

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2022 年 7 月 7 日 (07.07.2022)



(10) 国际公布号
WO 2022/143677 A1

(51) 国际专利分类号:
H04L 12/00 (2006.01)

(21) 国际申请号: PCT/CN2021/142132

(22) 国际申请日: 2021 年 12 月 28 日 (28.12.2021)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202011582831.6 2020 年 12 月 28 日 (28.12.2020) CN

(71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 蔡昕辰 (CAI, Xincheng); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 林钦亮 (LIN, Qinliang); 中国广

东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,

(54) **Title:** METHOD FOR USING INTERMEDIATE DEVICE TO PROCESS DATA, COMPUTER SYSTEM, AND INTERMEDIATE DEVICE

(54) 发明名称: 使用中间设备对数据处理的方法、计算机系统、及中间设备

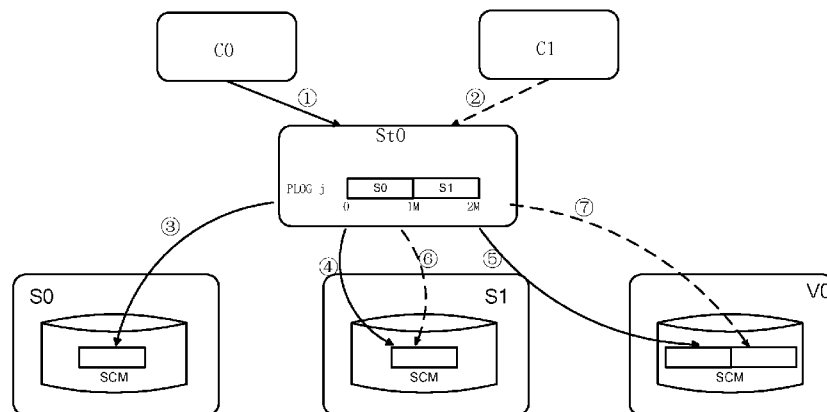


图 3

(57) **Abstract:** Provided in the embodiments of the present application are a computer system, a data processing method, and an intermediate device. The system comprises a computing node, a storage node, a check node, and the intermediate device; the computing node accesses the storage node and the check node by means of the intermediate device; a service is run on the computing node; the intermediate device is used, on the basis of a preset fragment length, for segmenting data to be written that is written into a logical storage space corresponding to the service so as to obtain a plurality of data; according to metadata of the logical storage space, a storage node for writing the data obtained by segmentation and a check node for writing verification data of the data to be written are determined; the data obtained by segmentation is written into a physical storage space corresponding to the logical storage space in the storage node; and the verification data is written into the physical storage space corresponding to the logical storage space in the check node. The computer system provided by the embodiments of the present application effectively reduces the load of the computing node, and reduces the time delay of data writing.

CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

— 包括国际检索报告(条约第21条(3))。

(57) 摘要：本申请实施例提供一种计算机系统、数据处理方法和中间设备，所述系统包括计算节点、存储节点、校验节点和中间设备，所述计算节点通过中间设备访问存储节点和校验节点；计算节点上运行业务；中间设备用于基于预设的分片长度对写入所述业务对应的逻辑存储空间的待写数据进行切分得到多个数据，根据所述逻辑存储空间的元数据确定写入所述切分得到的数据的存储节点及写入所述待写数据的校验数据的校验节点，将切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中，并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中。本申请实施例提供的计算机系统有效减少了计算节点的负载，且减少了数据写入的时延。

使用中间设备对数据处理的方法、计算机系统、及中间设备

技术领域

本申请涉及计算机技术领域，具体涉及一种使用中间设备对数据处理的方法、计算机系统、及中间设备。

背景技术

在多节点的计算机系统中，通常包括多个计算节点和多个存储节点。为了实现由多个计算节点并发地对应用的存储空间进行访问，相关技术中通过存储节点对应用的存储空间进行管理，而通过存储节点对应用的存储空间进行管理时，在计算节点对存储节点中存储的应用的数据访问时，会增加计算节点的负载，并影响数据处理的时延。例如，计算节点在基于数据分片存储方式对多个存储节点进行多分片写入的场景中，当计算节点需要将应用的数据写入应用对应的存储空间时，计算节点首先请求多个存储节点中的主存储节点在应用对应的存储空间中为所述数据分配写入地址，在获取写入地址后根据该写入地址对所述数据进行切分，再向所述写入地址对应的至少一个存储节点分别发送写入数据的各个部分的写请求。这样，在写入数据时，计算节点首先需要向所述主存储节点申请分配写入地址，在分配了写入地址之后，然后再向各个存储节点发送写入数据的一部分的写请求，即计算节点需要向存储节点发送至少两次请求，这不但增加了计算节点的负载，且增加了数据写入的时延。

发明内容

本申请实施例旨在提供一种数据处理的方法、计算机系统、及中间设备，通过在计算节点与存储节点之间连接的中间设备对应用的存储空间进行管理，有效减少了计算节点的负载，且减少了数据写入的时延。

为实现上述目的，本申请第一方面提供一种计算机系统，其特征在于，包括计算节点、存储节点、校验节点和中间设备，所述计算节点通过所述中间设备访问所述存储节点和所述校验节点；所述计算节点上运行业务；所述中间设备用于基于预设的分片长度对写入所述业务对应的逻辑存储空间的待写数据进行切分得到多个数据，根据所述逻辑存储空间的元数据确定写入所述切分得到的数据的存储节点及写入所述待写数据的校验数据的校验节点，将所述切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中，并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中。

通过将业务的逻辑存储空间的元数据存储在该中间设备上，以使中间设备通过所述元数据对所述业务的逻辑存储空间进行管理，以进行对存储节点和校验节点的操作，这样可以减轻计算节点的负载，并减少了数据写入的时延。例如，当所述操作为写操作时，计算节点只需要发送写请求至中间设备即可，中间设备根据所述元数据为所述写请求在所述逻辑存储空间中分配地址，并根据所分配的地址将数据写入存储节点，并将校验数据写入校验节点，由于计算节点只需要发送请求至中间设备，所以减轻了计算节点的负载，也减少了数据写入的时延。

在本申请第一方面的一种可能的实现方式中，所述逻辑存储空间的元数据包括所述逻辑

存储空间的标识, 所述逻辑存储空间的地址信息, 所述逻辑存储空间对应的 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的 M 个校验节点中的 M 个第二物理存储空间的地址信息, 其中, 所述第一物理存储空间的长度等于所述分片长度, 所述逻辑存储空间的长度等于 N 个所述分片长度之和。

通过在所述元数据中记录应用的逻辑存储空间的地址信息、存储节点中物理存储空间的地址信息以及校验节点中物理存储空间的地址信息, 中间设备可以将对逻辑存储空间的操作映射至对存储节点和校验节点中的物理存储空间的操作, 从而实现对存储在存储节点中的物理存储空间中的业务数据的操作。

在本申请第一方面的一种可能的实施方式中, 所述逻辑存储空间为持久化日志空间。

在本申请第一方面的一种可能的实施方式中, 所述中间设备为网络交换设备。

由于计算节点与存储节点和校验节点之间的通信会经过网络交换机, 所以将应用的存储空间的元数据存储在网络交换机中, 可以更加方便的实现对应用的存储空间的管理。

在本申请第一方面的一种可能的实现方式中, 所述第二物理存储空间的长度等于所述分片长度, 当写入所述逻辑存储空间的待写数据的量等于所述逻辑存储空间的长度时, 所述多个数据为 N 个数据分片, 所述中间设备还用于计算所述 N 个数据分片的 M 个校验分片, 所述中间设备用于将所述切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中, 并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中具体包括, 所述中间设备用于将所述 N 个数据分片分别写入所述 N 个存储节点中的所述第一物理存储空间中, 将所述 M 个校验分片分别写入所述 M 个校验节点中的所述第二物理存储空间中。

通过由中间设备计算数据分片的校验分片, 并将校验分片发送给校验节点, 计算节点只需要发送写请求至中间设备即可, 减轻了计算节点的负载, 减少了写请求的延时。

在本申请第一方面的一种可能的实现方式中, 所述第二物理存储空间的长度等于 N 个所述分片长度之和, 当写入所述逻辑存储空间的待写数据的量小于所述逻辑存储空间的长度时, 所述中间设备用于将所述切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中, 并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中具体包括, 所述中间设备用于将切分获取的多个数据分别写入对应的存储节点中的第一物理存储空间中, 将所述待写数据写入每个校验节点中的第二物理存储空间中。

通过由中间设备基于分片长度对待写数据进行切分, 并将切分得到的数据发送给对应的存储节点, 将待写数据发送给每个校验节点, 以使得校验节点进行对校验分片的计算, 计算节点只需要发送写请求至中间设备即可, 减轻了计算节点的负载, 减少了写请求的延时。

在本申请第一方面的一种可能的实现方式中, 所述待写数据为计算节点发送的一个写请求对应的数据, 所述计算节点通过多个报文将所述待写数据写入所述逻辑存储空间, 所述中间设备用于基于预设的分片长度对写入所述业务对应的逻辑存储空间的待写数据进行切分得到多个数据具体包括: 所述中间设备用于, 按照所述第一物理存储空间的长度从所述第一报文的待写数据切分出用于写入对应的第一存储节点中的第一物理存储空间的末尾地址的数据, 在所述第一报文不是所述写请求最后一个报文的情况中, 在缓冲区中存储所述第一报文的待写数据在切分之后的剩余数据, 以用于在与接收的所述写请求的下一个报文的部分待写数据拼接成预定长度数据之后再发送给另一对应的存储节点。

通过将报文的在切分之后剩余的数据缓存起来并与后一个报文的部分数据拼接为满报文 (即具有最大报文数据长度) 再进行发送, 避免了存储节点误判丢包的情况。

在本申请第一方面的一种可能的实现方式中, 所述中间设备还用于, 通过将所述第一报

文的序列号修改为与所述第一存储节点对应的序列号，生成用于在所述确定的写入地址中写入所述第一报文的部分待写数据的报文，并在与所述第一存储节点对应的发送队列中存储该生成的报文，以向所述第一存储节点发送该报文。

由于中间设备对写请求的待写数据进行 EC 分包，一个报文可能会被切分发送给两个存储节点，中间设备通过修改从计算设备接收的报文的序列号，避免出现中间设备从计算设备接收的报文的序列号的增加数与中间设备向存储节点发送的报文的序列号的增加数不一致的问题。

在本申请第一方面的一种可能的实现方式中，所述 M 个校验节点还用于，基于存储的所述逻辑存储空间对应的业务数据的 N 个数据分片，生成该业务数据的校验分片，为该校验分片分配第三物理存储空间，在所述第三物理存储空间中存储所述校验分片，并将所述第三物理存储空间的地址信息发送给所述中间设备；所述中间设备还用于在从所述 M 个校验节点分别接收写入校验分片的第三物理存储空间的地址信息之后，将所述逻辑存储空间的元数据中的 M 个第二物理存储空间的地址信息修改为所述 M 个第三物理存储空间的地址信息。

本申请第二方面提供一种数据处理方法，其特征在于，所述方法由中间设备执行，所述中间设备连接计算节点、N 个存储节点和 M 个校验节点，所述计算节点上运行业务，所述方法包括：接收所述计算节点发送的对所述业务的数据的写请求；基于预设的分片长度对所述写请求的待写数据进行切分得到多个数据；根据所述业务对应的逻辑存储空间的元数据，确定所述 N 个存储节点中的分别写入所述多个数据的多个存储节点及写入所述待写数据的校验数据的 M 个校验节点；请求所述多个存储节点将所述多个数据分别写入其与所述逻辑存储空间对应的物理存储空间中；请求所述 M 个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中。

在本申请第二方面的一种可能的实现方式中，所述逻辑存储空间的元数据包括所述逻辑存储空间的标识，所述逻辑存储空间的地址信息，所述逻辑存储空间对应的所述 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的所述 M 个校验节点中的 M 个第二物理存储空间的地址信息，其中，所述第一物理存储空间的长度等于所述分片长度，所述逻辑存储空间的长度等于 N 个所述分片长度之和。

在本申请第二方面的一种可能的实现方式中，所述方法还包括：从计算节点接收所述逻辑存储空间的信息；对所述 N 个存储节点分别发送第一物理存储空间分配请求，所述第一存储空间分配请求用于请求对所述逻辑存储空间中的一个分片长度的空间分配第一物理存储空间；对所述 M 个校验节点分别发送第二物理存储空间分配请求，所述第二存储空间分配请求用于请求对所述逻辑存储空间分配第二物理存储空间；从所述 N 个存储节点接收其为所述逻辑存储空间分配的第一物理存储空间的信息；从所述 M 个存储节点接收其为所述逻辑存储空间分配的第二物理存储空间的信息。

在本申请第二方面的一种可能的实现方式中，所述第二物理存储空间的长度等于 N 个所述分片长度之和，所述写请求的待写数据的量小于所述逻辑存储空间的长度，所述方法还包括：根据所述业务对应的逻辑存储空间的元数据，确定将所述多个数据分别写入所述多个存储节点的写入地址、及将所述待写数据写入每个校验节点的写入地址。

在本申请第二方面的一种可能的实现方式中，根据所述业务对应的逻辑存储空间的元数据，确定所述 N 个存储节点中的分别写入所述多个数据的多个存储节点及写入所述待写数据的校验数据的 M 个校验节点包括：为所述写请求在所述逻辑存储空间中分配第一地址；根据所述逻辑存储空间的元数据确定与所述第一地址对应的多个存储节点及各个存储节点中的写

入地址；根据所述逻辑存储空间的元数据确定每个所述校验节点中的与所述第一地址对应的写入地址。

在本申请第二方面的一种可能的实现方式中，接收所述计算节点发送的对所述业务的数据的写请求包括，接收所述计算节点发送的所述写请求的第一报文，所述根据所述逻辑存储空间的元数据确定与所述第一地址对应的多个存储节点中的第二地址包括，为所述第一报文的至少部分待写数据分配所述第一地址中的地址，根据所述存储空间的元数据确定所述分配给所述第一报文的地址对应的用于写入所述第一报文中的至少部分待写数据的第一存储节点和该第一存储节点中的写入地址。

在本申请第二方面的一种可能的实现方式中，所述基于预设的分片长度对所述写请求的待写数据进行切分得到多个数据包括，在所述第一报文的部分待写数据被分配的所述第一地址中的地址位于所述逻辑存储空间中的一个分片长度空间的末尾地址的情况下，从所述第一报文的待写数据切分出用于写入所述第一存储节点的写入地址的数据，在所述第一报文不是所述写请求最后一个报文的情况下，在缓冲区中存储所述第一报文的待写数据在切分之后的剩余数据，以用于在与接收的所述写请求的下一个报文的部分待写数据拼接成预定长度数据之后再发送给另一对应的存储节点。

