

IO 请求的调度方法及装置

技术领域

[01] 本说明书一个或多个实施例涉及计算机领域, 尤其涉及 IO 请求的调度方法及装置。

背景技术

[02] 对于部分数据存储系统, 例如基于日志结构合并树(Log-Structured Merge Tree, LSM) 结构的数据库, 来自应用程序的数据查询请求(也可称为前台任务)发起的 IO 请求需要占用磁盘的 IO 资源外, 还可能存在不定时的后台任务(例如转储任务或者合并任务)所发起的 IO 请求需要占用磁盘的 IO 资源。后台任务发起的 IO 请求占用过多的 IO 资源时, 将会增加前台任务发起的 IO 请求的响应时间, 造成系统卡顿而影响用户体验。

[03] 通常的, 为了确保系统的高效运行以提高用户体验, 部分 IO 请求可能需要能够尽快被处理。然而, 在确保用户具有较好的用户体验的情况下, 无论是前台任务发起的 IO 请求还是后台任务发起的 IO 请求均应当尽快被处理。

[04] 因此希望有一种新的技术方案, 以期能够更好的完成对 IO 请求进行调度。

发明内容

[05] 本说明书一个或多个实施例中提供了一种 IO 请求的调度方法及装置, 能够更好的完成对 IO 请求进行调度。

[06] 第一方面, 提供了一种 IO 请求的调度方法。该方法包括: 获取待调度的目标 IO 请求, 并确定所述 IO 目标请求是否属于被限流的 IO 请求。当所述目标 IO 请求属于未被限流的第一类 IO 请求时, 将当前时刻确定为所述目标 IO 请求的调度时刻; 当所述目标 IO 请求属于被限流的第二类 IO 请求时, 根据当前限流资源比例确定所述目标 IO 请求的调度时刻, 所述当前限流资源比例是当前时刻所属的当前时间周期中用于处理第二类 IO 请求的资源比例, 所述当前限流资源比例根据前一时间周期的前一限流资源比例和前一负载信息计算得到, 所述前一负载信息是前一时间周期内调度的第一类 IO 请求的负载信息, 所述调度时刻不小于当前时刻。然后将所述目标 IO 请求加入 IO 请求集合, 所述 IO 请求集合中的若干 IO 请求按照调度时刻从小到大的顺序依次被发送到磁盘进行处理。

[07] 在一种可能的实施方式中, 第一类 IO 请求包括前台 IO 请求, 第二类 IO 请求包括后台 IO 请求。

[08] 在一种可能的实施方式中, 将所述目标 IO 请求加入 IO 请求集合具体包括: 将所述目标 IO 请求加入通过最小堆来管理的 IO 请求集合。

[09] 在一种可能的实施方式中, 根据当前限流资源比例确定所述目标 IO 请求的调度时刻, 包括: 根据当前限流资源比例以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻, 所述期望调度时刻与当前限流资源比例负相关; 将当前时刻和所述期望调度时刻的较大值确定为所述目标 IO 请求的调度时刻。

[10] 在一种可能的实施方式中, 根据当前限流资源比例确定所述目标 IO 请求的调度时刻, 还包括: 根据磁盘的性能确定所述目标 IO 请求的请求带宽所对应的期望读写频次。相应的, 根据当前限流资源比例以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻, 具体包括: 根据所述期望读写频次、当前限流资源比例, 以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时

刻；其中，所述期望调度时刻与所述期望读写频次负相关。

[11]在一种可能的实施方式中，所述目标 IO 请求属于未被限流的第一类 IO 请求时，所述方法还包括：确定获取所述目标 IO 请求的第一时刻；在第二时刻，从所述 IO 请求集合中选择所述目标 IO 请求，并将所述目标 IO 请求发送到磁盘，其中在所述第二时刻，所述目标 IO 请求的调度时刻是所述 IO 请求集合包含的所有 IO 请求中的最小值；接收磁盘返回的请求结果；向所述目标 IO 请求对应的应用程序发送所述请求结果，并确定发送所述请求结果的第三时刻；根据所述第三时刻和所述第一时刻确定所述目标 IO 请求的响应时间。

[12]在一种可能的实施方式中，还包括：在达到当前时间周期的结束时刻时，根据当前时间周期内调度的第一类 IO 请求的当前负载信息，确定下一时间周期的下一限流资源比例；其中，当前负载信息具体包括，当前时间周期内调度的第一类 IO 请求的平均响应时间，以及当前时间周期内调度的第一类 IO 请求的平均请求带宽。

[13]在一种可能的实施方式中，根据当前时间周期内调度的第一类 IO 请求的当前负载信息，确定下一时间周期的下一限流资源比例，包括：根据磁盘的性能确定所述平均请求带宽对应的期望响应时间；当所述平均响应时间不小于所述期望响应时间时，按照预设步长减小所述当前限流资源比例，获得下一时间周期的下一限流资源比例；或者，当所述平均响应时间小于所述期望响应时间时，按照预设步长增大所述当前限流资源比例，获得下一时间周期的下一限流资源比例。

[14]在一种可能的实施方式中，下一时间周期的下一限流资源比例不大于第一预设阈值且不小于第二预设阈值。

[15]第二方面，提供了一种 IO 请求的调度方法，包括：获取 IO 请求集合，所述 IO 请求集合通过第一方面中任一项所述的方法得到，所述 IO 请求集合中包括若干 IO 请求，每个 IO 请求具有对应的调度时刻；从所述 IO 请求集合中选择调度时刻最小的 IO 请求作为当前 IO 请求；将所述当前 IO 请求发送到磁盘，使磁盘处理所述当前 IO 请求并返回请求结果。

[16]在一种可能的实施方式中，将所述当前 IO 请求发送到磁盘，具体包括：在所述当前 IO 请求对应的调度时刻小于当前时刻时，将所述当前 IO 请求发送到磁盘。

