(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
2017 年 7 月 6 日 (06.07.2017)

WIPO | PCT

(10) 国际公布号
WO 2017/113960 A1

(51) 国际专利分类号:
*G06F 3/06* (2006.01)

(21) 国际申请号: PCT/CN2016/103268

(22) 国际申请日: 2016 年 10 月 25 日 (25.10.2016)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201510998928.8 2015 年 12 月 28 日 (28.12.2015) 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: **邱鑫 (QIU, Xin)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 **许慧锋 (XU, Huifeng)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 **郭海涛 (GUO, Haitao)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 **刘洪广 (LIU, Hongguang)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 **刘华伟 (LIU, Huawei)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 **谭春毅 (TAN, Chunyi)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 **吉辛维克多 (GISSIN, Victor)**; 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, JP, KE, KG, 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,

[见续页]

(54) **Title**: DATA PROCESSING METHOD AND NVME STORAGE

(54) **发明名称** : 一种数据处理方法以及 NVMe 存储器

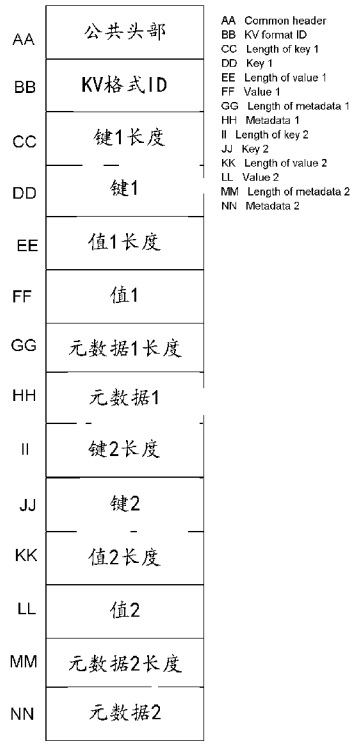



图 2

(57) **Abstract**: Provided are a data processing method and a storage device. The data processing method comprises: an NVMe storage receives an NVMe write command transmitted by a host, the NVMe write command carrying a key and a value pointer, the value pointer pointing to a first storage space, and the first storage space being used to store a value; the NVMe storage acquires, from the NVMe write command, the key and a length of the value and allocates a second storage space for the value according to the length of the value, the second storage space being in the NVMe storage; and the NVMe storage acquires the value from the host and saves the value in the second storage space.

(57) **摘要**: 一种数据处理方法以及存储装置，所述数据处理方法包括：NVMe 存储器接收主机发送的 NVMe 写命令，所述 NVMe 写命令中携带 key 和 value 指针，所述 value 指针指向第一存储空间，所述第一存储空间用于存储 value；所述 NVMe 存储器从所述 NVMe 写命令中获得所述 key 以及获得 value 长度，按照所述 value 长度为所述 value 分配第二存储空间，所述第二存储空间在所述 NVMe 存储器中；所述 NVMe 存储器从所述主机获得所述 value，把所述 value 保存在所述第二存储空间中。

WO 2017/113960 A1

SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, 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))。

# 一种数据处理方法以及NVMe存储器

## 技术领域

本发明实施例涉及存储领域，特别涉及NVMe领域。

## 背景技术

NVMe（Non-Volatile Memory Express,非易失性存储快速）协议是一种使用在存储系统中的高速接口协议，NVMe协议比SCSI协议提供更快的读写速度和更低的延迟，产业重视和普及程度越来越高。

随着信息技术的发展，经常使用对象存储（object storage）技术。一种常见的对象存储技术是键值 (key value，KV)存储。现有技术中，由于NVMe设备仅仅支持块（block）接口，因此如果主机（Host）要把KV数据存储到NVMe存储器中，其步骤是：主机要把KV命令（一般情况下由Key、Value和Metadata构成）转成块数据（例如把一个KV命名拆分/合并成一个或者至少两个块数据）；主机给块数据分配LBA地址；主机把块数据发送给NVMe存储器；NVMe存储器收到块数据后按照分配的LBA地址对块数据逐一进行存储。

然而，在上述步骤中，把KV数据转换成块数据，以及给块数据分配LBA地址会耗费主机大量的运算资源，导致存储系统的性能下降，对主机和存储控制器的运行效率也会造成影响。

## 发明内容

本发明提供数据处理方法、NVMe存储器以及存储系统的方案。可以提高把KV数据写入NVMe存储器的效率。在部分方案中，相应的，也会提高把KV数据从NVMe存储器读出来效率。

本发明实施例第一方面，提供一种数据处理方法，该方法包括：快速非易失性存储NVMe存储器接收主机发送的NVMe写命令，所述NVMe写命令中

携带key，所述NMVe写命令携带value指针，所述value指针指向所述主机中的第一存储空间，所述第一存储空间用于存储value，所述key与所述value属于同一个KV对；所述NVMe存储器从所述NVMe写命令中获得所述key，根据所述value指针获得value长度，按照所述value长度为所述value分配第二存储空间，所述第二存储空间在所述NVMe存储器中；所述NVMe存储器发送第一传输请求给所述主机，从所述主机获得所述value，把所述value保存在所述第二存储空间中。基于该方案，KV数据从主机传递到NVMe存储器的过程中，不需要把KV数据转换成块的形式，提高了KV数据的存储效率。

在第一方面的第一种可能的实施方式中，所述NVMe存储器发送第一传输请求给主机以及从所述主机获得所述value，具体包括：所述NVMe存储器发送DMA传输请求给所述主机，从所述主机获得所述value，所述DMA指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址，其中，所述NVMe存储器和所述主机用PCIe总线连接。该方案提供了使用DMA的方式存储value传输方式。

在第一方面的第二种可能的实施方式中，所述NVMe存储器发送第一传输请求给主机以及从所述主机获得所述value，具体包括：所述存储器发送RDMA传输请求给所述主机，从所述主机获得所述value，所述RDMA指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址，其中，所述NVMe存储器和所述主机用Fabric总线连接。该方案提供了使用RDMA的方式存储value传输方式。

在第一方面的第三种可能的实施方式中，其中：所述NVMe写命令中进一步携带KV数量的字段，所述KV数量的字段用于描述所述NVMe写命令中KV的数量，所述NVMe存储器从所述NVMe写命令中获得与KV数量相同数量的key，以及获得与KV数量相同数量的value。该方案可以支持正在同一个NVMe写命令中携带复数个KV。

在第一方面的第四种可能的实施方式中，其中：所述NVMe写命令中进一步携带KV格式的字段，其中，所述KV格式的字段描述所述NVMe写命令中字段的结构，所述NVMe存储器按照所述KV格式字段所定义的字段内容，从所述NVMe写命令中获取各个字段。该方案可以使得同一个NVMe设备支持多种格式传输KV的NVMe写命令报文，在一次特定的写操作过程中，使用其中一种格式的写命令报文。类似的，在读命令流程中，也可以使得同一个NVMe设备支持多种格式传输KV的NVMe读命令报文。

在第一方面的第五种可能的实施方式中，其中，所述NVMe写命令还携带所述value元数据指针，所述方法还包括：所述NVMe存储器根据所述元数据指针获得元数据长度，按照所述元数据长度为所述元数据分配第四存储空间，所述第四存储空间在所述NVMe存储器中；所述NVMe存储器通过所述第一传输请求从所述主机中获得所述元数据，把所述元数据保存在第四存储空间中。该方案描述了如何对KV中的元数据进行存储的具体方案。

结合第一方面的第五种可能的实施方式，在第六种可能的实施方式中，其中：所述元数据传输失败后，所述NVMe存储器释放为所述value分配的存储空间，以及释放为所述元数据分配的第四存储空间。该方案在元数据传输失败后，可以及时释放被value占用的资源。

结合第一方面的第七种可能的实施方式，其中，所述NVMe存储器发送第一传输请求给所述主机以及从所述主机获得所述value，具体包括下述其中一种：所述存储器发送至少两个DMA传输请求给所述主机以获得所述value，每个DMA传输请求用于请求获得所述value的一部分，当任意一个DMA传输请求执行失败，则NVMe存储器释放为所述value分配的存储空间，其中，所述NVMe存储器和所述主机用PCIe总线连接；或者，所述存储器发送至少两个RDMA传输请求给所述主机以获得所述value，每个RDMA传输请求用于请求获得所述value的一部分，当任意一个RDMA传输请求执行失败，则NVMe存储器释放为所述value分配的存储空间，其中，所述NVMe存储器和所述主

机用Frabic连接。该方案可以把同一个value分成多次RDMA传输请求进行传输，减小每次RDMA传输的数据量。

结合第一方面的第八种可能的实施方式，其中，所述方法还包括：生成所述key与所述第二存储空间的映射关系。该映射关系可以为后续的KV（尤其是value）的读取方案提供基础。

结合第一方面的第九种可能的实施方式，该方法之后，进一步包括：所述NVMe存储器从所述主机接收NVMe读命令，所述NVMe读命令中携带所述key。此外，还可以进一步包括：所述NVMe存储器根据所述key在所述映射关系中查询，获得存储所述value的所述第二存储空间的位置信息，把所述第二存储空间的位置信息发送给所述主机；所述NVMe存储器接收所述主机发送的第二传输请求，所述第二传输请求用于请求获得存储在所述第二存储空间的数据。该第一方面的第八种可能的实施方式可以实现对KV（或者value）的读取。

结合第一方面的第十种可能的实施方式，其中，所述NVMe存储器接收所述主机发送的第二传输请求之前，进一步包括：所述主机根据所述第二存储空间的大小预留所述主机中的第三存储空间。所述NVMe存储器发送所述value给所述主机之后，进一步包括：所述主机把收到的来自所述第二存储空间的数据写入所述第四存储空间空间。该方案描述了主机在读取KV（或者value）的过程中所执行的操作。