在本申请第二方面的一种可能的实现方式中，请求所述多个存储节点将所述多个数据分别写入其与所述逻辑存储空间对应的物理存储空间中包括，通过将所述第一报文的序列号修改为与所述第一存储节点对应的序列号，生成用于在所述第一存储节点的写入地址中写入所述第一报文中的至少部分待写数据的报文，并在与所述第一存储节点对应的发送队列中存储该生成的报文，以向所述第一存储节点发送该报文。

在本申请第二方面的一种可能的实现方式中，所述方法还包括：在请求所述M个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中之后，从所述M个校验节点分别接收写入校验分片的第三物理存储空间的地址信息，将所述逻辑存储空间的元数据中的M个第二物理存储空间的地址信息修改为所述M个第三物理存储空间的地址信息。

在本申请第二方面的一种可能的实现方式中，所述待写数据的量等于所述逻辑存储空间的长度，所述基于预设的分片长度对所述写请求的待写数据进行切分得到多个数据包括，基于所述分片长度对所述待写数据进行分片得到N个数据分片，所述方法还包括：基于所述N个数据分片生成M个校验分片；所述请求所述M个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中包括，请求所述M个校验节点分别写入所述M个校验分片中的一个校验分片。

在本申请第二方面的一种可能的实现方式中，所述方法还包括，在请求所述M个校验节点分别写入所述M个校验分片中的一个校验分片之后，从所述M个校验节点分别接收写入校验分片的第三物理存储空间的地址信息，将所述逻辑存储空间的元数据中的M个第二物理存储空间的地址信息修改为所述M个第三物理存储空间的地址信息。

本申请第三方面提供一种中间设备，所述中间设备连接计算节点、N个存储节点和M个校验节点，所述计算节点上运行业务，所述中间设备包括：接收单元，用于接收所述计算节点发送的对所述业务的数据的写请求；切分单元，用于基于预设的分片长度对所述数据进行切分得到多个数据；确定单元，用于根据所述业务对应的逻辑存储空间的元数据，确定所述N个存储节点中的分别写入所述多个数据的多个存储节点及写入所述待写数据的校验数据的M个校验节点；请求单元，用于请求所述多个存储节点将所述多个数据分别写入其与所述逻

辑存储空间对应的物理存储空间中，及请求所述 M 个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中。

在本申请第三方面的一种可能的实现方式中，所述逻辑存储空间的元数据包括所述逻辑存储空间的标识，所述逻辑存储空间的地址信息，所述逻辑存储空间对应的所述 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的所述 M 个校验节点中的 M 个第二物理存储空间的地址信息，其中，所述第一物理存储空间的长度等于所述分片长度，所述逻辑存储空间的长度等于 N 个所述分片长度之和。

在本申请第三方面的一种可能的实现方式中，所述第二物理存储空间的长度等于 N 个所述分片长度之和，所述待写数据的量小于所述逻辑存储空间的长度，所述确定单元还用于：根据所述业务对应的逻辑存储空间的元数据，确定将所述多个数据分别写入所述多个存储节点的写入地址、及将所述待写数据写入每个校验节点的写入地址。

在本申请第三方面的一种可能的实现方式中，所述待写数据的量等于所述逻辑存储空间的长度，所述切分单元还用于，对所述数据进行分片得到 N 个数据分片，所述中间设备还包括：生成单元，用于基于所述 N 个数据分片生成 M 个校验分片；所述请求单元还用于，请求所述 M 个校验节点分别写入所述 M 个校验分片中的一个校验分片。

本申请第四方面提供一种中间设备，包括处理单元和存储单元，所述存储单元中存储有可执行代码，所述处理单元执行所述可执行代码以实现如本申请第二方面和第三方面所述的方法。

本申请第五方面提供一种中间设备，包括：通信接口，用于与存储节点及计算节点进行数据传输；处理单元，用于对所述通信接口接收的数据进行处理，以执行如本申请第二方面和第三方面所述的方法。

附图说明

通过结合附图描述本申请实施例，可以使得本申请实施例更加清楚：

图 1 为本申请实施例所应用的计算机系统的架构图；

图 2 为本申请实施例提供的一种在计算机系统中写数据的方法流程图；

图 3 为计算节点 C0 和 C1 并行对 PLOG j 进行写入的过程示意图；

图 4 为计算节点 C0 以上述第二种数据写入方式对 PLOG j 进行写入的过程示意图；

图 5 为在计算机系统中写数据的过程示意图；

图 6 为本申请实施例提供的另一种在计算机系统中写数据的过程示意图；

图 7 为本申请实施例提供的中间设备的架构图；

图 8 为本申请实施例提供的一种中间设备的架构图；

图 9 为本申请实施例提供的一种中间设备的架构图；

图 10 为本申请实施例提供的云服务系统的示意图。

具体实施方式

下面将结合附图，对本申请实施例中的技术方案进行描述。

图 1 为本申请实施例所应用的计算机系统 100 的架构图。如图 1 中所示，该计算机系统 100 中包括计算层 12、网络层 13 和存储层 14。其中，计算层 12 包括多个计算节点，图 1 中示意示出计算节点 C0、C1 和 C2；网络层 13 包括多个中间设备，图 1 中示意示出中间设备 St0 和 St1；存储层 14 包括多个存储节点和至少一个校验节点，图 1 中示意示出存储节点 S0、S1

和校验节点 V0。其中，所述计算节点、存储节点和校验节点可以为物理服务器，或者也可以为虚拟机、容器等基于通用硬件资源抽象的虚拟实体。所述中间设备可以为可编程网络连接设备，例如交换机，下文中将交换机作为中间设备的示例进行描述。所述多个计算节点例如可以作为多个应用（Application，APP）的应用服务器，为用户终端的用户提供业务处理服务。所述多个存储节点例如可用于存储 APP 的业务数据，所述至少一个校验节点用于存储所述业务数据的校验数据。存储节点和校验节点中可通过内存或硬盘等存储介质来存储业务数据及其校验数据。由于非易失性内存介质不仅具有大容量、高访问速度的特点，同时还具备非易失的存储特性，所以本发明实施例中的存储介质为非易失性内存介质，所述非易失性内存例如包括存储级内存（Storage class memory，SCM）。所述非易失性内存介质只是举例说明，但本发明实施例并不仅限于所述非易失性内存。

在图 1 所示的多节点分布式的计算机系统中，每个计算节点与至少一个交换机连接，每个交换机与多个存储节点和至少一个校验节点连接，从而计算节点通过交换机可访问到多个存储节点和至少一个校验节点。另外，交换机与交换机之间直接或间接连接，从而使得计算节点可通过任一交换机访问存储节点。其中，计算节点与交换机之间的连接、交换机与存储节点之间的连接可以为各种连接方式，如远程直接内存访问（Remote direct memory access，RDMA）连接方式，或者传输控制协议（Transmission Control Protocol，TCP）连接方式。多个计算节点可通过交换机对存储节点进行对业务数据的写入、读取等操作。

在相关技术中，为了通过数据分片存储的方式保证数据可靠性，计算节点与存储节点和校验节点直接连接，计算节点通过对与多个存储节点中的存储空间对应的一段逻辑存储空间进行访问，从而访问各个所述存储节点中的存储空间。所述逻辑存储空间例如为持久化日志空间（Persistence LOG，PLOG）。该 PLOG 通过 PLOG 的唯一标识 PLOG ID 来标识，PLOG 上的存储数据采用追加写的形式进行存储，即，对于已经存储的数据不进行覆盖修改，而是将该修改追加存储到新的地址中。通常，PLOG 与存储节点中的 SCM、固态硬盘（Solid State Disk，SSD）等介质中的一段连续物理存储空间相对应，所述物理存储空间为存储节点对外部设备提供的存储空间，外部设备可以利用该存储空间进行数据的访问，对于 SCM 等介质来说所述物理存储空间即为数据实际存储的物理地址，但对于 SSD 等存储介质来说，并不是数据实际存储的物理地址。

在该相关技术中，在多个存储节点中的主存储节点中存储 PLOG 的元数据，该元数据中包括 PLOG 的 ID、该 PLOG 对应的多个存储节点中的存储空间的地址、该 PLOG 的未分配空间的地址等信息，所述主存储节点通过 PLOG 的元数据对 PLOG 进行管理。例如，当计算节点希望通过 PLOG 写入数据至多个存储节点中的至少一个存储节点时，计算节点首先请求主存储节点在 PLOG 中为所述数据分配地址，主存储节点在根据 PLOG 元数据在所述 PLOG 中未分配的存储空间为所述数据分配地址后，将所分配的地址返回给计算节点。计算节点在获取所分配的地址之后，在需要的情况下根据所分配的地址对所述待写的数据进行切分，再向所述写入地址对应的至少一个存储节点发别发送写入对应的所述数据的一部分的写请求，以将所述数据写入所述写入地址对应的各个存储节点。在该相关技术中，计算节点在向至少一个存储节点写入数据的过程中，需要与存储节点进行多次通信，增加了计算节点的负载，且增加了数据写入的时延。

在本申请实施例中，如图 1 中所示，计算节点通过交换机与存储节点 S0、S1 和校验节点 V0 连接，交换机作为中间设备存储有 PLOG 的元数据，并对 PLOG 的元数据进行管理，从而通过对 PLOG 的元数据的管理实现对 PLOG 的管理。所述交换机中存储的 PLOG 的元数据包括 PLOG

的 ID、该 PLOG 对应的各个存储节点中的存储空间的信息(例如可以为存储空间的起始地址及长度)、该 PLOG 的未分配空间的信息(例如可以为该 PLOG 未分配空间的起始地址)等,其中,该 PLOG 的空间长度等于存储节点 S0 和存储节点 S1 中相对于该 PLOG 分配的存储空间长度之和。当计算节点响应于用户对应用的操作需要对 PLOG 写入数据时,计算节点向交换机发送对 PLOG 写入数据的写请求。交换机在接收到该写请求之后,基于 PLOG 的元数据,对该写请求分配 PLOG 中的写入地址,基于该写入地址确定将该写请求中的待写数据写入的至少一个存储节点及其中的对应地址,在需要的情况中对该待写数据进行切分,并向各个确定的存储节点发送对其中的对应地址写入所述待写数据的至少部分的请求,以使各个存储节点将所述待写数据的至少部分写入所述存储节点的存储空间。在本申请实施例的技术实现中,通过将 PLOG 的元数据存储在交换机中,使得交换机来管理 PLOG,这样计算节点在将数据写入至少一个存储节点的时候,只要发送数据写请求至交换机即可,而不需要发送地址分配请求,如此,不但降低了计算节点的负载,而且降低了写请求的时延。

图 2 为本申请实施例提供的一种在计算机系统中写数据的方法流程图。

如图 2 所示,首先在步骤 S201,计算节点发送对 PLOG i 的写请求 1。

在本申请实施例中的流程图中,计算节点可以为图 1 中多个计算节点中的任一计算节点,交换机为与所操作的 PLOG 对应的交换机,存储节点和校验节点为与该 PLOG 对应的存储节点。在本申请实施例中,一个 PLOG 仅由一个交换机进行管理。在计算节点生成一个新的 PLOG 之后,该 PLOG 被确定归属于某个交换机,该确定的交换机即为该 PLOG 对应的交换机,所述对应的交换机中记录有该 PLOG 的元数据,并基于该元数据对该 PLOG 进行管理。所述 PLOG 对应的存储节点和校验节点为计算节点在生成一个新的 PLOG 之后为所述 PLOG 指定的存储节点和校验节点。确定 PLOG 所述归属的交换机的具体方法例如可以为:计算节点在生成新的 PLOG 之后,可对该 PLOG 的 ID 进行哈希计算,从而基于该 PLOG ID 的哈希值将该 PLOG 归属到某个交换机,并将该 PLOG 的信息发送到该交换机,以由该交换机对该 PLOG 进行管理。另一种确定 PLOG 所述归属的交换机的方法可以为:计算节点在生成新的 PLOG 之后,可将该 PLOG 的信息发给其连接的任一交换机,从而由多个交换机协商确定并记录该 PLOG 与交换机的映射关系,之后,接收到 PLOG 信息的交换机可基于所述映射关系将该 PLOG 的信息转发给对应的交换机,并由对应的交换机对该 PLOG 进行管理。

假设图 2 中的计算节点为图 1 中的计算节点 C0。计算节点 C0 中设置有 PLOG 功能层,该 PLOG 功能层对应用提供调用接口,所述调用接口例如包括 PLOG 申请接口、PLOG 写接口等等。当计算节点 C0 中的应用需要向计算机系统中写入数据时,首先,应用从 PLOG 功能层获取预先记录的当前可写入的 PLOG 的 ID,确定当前可写入的 PLOG 为 PLOG i,从而,应用调用 PLOG 功能层中的写接口,从而使得计算节点 C0 直接或间接向对应的交换机发送对 PLOG i 的写请求 1。

具体是,计算节点 C0 中的应用调用所述写接口,传入 PLOG i 的标识,待写数据的存放地址、待写数据的长度,PLOG 功能层根据传入的参数调用网卡的数据发送接口,根据待写数据的存放地址和待写数据的长度获取待写数据,基于该待写数据组装成写请求 1 的数据包(即报文)发送给交换机。根据设定的一次发送的最大报文长度,该写请求 1 可能以单个数据包发送给所述对应的交换机,或者以多个数据包发送给对应的交换机。其中,在该写请求 1 的数据包包括单个数据包的情况中,该写请求 1 中包括写请求 1 的编号、PLOG i 的标识、待写数据及待写数据的长度。在该写请求 1 的数据包包括多个数据包的情况中,每个数据包中包括写请求 1 的编号、PLOG i 的标识和待写数据中的一部分数据,另外,首个数据包中包括写

请求 1 中的待写数据的长度。

在步骤 S202, 交换机确定 PLOG i 的剩余空间是否足够。

如上文所述, 交换机 St0 与 PLOG i 相对应, 因此, 交换机 St0 中记录作为 PLOG i 的元数据, 以通过所述元数据对 PLOG i 进行管理。该 PLOG i 的元数据中例如包括 PLOG i 的标识、PLOG i 的地址信息 (例如 PLOG i 的未分配存储空间的起始地址及 PLOG i 的长度等)、PLOG i 对应的存储节点中的存储空间的地址信息等信息, 由于所述 PLOG i 为逻辑存储空间, 其中的地址为从 0 开始的偏移地址, 从而, 交换机通过以 PLOG i 的长度减去 PLOG i 的未分配存储空间的起始地址可获得 PLOG i 的剩余空间长度。

在接收到写请求 1 之后, 交换机将写请求 1 的待写数据的长度与 PLOG i 的剩余空间的长度进行比较, 以确定剩余空间是否足够用于写入该待写数据。如果剩余空间不足, 则交换机通知计算节点申请新的 PLOG 用于写入写请求 1 的待写数据, 如果剩余空间足够, 则对 PLOG i 对应的物理存储空间进行写入。