[17]第三方面，提供了一种 IO 请求的调度装置，包括：请求获取单元，配置为获取待调度的目标 IO 请求；分类确定单元，配置为确定所述目标 IO 请求是否属于被限流的 IO 请求；调度时刻确定单元，配置为当所述目标 IO 请求属于未被限流的第一类 IO 请求时，将当前时刻确定为所述目标 IO 请求的调度时刻；当所述目标 IO 请求属于被限流的第二类 IO 请求时，根据当前限流资源比例确定所述目标 IO 请求的调度时刻，所述当前限流资源比例是当前时刻所属的当前时间周期中用于处理第二类 IO 请求的资源比例，所述当前限流资源比例根据前一时间周期的前一限流资源比例和前一负载信息计算得到，所述前一负载信息是前一时间周期内调度的第一类 IO 请求的负载信息，所述调度时刻不小于当前时刻；请求调度单元，配置为将所述目标 IO 请求加入 IO 请求集合，所述 IO 请求集合中的若干 IO 请求按照调度时刻从小到大的顺序依次被发送到磁盘进行处理。

[18]在一种可能的实施方式中，第一类 IO 请求包括前台 IO 请求，第二类 IO 请求包括后台 IO 请求。

[19]在一种可能的实施方式中，所述请求调度单元，具体配置为将所述目标 IO 请求加入通过最小堆来管理的 IO 请求集合。

[20]在一种可能的实施方式中，所述调度时刻确定单元，包括：计算子单元，配置为根

据当前限流资源比例以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻, 所述期望调度时刻与当前限流资源比例负相关; 第一确定子单元, 配置为将当前时刻和所述期望调度时刻的较大值确定为所述目标 IO 请求的调度时刻。

[21] 在一种可能的实施方式中, 所述调度时刻确定单元还包括第二确定子单元, 配置为根据磁盘的性能确定所述目标 IO 请求的请求带宽所对应的期望读写频次; 所述计算子单元, 具体配置为根据所述期望读写频次、当前限流资源比例, 以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻; 其中, 所述期望调度时刻与所述期望读写频次负相关。

[22] 在一种可能的实施方式中, 所述装置还包括请求处理单元, 配置为所述目标 IO 请求属于未被限流的第一类 IO 请求时, 确定获取所述目标 IO 请求的第一时刻; 在第二时刻, 从所述 IO 请求集合中选择所述目标 IO 请求, 并将所述目标 IO 请求发送到磁盘, 其中在所述第二时刻, 所述目标 IO 请求的调度时刻是所述 IO 请求集合包含的所有 IO 请求中的最小值; 接收磁盘返回的请求结果; 向所述目标 IO 请求对应的应用程序发送所述请求结果, 并确定发送所述请求结果的第三时刻; 根据所述第三时刻和所述第一时刻确定所述目标 IO 请求的响应时间。

[23] 在一种可能的实施方式中, 还包括计算单元, 配置为在达到当前时间周期的结束时刻时, 根据当前时间周期内调度的第一类 IO 请求的当前负载信息, 确定下一时间周期的下一限流资源比例; 其中, 当前负载信息具体包括, 当前时间周期内调度的第一类 IO 请求的平均响应时间, 以及当前时间周期内调度的第一类 IO 请求的平均请求带宽。

[24] 在一种可能的实施方式中, 所述计算单元具体配置为根据磁盘的性能确定所述平均请求带宽对应的期望响应时间; 当所述平均响应时间不小于所述期望响应时间时, 按照预设步长减小所述当前限流资源比例, 获得下一时间周期的下一限流资源比例; 或者, 当所述平均响应时间小于所述期望响应时间时, 按照预设步长增大所述当前限流资源比例, 获得下一时间周期的下一限流资源比例。

[25] 在一种可能的实施方式中, 下一时间周期的下一限流资源比例不大于第一预设阈值且不小于第二预设阈值。

[26] 第四方面, 提供了一种 IO 请求的调度装置, 包括: 获取单元, 配置为获取 IO 请求集合, 所述 IO 请求集合通过第三方面中任一项所述的装置得到, 所述 IO 请求集合中包括若干 IO 请求, 每个 IO 请求具有对应的调度时刻; 选择单元, 配置为从所述 IO 请求集合中选择调度时刻最小的 IO 请求作为当前 IO 请求; 发送单元, 配置为将所述当前 IO 请求发送到磁盘, 使磁盘处理所述当前 IO 请求并返回请求结果。

[27] 在一种可能的实施方式中, 所述发送单元配置为在所述当前 IO 请求对应的调度时刻小于当前时刻时, 将所述当前 IO 请求发送到磁盘。

[28] 第五方面, 提供了一种计算机可读存储介质, 其上存储有计算机程序, 当所述计算机程序在计算设备中执行时, 计算设备实现第一方面或第二方面中任一项所述的方法。

[29] 第六方面, 提供了一种计算设备, 包括存储器和处理器, 所述存储器中存储有计算机程序, 所述处理器执行所述计算机程序时, 实现第一方面或第二方面中任一项所述的方法。

[30] 通过本说明书一个或多个实施例中提供的方法及装置, 通过将 IO 请求划分为未被限流的第一类 IO 和被限流 IO 的第二类 IO 请求, 根据前一时间周期内调度的第一类 IO 请求的负载信息, 动态调整当前时间周期内第二类 IO 请求的限流情况, 实现根据磁盘的负载自适应的分配 IO 资源。如此, 既可在磁盘高负载时降低第二类 IO 请求对 IO 资

源的占用，确保未被限流的第一类 IO 请求能够尽快被执行，即确保系统的高效运行以提高用户体验，还可在磁盘低负载时增加第二类 IO 对 IO 资源的占用，加速第二类 IO 请求的执行，从而实现更好的完成对 IO 请求进行调度。

