点的本地存储,直至完成当前迭代次数的迭代数据集信息列表中所有小文件数据的训练;

当满足迭代结束条件后,完成迭代训练。

[0122] 本发明实施例提供的数据集处理装置还可以包括:

驱逐处理单元,用于根据下一次迭代训练的迭代数据集信息列表和当前迭代次数的迭代数据集信息列表,对计算节点的本地存储中的小文件数据进行驱逐处理。

[0123] 其中,驱逐处理单元根据下一次迭代训练的迭代数据集信息列表和当前迭代次数的迭代数据集信息列表,对计算节点的本地存储中的小文件数据进行驱逐处理,具体可以包括:

以在下次迭代训练的迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

以当前迭代次数的迭代数据集信息列表中对应的小文件数据均已完成训练为高

优先级,调整驱逐优先级队列;

根据驱逐优先级队列,对计算节点的本地存储中的聚合文件进行驱逐处理。

[0124] 其中,驱逐处理单元根据驱逐优先级队列,对计算节点的本地存储中的聚合文件进行驱逐处理,具体可以包括:

若计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对计算节点的缓存中的聚合文件的第一待驱逐数量;

根据第一待驱逐数量,将计算节点的缓存中在驱逐优先级队列靠前的聚合文件迁移至计算节点的本地硬盘中;

若计算节点的本地硬盘空间已满,则根据第一待驱逐数量,确定对计算节点的本地硬盘中的聚合文件的第二待驱逐数量;

根据第二待驱逐数量,将计算节点的本地硬盘中在驱逐优先级队列靠前的聚合文件进行删除处理。

[0125] 或者,驱逐处理单元根据下一次迭代训练的迭代数据集信息列表和当前迭代次数的迭代数据集信息列表,对计算节点的本地存储中的小文件数据进行驱逐处理,具体可以包括:

以在下一次迭代训练的迭代数据集信息列表中索引位置靠后为高优先级,生成聚合文件的驱逐优先级队列;

若计算节点的缓存空间已满,则根据当前迭代次数中剩余批量训练所需的缓存空间,确定对计算节点的缓存中的聚合文件的第一待驱逐数量;

根据第一待驱逐数量,将驱逐优先级队列靠前且在当前迭代次数的迭代数据集信息列表中对的小文件数据均已完成训练的聚合文件迁移至计算节点的本地硬盘中;

若计算节点的本地硬盘空间已满,则根据第一待驱逐数量,确定计算节点的本地硬盘中聚合文件的第二待驱逐数量;

根据第二待驱逐数量,将计算节点的本地硬盘中在驱逐优先级队列靠前且在当前迭代次数的迭代数据集信息列表中对的小文件数据均已完成训练的聚合文件进行删除处理。

[0126] 本发明实施例提供的数据集处理装置还可以包括:

第一元数据更新单元,用于在接收到模型训练任务时,将聚合文件数据集的元数据分发至各计算节点以存储在计算节点的本地硬盘中,以在使用时将聚合文件数据集的元数据加载到计算节点的内存并根据时间戳维护更新。

[0127] 本发明实施例提供的数据集处理装置还可以包括:

第二元数据更新单元,用于在接收到模型训练任务时,将聚合文件数据集的元数据持久化保存到计算节点的缓存;

元数据维护单元,用于运行超时机制对计算节点的缓存中经过预设时间未使用的聚合文件数据集的元数据进行清空,或者,在模型训练任务完成后,对计算节点的缓存中的聚合文件数据集的元数据进行清空。

[0128] 在本发明实施例中,聚合文件数据集的元数据可以包括聚合文件的标识、聚合文件中的小文件数据名列表、聚合文件中的小文件数据目录结构、聚合文件中的小文件数据总数、聚合文件的大小、聚合文件中的小文件数据的修改时间戳。

[0129] 由于装置部分的实施例与方法部分的实施例相互对应,因此装置部分的实施例请参见方法部分的实施例的描述,这里暂不赘述。

[0130] 下面对本发明实施例八进行说明。

[0131] 图7为本发明实施例提供的一种数据集处理设备的结构示意图。

[0132] 如图7所示,本发明实施例提供的数据集处理设备包括:

存储器710,用于存储计算机程序711;