在步骤 S203, 交换机在上述步骤 S202 确定 PLOG i 的剩余空间不足的情况中, 向计算节点发送 PLOG i 空间不足的信息。

在步骤 S204, 计算节点生成 PLOG j。

例如, 在计算节点 C0 中, 应用在接收到上述 PLOG i 空间不足的信息之后, 可调用 PLOG 功能层中的 PLOG 申请接口, 以向 PLOG 功能层发送申请新的 PLOG 的请求。在该申请请求中, 指定了与该新申请的 PLOG 对应的 N 个存储节点和 M 个校验节点, 该 PLOG 的长度等信息。在计算机系统中, 可通过数据分片存储的方式保证数据存储的可靠性, 在该方式中例如可通过 (Redundant Arrays of Independent Disks, RAID) 算法及纠删码 (Erasure Coding, EC) 算法计算分别存储至所述 N 个存储节点的 N 个数据分片的 M 个校验分片, 然后将所述 N 个数据分片和所述 M 个校验分片分别存入到所述 N 个存储节点及所述 M 个校验节点中, 从而在任意存储节点出现故障的情况下, 可通过数据分片的校验分片恢复数据。对应于该数据分片存储方式, 假设存储节点中的分片的长度为预设分片长度, 可将 PLOG 的长度设置为 N 个分片长度之和, 通过基于 PLOG 的元数据将 PLOG 中的每个分片所占用的存储空间与每个存储节点中的具有所述分片长度的存储空间对应起来, 从而可以通过该 PLOG 进行对 N 个存储节点的分片存储的管理, 例如, 可基于 PLOG 的元数据进行对 N 个存储节点的写入或读取。另外, 通过在 PLOG 的元数据中记录校验节点中存储校验分片的地址, 当存储节点出现故障不能读取时, 可基于 PLOG 的元数据从校验节点中读取 PLOG 的校验分片以恢复 PLOG 的数据分片。

其中, 在多分片存储方式的场景下, 有两种写入数据的方式。在第一种方式中, 交换机 St0 对 M 个存储节点的存储空间先后写入不同写请求的数据作为 N 个数据分片, 并将所述 N 个数据分片 (即全部写请求的数据) 都写入每个校验节点中, 从而校验节点可基于其接收的 N 个数据分片计算对应的校验分片并存储该校验分片。在第二种方式中, 在待写数据的长度在预定范围内的情况中, 例如该长度大于 $(N-1)*a$ 且小于等于 $N*a$, 其中 a 为一个数据分片的长度, 从而, 交换机 St0 可在接收到该待写数据之后, 将该待写数据 (如果该待写数据的长度小于 $N*a$, 则将其补全为 $N*a$) 切分为 N 个数据分片, 计算该待写数据的 M 个校验分片, 并将所述 N 个数据分片和 M 个校验分片分别写入对应的存储节点和校验节点。

计算节点 C0 中的 PLOG 功能层在从应用接收到对新的 PLOG 的申请之后, 生成新的 PLOG ID, 并将该 ID 返回给应用。该新生成的 PLOG ID 可表示为 “PLOG j”, 该 ID 例如为根据 PLOG 的编号顺序生成, 例如, PLOG i 为 PLOG 1, 则确定 “PLOG j” 为 PLOG 2。可以理解, 本申请实施例不限于根据 PLOG 的编号顺序来确定 PLOG ID, 只要该 PLOG ID 可唯一标识该新生成的 PLOG

即可，例如，可将之前删除的 PLOG 的 ID 重新用作为该新的 PLOG 的 ID。

在步骤 S205，计算节点向交换机发送 PLOG j 的元数据信息。

计算节点 C0 的 PLOG 功能层在如上所述生成 PLOG j 之后，在一种实现方式中，计算节点 C0 可基于预定规则确定 PLOG j 对应的交换机，并将 PLOG j 的元数据信息发送给 PLOG j 对应的交换机；在另一种实施方式中，计算节点 C0 可将 PLOG j 的元数据信息发送给其连接的任一交换机，该交换机使得由多个交换机协商确定与 PLOG j 对应的交换机、并将 PLOG j 的元数据发送给与 PLOG j 对应的交换机。假设 PLOG j 对应的交换机为交换机 St0，即，PLOG j 的元数据信息被发送给交换机 St0。所发送的元数据信息中包括 PLOG j 的标识、PLOG j 对应的存储节点（即 S0、S1）和校验节点 V0、该 PLOG j 的长度等信息。

在步骤 S206，交换机向各个存储节点和各个校验节点发送存储空间分配请求。

交换机 St0 在接收到 PLOG j 的元数据信息之后，在本地存储 PLOG j 的元数据，并在元数据中增加 PLOG j 的未分配存储空间的信息，例如未分配存储空间的起始地址。由于 PLOG j 还未被使用，因此，此时的 PLOG j 未分配存储空间的起始地址为默认的偏移地址 0。

交换机 St0 可以表格的形式记录每个 PLOG 的元数据，如表 1 所示，为交换机 St0 中管理的 PLOG 的元数据。

PLOG 0	Offset=200, ...
PLOG 1	Offset=100, ...
...	
PLOG j	Offset=0, Volume=2M, S0: Addr= /S1: Addr= /V0: Addr=, ...
...	

表 1

如表 1 中所示，表 1 中的“Offset”表示 PLOG 的未分配存储空间的起始地址，“Volume”表示 PLOG 的大小（长度）。另外，在 PLOG j 的元数据中，“S0: Addr= / S1: Addr= / S2: Addr=”表示 PLOG j 与存储节点 S0、S1 和校验节点 V0 相对应，存储节点 S0、S1 和校验节点 V0 中与 PLOG j 对应的地址（Addr）尚未确定，因此为空。也就是说，PLOG j 对应于 2 个存储节点+1 个校验节点的分片存储方式。根据表 1，由于 PLOG j 的长度例如为 2 兆（M），由于 PLOG j 的长度等于 N 个分片的分片长度之和，这里 N=2，因此，分片长度为 1M，即在存储节点 S0 和存储节点 S1 中需要各分配 1M 的空间以与 PLOG j 相对应。

之后，交换机 St0 基于 PLOG j 的元数据中的 PLOG j 向对应的存储节点和校验节点发送存储空间分配请求，即向存储节点 S0、S1（图 2 中仅示出一个存储节点作为示意）和校验节点 V0 分别发送存储空间分配请求。所述存储空间分配请求中包括该 PLOG j 的长度，如上文所示，由于所述 PLOG j 的大小为 2M，因此，发送给存储节点 S0 和 S1 的存储空间分配请求用于请求在存储节点 S0 和存储节点 S1 中各分配 1M 的存储空间，该 1M 的存储空间即为 PLOG j 的一个数据分片的大小。

在确定通过上述第二种数据写入方式写入数据的情况中，交换机 St0 发送给校验节点的存储空间分配请求用于请求在校验节点 V0 中分配 1M 的存储空间，该 1M 的存储空间即为 PLOG j 的一个校验分片的大小。

在确定通过上述第一种数据写入方式进行写入的情况中，或者在不确定是通过第一种数据写入方式还是第二种数据写入方式进行写入的情况中，交换机 St0 发送给校验节点的存储空间分配请求用于请求在校验节点 V0 中分配 2M 的存储空间，该 2M 的存储空间也与 PLOG j 的 2M 空间相对应，从而用于在应用以上述第一种方式对 PLOG j 进行写入的过程中使得校验

节点存储写入 PLOG j 的全部数据，并使得校验节点基于存储的写入 PLOG j 的数据计算对应的校验分片。

在步骤 S207，存储节点和校验节点根据存储空间分配请求分配存储空间，并返回存储空间信息。

如上文所述，存储节点 S0、S1 和校验节点 V0 在分别接收到上述存储空间分配请求之后，将分别为 PLOG j 分配对应大小的本地的存储空间，并向交换机 St0 返回该存储空间信息。其中，存储节点 S0（或存储节点 S1）在接收到存储空间分配请求之后，在本地的存储介质中（例如 SCM）中获取未使用的 1M 大小的连续存储空间，将该存储空间标记为已分配，并将该存储空间的地址信息（例如起始地址或者地址范围）返回给交换机 St0。校验节点 V0 在接收到存储空间分配请求之后，在本地的存储介质中获取未使用的 2M 大小（或 1M 大小）的连续存储空间，将该存储空间标记为已分配，并将该存储空间的地址信息返回给交换机 St0。

在步骤 S208，交换机在 PLOG j 元数据中记录各个存储空间的地址信息。

交换机 St0 在从各个与 PLOG j 对应的存储节点和校验节点接收到存储空间的地址信息之后，记录到 PLOG j 的元数据中。具体是，交换机 St0 可将存储节点 S0、S1 和校验节点 V0 中的为 PLOG j 分配的存储空间的起始地址和地址长度分别记录到表 1 中的各个相应的“Addr”字段中。

另外，在计算节点中的应用首次使用的时候，计算节点会给应用分配一个初始 PLOG，并为该初始 PLOG 分配存储空间，对初始 PLOG 分配存储空间的过程与为 PLOG j 分配存储空间的过程相同，具体请参考步骤 S205-S208 的描述，在此不再赘述。

在完成上述各个步骤之后，也即完成了对 PLOG j 的创建过程。交换机在该创建过程完成之后，可将新生成的 PLOG j 通知给各个计算节点，从而各个计算节点可并行对该 PLOG j 写数据。

由于相对于上述两种数据写入方式，图 2 中的步骤 S209-S217 具有不同的具体实施方式，因此，下文中将相对于上述两种数据写入方式分别进行对步骤 S209-S219 的描述。

首先相对于上述第一种数据写入方式描述图 2 中的步骤 S209-S219。

图 3 为计算节点 C0 和 C1 并行对 PLOG j 进行写入的过程示意图。为了区别计算节点 C0 和 C1 的写入，在图 3 中，将与计算节点 C0 相关的操作用实线示出，将与计算节点 C1 相关的操作用虚线示出。另外，在图 3 中，如上文所述，在交换机 St0 中存储了 PLOG j 的元数据，图 3 中以数据条示意示出 PLOG j，假设 PLOG j 中地址从左向右从 0 增大到 2M，则该地址中的前 1M 的空间可设定为与存储节点 S0 相对应，该地址中的后 1M 的空间可设定为与存储节点 S1 相对应，其中，可将该 PLOG j 中的两个 1M 空间分别称为一个分片段。同时在与 PLOG j 对应的存储节点 S0、S1 和校验节点 V0 的 SCM 中分别为 PLOG j 分配了一段存储空间，该存储空间也以数据条示意示出，其中存储节点 S0 和 S1 中的存储空间为 1M 的长度，校验节点 V0 中的存储空间为 2M 的长度。

在步骤 S209，计算节点 C0 向交换机 St0 发送对 PLOG j 的写请求 1'。

计算节点 C0 中的应用在获取到新生成的 PLOG j 之后，可重新发起对该 PLOG j 的写请求 1'，以写入上述未成功写入的待写数据。所述写请求 1' 用于与上述对 PLOG i 的写请求 1 相区别。类似地，该写请求 1' 中包括 PLOG j 的标识和待写数据。

在步骤 S210，交换机 St0 基于分片长度对写请求 1' 的待写数据进行切分。

参考图 3，假设计算节点 C0 相比于计算节点 C1 在先通过图 3 中的操作①对交换机 St0 发送对 PLOG j 的写请求 1'，从而交换机 St0 在处理完写请求 1' 之后，再处理由计算节点

C1 发送的写请求。

相对于该写请求 1'，交换机 St0 首先在 PLOG j 的未分配地址中按照地址从小到大的顺序对该写请求 1' 的待写数据分配地址。一般情况下，地址空间用起始地址加地址空间的长度表示，所以，在交换机 St0 为写请求 1' 分配地址空间 1' 时，首先从 PLOG j 的元数据中获取变量 offset 的值，所述变量 offset 的值用于表示 PLOG j 中未分配地址空间中的起始地址，将所述变量 offset 的值作为所述写请求 1' 的起始地址，如此，分配给所述写请求 1' 的地址空间 1' 即为以所述变量 offset 的值为起始地址，长度为待写数据长度的地址空间。在本申请实施例中，所述地址空间也可以称为“地址”。在为所述写请求 1' 分配地址空间 1' 之后，则交换机 St0 将元数据中 PLOG j 的 offset 更新为当前值加上写请求 1' 的待写数据的长度。例如，假设写请求 1' 中的待写数据的长度为 1.7M，并且 PLOG j 的未分配地址的起始地址为 0，则交换机 St0 将 PLOG j 中的前 1.7M 地址作为地址空间 1' 分配给写请求 1'。该 1.7M 的地址对应于存储节点 S0 的 1M 存储空间和存储节点 S1 中的 0.7M 存储空间。

由于所述分配的地址与两个存储节点相对应，交换机 St0 根据预设的分片长度（即 1M）将所述写请求 1' 的待写数据切分为 1M 的数据和 0.7M 的数据，以分别写入存储节点 S0 和存储节点 S1。

在步骤 S211，交换机 St0 根据 PLOG j 的元数据确定写入数据的存储节点和写入校验数据的校验节点。

所述 PLOG j 的元数据中记录了与 PLOG j 对应的存储节点 S0 和存储节点 S1 中的 1M 空间的地址，与 PLOG j 对应的校验节点 V0 中的 2M 空间的地址，通过上述对写请求 1' 的地址分配可以得出，写入写请求 1' 的数据的存储节点包括存储节点 S0 和存储节点 S1。另外在上述第一种写入方式中，将写请求 1' 的待写数据写入校验节点作为校验数据，校验节点在 PLOG j 写满之后会根据本地存储的 PLOG j 的两个数据分片计算校验分片。因此，交换机可从 PLOG j 的元数据确定写入写请求 1' 的校验数据的校验节点 V0。

更具体地，交换机 St0 可根据对写请求 1' 分配的 PLOG j 中的地址，确定写入各个存储节点的地址和写入校验节点的地址。交换机通过将 PLOG j 元数据中记录的存储节点中的存储空间的起始地址与对写请求 1' 分配的地址中与该存储节点对应的地址相加，可获取存储节点中的写入地址。交换机通过将 PLOG j 元数据中记录的校验节点中的存储空间的起始地址与对写请求 1' 分配的地址相加，可获取校验节点中的写入地址。例如，根据写请求 1' 分配的 PLOG j 中的 1.7M 的地址，可确定写入存储节点 S0 的地址为所述全部 1M 存储空间，写入存储节点 S1 的地址为所述 1M 存储空间中的前 0.7M 空间，写入校验节点 V0 中的地址为所述 2M 空间中的前 1.7M 空间。