结合第一方面的第十一种可能的实施方式，所述NVMe存储器接收的所述主机发送的NVMe读命令中还携带所述主机的空闲空间信息，所述NVMe存储器从所述主机接收所述NVMe读命令之后，进一步包括：所述NVMe存储器判断所述主机的空闲存储空间是否大于等于所述第二存储空间，如果是，执行将所述第二存储空间的位置信息发送给所述主机的步骤，如果否，结束步骤。该方案在NVMe读命令中携带所述主机的空闲空间信息，主机不用在

收到NVMe存储器响应后判断空闲空间是否足够以及预留存储空间，可以减少NVMe存储器和主机之间的交互次数，提高了主机读取value的效率。需要说明的是，这里的空闲空间是一段地址连续的空间。

结合第一方面的第十二种可能的实施方式，其中:所述第一存储空间用第一存储空间的首地址和所述value长度描述；所述第二存储空间用第二存储空间的首地址描述。该方案描述了存储空间的描述方式。依靠该描述方式所描述的内容，可以定位读出value的存储位置和写入value的存储位置。

第二方面，本发明提供一种NVMe存储器的实施方式，包括控制器以及存储介质，所述控制器和所述存储介质连接，所述存储介质用于提供存储空间，所述处理器被配置为执行：接收主机发送的快速非易失性存储NVMe写命令，所述NVMe写命令中携带key，所述NMVe写命令携带value指针，所述value指针指向所述主机中的第一存储空间，所述第一存储空间用于存储value，所述key与所述value属于同一个KV对；从所述NVMe写命令中获得所述key，根据所述value指针获得value长度，按照所述value长度为所述value分配第二存储空间，所述第二存储空间在所述存储介质中；发送第一传输请求给所述主机，从所述主机获得所述value，把所述value保存在所述第二存储空间中。NVMe存储器被配置为为执行地一方面的方法以及第一方面各种实现方式的方法。

本发明第三方面提供一种数据处理方法的实施方法，该方法包括：非易失性存储快速NVMe存储器接收NVMe写命令，所述NVMe写命令的头部携带键key，所述NMVe命令还携带值value，所述key和所述value对应，所述key和所述value属于同一个KV对；所述NVMe存储器从所述NVMe写命令中获得所述key以及所述value；所述NVMe存储器把所述value保存在所述NVMe存储器的存储介质中。

结合第三方面的第一种可能的实施方式，所述NVMe写命令中进一步携

带KV数量的字段，所述KV数量的字段用于描述所述NVMe写命令中KV 的数量，所述NVMe存储器从所述NVMe写命令中获得与KV数量相同数量的key，以及获得与KV数量相同数量的value。

结合第三方面的第二种可能的实施方式，其中，所述NVMe写命令中进一步携带KV格式的字段，其中，所述KV格式的字段描述所述NVMe写命令中字段的结构，所述NVMe存储器按照所述KV格式字段所定义的字段内容，从所述NVMe写命令中获取各个字段。

结合第三方面的第三种可能的实施方式，其中，所述NVMe写命令中进一步携带所述key的长度，所述NVMe存储器从所述写命令中获得所述key具体包括：从所述key的预设起始位置，按照所述key的长度从所述写命令中获得所述key。

结合第三方面的第四种可能的实施方式，其中，所述NVMe写命令中进一步携带value偏移量以及所述value的长度，所述NVMe存储器从所述NVMe写命令中获得所述value具体包括：所述NVMe存储器从所述偏移量指示的位置，按照所述value的长度获得所述value。

结合第三方面的第五种可能的实施方式，该方法进一步包括：生成所述key与所述value存储空间的映射关系。

第三种可能的实现方式，是基于本申请第三方面的第五种可能实现方式，该方法之后，进一步包括：主机发送NVMe读命令给所述NVMe存储器，所述NVMe读命令中携带所述key；所述NVMe存储器从所述主机接收所述读命令，从所述NVMe读命令中获得所述key；所述NVMe存储器使用所述key从所述映射关系中查找所述value存储空间；所述NVMe存储器从使用所述value存储空间获得所述value；所述NVMe存储器生成所述NVMe读命令的响应消息发送给所述主机，所述响应消息携带所述value。

第四方面，本发明提供一种NVMe存储器的实施方式，包括控制器以及

存储介质。 所述控制器被配置为为执行地三方面的方法以及第三方面各种实现方式的方法。

本发明实施例第五方面，提供一种存储装置，该存储装置可以是物理设备，例如NVMe存储器；也可以是逻辑上设备，例如是运行在NVMe存储器的处理器中的程序，或者存储服务器中的程序。该装置包括：接口模块，用于接收主机发送的NVMe写命令，所述NVMe写命令中携带key，所述NMVe写命令携带value指针，所述value指针指向所述主机中的第一存储空间，所述第一存储空间用于存储value，所述key与所述value属于同一个KV对；处理模块，用于从所述NVMe写命令中获得所述key，根据所述value指针获得value长度，按照所述value长度为所述value分配第二存储空间；存储模块，用于发送第一传输请求给所述主机，从所述主机获得所述value，把所述value保存在所述第二存储空间中。所述第二存储空间可以由在所述NVMe存储器的存储介质提供，所述存储介质和所述NVMe存储器的处理器连接。基于该方案，KV数据从主机传递到NVMe存储器的过程中，不需要把KV数据转换成块的形式，提高了KV数据的存储效率。

在第五方面的第一种可能的实施方式中，所述存储模块发送第一传输请求给主机以及从所述主机获得所述value，具体包括：所述存储模块发送DMA传输请求给所述主机，从所述主机获得所述value，所述DMA指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址其中。其中，所述存储装置（存储装置是硬件）和所述主机用PCIe总线连接，或者所述存储装置（存储装置是软件）所在的NVMe存储器和所述主机用PCIe总线连接。该方案提供了使用DMA的方式存储value传输方式。

在第五方面的第二种可能的实施方式中，所述存储模块发送第一传输请求给主机以及从所述主机获得所述value，具体包括：所述存储模块发送RDMA传输请求给所述主机，从所述主机获得所述value，所述RDMA指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址其

中，所述存储装置（存储装置是硬件）和所述主机用Fabric总线连接，或者所述存储装置（存储装置是软件）所在的NVMe存储器和所述主机用Fabric总线连接。该方案提供了使用RDMA的方式存储value传输方式。

在第五方面的第三种可能的实施方式中，其中：所述NVMe写命令中进一步携带KV数量的字段，所述KV数量的字段用于描述所述NVMe写命令中KV的数量，所述NVMe存储器从所述NVMe写命令中获得与KV数量相同数量的key，以及获得与KV数量相同数量的value。该方案可以支持正在同一个NVMe写命令中携带复数个KV。

在第五方面的第四种可能的实施方式中，其中：所述NVMe写命令中进一步携带KV格式的字段，其中，所述KV格式的字段描述所述NVMe写命令中字段的结构，所述NVMe存储器按照所述KV格式字段所定义的字段内容，从所述NVMe写命令中获取各个字段。该方案可以使得同一个NVMe设备支持多种格式传输KV的NVMe写命令报文，在一次特定的写操作过程中，使用其中一种格式的写命令报文。类似的，在读命令流程中，也可以使得同一个NVMe设备支持多种格式传输KV的NVMe读命令报文。

在第五方面的第五种可能的实施方式中，其中，所述NVMe写命令还携带所述value元数据指针，所述方法还包括：所述NVMe存储器根据所述元数据指针获得元数据长度，按照所述元数据长度为所述元数据分配第四存储空间，所述第四存储空间在所述NVMe存储器中；所述NVMe存储器通过所述第一传输请求从所述主机中获得所述元数据，把所述元数据保存在第四存储空间中。该方案描述了如何对KV中的元数据进行存储的具体方案。

结合第五方面的第五种可能的实施方式，在第六种可能的实施方式中，其中：所述元数据传输失败后，所述处理模块还用于释放为所述value分配的存储空间，以及释放为所述元数据分配的第四存储空间。该方案在元数据传输失败后，可以及时释放被value占用的资源。

结合第五方面的第七种可能的实施方式，其中，所述存储模块发送第一传输请求给所述主机以及从所述主机获得所述value，具体包括下述其中一种：所述处理模块发送至少两个DMA传输请求给所述主机以获得所述value，每个DMA传输请求用于请求获得所述value的一部分，当任意一个DMA传输请求执行失败，则所述存储模块释放为所述value分配的存储空间，其中，所述NVMe存储器和所述主机用PCIe总线连接；或者，所述存储模块发送至少两个RDMA传输请求给所述主机以获得所述value，每个RDMA传输请求用于请求获得所述value的一部分，当任意一个RDMA传输请求执行失败，则所述存储模块释放为所述value分配的存储空间，其中，所述NVMe存储器和所述主机用Frabic连接。该方案可以把同一个value分成多次RDMA传输请求进行传输，减小每次RDMA传输的数据量。

结合第五方面的第八种可能的实施方式，其中，所述处理模块还用于：生成所述key与所述第二存储空间的映射关系。该映射关系可以为后续的KV（尤其是value）的读取方案提供基础。

结合第五方面的第九种可能的实施方式，所述接口模块还用于从所述主机接收NVMe读命令，所述NVMe读命令中携带所述key。所述处理模块，还用于根据所述key在所述映射关系中查询，获得存储所述value的所述第二存储空间的位置信息，把处理模块还用于，通过所述接口模块把所述第二存储空间的位置信息发送给所述主机；所述接口模块还用于接收所述主机发送的第二传输请求，所述第二传输请求用于请求获得存储在所述第二存储空间的数据。该第五方面的第八种可能的实施方式可以实现对KV（或者value）的读取。