处理器720,用于执行计算机程序711,该计算机程序711被处理器720执行时实现如上述任意一项实施例所述数据集处理方法的步骤。

[0133] 其中,处理器720可以包括一个或多个处理核心,比如3核心处理器、8核心处理器等。处理器720可以采用数字信号处理DSP(Digital Signal Processing)、现场可编程门阵列FPGA(Field-Programmable Gate Array)、可编程逻辑阵列PLA(Programmable Logic Array)中的至少一种硬件形式来实现。处理器720也可以包括主处理器和协处理器,主处理器是用于对在唤醒状态下的数据进行处理的处理单元,也称中央处理器CPU(Central Processing Unit);协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中,处理器720可以集成有图像处理GPU(Graphics Processing Unit),GPU用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中,处理器720还可以包括人工智能AI(Artificial Intelligence)处理器,该AI处理器用于处理有关机器学习的计算操作。

[0134] 存储器710可以包括一个或多个计算机可读存储介质,该计算机可读存储介质可以是非暂态的。存储器710还可包括高速随机存取存储器,以及非易失性存储器,比如一个或多个磁盘存储设备、闪存存储设备。本实施例中,存储器710至少用于存储以下计算机程序711,其中,该计算机程序711被处理器720加载并执行之后,能够实现前述任一实施例公开的数据集处理方法中的相关步骤。另外,存储器710所存储的资源还可以包括操作系统712和数据713等,存储方式可以是短暂存储或者永久存储。其中,操作系统712可以为Windows。数据713可以包括但不限于上述方法所涉及到的数据。

[0135] 在一些实施例中,数据集处理设备还可包括有显示屏730、电源740、通信接口750、输入输出接口760、传感器770以及通信总线780。

[0136] 本领域技术人员可以理解,图7中示出的结构并不构成对数据集处理设备的限定,可以包括比图示更多或更少的组件。

[0137] 本发明实施例提供的数据集处理设备,包括存储器和处理器,处理器在执行存储器存储的程序时,能够实现如上所述的数据集处理方法,效果同上。

[0138] 下面对本发明实施例九进行说明。

[0139] 需要说明的是,以上所描述的装置、设备实施例仅仅是示意性的,例如,模块的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个模块或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或模块的间接耦合或通信连接,可以是电性,机械或其它的形式。作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理模块,即可以位于一个地方,或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全

部模块来实现本实施例方案的目的。

[0140] 另外,在本发明各个实施例中的各功能模块可以集成在一个处理模块中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。

[0141] 集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,执行本发明各个实施例所述方法的全部或部分步骤。

[0142] 为此,本发明实施例还提供一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,计算机程序被处理器执行时实现如数据集处理方法的步骤。

[0143] 该计算机可读存储介质可以包括:U盘、移动硬盘、只读存储器ROM(Read-Only Memory)、随机存取存储器RAM(Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0144] 本实施例中提供的计算机可读存储介质所包含的计算机程序能够在被处理器执行时实现如上所述的数据集处理方法的步骤,效果同上。

[0145] 以上对本发明所提供的一种数据集处理方法、装置、设备及计算机可读存储介质进行了详细介绍。说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置、设备及计算机可读存储介质而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以对本发明进行若干改进和修饰,这些改进和修饰也落入本发明权利要求的保护范围内。