在步骤 S212，交换机 St0 向各个存储节点发送写入数据的写请求 2'。

参考图 3，交换机 St0 通过操作③向存储节点 S0 发送写入上述 1M 数据的写请求 2'，请求将该待写数据中的前 1M 数据写入存储节点 S0 中的所示数据条中，通过操作④向存储节点 S1 发送写入上述 0.7M 数据的写请求 2'，请求将该待写数据的后 0.7M 数据写入存储节点 S1 中的所示数据条中的前 0.7M 的地址中。

在步骤 S213，各个存储节点对 PLOG j 对应的存储空间写入对应的数据。

在步骤 S214，各个存储节点在完成上述写入之后，返回写请求 2' 的完成信息。

在步骤 S215，交换机 St0 向校验节点发送写入校验数据的写请求 3'。

参考图 3，交换机 St0 通过操作⑤向校验节点 V0 发送写入写请求 1' 的 1.7M 待写数据的写请求 3'，请求将该待写数据写入校验节点 V0 中的所示数据条中的前 1.7M 的地址中作

为校验数据。

在步骤 S216, 校验节点 V0 对 PLOG j 对应的存储空间写入校验数据。

具体是, 校验节点 V0 对 PLOG j 对应的 2M 存储空间的前 1.7M 空间写入所述写请求 1' 的待写数据。

在步骤 S217, 校验节点 V0 向交换机 St0 返回写请求 3' 的完成信息。

在步骤 S218, 交换机 St0 在确定存储节点 S0、S1 和校验节点 V0 都完成写入之后, 确定写请求 1' 之前接收的访问 PLOG j 的写请求都已经完成。

本申请实施例中, 通过根据交换机接收访问 PLOG j 的写请求的顺序向计算节点返回写请求, 实现写请求在 PLOG j 的地址空间的保序写入。由于交换机 St0 根据接收写请求的顺序为访问 PLOG j 的各个写请求分配 PLOG j 中的地址空间, 因此, 较早接收的写请求被分配的地址空间中的地址较小, 通过根据接收写请求的顺序向计算节点返回写请求, 也即根据写请求的写入地址的从小到大的顺序返回写请求, 从而可在 PLOG j 的地址空间保序写入, 防止存储节点中与 PLOG j 对应的存储空间出现空洞。

在本申请实施例中, 可基于写请求队列来进行该确定写请求 1' 之前接收的访问 PLOG j 的写请求都已经完成。交换机 St0 根据接收各个写请求时间顺序在该写请求队列中记录各个写请求, 当接收到新的写请求后, 则交换机 St0 将所述写请求的 ID 记录至所述写请求队列的队尾, 当写请求完成之后, 例如从存储节点和校验节点都接收到写请求的全部报文写完成确认信息之后, 在确定该写请求的 ID 为写请求队列的队列头的情况中, 向发送该写请求的计算节点返回写请求完成信息, 并将所述写请求的 ID 从所述写请求队列中移除。可以理解, 本申请实施例不限于通过写请求队列的方式来确认写请求 1' 之前接收的访问 PLOG j 的写请求都已经完成, 其他可以确定写请求 1' 之前接收的访问 PLOG j 的写请求都已经完成的方式也在本发明实施例提供的保护范围之内。

在步骤 S219, 交换机 St0 向计算节点 C0 返回写请求 1' 的完成信息。

交换机 St0 在确定写请求 1' 之前接收的访问 PLOG j 的写请求都已经完成之后, 向计算节点 C0 返回写请求 1' 的相关信息, 该相关信息中包括写请求 1' 成功完成信息, 以及写请求 1' 被分配的 PLOG j 中的写入地址。

计算节点 C0 中的应用在获取该写请求 1' 的成功完成信息之后, 将所述写请求 1' 的写入地址 (即 PLOG j, 地址空间 1') 记录在产生所述写请求 1' 的业务中, 以方便后续业务对写入所述写入地址的数据进行读取操作。当计算节点 C0 中的应用需要读取通过上述写请求 1' 写入的数据时, 可对交换机 St0 发送对 PLOG j 的地址空间 1' 的读请求, 交换机 St0 根据 PLOG j 的元数据, 可从存储节点 S0 和 S1 中的与地址空间 1' 对应的地址空间读取到该数据, 并将该数据返回给计算节点 C0。

当在上述步骤 S202 中判断 PLOG i 的剩余空间足够的情况中, 将与上文中处理写请求 1' 类似地通过执行步骤 S212-S217, 从而完成对写请求 1 的待写数据的写入, 在此不再赘述。

从上述步骤 S209-S219 可以看出, 由于交换机中存储了 PLOG j 的元数据, 所以计算节点只需要发送写请求 1' 至交换机 St0, 由交换机 St0 根据 PLOG j 的元数据为写请求 1' 分配地址空间 1', 且在为 PLOG j 分配了地址空间 1' 之后, 交换机 St0 不需要将该分配的地址返回给计算节点 C0, 从而有效减少了计算节点 C0 的负载, 并减少了数据的写入时延, 从而提升了计算机系统的处理效率。

参考图 3, 计算节点 C1 在计算节点 C0 进行操作①之后, 通过操作②对交换机 St0 发送对 PLOG j 的写请求, 从而, 交换机 St0 在处理写请求 1' 之后才开始处理来自计算节点 C1

的写请求。在处理该写请求的过程中,交换机 St0 在相对于该写请求确定 PLOG j 的剩余空间足够之后,由于 PLOG j 中已经分配给写请求 1' 1.7M 的地址,因此在 PLOG j 中从 1.7M 的起始地址开始为该写请求分配地址,根据该分配的地址,交换机 St0 通过图 3 中的操作⑥请求将写请求的待写数据写入存储节点 S1 中的所示数据条中的从 0.7M 开始的地址中,并通过图 3 中的操作⑦请求将写请求的待写数据写入校验节点 V0 中的所示数据条中的从 1.7M 开始的地址中。

在 PLOG j 写满或者写到预定范围的情况中,校验节点可自行或基于来自计算节点的指示计算 PLOG j 的校验分片。校验节点 V0 可根据 EC 算法或者 RAID 算法,从本地存储的 PLOG j 的两个数据分片计算出一个校验分片,该校验分片的长度为 1M,并将该校验分片另存到 1M 的存储空间中。之后,校验节点 V0 将存储校验分片的存储空间地址信息发送给交换机 St0。交换机 St0 在接收到该地址信息之后在 PLOG j 元数据中更新校验节点的对应存储空间地址信息。

下面相对于上述第二种数据写入方式描述图 2 中的步骤 S209-S219。

图 4 为计算节点 C0 以上述第二种数据写入方式对 PLOG j 进行写入的过程示意图。在图 4 中,与图 3 类似地,在交换机 St0 中存储了 PLOG j 的元数据,图 4 中以数据条示意示出 PLOG j,假设 PLOG j 中地址从左向右从 0 增大到 2M,则该地址中的前 1M 的空间可设定为与存储节点 S0 相对应,该地址中的后 1M 的空间可设定为与存储节点 S1 相对应。同时在与 PLOG j 对应的存储节点 S0、S1 和校验节点 V0 的 SCM 中分别为 PLOG j 分配了一段存储空间,该存储空间也以数据条示意示出,其中存储节点 S0 和 S1 中的存储空间为 1M 的长度,校验节点 V0 中的存储空间为 1M 的长度。

在步骤 S209,计算节点 C0 向交换机 St0 发送对 PLOG j 的写请求 1'。

计算节点 C0 中的应用在获取到新生成的 PLOG j 之后,可重新发起对该 PLOG j 的写请求 1',以写入上述未成功写入的待写数据。所述写请求 1' 用于与上述对 PLOG i 的写请求 1 相区别。类似地,该写请求 1' 中包括 PLOG j 的标识和待写数据。假设该写请求 1' 的待写数据的长度为 2M,即所述 PLOG j 的长度。

在步骤 S210,交换机 St0 对写请求 1' 的待写数据进行分片。

参考图 4,交换机 St0 在接收到写请求 1' 之后,根据第二种写入方式,将写请求 1' 的 2M 待写数据分为两个各 1M 的数据分片 (D1, D2),并基于该两个数据分片生成 1M 的校验分片 CH1。

在步骤 S211,交换机 St0 根据 PLOG j 的元数据确定写入数据分片的存储节点和写入校验数据的校验节点。

所述 PLOG j 的元数据中记录了与 PLOG j 对应的存储节点 S0 和存储节点 S1 中的 1M 空间的地址,与 PLOG j 对应的校验节点 V0 中的 1M 空间的地址,因此,交换机 St0 可根据 PLOG j 的元数据确定写入数据分片 D1 的存储节点 S0 中的写入地址 (即存储节点 S0 中的所述 1M 空间)、写入数据分片 D2 的存储节点 S1 中的写入地址 (即存储节点 S1 中的所述 1M 空间)、及写入校验分片 CH1 的校验节点 V0 中的写入地址 (即校验节点 V0 中的所述 1M 空间)。

在另一种实施方式中,由于初始尚未确定是通过第一种方式还是通过第二种方式进行写入,在所述校验节点中相对于 PLOG j 分配了 2M 存储空间,并在 PLOG j 元数据中记录了该 2M 存储空间的地址,在该情况中,交换机 St0 确定将所述校验分片 CH1 写入校验节点 V0,但并不确定将校验分片 CH1 写入校验节点 V0 的所述分配给 PLOG j 的 2M 空间中,校验节点 V0 在接收到写入校验分片的请求之后将重新为 PLOG j 分配 1M 的空间用于写入校验分片 CH1,

并将该重新分配的空间地址发送给交换机 St0, 交换机 St0 在接收到该重新分配的地址之后, 将 PLOG j 元数据中记录的校验节点 V0 的地址空间修改为该重新分配的地址。

在步骤 S212, 交换机 St0 向各个存储节点发送写入数据分片的写请求 2'。

在该第二种数据写入方式中, 由于交换机 St0 向各个存储节点发送写入数据分片 (即 1M) 的写请求 2', 因此, 在该写请求 2' 中只需要包括各个存储节点中的与 PLOG j 对应的存储空间的地址和对应的数据分片。参考图 4, 交换机 St0 向存储节点 S0 发送写入数据分片 D1 的写请求 2', 交换机 St0 向存储节点 S1 发送写入数据分片 D1 的写请求 2'。

在步骤 S213, 各个存储节点对 PLOG j 对应的存储空间写入对应的数据分片。

在步骤 S214, 各个存储节点在完成上述写入之后, 返回写请求 2' 的完成信息。

在步骤 S215, 交换机 St0 向校验节点发送写入校验数据的写请求 3'。

由于交换机 St0 向校验节点发送写入满校验分片 (即 1M) 的写请求 3', 因此, 在该写请求 3' 中只需要包括校验节点中的与 PLOG j 对应的存储空间的地址和对应的校验分片。参考图 4, 交换机 St0 向校验节点 V0 发送写入校验分片 CH1 的写请求 3'。

在步骤 S216, 校验节点 V0 对 PLOG j 对应的存储空间写入校验数据。

具体是, 校验节点 V0 对 PLOG j 对应的 1M 存储空间写入所述校验分片 CH1。

可以理解, 这里虽然仅示出了一个校验节点作为示例, 在包括多个校验节点的情况中, 交换机 St0 将多个校验分片分别发送给多个校验节点, 每个校验节点执行与校验节点 V0 相同的处理, 以写入与本校验节点对应的校验分片。

在步骤 S217, 校验节点 V0 向交换机 St0 返回写请求 3' 的完成信息。

在该第二种数据写入方式中, 步骤 S218 和步骤 S219 的执行与上文所述相同, 在此不再赘述。

在图 2-图 4 所示的本申请实施例中, 将原来由计算节点或存储节点运行的控制逻辑卸载到作为网络汇集点的可编程中间设备上, 计算节点在写入数据的过程中, 只需要与交换机进行一次通信, 就可以对存储节点和校验节点进行多分片写入, 有效降低了计算节点与交换机之间的通信次数, 减少了计算节点的负载, 且减少了数据写入的时延, 提升了访问效率。

如上文所述, 所述写请求可能以一个报文进行发送, 也可能基于预设的最大报文长度被拆分成多个报文进行发送。所述报文是网络中交换与传输的数据单元, 即数据发送端一次性发送的数据块。在计算机系统中可预先设定报文的最大报文长度和报文中的最大报文数据长度。下文中将参考图 5 和图 6 分别描述在写请求被拆分成多个报文的情况中实施上述第一种数据写入方式和第二种数据写入方式的具体过程。

图 5 为在计算机系统中写数据的过程示意图。该过程与上述第一种数据写入方式相对应。参考图 5, 假设根据预设的最大报文数据长度, 计算节点 C0 在接收到写请求 1' 之后将写请求 1' 拆成顺序排列的 $M_0 \sim M_5$ 六个报文。其中, 报文 $M_0 \sim M_5$ 中的每个报文都为对 PLOG j 的一个写请求, 其写请求编号都为写请求 1' 的写请求编号, 并且其中包括写请求 1' 的待写数据中的一部分数据, 另外, 其中的第一个报文 M_0 中包括写请求 1' 的待写数据的长度。图 5 中以标注报文名称的实线框示意示出各个报文中的待写数据的长度。

参考图 5, 计算节点 C0 首先向交换机 St0 发送对 PLOG j 的写请求 1' 的报文 M_0 。该报文 M_0 中包括写请求 1' 的部分数据, 该部分数据的长度例如等于预先设定的最大报文数据长度, 另外, 该报文 M_0 中包括写请求 1' 的待写数据的长度。

之后, 交换机 St0 根据报文 M_0 中包括的写请求 1' 的待写数据的长度在 PLOG j 中为写请求 1' 分配地址空间 1'。

对写请求 1' 分配地址空间 1' 的具体操作可参考上文的描述, 在此不再赘述。例如, 假设在为写请求 1' 分配地址之前, Offset 的值为 0, 即, PLOG j 当前的未分配地址空间中的起始地址为 0。如图 5 中所示, 假设写请求 1' 的待写数据的长度为 L_7 字节, 则交换机 St0 为写请求 1' 分配的写入地址空间为 $0 \sim L_7 - 1$ 。然后, 交换机 St0 将 PLOG j 的元数据中的变量 Offset 的值更新为 $0 + L_7 = L_7$, 即, 当前 PLOG j 中的未分配地址空间的起始地址为 L_7 。当 St0 在处理完写请求 1' 之后为其它写请求分配写入地址空间时, 则以地址空间 L_7 作为起始地址为所述写请求分配地址空间。