附图说明

[31]为了更清楚地说明本说明书实施例的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，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它的附图。

[32]图 1 为本说明书实施例中提供的一种 IO 请求的调度方法的流程图；

[33]图 2 为本说明书实施例中确定第二类 IO 请求的调度时刻的过程示意图；

[34]图 3 为本说明书实施例中通过最小堆管理 IO 请求集合的示意图；

[35]图 4 为本说明书实施例中提供的另一种 IO 请求的调度方法的流程图；

[36]图 5 为本说明书实施例中获取下一时间周期的资源比例的过程示意图；

[37]图 6 为本说明书实施例中提供的又一种 IO 请求的调度方法的流程图；

[38]图 7 为本说明书实施例中提供的一种 IO 请求的调度装置的结构示意图；

[39]图 8 为本说明书实施例中提供的另一种 IO 请求的调度装置的结构示意图。

具体实施方式

[40]下面结合附图，对本说明书所提供的各个非限制性实施例进行详细描述。

[41]包含基于 LSM 结构的数据库在内的部分数据存储系统，在单位时间内调度的 IO 请求通常包括前台 IO 请求和后台 IO 请求等两种类型的 IO 请求。

[42]以基于 LSM 结构的数据库为例。写入新数据的过程包括：首先，活跃 Memtable 作为 LSM 的内存数据结构，将负责应用程序对新数据的写入；活跃 Memtable 中写入相应大小的新数据后将会停止写入，并且被冻结为冻结 Memtable，此外还会创建新的活跃 Memtable 来负责新数据的写入。接着，系统后台可以调度转储任务，通过转储任务将冻结 Memtable 转换格式转储(Dump)到磁盘中生成 SSTable 结构文件，此后该冻结 Memtable 占用的内存可以被释放以供后续的活跃 Memtable 使用，其中 SSTable 结构文件也被表述为转储 SSTable。此外，还可以在转储 SSTable 积累到相应数量后，由系统后台调度合并任务将相应数量的转储 SSTable 合并成基线 SSTable 并写入磁盘。其中需要说明的是，转储任务或者合并任务将会通过向磁盘发起若干 IO 请求来实现向磁盘写入转储 SSTable 或者基线 SSTable，其中由转储任务和/或合并任务等后台任务发起的 IO 请求均属于后台 IO 请求。

[43]以基于 LSM 结构的数据库为例。应用程序读取数据时需要发起数据查询请求（数据查询请求属于前台任务），首先会基于数据查询请求查询内存中的 Memtable。如果内存中的 Memtable 并不包含数据查询请求期望查询的目标数据，则需要向磁盘发起若干 IO 请求来实现从磁盘中的 SSTable 读取目标数据，数据查询请求发起的 IO 请求属于前台 IO 请求。

[44]后台任务发起的 IO 请求和前台任务发起的 IO 请求将会对磁盘的 IO 资源形成竞争。例如在转储任务和合并任务执行期间，往往需要通过大量的后台 IO 请求实现向磁盘写

入大规模数据;如果此时前台任务较多,由于后台 IO 请求占用过多的磁盘带宽/IO 资源,将会导致前台任务所对应的前台 IO 请求的响应时间增加,进而造成整个前台任务的响应时间增加甚至超时失败。

[45]需要说明的是,数据查询请求查询多个数据源,或者说需要通过超过预设数量的 IO 请求来实现从磁盘中的 SSTable 读取某个数据查询请求期望读取的目标数据时,该数据查询请求可以被称为大查询请求。大查询请求会影响其它前台任务所发起的 IO 请求的响应时间,造成其它前台任务的响应时间增加甚至超时失败。

[46]通常需要对部分 IO 请求进行直接限流,例如对后台 IO 请求和/或大查询请求对应的 IO 请求进行直接限流,使其它 IO 请求能够尽快被处理,确保系统的高效运行以提高用户体验。然而,如果对部分 IO 请求进行直接限流,可能会造成被限流的 IO 请求始终无法被快速执行而带来其它问题,例如对后台 IO 请求进行直接限流则可能导致转储任务和合并任何执行较慢,甚至造成内存空间不足而无法提供更多的活跃 Memtable 来写入新数据。

[47]考虑到以上问题,本说明书实施例中至少提供了一种 IO 请求的调度方法和装置。通过将 IO 请求划分为未被限流的第一类 IO 和被限流 IO 的第二类 IO 请求,根据前一时间周期内调度的第一类 IO 请求的负载信息,动态调整当前时间周期内第二类 IO 请求的限流情况,实现根据磁盘的负载自适应的分配 IO 资源。如此,既可在磁盘高负载时降低第二类 IO 请求对 IO 资源的占用,确保未被限流的第一类 IO 请求能够尽快被执行,即确保系统的高效运行以提高用户体验,还可在磁盘低负载时增加第二类 IO 对 IO 资源的占用,加速第二类 IO 请求的执行,从而实现更好的完成对 IO 请求进行调度。

[48]图 1 为本说明书实施例中提供的一种 IO 请求的调度方法的流程图。该方法可以由计算设备执行,该方法至少可以包括如下步骤:步骤 101,获取待调度的目标 IO 请求。

[49]该目标 IO 请求可以包括前台 IO 请求或者后台 IO 请求。

[50]步骤 102,确定目标 IO 请求是否属于被限流的 IO 请求。

[51]可以将 IO 请求划分为两个分类,即划分为未被限流的第一类 IO 请求和被限流的第二类 IO 请求;具体如何划分第一类 IO 请求和第二类 IO 请求,可以结合实际业务需求进行灵活配置。例如,可以将全部的前台 IO 请求划分到未被限流的第一类 IO 请求,将全部的后台 IO 请求划分到被限流的第二类 IO 请求;或者,将前台 IO 请求中除大查询请求对应的 IO 请求以外的其它前台 IO 请求划分到第一类 IO 请求,将大查询请求对应的 IO 请求以及全部的后台 IO 请求划分到第二类 IO 请求。