[0146] 还需要说明的是,在本说明书中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

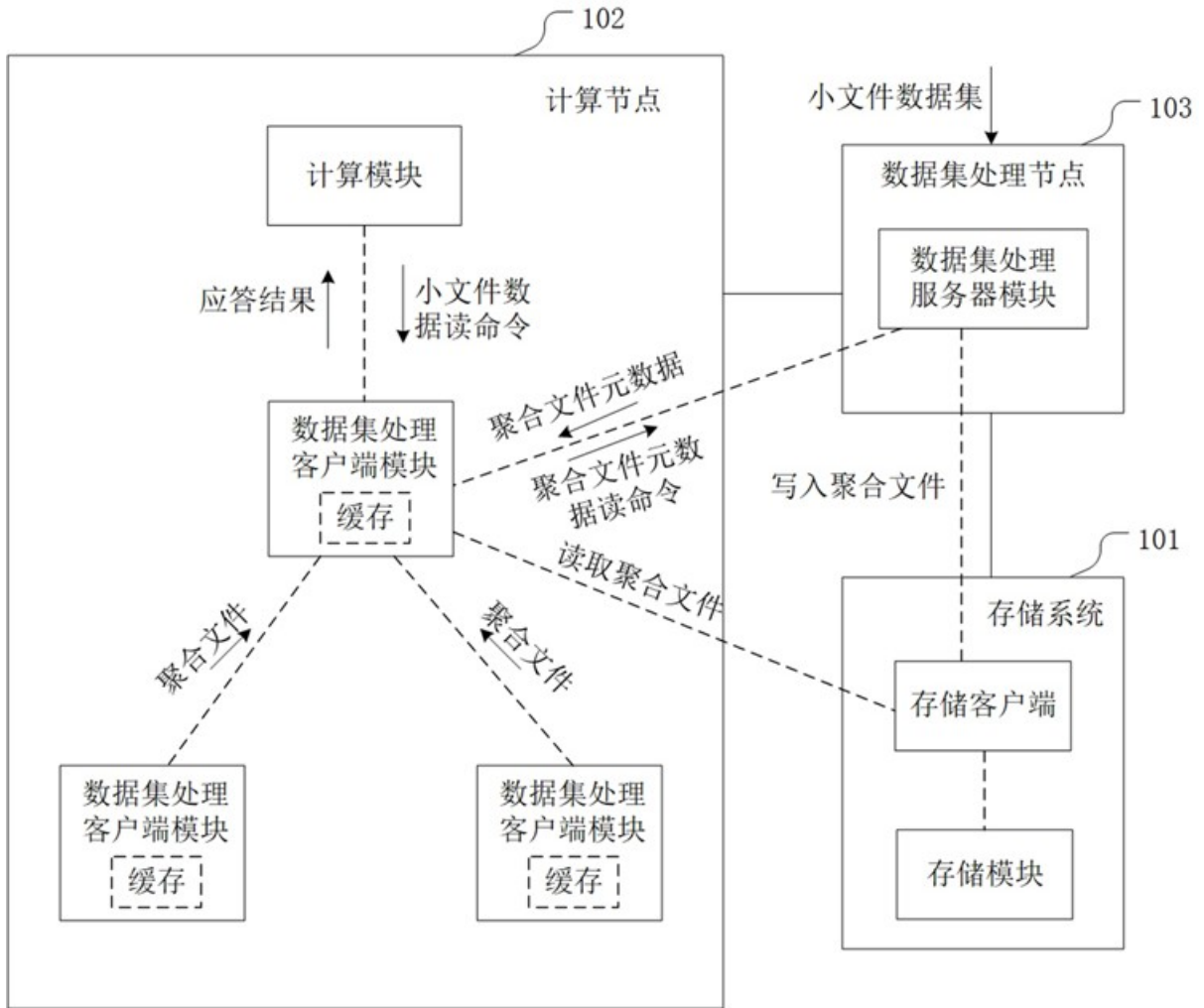