结合第五方面的第十种可能的实施方式，其中，所述接口模块在接收所述主机发送的第二传输请求之前，所述主机还用于：根据所述第二存储空间的大小预留所述主机中的第三存储空间。所述存储装置发送所述value给所述主机之后，所述主机还用于把收到的来自所述第二存储空间的数据写入所述

第四存储空间空间。该方案描述了主机在读取KV（或者value）的具有的功能。

结合第五方面的第十一种可能的实施方式，所述接口模块接收的所述主机发送的NVMe读命令中还携带所述主机的空闲空间信息，所述NVMe存储器从所述主机接收所述NVMe读命令之后，进一步包括：所述NVMe存储器判断所述主机的空闲存储空间是否大于等于所述第二存储空间，如果是，执行将所述第二存储空间的位置信息发送给所述主机的步骤，如果否，结束步骤。该方案在NVMe读命令中携带所述主机的空闲空间信息，主机不用在收到NVMe存储器响应后判断空闲空间是否足够以及预留存储空间，可以减少NVMe存储器和主机之间的交互次数，提高了主机读取value的效率。需要说明的是，这里的空闲空间是一段地址连续的空间。

结合第五方面的第十二种可能的实施方式，其中:所述第一存储空间用第一存储空间的首地址和所述value长度描述；所述第二存储空间用第二存储空间的首地址描述。该方案描述了存储空间的描述方式。依靠该描述方式所描述的内容，可以定位读出value的存储位置和写入value的存储位置。

第六方面：结合第五方面及第五方面的各种可能的实现方式提供的存储装置，本发明还还提供存储系统的实施方式，存储系统包括主机和存储装置。

本发明第七方面提供一种存储装置，该存储装置可以是物理设备，例如NVMe存储器；也可以是逻辑上设备，例如是运行在NVMe存储器的处理器中的程序，或者存储服务器中的程序。该装置包括:接口模块，用于接收NVMe写命令，所述NVMe写命令的头部携带键key，所述NMVe命令还携带值value，所述key和所述value对应，所述key和所述value属于同一个KV对；处理模块，用于从所述NVMe写命令中获得所述key以及所述value；存储模块，用于把所述value保存在所述NVMe存储器的存储介质中。

结合第七方面的第一种可能的实施方式，所述NVMe写命令中进一步携

带KV数量的字段，所述KV数量的字段用于描述所述NVMe写命令中KV 的数量，所述处理模块从所述NVMe写命令中获得与KV数量相同数量的key，以及获得与KV数量相同数量的value。

结合第七方面的第二种可能的实施方式，其中，所述NVMe写命令中进一步携带KV格式的字段，其中，所述KV格式的字段描述所述NVMe写命令中字段的结构，所述处理模块按照所述KV格式字段所定义的字段内容，从所述NVMe写命令中获取各个字段。

结合第七方面的第三种可能的实施方式，其中，所述NVMe写命令中进一步携带所述key的长度，所述处理模块从所述写命令中获得所述key具体包括：从所述key的预设起始位置，按照所述key的长度从所述写命令中获得所述key。

结合第七方面的第四种可能的实施方式，其中，所述NVMe写命令中进一步携带value偏移量以及所述value的长度，所述处理模块从所述NVMe写命令中获得所述value具体包括：所述处理模块从所述偏移量指示的位置，按照所述value的长度获得所述value。

结合第七方面的第五种可能的实施方式，所述存储模块进一步用于：生成所述key与所述value存储空间的映射关系。

第七种可能的实现方式，是基于第七方面的第五种可能实现方式，提供一种主机的说是方式：主机用于发送NVMe读命令给所述NVMe存储器，所述NVMe读命令中携带所述key；所述接口模块还用于从所述主机接收所述读命令，从所述NVMe读命令中获得所述key；所述处理模块还用于使用所述key从所述映射关系中查找所述value存储空间；所述处理模块还用于从使用所述value存储空间获得所述value；所述处理模块还用于，生成所述NVMe读命令的响应消息，通过所述接口模块把响应消息发送给所述主机，所述响应消息携带所述value。

第八方面：结合第七方面及第七方面的各种可能的实现方式提供的存储装置，本发明还还提供存储系统的实施方式，存储系统包括主机和存储装置。

第九方面：结合第二方面及第二方面的各种可能的实现方式提供的存储器，本发明还还提供存储系统的实施方式，存储系统包括主机和NVMe存储器。

第十方面：结合第四方面及第四方面的各种可能的实现方式提供的存储器，本发明还还提供存储系统的实施方式，存储系统包括主机和存储装置。

基于本发明提供数据处理方法、NVMe存储器以及存储系统的方案。可以提高把KV数据写入NVMe存储器的效率。

**附图说明**

图1是一种数据处理方法实施例的流程图；

图2是一种NVMe命令格式示意图；

图3是一种NVMe命令格式示意图；

图4是一种NVMe命令格式示意图；

图5是一种NVMe命令格式示意图；

图6是一种key和value存储位置、metadata存储位置映射关系示意图；

图7是一种数据处理系统硬件拓扑实施例拓扑图；

图8是一种数据处理方法实施例的流程图；

图9是一种命令格式示意图；

图10是一种命令格式示意图；

图11是一种命令格式示意图；

图12是一种写数据方法实施例流程图；

图13是一种读数据方法实施例流程图；

图14是一种写数据方法实施例流程图；

图15一种数据处理系统实施例逻辑功能图。

**具体实施方式**

快速非易失性存储（Non-Volatile Memory Express，NVMe）是一种逻辑设备接口，可以支持使用PCIe总线访问非易失性存储介质。NVMe接口可以用于flash介质的存储器，例如固态硬盘SSD。NVMe接口的设备，称为NVMe设备。NVMe存储器是NVMe设备的一种，指具有存储功能的NVMe设备，下面主要以NVMe存储器（本发明实施例中，把NVME存储器简称存储器）为例进行介绍。

KV存储，也称为K/V存储，是存储技术的一种。在KV存储中，键-值（key-value/key value）对是基本数据模型。Key-value对可以包括键（key）和值（value）。对key-value对进行扩展之后，Key-value对还可以包括元数据（Metadata）。key唯一标记一个value。

如果主机以块的方式把KV数据发送给NVMe存储器进行存储，主机需要执行的操作包括：首先，把KV数据转换为块的形式；然后，给块分配存储器上的逻辑块地址（Logic Block Address，LBA）；接着，生成写命令，写命令中携带块以及块的LBA；NVMe存储器收到块后，按照LBA对应的物理地址对块进行存储。如果要读取这些KV数据，需要先读取块数据，然后把块数据转换成KV数据。写入KV数据过程中，分配LBA地址以及把KV数据转换成块数据需要耗费主机大量的时间和系统资源，增加系统延迟。读取KV数据的过程中，主机需要发送待读块的位置信息（例如起始地址+长度）给NVMe存储器，获得到块数据后，把块数据重新转换成KV数据也会耗费主机大量的运算资源，增加系统延迟。

本申请可以适用NVMe场景以及NVMe Over Fabric（NOF）的场景。NVMe场景中，主机和存储器通过总线（例如PCIe总线）连接，在这种场景下，存

储器作为主机的一个组件，例如主机是服务器，存储器是服务器中的NVMe接口固态硬盘（SSD）。NOF场景中，主机和存储器通过fabric（例如以太网、FC等）连接。主机包括内存和处理器以及接口，主机和存储器这二者可以是两个独立的设备。本发明实施例中，主机是读数据和写数据的发起者，因此称作发起者（initiator）；存储器是读请求或者写请求的响应者，也称作目标器（target）。

本发明实施例可以扩展现有的NVMe协议，提出了新的NVMe命令。新的NVMe命令对现有的NVMe命令（例如NVMe标准协议1.2.1中所定义的NVMe命令，或者NVMe over Fabric 标准协议1.0所定义的NVMe命令）进行了的扩展和优化，使得NVMe设备可以直接支持KV的接口。

由于NVMe设备可以直接支持KV的接口，因此存储器和主机之间可以直接通过NVMe协议传输KV数据。主机可以通过NVMe协议直接把KV数据写入存储器，或者通过NVMe协议直接从存储器中读出KV。因此，存储器和主机之间避免了像现有技术那样在KV格式和Block格式之间的反复转换，降低了系统复杂度，提高了系统性能。在没有特别说明情况下，本发明各实施例中的NVMe命令均是指扩展的NVMe命令。

在一种可选的实现方式中，key直接携带在NVMe命令的命令头中，因此，直接读取NVMe的命令头（不需要解析NVMe命令的净荷）就可以获得key。

对于value，可以携带在NVMe命令中，读取NVMe命令可以获得所述value。此外，NVMe命令也可以不直接携带value，而是携带value的指针，value指针直接或者间接的指向了value的存储空间，通过DMA/RDMA技术可以从value的存储空间获得所述value。

下面对本发明各实施例可能出现的名词进行示例性介绍。

在现有的NVMe 写命令中，增加如下参数， 或者定义全新的NVMe KV写命令，其中可以包含如下参数。

Number of KV: 一个NVMe命令中传输的KV的数量。 KV包括key和与key对应的value。或者KV包括key和value指针，value指针指向value存储空间，value和key对应。