[52]当目标 IO 请求属于未被限流的第一类 IO 请求时,执行步骤 103,将当前时刻确定为 IO 请求的调度时刻,并接着执行步骤 105。

[53]当目标 IO 请求属于被限流的第二类 IO 请求时,执行步骤 104,根据当前限流资源比例确定目标 IO 请求的调度时刻,并接着执行步骤 105。

[54]当前限流资源比例是当前时刻所属的当前时间周期中用于处理第二类 IO 请求的资源比例,该比例越大,则允许更多的 IO 资源被用于处理第二类 IO 请求。在一个示例中,当前限流资源比例根据前一时间周期的前一限流资源比例和前一负载信息计算得到,前一负载信息是前一时间周期内调度的第一类 IO 请求的负载信息,并且该目标 IO 请求的调度时刻不小于当前时刻。

[55]在一个较为具体的示例中,如图 2 所示,步骤 104 可以包括:步骤 1041,根据磁盘的性能确定目标 IO 请求的请求带宽对应的期望读写频次。

[56]可以通过 io_benchmark 或者其它磁盘性能测量工具,在初始化计算设备的数据存储

系统时对磁盘的性能进行测量，以获得磁盘的性能。

[57]举例来说，可以测量磁盘处理样本请求带宽（后续利用 io_size 表征单个 IO 请求的请求带宽）的前台 IO 请求时，样本 io_size 对应的样本响应时间（后续利用 io_rt 表征单个 IO 请求的响应时间）和样本 IOPS（IO request count per second，每秒 IO 请求计数）。接着对该组样本 io_size、样本 io_rt 和样本 IOPS 进行存储，例如存储至配置文件 io_resource.conf 中。测量并存储多组样本 io_size、样本 io_rt 和样本 IOPS 之后，通常来说不同组中的样本 io_size 和样本 IOPS 的乘积相等或者近似于相等，并且样本 io_size 和样本 IOPS 的乘积能够用于表征磁盘提供的 IO 资源/带宽。因此，可以通过对多个组中的样本 io_size 和样本 IOPS 进行线性拟合，例如进行对数线性拟合，得到样本 io_size 和样本 IOPS 之间的第一线性关系，即得到变量 io_size 和变量 IOPS 之间的第一线性关系；以及，通过对多个组中的样本 io_size 和样本 io_rt 进行线性拟合，例如进行简单线性拟合，得到样本 io_size 和样本 io_rt 之间的第二线性关系，即得到变量 io_size 和变量 io_rt 之间的第二线性关系。最后，对第一线性关系和第二线性关系进行存储，以便后续对第一线性关系和第二关系进行使用，其中第一线性关系和第二线性关系可以用于描述磁盘的性能。

[58]相应的，步骤 1041 中具体可以将目标 IO 请求的 io_size 代入第一线性关系中的变量 io_size，计算得到第一线性关系中变量 IOPS 的取值，变量 IOPS 的取值即为目标 IO 请求的 io_size 对应的期望读写频次。

[59]步骤 1042，根据期望读写频次、当前限流资源比例，以及获取的上一个第二类 IO 请求的调度时刻，计算目标 IO 请求的期望调度时刻。

[60]目标 IO 请求的期望调度时刻可以是在上一个第二类 IO 请求的调度时刻上加上期望时间差，该期望时间差与期望读写频次负相关，也与当前限流资源比例负相关。换言之，期望读写频次越大，当前限流资源比例越大，则预期会在处理上一个第二类 IO 请求后更短的时间内，处理本次获取的目标 IO 请求，因此期望时间差越小。

[61]例如，可以通过如下公式 1 计算目标 IO 请求的期望调度时刻：

$$\text{Expect_deadline} = \text{last_deadline} + 1 / (a * \text{Expect_IOPS}) \quad (1)$$

[62]其中，Expect_deadline 表征目标 IO 请求的期望调度时刻，last_deadline 表征在先获取的上一个第二类 IO 请求的调度时刻，a 表征当前限流资源比例，Expect_IOPS 表征期望读写频次。

[63]需要说明的是，步骤 1041 是可选地。与此相应的，在步骤 1042 中可以根据当前限流资源比例以及上一个第二类 IO 请求的调度时刻，计算本次获取的目标 IO 请求的期望调度时刻，例如前述公式 1 中的 Expect_IOPS 可以作为非必要的参数而被删除。

[64]在步骤 1043，将当前时刻和期望调度时刻中的较大值确定为目标 IO 请求的调度时刻。

[65]当前时刻是较大值时，说明磁盘负载可能相对较小，当前时刻作为目标 IO 请求的调度时刻能够使 IO 请求尽快被处理。

[66]期望调度时刻是较大值时，说明获取的上一个第二类 IO 请求可能还未被发送到磁盘进行处理，磁盘负载可能相对较大，将期望调度时刻作为目标 IO 请求的调度时刻能够对第二类 IO 请求进行限流，单位时间内能够有更多的 IO 资源被用于处理第一类 IO 请求，同时还能确保获取的前一个第二类 IO 请求相比于目标 IO 请求能够更早的被处理。

[67]回到图 1，在步骤 105，将目标 IO 请求加入 IO 请求集合。

[68]IO 请求集合中的若干 IO 请求按照调度时刻从小到大的顺序依次被发送到磁盘进行处理。在一个较为具体的示例中，为了方便后续过程中快速的实现从 IO 请求集合中选择调度时刻最小的 IO 请求，IO 请求集合中可以按照调度时刻由小到大的顺序，对 IO 请求集合中的若干 IO 请求进行排序以形成 IO 请求序列；或者，通过堆结构中的最小堆来管理 IO 请求集合，最小堆是已经按照调度时刻的大小进行排序的完全二叉树，其中作为根节点的 IO 请求的调度时刻是若干 IO 请求中的最小值。