图 1

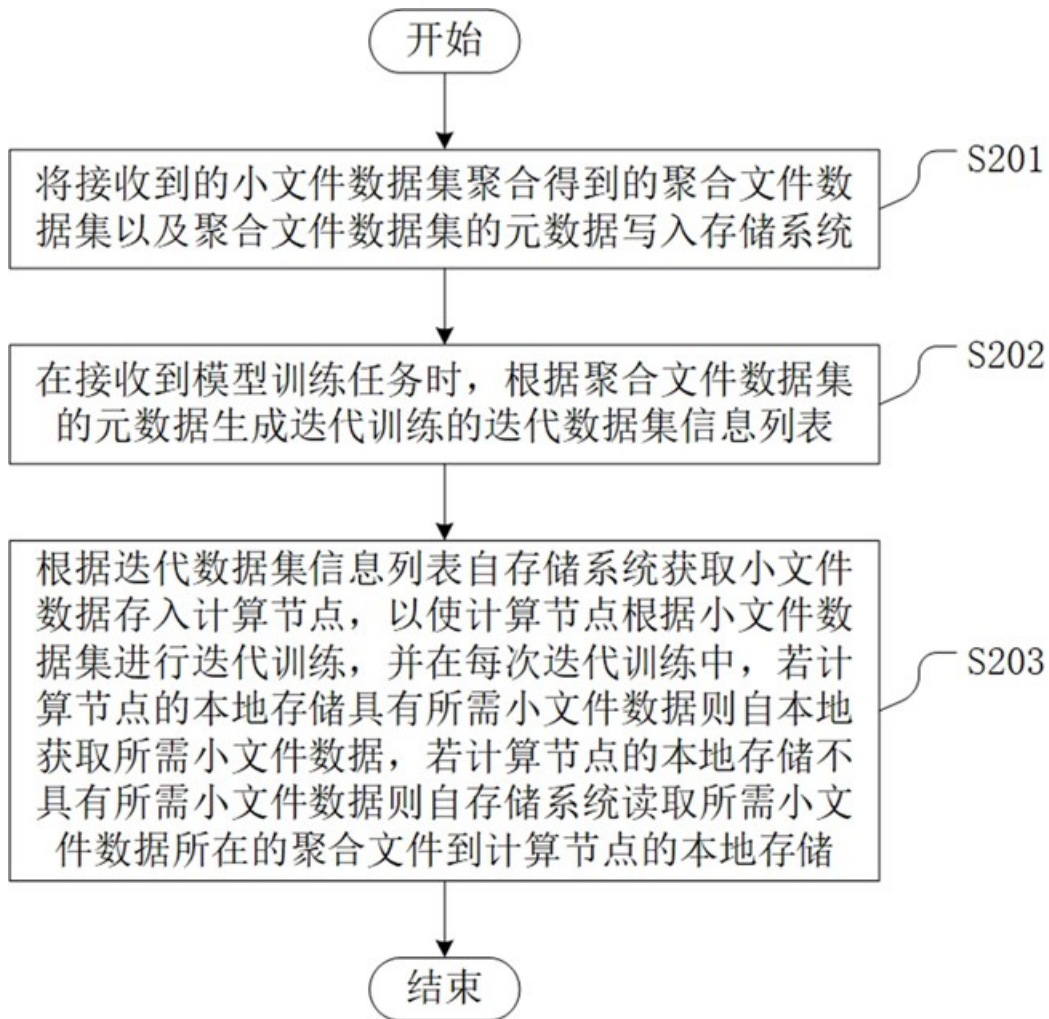


图 2

小文件数据集 数据集处理 存储系统 存储系统 存储系统
服务器模块 存储系统 客户端 存储模块