Value（值）: 数据，可以被存储到存储器中，或者从存储器中读取。例如一个电影。

Key（键）: 唯一标识一个Value，key也可以称为关键码。Key与对应的value的组合可以称为KV或者KV对（pair）。

KV（key value）: key和value的组合，也称为KV对（KV pair）。可选的，KV中还包括metadata。

metadata（元数据）: 用于描述value的属性。例如: 如果Value是一部电影， 元数据可以包括不限于: 电影名称、时长、主演等信息。

Common Header（公共头部）: 是命令头（Header）的一部分，指命令头中和现有NVMe命令头相同的部分。

KV Format ID（KV格式ID）: 指示当前NVMe命令的格式。或者说定义NVMe命令中各个字段的内容。传输KV的NVMe命令可以有多种命令格式，通过KV Format ID来区分。不同的命令格式的NVMe报文可以有不同的字段，或者不同的字段排列顺序。

Key Length: 描述key的长度。

Value Length: 描述value的长度。

Metadata Length: 描述Metadata的长度。

Key Offset: 描述key在NVMe命令中的偏移位置。

Value Offset: 描述value在NVMe命令中的偏移位置。

Metadata Offset: 描述元数据在NVMe命令中的偏移位置。

DPTR（Data Pointer）：数据指针，指向待传输的数据。

MPTR（Metadata Pointer）：元数据指针，指向待传输的元数据。

PRP（Physical Region Page） Entry：物理区域页实例（或者称为物理区域页格式条目），可以记录指针。PRP是NVMe协议中常用的两种数据传输协议，可以用于NVMe over PCIe架构。PRP entry可以指向物理内存页（memory page）的指针。

SGL（Scatter Gather List）Entry：分散搜集表实例（或者称为分散搜集表格式条目），可以记录指针。SGL是NVMe协议中常用的两种数据传输协议之一，可以用于NVMe over PCIe/Fabric架构。

参见图1，本发明提供一种数据处理方法实施例，可以用于NVMe存储器和主机之间。

11，主机生成NVMe写命令（后文简称写命令）发送给存储器，写命令中携带value以及与value对应的key。写命令通过主机的NVMe接口发送给NVMe存储器。key和value可以携带在净荷中，也可以携带在命令头中。

本步骤中，Key可以携带在命令的头（header）中，也可以携带在净荷（payload）中。本发明各实施例中，在没特别声明的情况下，写命令/读命令中携带的字段既可以在命令头中，也可以在净荷中。例如，value可以携带在命令头中，也可以携带在净荷中。

可选的，如果写命令中携带不止一个KV，那么写命令中还可以携带number of KV，用于描述KV的数量。存储器收到写命令后，可以按照number of KV指示的KV数量对KV进行读取，读取完所有KV后停止读取。此外，还可以使用结束符来指示KV的结束，读到结束符后停止读取。

可选的，写命令进一步携带KV Format ID，传输KV的NVMe命令可以有多种命令格式，不同的命令格式中，字段可以不同，字段的位置也可以不同。KV Format ID用于指示本命令所使用的命令格式。

key在写命令的命令头中的位置可以是不固定的，也可以是预先设定的固定位置。如果key在写命令中的位置是不固定的，写命令中可以进一步携带key位置信息。key位置信息可以是key length与key offset的组合，key offset描述key在写命令中的起始位置。key位置信息也可以是key在写命令中的起始位置和key在写命令中结束位置的组合。如果key在写命令中的起始位置预先设定，那么key位置信息可以是key length，key length描述key的长度。

有两种方案可以预先设定关于key的起始位置。一种是预先设定key在命令中的位置，例如从命令的首个bit开始，从第20bits开始是key；另外一种是预先设定key和其他字段的相对位置，例如key字段在key length字段之后，并且key字段与key length字段相邻。类似的，本实施例和其他实施例中涉及到的“预先设定”的位置均可以使用这两种方案，例如本实施例后续将要介绍的value、metadata的起始位置。

key可以是定长或者变长。key是定长，指每个命令中的key长度相同。Key是变长是指不同的命令中，key的长度可以不同。在key为定长的情况下，命令的接收方可以忽略位置信息（例如key length），或者写命令可以不屑道位置信息。存储器收到写命令后，直接按照预设的起始位置，按固定的长度读取key即可。

value在写命令中的位置可以是固定的也可以是不固定的。如果value在写命令中的位置不是固定的，写命令中可以进一步携带value位置信息。value位置信息可以是value length，value在写命令中的起始位置预先设定，因此可以不携带在写命令中。从value的起始位置开始，以value length作为读取长度，就可以可从写命令中读取出value。value位置信息可以是value length与value offset的组合，value offset描述value在写命令中起始位置。value位置信息也可以是value在写命令中的起始位置和value在写命令中结束位置这二者的组合。

value可以是定长或者变长。当value为固定长度的时候，命令的接收方可

以忽略位置信息（例如value length），或者写命令中不携带位置信息。命令的接收方直接按照预设的起始位置，按固定的长度读取value即可。

可选的，写命令中进一步携带value的metadata。metadata可以携带在写命令的命令头中或者净荷中。

metadata在写命令中的位置可以是固定的。如果metadata在写命令中的位置不是固定的，写命令中可以进一步携带metadata位置信息。Metadata位置信息和value位置信息类似，因此可以参见value位置信息的描述，此处不做赘述。同样的，metadata可以是定长或者变长。

参见图2是一种可选的写命令的命令格式，common header是NVMe命令头的一部分。该命令携带2个KV，因此有2个key，分别是key1和key2；有2个value，分别是value1和value2；2个key length，分别是key1 length和key2 length；2个value length，分别是value1 length以及value2 length。此外，该命令还携带2个metadata，分别是metadata1和metadata2；以及对应的metadata length1和metadata length2。

参见图3，是另一NVMe命令格式示意图。和图2相比，增加了value位置信息（value1 offset，value2 offset），以及metadata位置信息（metadata1 offset，metadata1 length，metadata2 offset，metadata2 length），还增加了number of KV，在图3示例的命令中携带了2个KV，因此number of KV的值是2。NVMe命令中的位置信息描述的都是字段在NVMe中的位置。图3中以箭头来表示：通过偏移量可以找到待写数据的位置（例如起始位置）。例如，在读出value1 offset字段后，value1 offset字段的值描述了value1在命令中的偏移位置，从而可以读出value1 offset。图3的命令格式中，字段的分布按字段的类型确定，同类型的字段相邻。例如value和metadata都属于数据部分，因此相邻；value位置信息和metadata位置信息都属于位置信息，因此相邻。

参见图4，是另一NVMe命令格式示意图。和图3相比，字段按照KV归属

分布，同一个KV的字段相邻。图4的命令格式中，先是KV1的字段，然后是KV2的字段。此外，在图4的命令格式中，value 和value length; metadata和metadata length。

参见图5，是另一NVMe命令格式示意图。图5中，同一个KV的字段相邻。此外，value length和metadata length相邻，value和metadata相邻。

12，存储器的NVMe接口接收写命令，存储器从写命令中获得key和value。如果写命令中携带有metadata，则还从写命令中获得metadata。

存储器由控制器和存储介质组成，控制器中包括处理器，可选的，还可以包括内存。存储介质例如是闪存或者磁盘。存储器也可以是具有管理能力的硬盘，称为智能硬盘。

如果多种类型的NVMe命令格式（例如图2和图3分别是不同类型的NVMe命令格式）被同一个存储器支持，则存储器通过KV Format ID可以确定收到的命令是哪一种格式。然后按照这个格式中key、value等字段的位置读出key和value等内容。不同的NVMe命令格式，数据的位置关系可以不同。例如，有的格式中，key的位置是固定的；有的格式中，key的位置不是固定的，由key位置信息确定。

如果写命令中携带一个KV，则读出这一个KV。如果写命令中携带至少两个KV，则读出这至少两个KV。存储器可以通过number of KV字段获知KV的数量，在读出相应数量的KV后完成读操作。除了使用number of KV外来标记什么情况下完成读操作外，也可以在命令中增加结束符来，在读到结束符后表示整个命令中的KV已经读取完毕。

如果key的位置是固定的，则存储器通过固定位置读出key。如果key的位置不固定，则按照key位置信息从写命令中读出key。例如，key的位置信息是key length，key在写命令中的起始位置预先设定，从key的起始位置开始，以key length作为读取长度连续的读出数据，即可获得写命令中携带的key。例

如，key的位置信息是key length与key offset的组合，则以key offset作为key的起始位置，以key length以key length作为读取长度连续的读出数据，即可获得写命令中携带的key。例如，key的位置信息是起始位置和结束位置共同组成，则读取起始位置和结束位置之间的数据，即可获得key。

关于读取value和metadata的读取方式，均可以参考key的读取方式，此处不再赘述。在读出key/value/metadata时，存储器可以获得key length/value length/metadata length。

以图2的命令格式为例，command header、KV format ID、key length以及value length这几个字段的长度是固定的，各个字段的相对位置也是固定的，因此可以不需要key offset、value offset等字段来确定key、value的起始位置。存储器读取写命令各个字段的顺序是：读取command header，读取KV format ID，读取key 1 length，按照key 1 length所记录的数值读取key1，读取value1 length，按照value1 length读取value1，读取key2 length，按照key2 length读取key2，读取value2 length，按照value2 length读取value2。。

13，存储器对value进行存储。具体而言，是存储器的控制器把key和value存储到非易失性存储介质中。存储器的接口收到value后，先发送到处理器。本步骤中，处理器把value暂存在内存中，然后从内存下发到非易失性存储介质。

此外，存储器还可以记录key和value存储空间的映射关系。映射关系的可以由存储器的存储介质进行存储；也可以是发送给主机，由主机进行存储。后文以由存储器进行存储进行举例。