[69]举例来说，IO 请求集合中包括 IO 请求 A1、A2、A3、A4 以及本次加入 IO 请求集合中的目标 IO 请求 A5，假设按照调度时刻由小到大的顺序排列时可以形成序列 A1、A5、A2、A3、A4，通过最小堆来管理 IO 请求集合时，则可能形成如图 3 中所示根节点为 A1 的完全二叉树。其中需要说明的是，该完全二叉树仅仅是示例性，即除了根节点为 A1 外，其它节点的所在位置可能不同于如图 3 中所示的完全二叉树。

[70]在一些实施例中，在图 1 中获取的目标 IO 请求属于未被限流的第一类 IO 请求时，还可以进一步根据该目标 IO 请求从 IO 请求集合中被读出处理的情况，确定其响应时间，该响应时间用于确定第一类 IO 请求的负载信息，从而更新前述的限流资源比例。下面参考图 4，描述属于第一类 IO 请求的目标 IO 请求的处理过程。如图 4 所示，该处理过程在图 1 基础上，还可以包括进一步的处理步骤，下面详细描述。

[71]在通过前述步骤 102 确定目标 IO 请求为第一类 IO 请求的情况下，还执行步骤 401，确定获取目标 IO 请求的当前时刻，记为第一时刻。

[72]此外，如前所述，在步骤 105 将前述目标 IO 请求加入到 IO 请求集合中，并且该 IO 请求集合按照调度时刻从小到大的顺序依次读出 IO 请求。也就是说，每次从 IO 请求集合读取 IO 请求进行处理时，IO 请求集合将会“弹出”当前调度时刻最小的 IO 请求。如此，随着调度时刻更小的 IO 请求依次被读出，在某些时刻，该目标 IO 请求成为 IO 请求集合中调度时刻最小的 IO 请求。此时，从 IO 请求集合中读取 IO 请求时，将读出上述目标 IO 请求。将该时刻记为第二时刻。

[73]于是，在步骤 402，在第二时刻，从 IO 请求集合中选择上述目标 IO 请求，并将其发送到磁盘。其中，在第二时刻，上述目标 IO 请求的调度时刻是 IO 请求集合包含的所有 IO 请求中的最小值。

[74]步骤 403，接收磁盘返回的请求结果。

[75]步骤 404，向目标 IO 请求对应的应用程序发送请求结果，并确定发送请求结果的第三时刻。

[76]步骤 405，根据第三时刻和第一时刻确定目标 IO 请求的响应时间。

[77]即将第三时刻和第二时刻的时间差确定为目标 IO 请求的响应时间。

[78]在一些实施例中，为了方便后续对下一时间周期的下一限流资源比例进行更新，在步骤 405 之后还可以执行步骤 406，根据目标 IO 请求的响应时间和请求带宽，更新已调度的所有第一类 IO 请求的累计请求带宽和累计响应时间，以及更新已调度的所有第一类 IO 请求的累计数量。

[79]举例来说，可以定义用于存储累计请求带宽的参数 `cumulative_size`，定义用于存储累计响应时间的参数 `cumulative_rt`，以及定义用于存储累计数量的参数 `cumulative_count`，其中前述三个参数的初始值均可以为 0。步骤 406 中具体根据目标 IO 请求的请求带宽和响应时间，对 `cumulative_size` 以及 `cumulative_rt` 的值进行更新，并将 `cumulative_count` 的值加 1。

[80]在一些实施例中，还可以对当前处理的目标 IO 请求的响应时间和请求带宽进行独

立存储,对当前时间周期内已调度的第一类 IO 请求的数量进行独立更新,以便根据这些数据确定当前时间周期内调度的第一类 IO 请求的当前负载信息,并基于当前负载信息获取下一时间周期的下一限流资源比例。

[81]与之相应的,当达到当前时间周期的结束时刻时,还可以执行步骤 407,确定当前时间周期内调度的第一类 IO 请求的当前负载信息。以及执行步骤 408,根据当前负载信息获取下一时间周期的下一限流资源比例。

[82]单个时间周期的时长可以灵活配置,例如配置为 100ms 或者其它数值。

[83]当前负载信息包括当前时间周期内调度的第一类 IO 请求的平均响应时间,以及当前时间周期内调度的第一类 IO 请求的平均请求带宽。

[84]举例来说,达到前一时间周期的结束时刻时,cumulative_count 的取值为 last_count、cumulative_size 的值为 last_size、cumulative_rt 的值为 last_rt。达到当前时间周期的结束时刻时,cumulative_count 的值为 current_count、cumulative_size 的值为 current_size、cumulative_rt 的值为 current_rt。则,

[85]可以通过如下公式 2 计算平均请求带宽:

$$\text{avg_size} = (\text{current_size} - \text{last_size}) / (\text{current_count} - \text{last_count}) \quad (2)$$

[86]其中,avg_size 表征平均请求带宽。

[87]可以通过如下公式 3 计算平均响应时间:

$$\text{avg_rt} = (\text{current_rt} - \text{last_rt}) / (\text{current_count} - \text{last_count}) \quad (3)$$

[88]其中,avg_rt 表征平均响应时间。

[89]在一个较为具体的示例中,请参考图 5,步骤 408 可以包括:步骤 4081,根据磁盘的性能确定平均请求带宽对应的期望响应时间。

[90]如前所述,磁盘的性能可通过第一线性关系和第二线性关系来描述。这里可以将平均请求带宽代入第二线性关系中的变量 io_size,计算出第二线性关系中变量 io_rt 的取值,变量 io_rt 的取值即为平均请求带宽所对应的期望响应时间。