交换机 St0 在为写请求 1' 分配地址空间 1' 之后, 在地址空间 1' 中为报文 M_0 分配地址空间 11'。

参考图 5, 在交换机中可通过变量 offset1 记录该过程中对地址空间 1' 的分配。该变量 offset1 可在 PLOG j 的元数据中进行记录, 或者可以在交换机 St0 的其它存储地址进行记录。变量 offset1 的初始值为地址空间 1' 的起始地址 (即 0)。假设报文 M_0 的待写数据的长度为 L_1 , 因此, 交换机 St0 将地址空间 1' 中的 $0 \sim L_1 - 1$ 的地址作为地址空间 11' 分配给报文 M_0 , 并将变量 offset1 的值改为 L_1 。

之后, 交换机 St0 确定与地址空间 11' (即地址空间 $0 \sim L_1 - 1$) 对应的存储节点中的地址空间 2'。

参考图 5 中的交换机 St0 中的 PLOG j 的数据条, 地址空间 $0 \sim L_1 - 1$ 在 PLOG j 的 $0 \sim 1M$ 的地址空间中, 即, 地址空间 $0 \sim L_1 - 1$ 对应于存储节点 S0。

在一种实现方式中, 交换机可基于地址空间 11' 的起始地址和 PLOG j 的元数据中记录的存储节点中 S0 的对应存储空间的起始地址来计算出存储节点 S0 中与地址空间 11' 对应的地址空间 2' 的起始地址, 并且可确定该地址空间 2' 的长度为报文 M_0 的待写数据的长度。例如, 假设存储节点 S0 中与 PLOG j 对应的存储空间的起始地址为 100, 而如上文所述, 地址空间 11' 的起始地址为 0, 从而地址空间 2' 的起始地址 $= 100 + 0 = 100$, 并且地址空间 2' 的长度为报文 M_0 的待写数据的长度 L_1 。

可以理解, 在上述实施方式中, 通过确定地址空间 2' 的起始地址和地址空间长度来确定地址空间 2', 本申请实施例不限于此, 其他可以确定地址空间 2' 的方式也在本发明实施例提供的保护范围之内。

交换机 St0 在确定存储节点 S0 中与地址空间 11' 对应的地址空间 2' 的起始地址之后, 生成写请求 2' 并将其发送给存储节点 S0, 该写请求 2' 包括地址空间 2' 的信息 (例如起始地址) 和报文 M_0 中的待写数据。

其中, 在 RDMA 连接方式中, 交换机 St0 通过建立与计算节点对应的队列对 (Queue Pairs, QP) (图 4 中未示出) 和三个分别与存储节点 S0、S1 和校验节点 V0 对应的 QP, 维护连接信息。所述 QP 中包括发送队列 (Send Queue, SQ) 和接收队列 (Receive Queue, RQ)。图 5 中示意示出与存储节点 S0、S1 和校验节点 V0 分别对应的三个发送队列 SQ_{S0} 、 SQ_{S1} 和 SQ_{V0} 。交换机 St0 在基于从计算节点 C0 接收的报文生成发送至各个存储节点的写请求的过程中, 由于交换机将从计算节点接收的多个报文分成至少一批以分别发送至至少一个存储节点, 并且在该分批的过程中可能存在将单个报文切分为两个报文以分别发送给不同的存储节点的可能, 因此, 交换机从计算节点接收的报文数与发送至各个存储节点的报文数通常是不同的。因此, 交换机 St0 将报文 M_0 中的数据包序列号 (package serial number, PSN) 替换为与发送队列 SQ_{S0} 对应的 PSN 号以生成写请求 2', 并将该写请求 2' 顺序放入发送队列 SQ_{S0} 中 (图 5 中以发送队列 SQ_{S0} 中的标注 M_0 的方框示出该写请求 2'), 以用于发送给存储节点 S0 的网卡。同

时, 交换机 St0 对发送队列 SQ_{s0} 对应的 PSN 号加 1, 以用于分配给下一个将通过发送队列 SQ_{s0} 发送的报文 (或写请求)。

在发送了与报文 M₀ 对应的写请求 2' 之后, 交换机 St0 可在本地的与 PLOG j 对应的写请求队列中对写请求 1' 进行记录, 以用于在数据写入存储节点和校验节点后, 按照写请求的接收顺序返回写请求。

存储节点 S0 在接收到写请求 2' 之后, 响应于该写请求 2' 在与 PLOG j 对应的存储空间中的地址空间 2' 中写入写请求 2' 中的待写数据。在 RDMA 连接方式中, 存储节点 S0 的网卡在接收到写请求 2' 之后例如通过单边写的方式将写请求 2' 中的待写数据写入存储节点 S0 中地址空间 2' 中。之后, 存储节点 S0 向交换机 St0 返回写请求 2' 完成信息。

同时, 交换机 St0 还对报文 M₀ 中的待写数据进行对校验节点的写入。为此, 交换机 St0 在地址空间 1' 中为报文 M₀ 中的待写数据分配地址空间 12'。如图 5 所示, 可通过交换机 St0 中设置的变量 offset2 进行该分配。该变量 offset2 的初始值例如为地址空间 1' 的起始地址 (这里为 0), 根据报文 M₀ 的长度 L₁, 对报文 M₀ 中的待写数据分配 PLOG j 中的地址空间 0~L₁-1, 同时将 offset2 的值更新为当前值+报文 M₀ 的长度 L₁, 即 0+L₁=L₁。

之后, 交换机 St0 根据元数据中记录的校验节点 V0 的对应存储空间的起始地址确定校验节点 V0 中与地址空间 12' 对应的地址空间 3', 并基于地址空间 3' 和报文 M₀ 中的待写数据生成写请求 3', 并将写请求 3' 发送给校验节点 V0, 图 5 中以 SQ_{v0} 队列中的标注 M₀ 的方框示意示出将发送给校验节点 V0 的写请求 3'。

校验节点 V0 在接收到写请求 3' 之后, 如图 5 中所示, 对与 PLOG j 对应的存储空间中的地址空间 3' 写入报文 M₀ 中的待写数据。在该实例中只包括一个校验节点 V0 作为示例, 在实际中可能存在多个校验节点, 在存在多个校验节点的情况中, 每个校验节点都写入报文 M₀ 的待写数据, 也就是说, 在该存储方式中, 在校验节点计算 PLOG j 的校验分片之前, 每个校验节点都相当于是用于存储 PLOG j 的全部数据的一个副本节点。

参考图 5, 计算节点 C0 在向交换机 St0 发送报文 M₀ 之后, 依次向交换机 St0 发送写请求 1' 的其它报文。其中, 相对于报文 M₁ 和 M₂ 所执行的写入过程与上文相对于报文 M₀ 执行的写入过程基本相同, 在此不再赘述。

计算节点 C0 在向交换机 St0 发送报文 M₃ 之后, 参考图 5 示出的 PLOG j 的数据条, 由于 PLOG j 的 0~1M 的空间不足以存储报文 M₃ 中的待写数据, 因此, 交换机 St0 只对报文 M₃ 分配 PLOG j 的 0~1M 空间中的剩余空间 L₃~L₄-1 作为分配给报文 M₃ 的地址空间, 同时修改变量 offset1 的值。

之后, 交换机 St0 根据分配给报文 M₃ 的地址空间的长度从报文 M₃ 的待写数据中切分出前半部分作为写请求 2' 的待写数据生成写请求 2', 并与上文类似地将写请求 2' 放入发送队列 SQ_{s0} 中, 以发送给存储节点 S0 的网卡。同时, 由于报文 M₃ 不是写请求 1' 的最后一个报文, 因此, 交换机 St0 将报文 M₃ 的剩余数据保存到缓冲区中, 以用于与写请求 1' 的下一个报文的部分数据拼接成整报文 (即具有最大报文数据长度) 写入到对应的存储节点。

存储节点 S0 在接收到与报文 M₃ 的前半部分待写数据对应的写请求 2' 之后, 对地址空间 2' 写入所述前半部分待写数据并向交换机 St0 返回写请求 2' 完成信息。之后, 交换机 St0 和校验节点 V0 执行上述与针对报文 M₀ 的类似的过程, 以在校验节点 V0 的对应存储空间中写入报文 M₃ 的全部待写数据, 即, 在对校验节点进行写入时, 不需要对报文 M₃ 的待写数据进行切分。

计算节点 C0 在向交换机 St0 发送报文 M₄ 之后, 交换机 St0 确定缓冲区中保存有报文 M₃

的后半部分,并且交换机 St0 根据变量 offset1 的值可确定上述地址空间 1' 的未分配地址当前对应于存储节点 S1,因此,交换机 St0 确定 PLOG j 的对应于存储节点 S1 的剩余地址空间与最大报文数据长度中的最小值。这里,最大报文数据长度最小,从而交换机 St0 从报文 M₄ 的待写数据切分出一部分待写数据与报文 M₃ 的后半部分拼接为最大报文数据长度的拼接数据,并在地址 1' 中为该拼接数据分配地址空间(即图 5 中 PLOG j 的地址 L₄~L₅-1),在该分配之后,将 offset1 的值修改为 L₅。同时,交换机 St0 将报文 M₄ 中剩余的待写数据保存到缓冲区中。在另一种情况中,如果 PLOG j 的对应于存储节点 S1 的地址空间的剩余地址空间比最大报文数据长度小,则根据 PLOG j 的对应于存储节点 S1 的地址空间的剩余地址空间来在地址 1' 中分配发送的下一个报文的地址空间,并且根据所述剩余地址空间对缓冲区中的数据和报文 M₄ 中的待写数据进行处理,以使得将发送的下一个报文的待写数据的长度等于所述剩余地址空间的长度。

在分配地址空间 L₄~L₅-1 之后,交换机 St0 首先可根据地址 L₄~L₅-1 确定该地址对应于存储节点 S1。在多分片存储方式中,对于 N 个数据分片的情况,可将 N 个存储节点编号为节点 0~节点 N-1,节点 i 对应的 PLOG 中的地址范围为 [i*分片尺寸, (i+1)*分片尺寸-1],因此,根据分配的 PLOG j 中的地址所在的范围可确定该地址对应于哪个节点,或者根据地址除以分片尺寸的商可确定该地址对应于哪个节点。在图 5 所示实例中,存储节点 S1 (即节点 1) 对应的 PLOG j 中的地址范围为 [1M, 2M-1],地址 L₅=1M,地址 L₅ 除以分片尺寸(即 1M)的商为 1,可确定对应于节点 1。之后,交换机 St0 根据 PLOG j 元数据中的存储节点 S1 的对应存储空间的起始地址,可确定与存储节点 S1 中与地址 L₄~L₅-1 对应的地址空间。之后,交换机 St0 根据上述确定的存储节点 S1 中的地址空间可基于上述生成的拼接数据生成写请求 2',并将该写请求 2' 发送给存储节点 S1。存储节点 S1 在接收到写请求 2' 之后对地址空间 2' 写入所述拼接数据并返回写请求 2' 完成的信息。之后,交换机 St0 在校验节点 V0 的对应存储空间中写入报文 M₄ 的待写数据。

通过将报文 M₃ 的后半部分与报文 M₄ 的前半部分拼接为满报文发送给存储节点,避免了存储节点在接收到非写请求的最后报文的不满报文时误判丢包的情况。

计算节点 C0 在向交换机 St0 发送报文 M₅ 之后,交换机 St0 首先与处理报文 M₄ 类似的,对报文 M₄ 的部分待写数据与报文 M₅ 的部分待写数据的拼接数据分配地址空间,以向存储节点 S1 写入所述拼接数据。交换机 St0 在向存储节点 S1 写入所述拼接数据之后,由于报文 M₅ 为写请求 1' 的最后一个报文,因此,交换机 St0 不将报文 M₅ 的剩余待写数据保存到缓冲区中,而是将该部分数据写入存储节点 S1 的对应的地址空间中。之后,交换机 St0 在校验节点 V0 的对应存储空间中写入报文 M₅ 的待写数据。

交换机 St0 在完成对写请求 1' 各个报文的处理之后,可执行图 2 中的步骤 S218 和 S219,在此不再赘述。

图 6 为本申请实施例提供的另一种在计算机系统中写数据的过程示意图。该过程与上述第二种数据写入方式相对应。

如上文参考图 2 的描述,计算节点 C0 在将写请求 1 发送给交换机 St0 之后,在基于写请求 1 的待写数据的长度确定 PLOG i 的剩余空间足够之后,交换机 St0 可基于该待写数据的长度确定是执行图 5 所示的过程还是执行图 6 所示的过程进行数据写入。如果待写数据的长度在预定范围内(例如对于图 6 所示的长度为 2M 的 PLOG i,可将该预定范围预设为 1.9M~2M-1),则交换机 St0 可确定执行图 6 所示的过程,否则,交换机 St0 可确定执行图 5 所示的过程。在交换机 St0 确定执行图 6 所示的过程的情况中,交换机 St0 在接收到写请求 1 的第一个报

文之后将等待接收了写请求 1 的全部报文之后, 再进行后续的处理。参考图 6, 假设写请求 1 的报文包括报文 $M_0 \sim M_6$ 七个报文, 这七个报文中的待写数据的长度和 (即写请求 1 的待写数据的长度) 例如等于 $PLOG\ i$ 的长度 (即 $2M$)。

交换机 St0 在接收了写请求 1 的全部七个报文之后, 将写请求 1 的待写数据切分为 N 个数据分片。

在图 6 所示实例中, $PLOG\ i$ 与两个存储节点和一个校验节点对应, 即该分片存储为 2 数据分片+1 校验分片的分片方式, 因此, 交换机 St0 将写请求 1 的待写数据切分为 2 个数据分片, 即在 $PLOG\ i$ 中示意示出的在 $0 \sim 1M-1$ 范围内的数据分片 D1 和在 $1M \sim 2M-1$ 范围内的数据分片 D2。在写请求 1 的待写数据小于 $PLOG\ i$ 的长度的情况中, 可通过对该待写数据的预定位置 (例如前部或后部等) 补零, 从而把该待写数据的长度扩充为 $PLOG\ i$ 的长度, 然后再进行数据分片的切分和后续对校验分片的生成。

之后, 交换机 St0 基于 N 个数据分片生成 M 个校验分片。

交换机 St0 可根据 EC 算法或者 RAID 算法基于 N 个数据分片生成 M 个校验分片。在图 6 所示实例中, 交换机 St0 基于 2 个数据分片 D1 和 D2 生成一个校验分片 CH1, 该校验分片 CH1 的长度也为 $1M$ 。