如果写命令中还携带metadata，则也对metadata进行存储，并记录key和metadata存储空间的映射关系。

value存储空间可以用value 存储空间的起始地址和value长度描述。metadata存储空间可以包括metadata存储空间的起始地址和metadata data长度

描述。参见图6是映射关系的示意图，包括key和value存储空间的起始地址的映射，key和value长度的映射，key和metadata存储空间的起始地址的映射，key和metadata长度的映射。可以把key作为索引在映射关系中查找对应的value存储空间的起始地址、value长度、metadata存储空间的起始地址以及metadata长度。图6的示意图所描述的内容记录在KV管理单元中。如果value和metadata定长，value长度和metadata长度是可选的。

存储空间可以是逻辑位置或者物理位置。只要存储器的控制器使用存储空间可以从存储介质中读取value和metadata即可。

上面步骤11-步骤13是写KV的过程，接下来的步骤14-步骤16是读KV的过程。这两个过程是相互独立的，读命令所请求的KV和写命令所写入的KV可以不是同一个。

14，主机生成NVMe读命令（后文简称读命令），发送读命令给存储器，读命令中携带所述key。key可以携带在读命令的命令头中。

生成读命令的主机和生成写命令的主机可以是同一个主机，也可以是不同主机。读命令的命令格式参考步骤11中写命令的格式以及图2。读命令包括command header，KV format ID和key。可选的，读命令中携带key位置信息。一个读命令中可以携带一个key也可以携带至少两个key。当携带至少两个key时，可以携带number of KV字段，以描述携带的key的数量。

读命令中可以没有value字段以及其他和value相关的字段。此外，读命令中可以没有metadata字段以及其他和metadata相关的字段。

15，存储器通过NVMe接口，接收读命令，从读命令中获得key。从存储器存储的KV管理单元中，查找key对应的value存储空间。从value存储空间中获得value并构造读命令的响应消息发送给主机，响应消息中携带value。

参考步骤12，按照命令格式的不同，获得key的方式也略有不同。例如，可以从读命令的固定位置获得key，也可以按照读命令中携带的位置信息从

写命令中获得key。

类似的，如果存储器中存储有与key对应的metadata，可以按照类似的方法查询metadata的存储空间并获得metadata。

需说明的是，如果KV管理单元存储在主机中，则使用key在主机中查询value存储空间，把value存储空间方式发送给存储器，存储器依靠value存储空间获得value发送给主机。

16，主机接收value并存储。例如可以存储在主机的内存（例如缓存）中，或者存储在主机的非易失性存储介质中。

上述步骤11-16中， value和metadata都携带在NVMe命令中，不用进行KV和block之间的转换，具有简单快捷的优点。读命令不用携带value和/或metadata的LBA，因此读value和/或metadata的过程也更为快捷。

参见图7，是本发明的数据处理系统硬件拓扑实施例，主机71和存储器72通信。主机71包括处理器711、内存712以及接口713。存储器72包括接口721、控制器722以及存储介质723。接口713和接口721通过通信链路73连接，通信链路73例如是PCIe总线、光纤通道FC或者以太网。

主机71所执行的操作可以由处理器711执行，例如处理器711通过运行内存712中的程序，可以执行步骤11、14和16，内存712和处理器711相对独立，也可以集成在一起。存储器72包括控制区721和存储介质722。存储器72所执行的操作由存储器的控制器721执行。具体而言，可以由处理器7211运行内存7212中的程序来执行存储器的操作，例如存储器的处理器用于执行步骤12、13和15。在一些情况下，例如当处理器是FPGA的时候，可以没有内存，直接由处理器执行相应的操作。

类似的，后续的方法中使用了相同的硬件，不同在于，主机和存储器用于执行的操作不同。例如，步骤21、24和27由主机71执行， 23和25由存储器72执行。步骤22和26中，一部分操作由主机71执行，另外一部分操作由存

储器72执行。

参见图8，本发明还提供一种实施方式，写命令中不直接携带value或者metadata，而是在NVMe命令中携带指针，通过指针指向的存储空间可以获得待写value和/或metadata，或者采用多级指针的方式，一个指针指向另外一个指针，从另外一个指针所指向的存储空间获得待写value和/或metadata。这种实施方式对于待写的value和/或待写的metadata的大小没有限制。

主机和存储器可以通过IP、FC等网络连接，可以运行在Fabric架构，也称为NOF（NVMe Over Fabric）架构。在NOF架构下，存储器可以通过远程直接数据存取（Remote Direct Memory Access，RDMA）的方式获得value 和/或metadata。如果存储器在主机内部，存储器还可以通过直接数据存取（Direct Memory Access，DMA）的方式从主机获得value和/或metadata。类似的，主机也可以通过RDMA/DMA的方式从存储器直接获得value和/或metadata。

步骤21，主机构造写命令，写命令中携带KV，KV包括value指针以及value的key。写命令通过主机的NVMe接口发送给NVMe存储器。所述key和所述value属于同一个KV对。

value指针可以携带在净荷中，也可以携带在命令头中。类似的，写命令或者读命令中携带的其他数据既可以携带在命令头中，也可以携带在净荷中。

在其他实施例中，写命令也可以携带key指针而不携带key。根据key指针获得key的方案，和通过value指针获得value的原理相同，因此下面不做赘述。仅以写命令携带key为例进行介绍。

value指针直接或者间接的指向了value的存储空间（为了方便描述，后文也称作第一存储空间），因此通过value指针可以获得value。value指针指向value存储空间（这种情况也被视为value指针指向了value）的情况下，可以从value指针指向的存储空间获得value。另外一种情况是，value指针指向第一指针，而第一指针指向value的存储空间，从value指针找到第一指针，从第

一指针指向的存储空间可以获得value。后一种情况中，可以携带尺寸更大的value。在其他实施例中，指针之间的引用还可以有更多的层级，只要最终能找到value的存储空间即可，例如value指针指向第一指针，第一指针指向第A1指针，第A1指针指向第A2指针，……，第AN-1指针指向第AN指针，第AN指针指向value的存储空间，N是大于等于2的整数。value指针可以携带在写命令的命令头或者净荷中。

所述第一存储空间可以用第一存储空间的首地址和所述value长度描述，也可以用第一存储空间的首地址和末地址描述。例如：value指针记录第一存储空间的首地址和value长度；或者，value指针指向了另一个指针，所述另一个指针记录第一存储空间的首地址和value长度。第一存储空间可以是逻辑位置或者物理位置。只要存储器使用第一存储空间可以从存储介质中读取存储的数据即可。

可选的，如果写命令中携带不止一个KV，那么写命令中还可以携带Number of KV。用于描述KV的数量。

可选的，写命令进一步携带KV Format ID，传输KV的NVMe命令可以有多种命令格式，不同的命令格式中，字段可以不同，字段的位置也可以不同。KV Format ID指示本命令所使用的命令格式。

key在写命令的命令头中的位置可以是固定的，也可以是预先设定的固定位置。key的特征可以参见步骤11对key的描述，此处不做赘述。

可选的，写命令中进一步携带metadata指针。和value指针类似，metadata指针直接或者间接指向metadata存储空间，因此通过metadata指针可以获得metadata。metadata指针可以携带在写命令的命令头或者净荷中。

metadata指针直接或者间接的指向了metadata的存储空间，通过metadata指针可以获得metadata。metadata指针直接指向metadata（具体而言是metadata的存储空间）的情况下，可以从metadata指针指向的存储空间获得metadata；

另外一种情况是，metadata指针指向第二指针，而第二指针指向metadata的存储空间，从metadata指针找到第二指针，从第二指针指向的存储空间可以获得metadata。后一种情况中，可以携带更长的metadata。在其他实施例中，指针之间的引用还可以有更多的层级，只要最终能找到metadata的存储空间即可，例如metadata指针指向第二指针，第二指针指向第B1指针，第B1指针指向第B2指针，……，第BN-1指针指向第BN指针，第BN指针指向metadata的存储空间，N是大于等于2的整数。

此外，metadata可以不通过指针的方式传输，而是直接携带在命令中，具体可以参考步骤11以及图2、图3中metadata的携带方式。

value指针可以是数据指针（Data Pointer，DPTR）。metadata指针可以是元数据指针（Metadata Pointer，PMTR）。

参见图9，是一个命令格式图。图9的命令中携带2个KV，KV中key直接携带在写命令中，value是以指针的方式携带在写命令中。分别是key1和DPTR1（指向value1），以及key2和DPTR2（指向value2）。此外，还通过指针的方式携带了这2个KV的metadata，其中MPTR1指向metadata1，MPTR2指向metadata2。本发明实施例中，每个命令可以携带1个KV，也可以携带2组以上的KV。

参见图10，是另外一个命令格式图。命令中的指针（DPTR，MPTR）没有直接指向value或者metadata，而是指向PRP entry或者SGL entry，PRP entry或者SGL entry是链表中的节点，每个PRP/SGL entry指向了另外一个地址，PRP/SGL entry指向的地址中存储了value或者metadata。也就说，写命令通过二级指针的方式指向了value或者metadata。在图10的方案中，key携带在命令中，具体而言是写的在命令头中。