[91]步骤 4082,确定平均响应时间是否小于期望响应时间。

[92]如果是,则执行步骤 4083,按照预设步长增大当前限流资源比例,获得下一时间周期的限流资源比例。

[93] 如果否,则执行步骤 4084,按照预设步长减小当前限流资源比例,获得下一时间周期的限流资源比例。

[94] 如果平均响应时间小于期望响应时间,说明磁盘负载相对较低,增大限流资源比例,减小对第二类 IO 请求的限流力度,下一时间周期内能够有更多的 IO 资源被用于处理第二类 IO 请求。如果平均响应时间不小于期望响应时间,说明磁盘负载相对较高,减小限流资源比例,增大对第二类 IO 请求的限流力度,下一时间周期内能够有更多的 IO 资源被用于处理第一类 IO 请求。

[95] 在一些实施例中,可以预先设置限流资源比例的取值范围,即预先设置限流资源比例的取值范围不大于第一预设阈值且不小于第二预设阈值。相应的,如果当前时间周期的当前限流资源比例为第一预设阈值或第二预设阈值,则将当前限流资源比例继续作为下一时间周期的限流资源比例,不再通过步骤 4083 或步骤 4084 重新获得下一时间周期的限流资源比例。

[96] 第一预设阈值、第二预设阈值以及预设步长可以灵活配置，例如将第一预设阈值配置为 50%，将第二预设阈值配置为 10%，将预设步长配置为 1%。

[97] 与前述方法实施例对应的，本说明书实施例中还提供了另一种 IO 请求的调度方法，该方法由计算设备执行。如图 6 所示，该方法包括：步骤 601，获取 IO 请求集合。

[98] 该 IO 请求集合中包括若干 IO 请求，每个 IO 请求均是通过如图 1 或图 4 所示实施例中的方式加入到 IO 请求集合的。

[99] 步骤 602，从所述 IO 请求集合中选择调度时刻最小的 IO 请求作为当前 IO 请求。其中，当前 IO 请求可能属于未被限流的第一类 IO 请求，也可能属于被限流的第二类 IO 请求。

[100] 步骤 603，将当前 IO 请求发送到磁盘。

[101] 在一种可能的实施方式中，选择当前 IO 请求之后，如果存在能够用于将当前 IO 请求发送到磁盘的 IO 资源，则立即执行步骤 603。

[102] 在一种可能的实施方式中，选择当前 IO 请求之后，还可以确定当前 IO 请求的调度时刻是否小于当前时刻。如果当前 IO 请求的调度时刻小于当前时刻，并且存在能够用于将当前 IO 请求发送到磁盘的 IO 资源，则执行步骤 602。如果当前 IO 请求的调度时刻不小于当前时刻，即使存在能够用于将当前 IO 请求发送到磁盘的 IO 资源，也将再次通过步骤 602 从 IO 请求集合中重新选择调度时刻最小的 IO 请求作为新的当前 IO 请求；对于已经选择的并且未能被发送到磁盘的 IO 请求，可以在当前时刻大于该 IO 请求的调度时刻时，通过步骤 603 将该 IO 请求发送到磁盘。

[103] 当前 IO 请求属于第一类 IO 请求时，从 IO 请求集合中选择当前 IO 请求的第二时刻大于当前 IO 请求的调度时刻，当前时刻必然大于当前 IO 请求的调度时刻。前述的两种实施方式均可确保当前 IO 请求能够尽快被发送到磁盘进行处理。

[104] 当前 IO 请求属于第二类 IO 请求时，从 IO 请求集合中选择当前 IO 请求之后的当前时刻可能小于当前 IO 请求的调度时刻。当前时刻小于当前 IO 请求的调度时刻，说明磁盘在当前时刻的负载可能相对较小。如果将当前 IO 请求直接发送到磁盘进行处理，能够增加第二类 IO 请求在单位时间内占用的 IO 资源，加速第二类 IO 请求的处理；如果直到当前时刻小于当前 IO 请求的调度时刻时，才将当前 IO 请求发送到磁盘进行处理，能够确保第二类 IO 请求不会过多的占用 IO 资源，第一类 IO 请求能够尽快被发送到磁盘进行处理。

[105] 需要说明的是，当前 IO 请求属于第一类 IO 请求时，该方法还可以包括前述如图 4 所示方法实施例中步骤 403 至步骤 406 中的部分或全部。

[106] 与前述方法实施例基于相同的构思，本说明书实施例中还提供了一种 IO 请求的调度装置。如图 7 所示，该装置包括：请求获取单元 701，配置为获取待调度的目标 IO 请求；分类确定单元 703，配置为确定所述目标 IO 请求是否属于被限流的 IO 请求；调度时刻确定单元 705，配置为当所述目标 IO 请求属于未被限流的第一类 IO 请求时，将当前时刻确定为所述目标 IO 请求的调度时刻；当所述目标 IO 请求属于被限流的第二类 IO 请求时，根据当前限流资源比例确定所述目标 IO 请求的调度时刻，所述当前限流资源比例是当前时刻所属的当前时间周期中用于处理第二类 IO 请求的资源比例，所述当前限流资源比例根据前一时间周期的前一限流资源比例和前一负载信息计算得到，所述前一负载信息是前一时间周期内调度的第一类 IO 请求的负载信息，所述调度时刻不小于当前时刻；请求调度单元 707，配置为将所述目标 IO 请求加入 IO 请求集合，所述 IO 请求集合中的若干 IO 请求按照调度时刻从小到大的顺序依次被发送到磁盘进行处理。