之后, 交换机 St0 对各个存储节点写入对应的数据分片, 并对各个校验节点写入对应的校验分片。

具体是, 为了发送各个数据分片和校验分片, 交换机 St0 需要根据最大报文数据长度将各个数据分片和校验分片分为多个部分, 以通过多个报文进行发送。例如, 如图 6 所示, 对于存储节点 S0, 交换机 St0 首先从数据分片 D1 中分出报文 M_0 中的待写数据, 之后, 存储节点 S0 在具有分片长度的地址中对数据分片 D1 的部分数据 (即图 6 中报文 M_0 中的待写数据) 分配地址空间 11。所述分片长度即为 $1M$, 与图 5 所示类似地, 可通过变量来记录对该分片长度地址的分配。之后, 交换机 St0 根据 $PLOG\ i$ 的元数据中记录的存储节点 S0 中的对应存储空间的起始地址, 确定存储节点 S0 中与地址空间 11 对应的地址空间 2。在确定地址空间 2 之后, 交换机 St0 可基于地址空间 2 和所述部分数据生成写请求 2, 并向存储节点 S0 发送对地址空间 2 的写请求 2。如图 6 所示, 交换机 St0 在生成该写请求 2 之后, 可将该写请求 2 (图 6 中以标注 M_0 的方框示意示出) 放入与存储节点 S0 对应的发送队列 SQ_{s0} 中以用于发送给存储节点 S0。存储节点 S0 在接收到写请求 2 之后, 对地址空间 2 写入报文 M_0 中的待写数据, 然后返回写请求 2 的完成信息。在完成对报文 M_0 中的待写数据的写入之后, 交换机 St0 可通过重复上述过程, 从而将分片 D1 中其它部分数据分别写入存储节点 S0。其中, 如图 6 中所示, 由于在切分数据分片的过程中, 将报文 M_3 的待写数据切分为写入存储节点 S0 和存储节点 S1 的两部分, 因此, 发送给存储节点 S0 的写请求 1 的最后一个报文只包括报文 M_3 的前半部分待写数据。

在进行对分片 D1 的写入之后, 交换机 St0 可同样地通过多次重复上述过程, 而将数据分片 D2 中的四个报文发送给存储节点 S1, 从而存储节点 S1 在其与 $PLOG\ i$ 对应的存储空间中写入数据分片 D2。其中, 在图 6 中的与存储节点 S1 对应的发送队列 SQ_{s1} 中, 示出了分片 D2 的四个报文的待写数据, 其中, 第一个报文的待写数据为报文 M_3 的部分待写数据与报文 M_4 的部分待写数据的拼接数据, 第二个报文的待写数据为报文 M_4 的部分待写数据与报文 M_5 的部分待写数据的拼接数据, 第三个报文的待写数据为报文 M_5 的部分待写数据与报文 M_6 的部分待写数据的拼接数据, 第四个报文的待写数据为报文 M_6 的部分待写数据。

在完成对数据分片至存储节点的写入之后, 交换机 St0 可进行对校验节点写入校验分片。

具体是, 交换机 St0 首先根据 PLOG i 的元数据确定校验节点 V0 中与 PLOG i 对应的地址空间 3, 该地址空间 3 即为校验节点 V0 中初始分配给 PLOG i 的 2M 空间。之后, 交换机 St0 生成写请求 3, 并将写请求 3 和地址空间 3 的信息发送给校验节点 V0。该写请求 3 中的待写数据为从校验分片中切分的具有最大报文数据长度的数据, 在图 6 中以标注为 P₀ 的方框示意示出, 并且该写请求 3 请求校验节点为校验分片分配 1M 的存储空间, 并将该待写数据写入到该新分配的存储空间中。校验节点 V0 在接收到该写请求 3 之后为校验分片分配存储空间, 将数据 P₀ 写入到该新分配的存储空间中, 并向交换机 St0 返回写请求 3 的完成信息和为校验分片分配的存储空间的地址信息。交换机 St0 在接收到为校验分片分配的存储空间的地址信息之后, 在 PLOG i 的元数据中更新校验节点的存储空间的地址信息。

对于校验分片中的后续每个报文, 可分别通过与上述写入数据 P₀ 相同的过程而写入到校验节点中。在此不再赘述。

交换机 St0 在确定对存储节点和校验节点的写入都完成之后, 可执行图 2 中的步骤 S218 和 S219, 在此不再赘述。

图 7 为本申请实施例提供的中间设备的架构图, 所述中间设备连接计算节点、N 个存储节点和 M 个校验节点, 所述计算节点上运行业务, 所述中间设备包括:

接收单元 71, 用于接收所述计算节点发送的对所述业务的数据的写请求;

切分单元 72, 用于基于预设的分片长度对所述数据进行切分得到多个数据;

确定单元 73, 用于根据所述业务对应的逻辑存储空间的元数据, 确定所述 N 个存储节点中的分别写入所述多个数据的多个存储节点及写入所述待写数据的校验数据的 M 个校验节点;

请求单元 74, 用于请求所述多个存储节点将所述多个数据分别写入其与所述逻辑存储空间对应的物理存储空间中, 及请求所述 M 个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中。

在一种可能的实现方式中, 所述逻辑存储空间的元数据包括所述逻辑存储空间的标识, 所述逻辑存储空间的地址信息, 所述逻辑存储空间对应的所述 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的所述 M 个校验节点中的 M 个第二物理存储空间的地址信息, 其中, 所述第一物理存储空间的长度等于所述分片长度, 所述逻辑存储空间的长度等于 N 个所述分片长度之和。

在一种可能的实现方式中, 所述第二物理存储空间的长度等于 N 个所述分片长度之和, 所述待写数据的量小于所述逻辑存储空间的长度, 所述确定单元 73 还用于: 根据所述业务对应的逻辑存储空间的元数据, 确定将所述多个数据分别写入所述多个存储节点的写入地址、及将所述待写数据写入每个校验节点的写入地址。

在一种可能的实现方式中, 所述待写数据的量等于所述逻辑存储空间的长度, 所述切分单元 72 还用于, 对所述数据进行分片得到 N 个数据分片, 所述中间设备还包括: 生成单元 75, 用于基于所述 N 个数据分片生成 M 个校验分片; 所述请求单元 74 还用于, 请求所述 M 个校验节点分别写入所述 M 个校验分片中的一个校验分片。

图 8 为本申请实施例提供的一种中间设备的架构图, 包括处理单元 81 和存储单元 82, 所述存储单元 82 中存储有可执行代码, 所述处理单元 81 执行所述可执行代码以实现如本申请图 2~图 6 所示的方法。

图 9 为本申请实施例提供的一种中间设备的架构图, 包括: 通信接口 91, 用于与存储节点及计算节点进行数据传输; 处理单元 92, 用于对所述通信接口接收的数据进行处理, 以执

行如本申请图 2~图 6 所示的方法。

可以理解,上文所述计算节点、存储节点和校验节点可以为物理服务器,也可以为云服务器(如虚拟服务器)。图 10 为本申请实施例提供的云服务系统的示意图。参见图 10,所述系统包括:计算设备 1001、存储设备 1002 和中间设备 1003。所述计算设备 1001 包括硬件层 10016 和运行在硬件层 10016 之上的虚拟机监控器(VMM)10011,以及多个虚拟机(VM_i)10012。任一所述虚拟机 10012 可以作为云服务系统的一个虚拟的计算节点。所述存储设备 1002 与计算设备 1001 类似地包括硬件层和运行在硬件层之上的虚拟机监控器(VMM),以及多个虚拟机(VM_j),任一所述虚拟机可以作为云服务系统的一个虚拟的存储节点或校验节点。下文将详细描述计算设备 1001 的组成作为示例。

具体的,虚拟机 10012 是通过虚拟机软件在公共硬件资源上模拟出的虚拟的计算机(服务器),虚拟机上可以安装操作系统和应用程序,虚拟机还可访问网络资源。对于在虚拟机中运行的应用程序而言,虚拟机就像是在真正的计算机中进行工作。

硬件层 10016 是虚拟化环境运行的硬件平台,可以由一个或多个物理主机的硬件资源抽象得到的。其中,硬件层可包括多种硬件,例如,硬件层 10016 包括处理器 10014(例如 CPU)和存储器 10015,还可以包括网卡(即 NIC)10013、高速/低速输入/输出(I/O, Input/Output)设备,及具有特定处理功能的其它设备。其中,存储器 10015 可以为易失性存储器(volatile memory),例如随机存取存储器(random-access memory, RAM)、动态随机存取存储器(Dynamic random-access memory, DRAM);所述存储器 10015 也可以为非易失性存储器(non-volatile memory),例如只读存储器(read-only memory, ROM),快闪存储器(flash memory),硬盘(hard disk drive, HDD)、固态硬盘(solid-state drive, SSD)、存储级内存(Storage class memory, SCM)等;所述存储器 10015 还可以包括上述种类的存储器的组合。虚拟机 10012 基于 VMM10011,以及硬件层 10016 提供的硬件资源,运行可执行程序,以实现上述实施例中由计算节点执行的方法步骤。为了简洁,在此不再赘述。

需要理解,本文中的“第一”,“第二”等描述,仅仅为了描述的简单而对相似概念进行区分,并不具有其他限定作用。

本领域的技术人员可以清楚地了解到,本申请提供的各实施例的描述可以相互参照,为描述的方便和简洁,例如关于本申请实施例提供的各装置、设备的功能以及执行的步骤可以参照本申请方法实施例的相关描述,各方法实施例之间、各装置实施例之间也可以互相参照。

本领域技术人员可以理解:实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时,执行包括上述各方法实施例的全部或部分步骤。

在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时,全部或部分地产生按照本发明实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输,例如,所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。

在本申请所提供的几个实施例中，应该理解到，所揭露的装置和方法，在没有超过本申请的范围内，可以通过其他方式实现。例如，以上所描述的实施例仅仅是示意性的，例如，所述模块或单元的划分，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，例如多个单元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。其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。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。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应所述以权利要求的保护范围为准。

权 利 要 求 书

1. 一种计算机系统，其特征在于，包括计算节点、存储节点、校验节点和中间设备，所述计算节点通过所述中间设备访问所述存储节点和所述校验节点；

所述计算节点上运行业务；

所述中间设备用于基于预设的分片长度对写入所述业务对应的逻辑存储空间的待写数据进行切分得到多个数据，根据所述逻辑存储空间的元数据确定写入所述切分得到的数据的存储节点及写入所述待写数据的校验数据的校验节点，将所述切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中，并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中。

2. 根据权利要求 1 所述的系统，其特征在于，所述逻辑存储空间的元数据包括所述逻辑存储空间的标识，所述逻辑存储空间的地址信息，所述逻辑存储空间对应的 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的 M 个校验节点中的 M 个第二物理存储空间的地址信息，其中，所述第一物理存储空间的长度等于所述分片长度，所述逻辑存储空间的长度等于 N 个所述分片长度之和。

3. 根据权利要求 1 或 2 所述的系统，其特征在于，所述第二物理存储空间的长度等于所述分片长度，当写入所述逻辑存储空间的待写数据的量等于所述逻辑存储空间的长度时，所述切分得到的多个数据为 N 个数据分片，所述中间设备还用于计算所述 N 个数据分片的 M 个校验分片，

所述中间设备用于将所述切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中，并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中包括，所述中间设备用于将所述 N 个数据分片分别写入所述 N 个存储节点中的所述第一物理存储空间中，将所述 M 个校验分片分别写入所述 M 个校验节点中的所述第二物理存储空间中。

4. 根据权利要求 1 或 2 所述的系统，其特征在于，所述第二物理存储空间的长度等于 N 个所述分片长度之和，当写入所述逻辑存储空间的待写数据的量小于所述逻辑存储空间的长度时，所述中间设备用于将所述切分得到的数据写入存储节点中与所述逻辑存储空间对应的物理存储空间中，并将所述校验数据写入校验节点与所述逻辑存储空间对应的物理存储空间中包括，所述中间设备用于将切分获取的多个数据分别写入对应的存储节点中的第一物理存储空间中，将所述待写数据写入每个校验节点中的第二物理存储空间中。

5. 一种数据处理方法，其特征在于，所述方法由中间设备执行，所述中间设备连接计算节点、N 个存储节点和 M 个校验节点，所述计算节点上运行业务，所述方法包括：

接收所述计算节点发送的对所述业务的数据的写请求；

基于预设的分片长度对所述写请求的待写数据进行切分得到多个数据；

根据所述业务对应的逻辑存储空间的元数据，确定所述 N 个存储节点中的分别写入所述多个数据的多个存储节点及写入所述待写数据的校验数据的 M 个校验节点；

请求所述多个存储节点将所述多个数据分别写入其与所述逻辑存储空间对应的物理存储空间中；

请求所述 M 个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中。

6. 根据权利要求 5 所述的方法, 其特征在于, 所述逻辑存储空间的元数据包括所述逻辑存储空间的标识, 所述逻辑存储空间的地址信息, 所述逻辑存储空间对应的所述 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的所述 M 个校验节点中的 M 个第二物理存储空间的地址信息, 其中, 所述第一物理存储空间的长度等于所述分片长度, 所述逻辑存储空间的长度等于 N 个所述分片长度之和。

7. 根据权利要求 5 或 6 所述的方法, 其特征在于, 还包括:

从计算节点接收所述逻辑存储空间的信息;

对所述 N 个存储节点分别发送第一物理存储空间分配请求, 所述第一存储空间分配请求用于请求对所述逻辑存储空间中的一个分片长度的空间分配第一物理存储空间;

对所述 M 个校验节点分别发送第二物理存储空间分配请求, 所述第二存储空间分配请求用于请求对所述逻辑存储空间分配第二物理存储空间。

8. 根据权利要求 5-7 任一项所述的方法, 其特征在于, 所述第二物理存储空间的长度等于 N 个所述分片长度之和, 所述写请求的待写数据的量小于所述逻辑存储空间的长度, 所述方法还包括: 根据所述业务对应的逻辑存储空间的元数据, 确定将所述多个数据分别写入所述多个存储节点的写入地址、及将所述待写数据写入每个校验节点的写入地址。

9. 根据权利要求 8 所述的方法, 其特征在于, 根据所述业务对应的逻辑存储空间的元数据, 确定将所述多个数据分别写入所述多个存储节点的写入地址、及将所述待写数据写入每个校验节点的写入地址包括:

为所述写请求在所述逻辑存储空间中分配第一地址;

根据所述逻辑存储空间的元数据确定与所述第一地址对应的多个存储节点及各个存储节点中的写入地址;

根据所述逻辑存储空间的元数据确定每个所述校验节点中的与所述第一地址对应的写入地址。

10. 根据权利要求 5-7 任一项所述的方法, 其特征在于, 所述待写数据的量等于所述逻辑存储空间的长度, 所述基于预设的分片长度对所述写请求的待写数据进行切分得到多个数据包括, 基于所述分片长度对所述待写数据进行分片得到 N 个数据分片,

所述方法还包括: 基于所述 N 个数据分片生成 M 个校验分片;

所述请求所述 M 个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中包括, 请求每个所述校验节点写入所述 M 个校验分片中的一个对应的校验分片。

11. 一种中间设备, 所述中间设备连接计算节点、N 个存储节点和 M 个校验节点, 所述计算节点上运行业务, 所述中间设备包括:

接收单元, 用于接收所述计算节点发送的对所述业务的数据的写请求;

切分单元，用于基于预设的分片长度对所述数据进行切分得到多个数据；

确定单元，用于根据所述业务对应的逻辑存储空间的元数据，确定所述 N 个存储节点中的分别写入所述多个数据的多个存储节点及写入所述待写数据的校验数据的 M 个校验节点；

请求单元，用于请求所述多个存储节点将所述多个数据分别写入其与所述逻辑存储空间对应的物理存储空间中，及请求所述 M 个校验节点分别将所述待写数据的校验数据写入其与所述逻辑存储空间对应的物理存储空间中。

12. 根据权利要求 11 所述的中间设备，其特征在于，所述逻辑存储空间的元数据包括所述逻辑存储空间的标识，所述逻辑存储空间的地址信息，所述逻辑存储空间对应的所述 N 个存储节点中的 N 个第一物理存储空间的地址信息、以及所述逻辑存储空间对应的所述 M 个校验节点中的 M 个第二物理存储空间的地址信息，其中，所述第一物理存储空间的长度等于所述分片长度，所述逻辑存储空间的长度等于 N 个所述分片长度之和。

13. 根据权利要求 11 或 12 所述的中间设备，其特征在于，所述第二物理存储空间的长度等于 N 个所述分片长度之和，所述待写数据的量小于所述逻辑存储空间的长度，所述确定单元还用于：根据所述业务对应的逻辑存储空间的元数据，确定将所述多个数据分别写入所述多个存储节点的写入地址、及将所述待写数据写入每个校验节点的写入地址。

14. 根据权利要求 11 或 12 所述的中间设备，其特征在于，所述待写数据的量等于所述逻辑存储空间的长度，所述切分单元还用于，对所述数据进行分片得到 N 个数据分片，