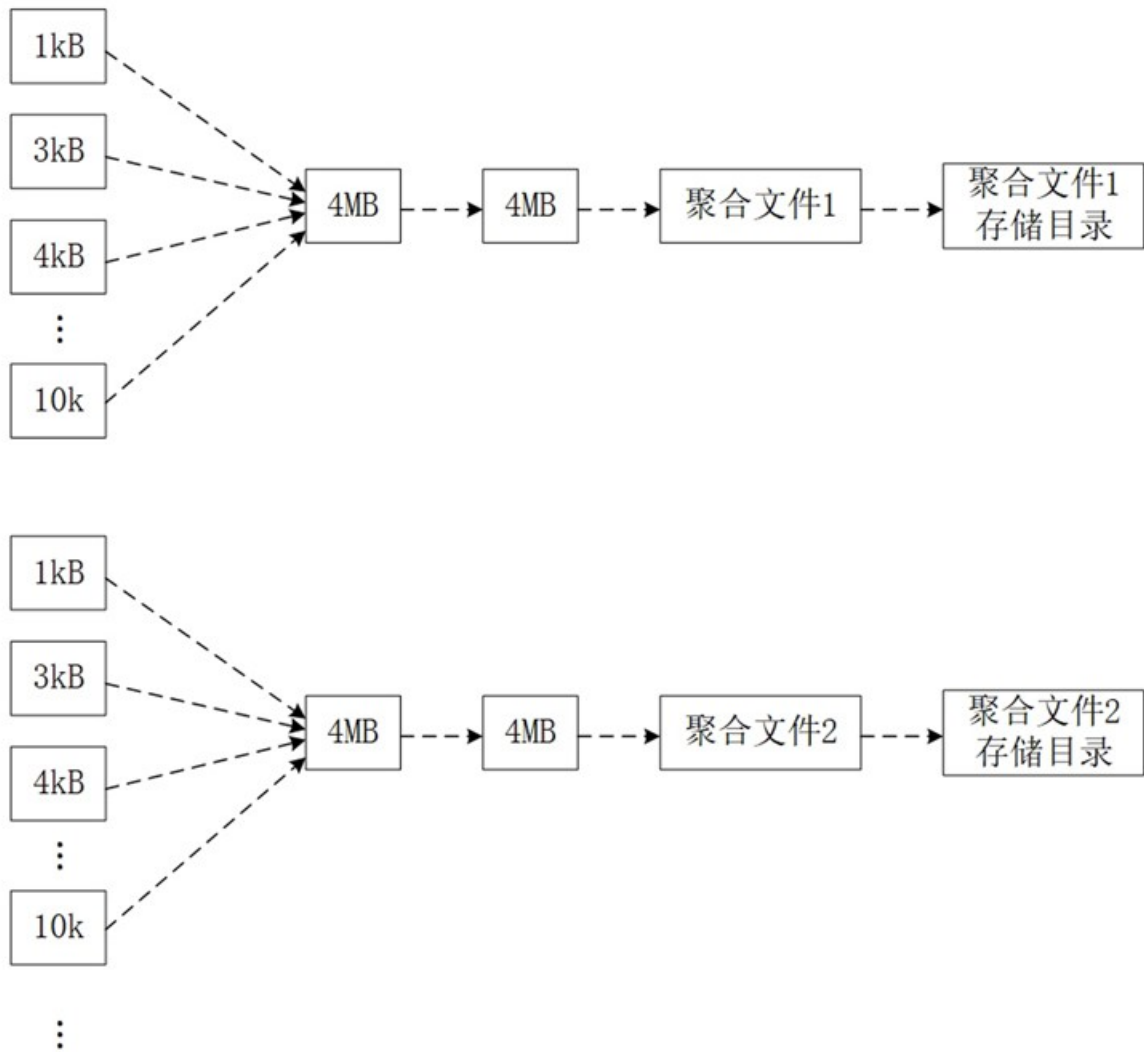


图 3

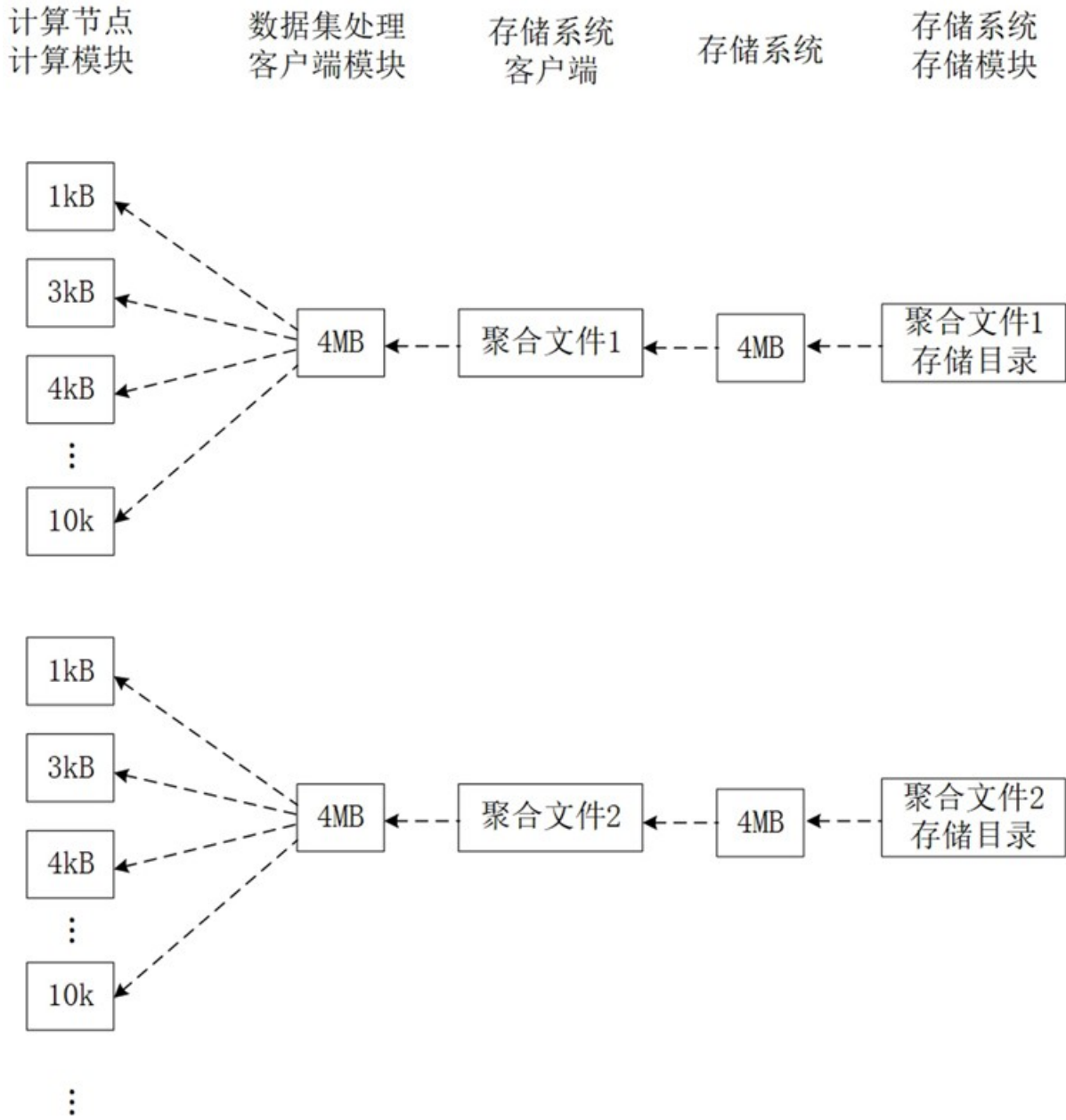


图 4