参见图11的命令格式，和图10不同之处在于，图10的key没有直接携带在命令中，而是通过指针的方式传输，具体如何携带可以参考value指针的介

绍。key指针可以携带在命令的命令头中。

具体而言，DPTR指向PRP/SGL entry1，而PRP/SGL entry2和PRP/SGL entry1 属于同一个链表，因此找到PRP/SGL entry1即可以找到PRP/SGL entry2。因此，也可以认为DTPR指向了PRP/SGL entry1以及PRP/SGL entry2。PRP/SGL entry1指向了value1，PRP/SGL entry2指向了value2。类似的，MPTR指向了PRP/SGL entry3以及PRP/SGL entry4，PRP/SGL entry3指向了metadata1，PRP/SGL entry4指向了metadata2。依靠DPTR可以找到value1的存储空间和value2的存储空间；依靠MPTR，可以找到metadta1的存储空间和metadata2的存储空间。PRP entry和SGL entry均是链表中的节点，PRP适用于普通NVMe（用PCIe连接主机和存储器），SGL适用于NOF（用Fabic连接主机和存储器）。RDMA可以使用PRP，DMA可以用PRP或者SGL。

PRP通过一系列的指向memory page指针，将数据所在的位置传输给NVMe设备，NVMe设备收到这些地址后，通过DMA的方式即可将数据从主机读取到NVMe设备上。SGL传输机制相对于PRP更加灵活一些，可以指定传输的长度，能够在地址连续的传输过程中跳过一些地址空间进行传输。SGL将需要传输数据的地址传输给NVMe设备，NVMe设备收到这些地址之后，通过DMA的方式将数据从主机读取到NVMe设备上。

此外，相较于图9，图10中还增加了number of KV字段以及key length字段，在步骤11、步骤12中已有介绍，不再赘述。

步骤22，存储器通过NVMe接口接收写命令。存储器通过value指针获得value的长度，按照value长度为value分配存储空间（为了方便描述，后文称作第二存储空间），把分配的存储空间的位置信息发送给主机。存储器发送传输请求（第一传输请求）给主机，从主机的第一存储空间获得value。主机和存储器是PCIe连接，则第一传输请求可以是DMA传输请求，例如单个DMA传输请求。主机和存储器是Fabric连接，则第一传输请求可以是RDMA传输

请求，例如单个RDMA传输请求。第一传输请求把第一存储空间的地址作为访问地址，第二存储空间的地址作为写入地址。第一传输请求中，访问地址可以是：第一存储空间的首地址+value长度，写入地址可以是：第二存储空间的首地址（或者是第二存储空间的首地址+value长度）。

存储器发送DMA/RDMA传输请求给主机，主机收到传输请求后执行DMA/RDMA传输，把value发送给存储器。

预先分配第二存储空间是可选的，存储器也可以不提前分配第二存储空间，直接从主机获得value，然后再为value分配第二存储空间。这种情况下，第一传输请求不携带写入地址。

如果写命令中携带有metadata指针，则还为metadata分配存储空间（第四存储空间）。由于metadata指针的处理方式和value指针相似，因此后续仅对value指针进行介绍。

在按照value指针获得value之前，按照value长度在存储器中中分配用于存储value的存储空间（第二存储空间），第二存储空间不小于value长度。第二存储空间的可以用首地址+末地址描述，也可以用首地址+value长度描述，也可以仅用首地址描述。value长度可以由value存储空间确定。例如写命令中，第一存储空间由存储value的空间的起始地址+value长度描述，则可以直接从写命令中获得value长度。

存储器由控制器和存储介质组成，控制器中包括处理器，存储介质例如是闪存或者磁盘。

command header、KV format ID、key以及key length这几个字段在步骤12中已有介绍，如何获得它们，以及如何利用他们获得相应信息此处不再赘述。

和步骤11- 16所介绍的实施例不同的是，本实施例中不是直接从命令中获得value和/或metadata，而是依靠指针来获得value和/或metadata。

写命令中携带的value指针直接指向了第一存储空间，例如图9的命令格

式，写命令中的指针DPTR指向了value存储空间（第一存储空间）。具体而言，指针DPTR指向了主机的一段内存地址范围，这一段内存地址范围存储了所述value。则按照DPTR描述的存储空间，以DMA或者RDMA的方式从主机内存地址获得value和。按照同样的方式，可以通过MPTR获得metadata。

如果写命令中的value指针间接指向value，可以参考图10的写命令格式：value指针直接指向了另外一个指针，另外一个指针指向value。写命令中的指针DPTR指向了PRP/SGL entry，PRP/SGL entry指向了主机内存地址，主机内存地址是value的存储空间。存储器先通过DPTR找到PRP/SGL entry，然后通过PRP/SGL entry找到位于主机内存的value存储空间，以DMA或者RDMA的方式从主机内存中获得所述value。

如果metadata指针携带在写命令中，则获得metadata的方式和获得value的方式相同。如果metadata直接携带在写命令中，则可以参照步骤12从写命令中直接获得metadata。

存储器发送第一传输请求给主机，从主机的第一存储空间获得value。如果主机和存储器是PCIe连接，第一传输请求可以是DMA传输请求；如果主机和存储器是Fabric连接，第一传输请求可以是RDMA传输请求。具体而言，可以分为两个步骤：（1）主机发送传输请求给主机，传输请求中携带第一存储空间作为访问地址，携带所述第二存储空间作为写入地址。具体而言，传输请求中携带的第一存储空间，可以由第一存储空间的首地址和所述value长度描述；第二存储空间用第二存储空间的首地址描述。（2）主机收到传输请求后，从第一存储空间读出数据，发送给存储器的第二存储空间进行存储。

本步骤中，可以传输key、value以及metadata通过单个DMA/RDMA传输。也可以把key、value和metadata分别进行传输。

步骤23，存储器对value进行存储。具体而言，可以是存储器的控制器把key和value存储到存储器的存储介质中。如果步骤22中还获得有metadata，也

对metadata进行存储。

例如从第二存储空间的首地址开始，把value连续的存入存储器的存储介质中。

此外，还可以记录key和value存储空间的映射关系，参见图6，图6中value存储空间由首地址+value长度描述。映射关系记录在KV管理单元中。value长度和metadata长度是可选的，因为这value和metadata都可以是固定长度。

上述步骤21-23中，value以指针的方式进行传输。在其他实施例中，key也可以用指针（直接指针或者间接指针）的方式传输，具体可以参照value的传输方式，此处不赘述。

上述步骤21-23是写命令的过程，步骤24-26是读命令的执行过程，这两个过程是相互独立的。读命令所请求的KV和写命令所写入的KV可以不是同一个。读命令和写命令的命令格式相似，区别在于读命令不携带value指针和metadata指针。

24，主机生成读命令发送给存储器，读命令中携带所述key。key可以携带在读命令的命令头中。

生成读命令的主机和生成写命令的主机可以是同一个主机，也可以是不同主机。读命令的格式参考步骤11中写命令的格式以及图2，读命令包括command header，KV format ID和key。可选的，读命令携带key或者key指针。一个读命令中可以携带一个key也可以携带至少两个key。当携带至少两个key时，可以携带number of KV字段，以描述携带的key的数量。

读命令请求读出value和/或metadata。

和写命令相比，读命令中可以没有value字段以及value位置信息，读命令中可以没有metadata字段以及metadata位置信息。

25，存储器通过NVMe接口接收读命令，从读命令中获得key。使用获得

的key，从存储器存储的映射关系中，查找value的存储空间并发送给主机。value在存储器中的存储空间是第二存储空间。

如果key直接携带在读命令中，参考步骤12，按照命令格式的不同，从读命令中获得key的方式也略有不同。例如，可以从读命令的固定位置获得key；如果key在读命令中的位置不固定，可以按照读命令中携带的位置信息从读命令中获得key。

可选的，如果存储器中存储有与key对应的metadata，还可以获得metadata的存储空间并发送给主机。

26，主机按照value长度给value分配存储空间（命名为第三存储空间）。主机构造传输请求（第二传输请求），发送传输请求给存储器，从存储器第二存储空间中获得value。

主机和存储器是PCIe连接，则第二传输请求可以是DMA传输请求，例如单个DMA传输请求；主机和存储器是Fabric连接，则第二传输请求可以是RDMA传输请求，例如单个RDMA传输请求。传输请求中，访问地址可以是：第二存储空间的首地址+value长度，写入地址可以是第三存储空间的首地址（或者是第三存储空间的首地址+value长度）。由前述步骤可知第二存储空间用于存储所述value，因此所述value长度和第二存储空间大小相同。也就是说第二存储空间的大小等于第三存储空间的大小。

value长度可以由value存储空间确定。例如在读命令中中，第二存储空间由存储value的空间的起始地址+value长度描述，则可以直接从写命令中获得value长度。

主机发送DMA/RDMA传输请求给存储器，存储器收到传输请求后执行DMA/RDMA传输，把value发送给主机。

预先分配第三存储空间是可选的，主机也可以不提前分配第三存储空间，直接从存储器获得value，然后再为value分配第三存储空间。这种情况下，传

输请求不携带写入地址。

类似的，主机可以按照metadata长度给metadata分配存储空间。并从存储器获取metadata。

27，主机把收到的value存入第三存储空间。例如从第三存储空间的首地址开始，把value连续的存入主机。第三存储空间例如在主机的内存中。

采用把value存入主机的的相类似的方法，把所述metadata存入主机。

需要说明的是，在其他实施例中，可以不使用DMA或者RDMA，把步骤26、27修改为：主机给所述value分配存储空间，存储器从所述第二存储空间中读出所述value，把读出的value写入所述第三存储空间。

此外，步骤26的另外一种可选方案是：所述NVMe存储器接收的所述主机发送的NVMe读命令中还可以进一步携带所述主机的空闲空间信息，空闲空间信息用于描述主机中连续的空闲存储空间的大小。所述NVMe存储器从所述主机接收所述NVMe读命令之后，进一步包括：所述NVMe存储器判断所述主机的空闲存储空间是否大于等于所述第二存储空间，如果是，执行将所述第二存储空间的位置信息发送给所述主机的步骤，如果否，结束步骤。这种方案中， 可以把主机中连续的空闲存储空间称为所述第三存储空间。