所述中间设备还包括：生成单元，用于基于所述 N 个数据分片生成 M 个校验分片；

所述请求单元还用于，请求每个所述校验节点写入所述 M 个校验分片中的一个对应校验分片。

15. 一种中间设备，包括处理单元和存储单元，所述存储单元中存储有可执行代码，所述处理单元执行所述可执行代码以实现权利要求 5-10 任意一项所述的方法。

16. 一种中间设备，包括：

通信接口，用于与存储节点及计算节点进行数据传输；

处理单元，用于对所述通信接口接收的数据进行处理，以执行权利要求 5-10 任意一项所述的方法。

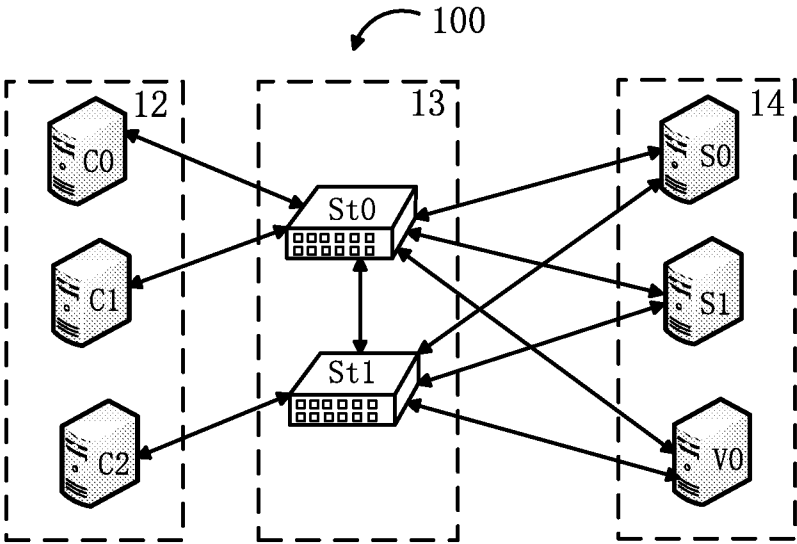


图 1

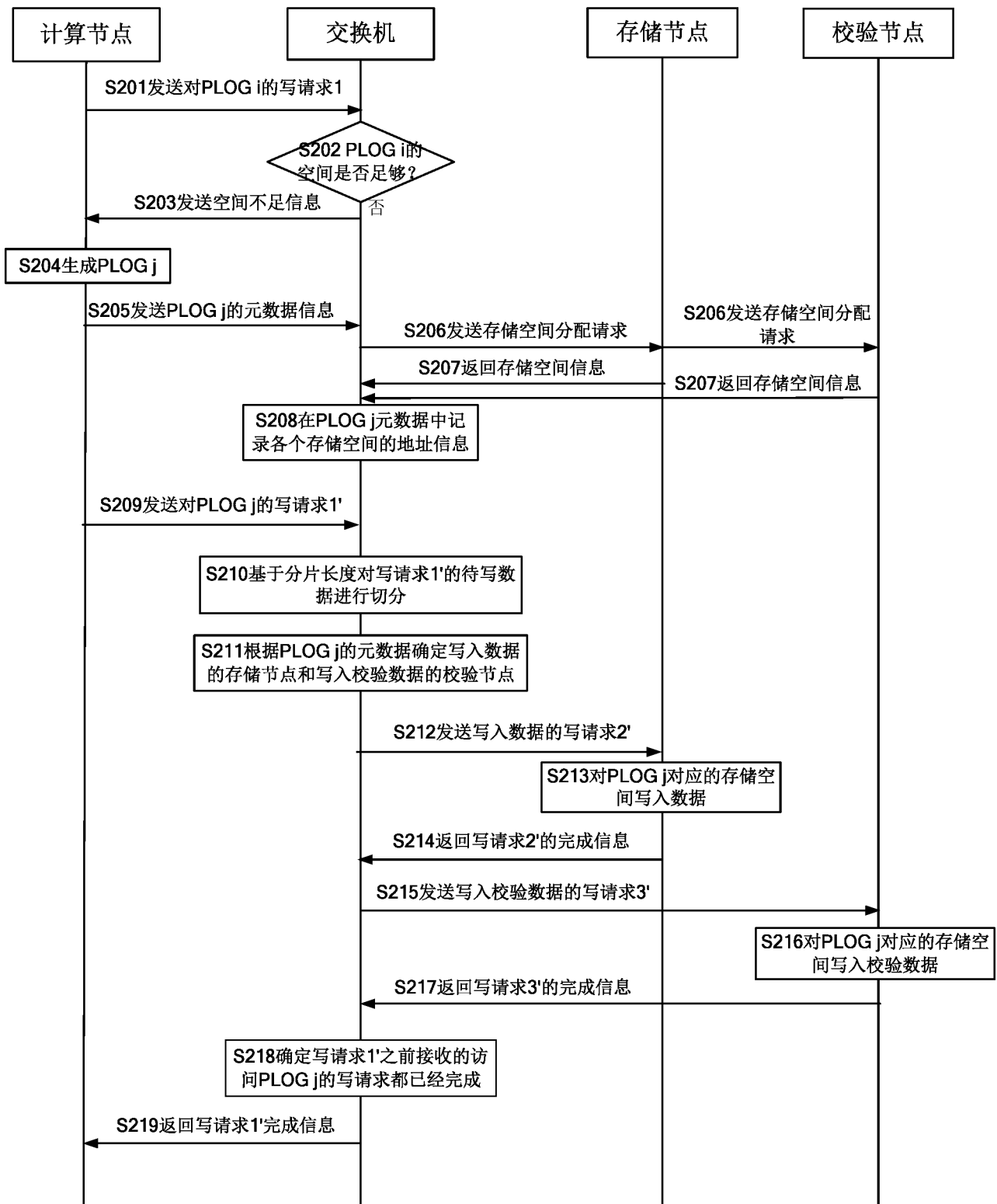


图 2

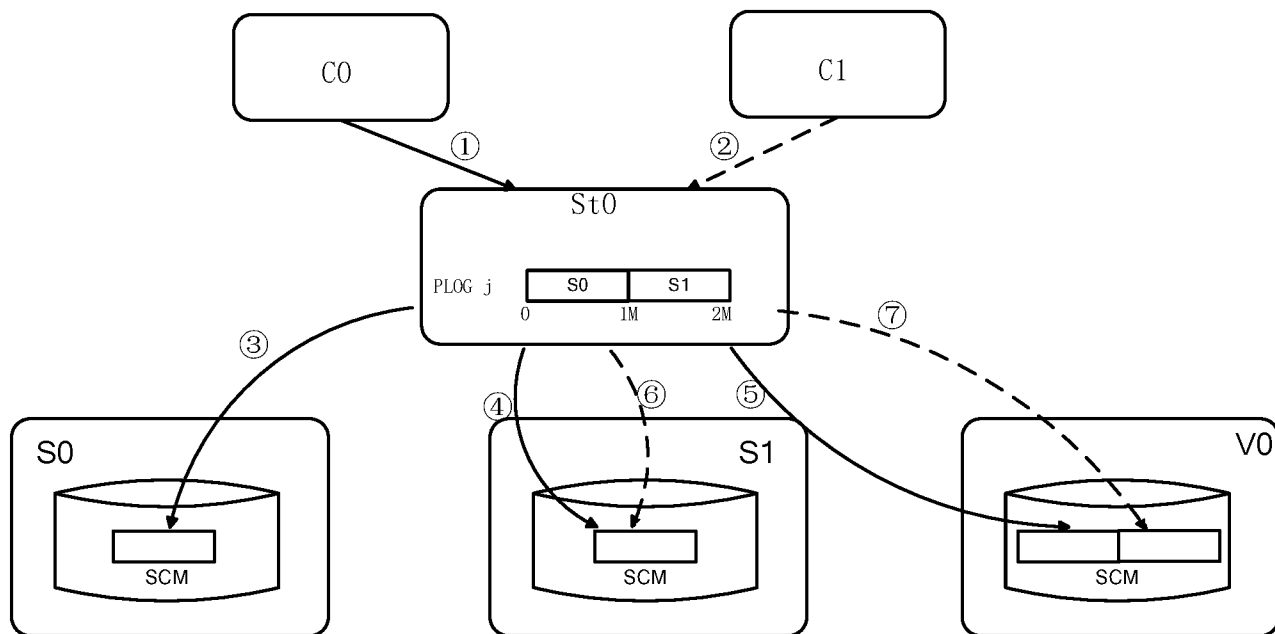


图 3

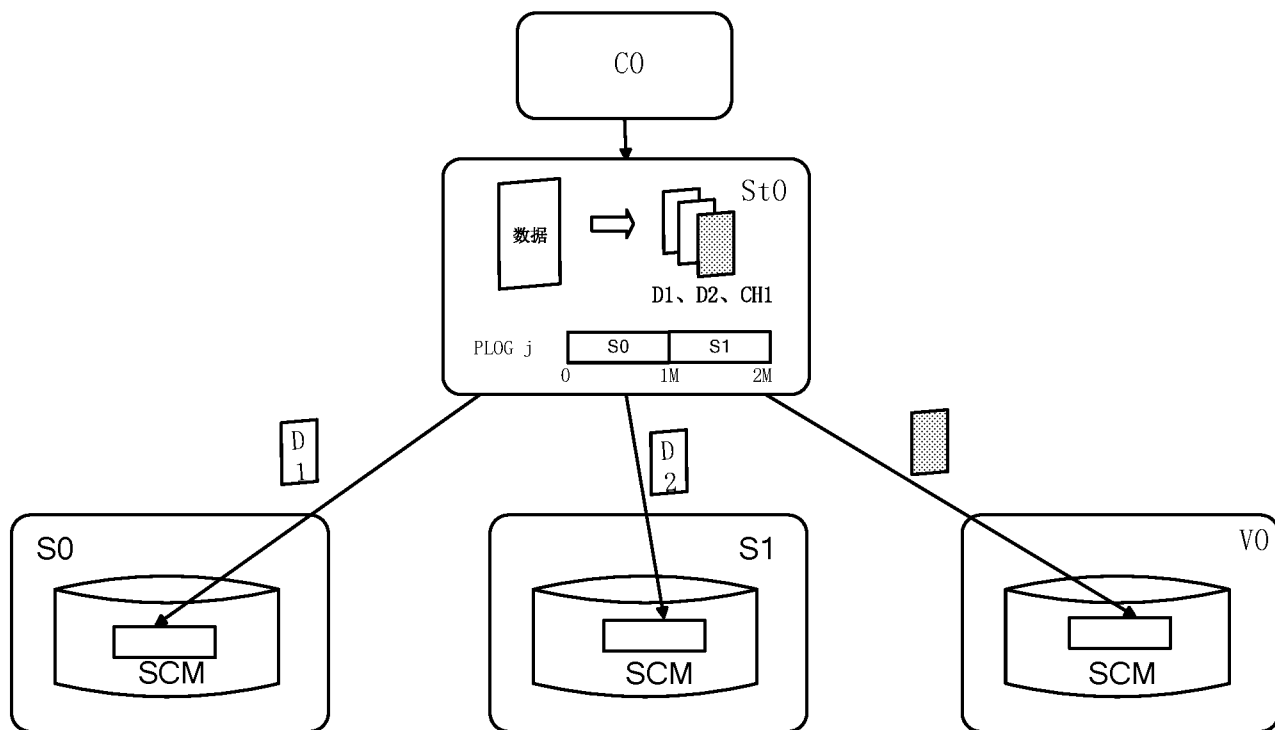


图 4

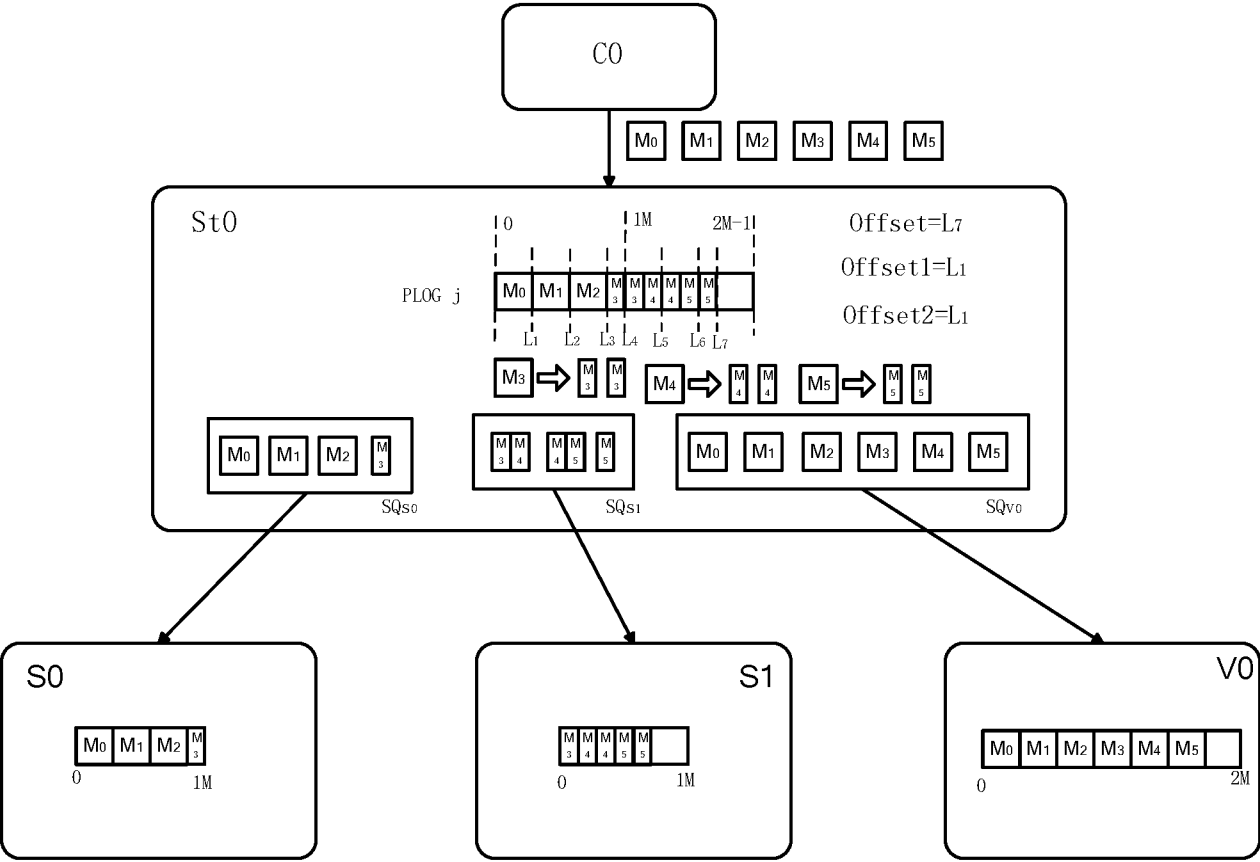


图 5

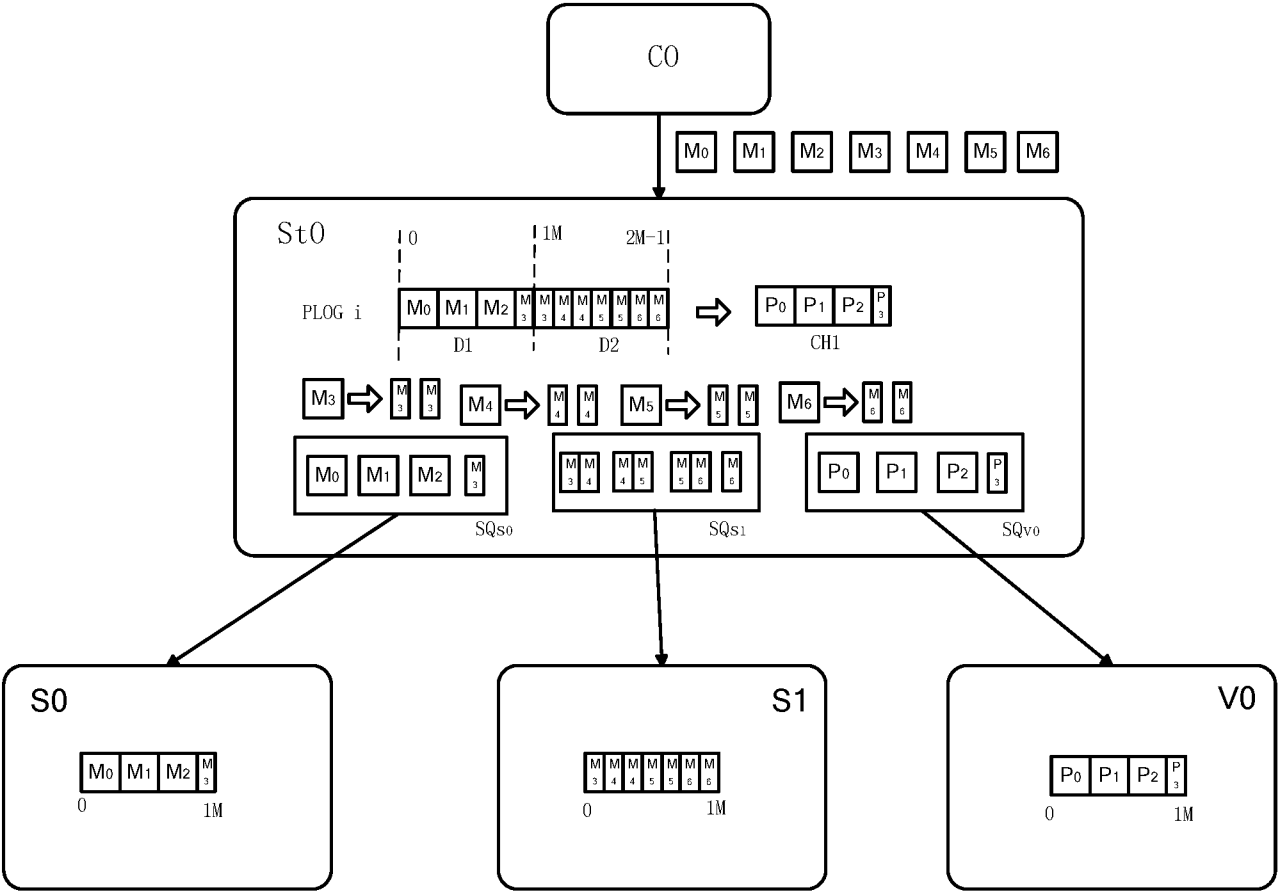


图 6

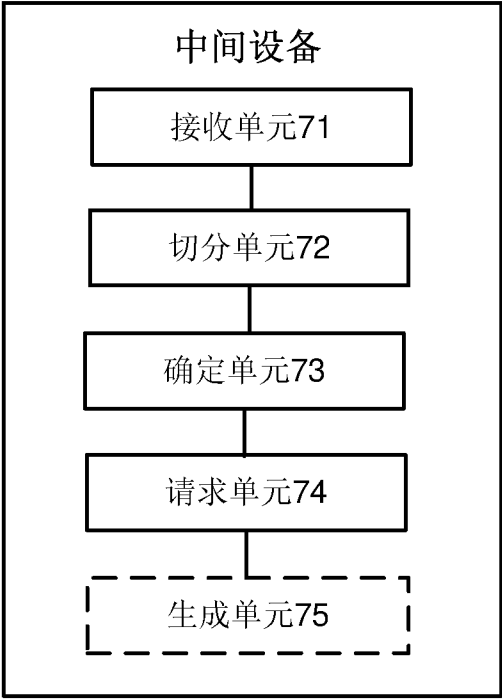


图 7

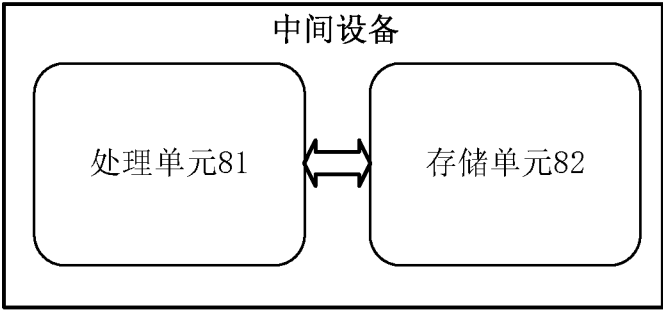


图 8

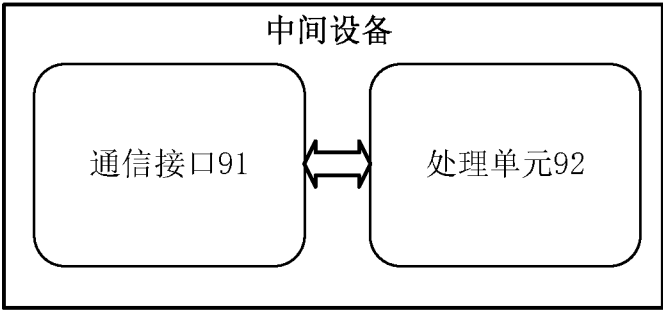


图 9

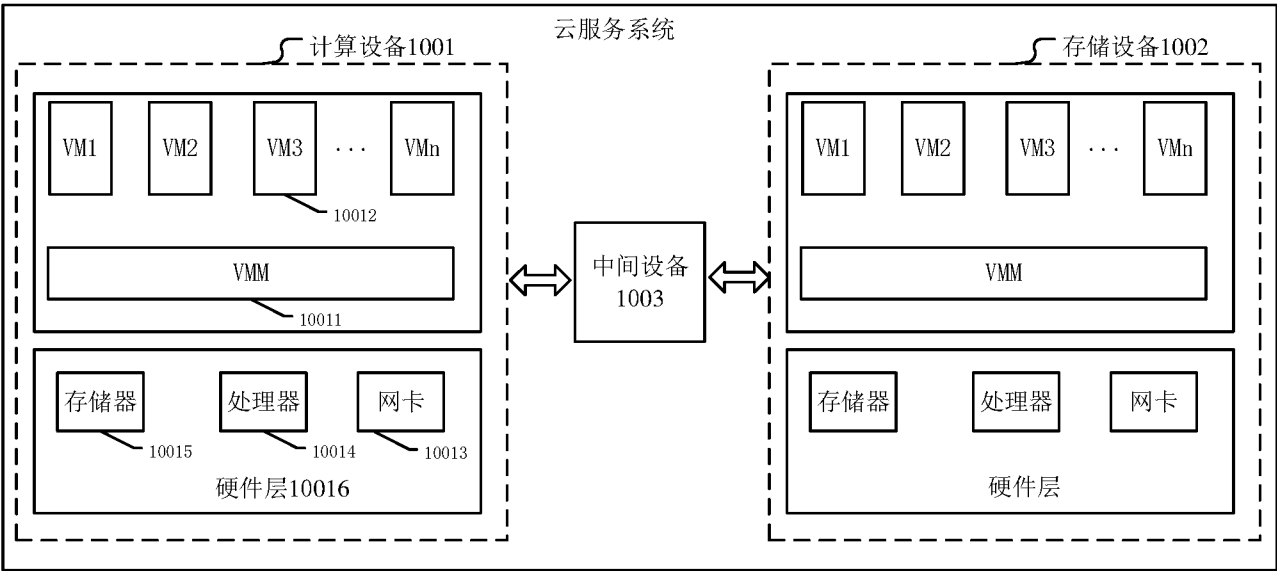


图 10

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/142132

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; CNTXT; CNKI; VEN; USTXT; EPTXT; WOTXT; 3GPP: 节点, 存储, 校验, 中间, 划分, 数据, 分片, 空间, 长度, node, store, check, intermediate, divide, data, slice, space, length

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 109074227 A (HUAWEI TECHNOLOGIES CO., LTD.) 21 December 2018 (2018-12-21) description, paragraphs [0002]-[0118]	1-16
A	CN 103761058 A (TIANJIN ZHONGKE BLUEWHALE INFORMATION TECHNOLOGY CO., LTD.) 30 April 2014 (2014-04-30) entire document	1-16
A	US 10620868 B1 (HITACHI LTD.) 14 April 2020 (2020-04-14) entire document	1-16



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

10 March 2022

Date of mailing of the international search report

18 March 2022

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088, China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2021/142132

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	109074227	A	21 December 2018	CA	2978927	A1	25 May 2018
				CA	2978927	C	17 September 2019
				WO	2018094704	A1	31 May 2018
				AU	2016397189	A1	14 June 2018
				AU	2016397189	B2	25 July 2019
				US	2018364920	A1	20 December 2018
				US	10303374	B2	28 May 2019
				BR	112017020736	A2	17 July 2018
				BR	112017020736	B1	19 January 2021
				BR	112017020736	B8	17 August 2021
				EP	3352071	A1	25 July 2018
				EP	3352071	A4	25 July 2018
				EP	3352071	B1	28 August 2019
				JP	2019504369	A	14 February 2019
				JP	6526235	B2	05 June 2019
				SG	11201707304 R	A	28 June 2018
CN	103761058	A	30 April 2014	None			
US	10620868	B1	14 April 2020	US	2020379668	A1	03 December 2020
				US	10761764	B1	01 September 2020
				US	2020301599	A1	24 September 2020

国际检索报告

国际申请号

PCT/CN2021/142132

A. 主题的分类

H04L 12/00 (2006.01) i

按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))

CNABS; CNTXT; CNKI; VEN; USTXT; EPTXT; WOTXT; 3GPP: 节点, 存储, 校验, 中间, 划分, 数据, 分片, 空间, 长度,
node, store, check, intermediate, divide, data, slice, space, length

C. 相关文件

类 型*

引用文件, 必要时, 指明相关段落

相关的权利要求

X

CN 109074227 A (华为技术有限公司) 2018年12月21日 (2018 - 12 - 21)
说明书[0002]-[0118]段

1-16

A

CN 103761058 A (天津中科蓝鲸信息技术有限公司) 2014年4月30日 (2014 - 04 - 30)
全文

1-16

A

US 10620868 B1 (HITACHI LTD.) 2020年4月14日 (2020 - 04 - 14)
全文

1-16

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的
公布日而引用的或者因其他特殊理由而引用的文件 (如具体
说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解
发明之理论或原理的在后文件“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是
新颖的或不具有创造性“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并
且这种结合对于本领域技术人员为显而易见时, 要求保护的发
明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2022年3月10日

国际检索报告邮寄日期

2022年3月18日

ISA/CN的名称和邮寄地址

中国国家知识产权局 (ISA/CN)

中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

李晓茜

电话号码 62089456

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2021/142132

检索报告引用的专利文件				公布日 (年/月/日)		同族专利	公布日 (年/月/日)
CN	109074227	A	2018年12月21日	CA	2978927	A1	2018年5月25日
				CA	2978927	C	2019年9月17日
				WO	2018094704	A1	2018年5月31日
				AU	2016397189	A1	2018年6月14日
				AU	2016397189	B2	2019年7月25日
				US	2018364920	A1	2018年12月20日
				US	10303374	B2	2019年5月28日
				BR	112017020736	A2	2018年7月17日
				BR	112017020736	B1	2021年1月19日
				BR	112017020736	B8	2021年8月17日
				EP	3352071	A1	2018年7月25日
				EP	3352071	A4	2018年7月25日
				EP	3352071	B1	2019年8月28日
				JP	2019504369	A	2019年2月14日
				JP	6526235	B2	2019年6月5日
				SG	11201707304R	A	2018年6月28日
CN	103761058	A	2014年4月30日	无			
US	10620868	B1	2020年4月14日	US	2020379668	A1	2020年12月3日
				US	10761764	B1	2020年9月1日
				US	2020301599	A1	2020年9月24日