[107] 在一种可能的实施方式中，第一类 IO 请求包括前台 IO 请求，第二类 IO 请求包括后台 IO 请求。

[108] 在一种可能的实施方式中，所述请求调度单元 707，具体配置为将所述目标 IO 请求加入通过最小堆来管理的 IO 请求集合。

[109] 在一种可能的实施方式中，所述调度时刻确定单元 705，包括：计算子单元，配置为根据当前限流资源比例以及获取的上一个第二类 IO 请求的调度时刻，计算所述目标 IO 请求的期望调度时刻，所述期望调度时刻与当前限流资源比例负相关；第一确定子单元，配置为将当前时刻和所述期望调度时刻的较大值确定为所述目标 IO 请求的调度时刻。

[110] 在一种可能的实施方式中，所述调度时刻确定单元 705 还包括第二确定子单元，配置为根据磁盘的性能确定所述目标 IO 请求的请求带宽所对应的期望读写频次；所述计算子单元，具体配置为根据所述期望读写频次、当前限流资源比例，以及获取的上一个第二类 IO 请求的调度时刻，计算所述目标 IO 请求的期望调度时刻；其中，所述期望调度时刻与所述期望读写频次负相关。

[111] 在一种可能的实施方式中，所述装置还包括请求处理单元 709，配置为在所述目标 IO 请求属于未被限流的第一类 IO 请求时，确定获取所述目标 IO 请求的第一时刻；在第二时刻，从所述 IO 请求集合中选择所述目标 IO 请求，并将所述目标 IO 请求发送到磁盘，其中在所述第二时刻，所述目标 IO 请求的调度时刻是所述 IO 请求集合包含的所有 IO 请求中的最小值；接收磁盘返回的请求结果；向所述目标 IO 请求对应的应用程序发送所述请求结果，并确定发送所述请求结果的第三时刻；根据所述第三时刻和所述第一时刻确定所述目标 IO 请求的响应时间。

[112] 在一种可能的实施方式中，还包括计算单元 711，配置为在达到当前时间周期的结束时刻时，根据当前时间周期内调度的第一类 IO 请求的当前负载信息，确定下一时间周期的下一限流资源比例；其中，当前负载信息具体包括，当前时间周期内调度的第一类 IO 请求的平均响应时间，以及当前时间周期内调度的第一类 IO 请求的平均请求带宽。

[113] 在一种可能的实施方式中，所述计算单元 711 具体配置为根据磁盘的性能确定所述平均请求带宽对应的期望响应时间；当所述平均响应时间不小于所述期望响应时间时，按照预设步长减小所述当前限流资源比例，获得下一时间周期的下一限流资源比例；或者，当所述平均响应时间小于所述期望响应时间时，按照预设步长增大所述当前限流资源比例，获得下一时间周期的下一限流资源比例。

[114] 在一种可能的实施方式中，下一时间周期的下一限流资源比例不大于第一预设阈值且不小于第二预设阈值。

[115] 与前述方法实施例基于相同的构思，本说明书实施例中还提供了一种 IO 请求的调度装置。如图 8 所示，该装置包括：获取单元 801，配置为获取 IO 请求集合，所述 IO 请求集合通过如图 1 或图 4 中所示方法得到，所述 IO 请求集合中包括若干 IO 请求，每个 IO 请求具有对应的调度时刻；选择单元 803，配置为从所述 IO 请求集合中选择调度时刻最小的 IO 请求作为当前 IO 请求；发送单元 805，配置为将所述当前 IO 请求发送到磁盘，使磁盘处理所述当前 IO 请求并返回请求结果。

[116] 在一种可能的实施方式中，所述发送单元 805 配置为在所述当前 IO 请求对应的调度时刻小于当前时刻时，将所述当前 IO 请求发送到磁盘。

[117] 本领域技术人员应该可以意识到，在上述一个或多个示例中，本说明书所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时，可以将这

些功能所对应的计算机程序存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令/代码进行传输，以便这些功能所对应的计算机程序被计算机执行时，通过计算机实现本说明书任意一个实施例中所述的方法。

[118] 本说明书实施例中还提供了一种计算机可读存储介质，其上存储有计算机程序，当所述计算机程序在计算设备中执行时，计算设备执行本说明书任意一个实施例中提供的 IO 请求的调度方法。

[119] 本说明书实施例中还提供了一种计算设备，包括存储器和处理器，所述存储器中存储有可执行代码，所述处理器执行所述可执行代码时，实现本说明书任意一个实施例中提供的 IO 请求的调度方法。

[120] 本说明书中的各个实施例均采用递进的方式描述，各个实施例中相同、相似的部分互相参见即可，每个实施例中重点说明的都是与其他实施例的不同之处。尤其，对于装置实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[121] 上述对本说明书特定实施例进行了描述。其它实施例在本申请要求保护的范围内。在一些情况下，部分动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

[122] 以上所述的具体实施方式，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施方式而已，并不用于限定本发明的保护范围，凡在本发明的技术方案的基础之上，所做的任何修改、等同替换、改进等，均应包括在本发明的保护范围之内。

权利要求书

1. 一种 IO 请求的调度方法, 包括:

获取待调度的目标 IO 请求, 并确定所述目标 IO 请求是否属于被限流的 IO 请求;

当所述目标 IO 请求属于未被限流的第一类 IO 请求时, 将当前时刻确定为所述目标 IO 请求的调度时刻;

当所述目标 IO 请求属于被限流的第二类 IO 请求时, 根据当前限流资源比例确定所述目标 IO 请求的调度时刻, 所述当前限流资源比例是当前时刻所属的当前时间周期中用于处理第二类 IO 请求的资源比例, 所述当前限流资源比例根据前一时间周期的前一线流资源比例和前一负载信息计算得到, 所述前一负载信息是前一时间周期内调度的第一类 IO 请求的负载信息, 所述调度时刻不小于当前时刻;