参见图12，以同一个命令中携带value指针和metadata指针为例，对步骤21-步骤23进行介绍。

步骤31，主机构造写命令。写命令中携带key。此外，写命令中携带value源地址链表和metadata源地址链表。源地址链表拥有指针的功能，value和metadata的源地址链表分别指向value的存储空间和metadata的存储空间。

步骤32，主机发送写命令给NVMe设备。

步骤33，NVMe设备收到写命令后，从写命令中解析出value源地址链表和metadata的源地址链表。链表例如是PRP entry或者SGL entry。

步骤34，NVMe设备从value源地址链表中解析出value长度，从metadata的源地址链表中解析出metadata长度。源地址链表描述了待读取数据的存储空间，因此可以从中解析出value长度和metadata长度。

步骤35，NVMe设备向主机发送DMA或者RDMA请求。请求中携带value源地址链表、metadata源地址链表、value目的地址链表以及metadata目的地址链表。 value目的地址链表描述NVMe设备预留的、用于存储value的存储空间，metadata目的地址链表描述NVMe设备预留的、用于存储metadata的存储空间。

步骤36，主机接收到DMA/RDMA请求后，按照步骤35的源地址和目的地址，向NVMe设备发送key。

步骤37，NVMe设备把value和metadata存储在介质中。

步骤38，NVMe设备记录value的存储空间和metadata的存储空间。存储空间和key建立映射，以便后续使用key查找value存储空间和metadata存储空间。映射关系可以存储在KV管理单元中，KV管理单元还可以记录value length和metadata length。

上述步骤中，key直接携带在命令中。在其他实施方式中，命令也可以不携带key而是携带key指针。存储器收到命令后，按照key指针获得key。获得key的具体步骤可以参考步骤22获得value的过程，由于原理类似，此处不再赘述。

参见图13，以同一个命令中携带value指针和metadata指针为例，对步骤24-27的具体描述。

步骤41，主机构造读命令，读命令中携带有key。

步骤42，主机把携带有key的读命令中给NVMe设备。

步骤43，NVMe设备从KV管理单元查找所述key对应的value length和

metadata length。其中，KV管理单元记录有value存储空间和metadata存储空间。

以value存储空间为例，value存储在NVMe设备中，value存储空间由value首地址+末地址描述，或者由首地址+value length。因此，获得value存储空间后，可以获得value length。类似的，可以获得metadata length。

步骤44，NVMe设备发送响应消息给主机，响应消息中携带value length和metadata length。

步骤45，主机按照value length和metadata length，给value和metadata分配存储空间，作为将来存储value和metadta之用。

步骤46，主机向NVMe设备发送DMA/RDMA请求，请求中携带本次传输的源地址和目的地址。源地址是NVMe中的value存储空间和metadata存储空间，目的地址是主机为value预留的存储空间，以及为metadata预留的存储空间。

步骤47，收到主机的DMA/RDMA请求后，NVMe设备按照DMA/RDMA协议发送value以及metadata。

步骤48，主机接收value和metadata。

步骤49，主机存储收到的value和metadata，例如存储在主机内存中。

DMA或者RDMA通常用于读取一段连续的地址空间的数据，当存储value的存储空间和存储metadata的存储空间不连续的时候，可以分作两次DMA/RDM传输进行处理。

参见图14，本发明另外提供一个基于步骤21-23的实施例，对基于图12的实施例进行改进。改进之处在于：同一个命令的key、value和metadata分成三次DMA/RDMA传输获得。

可选的，如果其中任意一个传输失败，意味着这个KV传输不成功，因

此已传输成功的数据可以删除，上传错误码。取消后续数据传输。并且可以释放之前为KV（key、value和metadata）分配的存储空间。

步骤51，构造NVMe写命令。

写命令中携带key。此外，写命令中携带value源地址链表和metadata源地址链表。源地址链表拥有指针的功能，value和metadata的源地址链表分别指向value的存储空间和metadata的存储空间。

步骤52，主机发送写命令给NVMe设备。

步骤53，NVMe设备收到写命令后，从写命令中解析出value源地址链表和metadata的源地址链表。链表例如是PRP entry或者SGL entry。

步骤54，NVMe设备从key源地址链表中解析出key长度，从value源地址链表中解析出value长度，从metadata的源地址链表中解析出metadata长度。为key、value、metadata预留存储空间。源地址链表描述了待读取数据的存储空间，因此可以从中解析出value长度和metadata长度。

步骤55，NVMe设备向主机发送DMA/RDMA请求。请求中携带key源地址链表和key目的地址链表。key目的地址链表描述NVMe设备预留的、用于存储key的存储空间。

步骤56，主机接收到DMA/RDMA请求后，按照步骤55的源地址和目的地址，向NVMe设备发送key。

步骤57，如果key传输失败，则NVMe设备取消后续value、metadata的传输。可选的，释放为key、value和metadata预留的存储空间。上报错误码给主机。如果key传输成功，则继续后续步骤58。

步骤58，NVMe设备向主机发送DMA/RDMA请求。请求中携带value源地址链表和value目的地址链表。value目的地址链表描述NVMe设备预留的、用于存储value的存储空间。

步骤59，主机接收到DMA/RDMA请求后，按照步骤58的源地址和目的地址，向NVMe设备发送value。

步骤60，如果value传输失败，则NVMe设备取消后续metadata的传输。可选的，释放为value和metadata预留的存储空间。可选的，删除已经传输的key。上报错误码给主机。如果value传输成功，则继续后续步骤61。

步骤61，NVMe设备向主机发送DMA/RDMA请求。请求中携带metadata源地址链表和key目的地址链表。metadata目的地址链表描述NVMe设备预留的、用于存储key的存储空间。

步骤62，主机接收到DMA/RDMA请求后，按照步骤61的源地址和目的地址，向NVMe设备发送metadata。

步骤63，如果metadata传输失败，可选的，释放为metadata预留的存储空间。可选的，删除已经传输的key和value。上报错误码给主机。如果value传输成功，则继续后续步骤64。

步骤64，存储value和metadata到存储器。步骤60和63所获得的value和metadata在内存中，本步骤把value和metadata存储到非易失性介质中，例如NVMe设备的硬盘。

步骤65，记录value的存储空间和metadata的存储空间。

从图14的步骤55-步骤63可以看出，key、value和metadata分别通过不同的DMA传输获得，如果其中任意一项数据传输失败，就取消余下数据的传输，已传输成功的数据可以删除。图14的其余步骤和图12相似，前文已有介绍，此处不再赘述。

类似的，除了value和metadata可以分开传输。value（或者metadata）也可以分开多次传输，例如把一个value分成两次以上DMA/RDMA传输。任意一次传输失败，则判定整个KV传输失败，释放为key、value和metadata预留的存储空间。当所述NVMe存储器和所述主机用PCIe总线连接，所述存储器

发送至少两个DMA传输请求给所述主机以获得所述value，每个DMA传输获得所述value的一部分。当所述NVMe存储器和所述主机用Frabic连接，所述存储器发送至少两个RDMA传输请求给所述主机以获得所述value，每次RDMA传输获得所述value的一部分。

value（或metadata）分成多次传输的方案，可以和步骤55 -步骤63中所描述的value和metadata分开传输的方案合并使用。这种情况下，任意一次传输失败（或者说，传输请求执行失败），则判定整个KV传输失败，释放为key、value和metadata预留的存储空间。

图14的流程描述的是对写命令的处理方式，DMA/RDMA传输请求的发起者是NVMe设备。类似的，在读命令的处理流程中，也可以采用类似的方案，key、value和metadata分别通过不同的DMA传输获取。不同之处在于，传输请求的发起者是主机，而通过DMA/RDMA传输key、value和metadata的是NVMe设备。

参见图15，本发明提供一种数据处理系统，包括主机装置80和存储装置90，主机装置80和存储装置90通过PCIe或者Fabric连接。主机装置80可以把KV数据读出或者写入存储装置90。

该主机装置80可以是物理上设备或者逻辑装置，包括收发模块801，管理模块802和缓存模块803。存储装置90可以是物理上设备或者逻辑装置，包括接口模块901、处理模块902和存储模块903。收发模块801和接口模块901通信。主机装置80拥有前述主机的功能，存储装置90拥有前述存储器的功能。

下面对主机装置80和存储装置90的功能进行简单介绍。需要说明的是，由于这两个装置（以及相应模块）的功能在方法流程中已有详细说明，因此这里仅做简单介绍。

该存储装置90包括：接口模块901，用于接收主机发送的NVMe写命令，所述NVMe写命令中携带key，所述NMVe写命令携带value指针，所述value

指针指向所述主机中的第一存储空间，所述第一存储空间用于存储value，所述key与所述value属于同一个KV对；处理模块902，用于从所述NVMe写命令中获得所述key，根据所述value指针获得value长度，按照所述value长度为所述value分配第二存储空间；存储模块903，用于发送第一传输请求给所述主机，从所述主机获得所述value，把所述value保存在所述第二存储空间中。所述第二存储空间可以由在所述NVMe存储器的存储介质提供，所述存储介质和所述NVMe存储器的处理器连接。基于该方案，KV数据从主机传递到NVMe存储器的过程中，不需要把KV数据转换成块的形式，提高了KV数据的存储效率。

或者该存储装置90包括：接口模块901，用于接收NVMe写命令，所述NVMe写命令的头部携带键key，所述NMVe命令还携带值value，所述key和所述value对应，所述key和所述value属于同一个KV对；处理模块902，用于从所述NVMe写命令中获得所述key以及所述value；存储模块903，用于把所述value保存在所述NVMe存储器的存储介质中。