将所述目标 IO 请求加入 IO 请求集合, 所述 IO 请求集合中的若干 IO 请求按照调度时刻从小到大的顺序依次被发送到磁盘进行处理。

2. 根据权利要求 1 所述的方法, 其中, 第一类 IO 请求包括前台 IO 请求, 第二类 IO 请求包括后台 IO 请求。

3. 根据权利要求 1 所述的方法, 其中, 将所述目标 IO 请求加入 IO 请求集合具体包括: 将所述目标 IO 请求加入通过最小堆来管理的 IO 请求集合。

4. 根据权利要求 1 所述的方法, 其中, 根据当前限流资源比例确定所述目标 IO 请求的调度时刻, 包括:

根据当前限流资源比例以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻, 所述期望调度时刻与当前限流资源比例负相关;

将当前时刻和所述期望调度时刻的较大值确定为所述目标 IO 请求的调度时刻。

5. 根据权利要求 4 所述的方法, 其中, 根据当前限流资源比例确定所述目标 IO 请求的调度时刻, 还包括: 根据磁盘的性能确定所述目标 IO 请求的请求带宽所对应的期望读写频次;

根据当前限流资源比例以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻, 具体包括: 根据所述期望读写频次、当前限流资源比例, 以及获取的上一个第二类 IO 请求的调度时刻, 计算所述目标 IO 请求的期望调度时刻; 其中, 所述期望调度时刻与所述期望读写频次负相关。

6. 根据权利要求 1 至 5 中任一项所述的方法, 其中, 当所述目标 IO 请求属于未被限流的第一类 IO 请求时, 所述方法还包括:

确定获取所述目标 IO 请求的第一时刻;

在第二时刻, 从所述 IO 请求集合中选择所述目标 IO 请求, 并将所述目标 IO 请求发送到磁盘; 其中, 在所述第二时刻, 所述目标 IO 请求的调度时刻是所述 IO 请求集合包含的所有 IO 请求中的最小值;

接收磁盘返回的请求结果;

向所述目标 IO 请求对应的应用程序发送所述请求结果, 并确定发送所述请求结果的第三时刻;

根据所述第三时刻和所述第一时刻确定所述目标 IO 请求的响应时间。

7. 根据权利要求 6 所述的方法, 其中, 还包括: 在达到当前时间周期的结束时刻时, 根据当前时间周期内调度的第一类 IO 请求的当前负载信息, 确定下一时间周期的下一限流资源比例; 其中, 当前负载信息具体包括, 当前时间周期内调度的第一类 IO 请求的平均响应时间, 以及当前时间周期内调度的第一类 IO 请求的平均请求带宽。

8. 根据权利要求 7 所述的方法, 其中, 根据当前时间周期内调度的第一类 IO 请求的当前负载信息, 确定下一时间周期的下一限流资源比例, 包括:

根据磁盘的性能确定所述平均请求带宽对应的期望响应时间;

当所述平均响应时间不小于所述期望响应时间时, 按照预设步长减小所述当前限流

资源比例,获得下一时间周期的下一限流资源比例;或者,当所述平均响应时间小于所述期望响应时间时,按照预设步长增大所述当前限流资源比例,获得下一时间周期的下一限流资源比例。

9. 根据权利要求 7 所述的方法,其中,下一时间周期的下一限流资源比例不大于第一预设阈值且不小于第二预设阈值。

10. 一种 IO 请求的调度方法,包括:

获取 IO 请求集合,所述 IO 请求集合通过权利要求 1 至 9 中任一项所述的方法得到,所述 IO 请求集合中包括若干 IO 请求,每个 IO 请求具有对应的调度时刻;

从所述 IO 请求集合中选择调度时刻最小的 IO 请求作为当前 IO 请求;

将所述当前 IO 请求发送到磁盘,使磁盘处理所述当前 IO 请求并返回请求结果。

11. 根据权利要求 10 所述的方法,其中,将所述当前 IO 请求发送到磁盘,具体包括:在所述当前 IO 请求对应的调度时刻小于当前时刻时,将所述当前 IO 请求发送到磁盘。

12. 一种 IO 请求的调度装置,包括:

请求获取单元,配置为获取待调度的目标 IO 请求;

分类确定单元,配置为确定所述目标 IO 请求是否属于被限流的 IO 请求;

调度时刻确定单元,配置为当所述目标 IO 请求属于未被限流的第一类 IO 请求时,将当前时刻确定为所述目标 IO 请求的调度时刻;当所述目标 IO 请求属于被限流的第二类 IO 请求时,根据当前限流资源比例确定所述目标 IO 请求的调度时刻,所述当前限流资源比例是当前时刻所属的当前时间周期中用于处理第二类 IO 请求的资源比例,所述当前限流资源比例根据前一时间周期的前一限流资源比例和前一负载信息计算得到,所述前一负载信息是前一时间周期内调度的第一类 IO 请求的负载信息,所述调度时刻不小于当前时刻;

请求调度单元,配置为将所述目标 IO 请求加入 IO 请求集合,所述 IO 请求集合中的若干 IO 请求按照调度时刻从小到大的顺序依次被发送到磁盘进行处理。

13. 一种 IO 请求的调度装置,包括:

获取单元,配置为获取 IO 请求集合,所述 IO 请求集合通过权利要求 12 所述的装置得到,所述 IO 请求集合中包括若干 IO 请求,每个 IO 请求具有对应的调度时刻;

选择单元,配置为从所述 IO 请求集合中选择调度时刻最小的 IO 请求作为当前 IO 请求;

发送单元,配置为将所述当前 IO 请求发送到磁盘,使磁盘处理所述当前 IO 请求并返回请求结果。

14. 一种计算设备,包括存储器和处理器,所述存储器中存储有计算机程序,所述处理器执行所述计算机程序时,实现权利要求 1 至 9 或者权利要求 10 至 11 中任一项所述的方法。

15. 一种计算机可读存储介质,其上存储有计算机程序,当所述计算机程序在计算设备中执行时,计算设备执行权利要求 1 至 9 或者权利要求 10 至 11 中任一项所述的方法。