在一种实施方式中，该主机装置80的包括：收发模块801，用于发送NVMe读命令给存储装置90，所述NMVe命令携带key；所述收发模块801，还用于从所述NVMe存储器接收所述NVMe读命令的响应消息，所述响应消息携带第二存储空间的位置信息；管理模块802，用于按照所述value位置信息中的value长度给value预留第三存储空间；所述收发模块801，还用于发送传输请求给所述NVMe存储器，所述传输请求的访问地址是所述第二存储空间，以及从所述NVMe存储器获得所述value；缓存模块803，把所述value保存在所述主机装置80的第三存储空间中。基于此模块结构，主机装置可以可以读取存储装置读取value和或metadata。

结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术

人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。

以上所述，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，基于本发明的变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应所述以权利要求的保护范围为准。

# 权 利 要 求

1、一种数据处理方法，其特征在于，该方法包括:

快速非易失性存储 NVMe 存储器接收主机发送的 NVMe 写命令，所述 NVMe 写命令中携带 key，所述 NMVe 写命令携带 value 指针，所述 value 指针指向所述主机中的第一存储空间，所述第一存储空间用于存储 value，所述 key 与所述 value 属于同一个 KV 对;

所述 NVMe 存储器从所述 NVMe 写命令中获得所述 key，根据所述 value 指针获得 value 长度，按照所述 value 长度为所述 value 分配第二存储空间，所述第二存储空间在所述 NVMe 存储器中;

所述 NVMe 存储器发送第一传输请求给所述主机，从所述主机获得所述 value，把所述 value 保存在所述第二存储空间中。

2、如权利要求 1 所述的数据处理方法，其中，所述 NVMe 存储器发送第一传输请求给主机以及从所述主机获得所述 value，具体包括:

所述 NVMe 存储器发送 DMA 传输请求给所述主机，从所述主机获得所述 value，所述 DMA 指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址，其中，所述 NVMe 存储器和所述主机用 PCIe 总线连接。

3、如权利要求 1 所述的数据处理方法，其中，所述 NVMe 存储器发送第一传输请求给主机以及从所述主机获得所述 value，具体包括:

所述存储器发送 RDMA 传输请求给所述主机，从所述主机获得所述 value，所述 RDMA 指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址，其中，所述 NVMe 存储器和所述主机用 Fabric 总线连接。

4、如权利要求 1 所述的数据处理方法，其中:

所述 NVMe 写命令中进一步携带 KV 数量的字段，所述 KV 数量的字段用于描述所述 NVMe 写命令中 KV 的数量，所述 NVMe 存储器从所述 NVMe

写命令中获得与 KV 数量相同数量的 key，以及获得与 KV 数量相同数量的 value。

5、如权利要求 1 所述的数据处理方法，其中：

所述 NVMe 写命令中进一步携带 KV 格式的字段，其中，所述 KV 格式的字段描述所述 NVMe 写命令中字段的结构，所述 NVMe 存储器按照所述 KV 格式字段所定义的字段内容，从所述 NVMe 写命令中获取各个字段。

6、如权利要求 1 所述的数据处理方法，其中，所述 NVMe 写命令还携带所述 value 元数据指针，所述方法还包括:

所述 NVMe 存储器根据所述元数据指针获得元数据长度，按照所述元数据长度为所述元数据分配第四存储空间，所述第四存储空间在所述 NVMe 存储器中；

所述 NVMe 存储器通过所述第一传输请求从所述主机中获得所述元数据，把所述元数据保存在第四存储空间中。

7、如权利要求 6 所述的数据处理方法，其中：

所述元数据传输失败后，所述 NVMe 存储器释放为所述 value 分配的存储空间，以及释放为所述元数据分配的第四存储空间。

8、如权利要求 1 所述的数据处理方法，其中，所述 NVMe 存储器发送第一传输请求给所述主机以及从所述主机获得所述 value，具体包括下述其中一种:

所述存储器发送至少两个 DMA 传输请求给所述主机以获得所述 value，每个 DMA 传输请求用于请求获得所述 value 的一部分，当任意一个 DMA 传输请求执行失败，则 NVMe 存储器释放为所述 value 分配的存储空间，其中，所述 NVMe 存储器和所述主机用 PCIe 总线连接；或者

所述存储器发送至少两个 RDMA 传输请求给所述主机以获得所述 value，每个 RDMA 传输请求用于请求获得所述 value 的一部分，当任意一个 RDMA 传输请求执行失败，则 NVMe 存储器释放为所述 value 分配的存储空间，其

中，所述 NVMe 存储器和所述主机用 Frabic 连接。

9、如权利要求 1 所述的数据处理方法，其中，所述方法还包括:

生成所述 key 与所述第二存储空间的映射关系。

10、如权利要求 9 所述的数据处理方法，该方法之后，进一步包括:

所述 NVMe 存储器从所述主机接收 NVMe 读命令，所述 NVMe 读命令中携带所述 key;

所述 NVMe 存储器根据所述 key 在所述映射关系中查询，获得存储所述 value 的所述第二存储空间的位置信息，把所述第二存储空间的位置信息发送给所述主机;

所述 NVMe 存储器接收所述主机发送的第二传输请求，所述第二传输请求用于请求获得存储在所述第二存储空间的数据。

11、如权利要求 10 所述的数据处理方法，其中:

所述 NVMe 存储器接收所述主机发送的第二传输请求之前，进一步包括:所述主机根据所述第二存储空间的大小预留所述主机中的第三存储空间;

所述 NVMe 存储器发送所述 value 给所述主机之后，进一步包括: 所述主机把收到的来自所述第二存储空间的数据写入所述第四存储空间空间。

12、如权利要求 10 所述的数据处理方法，所述 NVMe 存储器接收的所述主机发送的 NVMe 读命令中还携带所述主机的空闲空间信息，所述 NVMe 存储器从所述主机接收所述 NVMe 读命令之后，进一步包括:

所述 NVMe 存储器判断所述主机的空闲存储空间是否大于等于所述第二存储空间，如果是，执行将所述第二存储空间的位置信息发送给所述主机的步骤，如果否，结束步骤。

13、如权利要求 1 所述的数据处理方法，其中:

所述第一存储空间用第一存储空间的首地址和所述 value 长度描述;

所述第二存储空间用第二存储空间的首地址描述。

14、一种 NVMe 存储器，包括控制器以及存储介质，所述控制器和所述

存储介质连接，所述存储介质用于提供存储空间，其特征在于，所述处理器被配置为执行:

接收主机发送的快速非易失性存储 NVMe 写命令，所述 NVMe 写命令中携带 key，所述 NMVe 写命令携带 value 指针，所述 value 指针指向所述主机中的第一存储空间，所述第一存储空间用于存储 value，所述 key 与所述 value 属于同一个 KV 对;

从所述 NVMe 写命令中获得所述 key，根据所述 value 指针获得 value 长度，按照所述 value 长度为所述 value 分配第二存储空间，所述第二存储空间在所述存储介质中;

发送第一传输请求给所述主机，从所述主机获得所述 value，把所述 value 保存在所述第二存储空间中。

15、如权利要求 14 所述的 NVMe 存储器，其中， 发送第一传输请求给主机以及从所述主机获得所述 value，具体包括:

所述 NVMe 存储器和所述主机用 PCIe 总线连接，发送 DMA 传输请求给所述主机，从所述主机获得所述 value，所述 DMA 指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址。

16、如权利要求 14 所述的 NVMe 存储器，其中，发送第一传输请求给主机以及从所述主机获得所述 value，具体包括:

所述 NVMe 存储器和所述主机用 Fabric 总线连接，发送 RDMA 传输请求给所述主机，从所述主机获得所述 value，所述 RDMA 指令携带所述第一存储空间作为访问地址，携带所述第二存储空间作为写入地址。

17、如权利要求 14 所述的 NVMe 存储器，其中:

所述 NVMe 写命令中进一步携带 KV 数量的字段，所述 KV 数量的字段用于描述所述 NVMe 写命令中 KV 的数量，所述 NVMe 存储器从所述 NVMe 写命令中获得与 KV 数量相同数量的 key，以及获得与 KV 数量相同数量的 value。

18、如权利要求 14 所述的 NVMe 存储器，其中：

所述 NVMe 写命令中进一步携带 KV 格式的字段，其中，所述 KV 格式的字段描述所述 NVMe 写命令中字段的结构，所述 NVMe 存储器按照所述 KV 格式字段所定义的字段内容，从所述 NVMe 写命令中获取各个字段。

19、如权利要求 14 所述的 NVMe 存储器，其中，所述 NVMe 写命令还携带所述 value 元数据指针，所述处理器还被配置为：

根据所述元数据指针获得元数据长度，按照所述元数据长度为所述元数据分配第四存储空间，所述第四存储空间在所述 NVMe 存储器中；

通过所述第一传输请求从所述主机中获得所述元数据，把所述元数据保存在第四存储空间中。

20、如权利要求 19 所述的 NVMe 存储器，其中，所述处理器还被配置为：

所述元数据传输失败后，则释放为所述 value 分配的存储空间，以及释放为所述元数据分配的第四存储空间。

21、如权利要求 14 所述的 NVMe 存储器，其中，发送第一传输请求给所述主机以及从所述主机获得所述 value，具体包括下述其中一种：

发送至少两个 DMA 传输请求给所述主机以获得所述 value，每个 DMA 传输请求用于请求获得所述 value 的一部分，当任意一个 DMA 传输请求执行失败，则 NVMe 存储器释放为所述 value 分配的存储空间，其中所述 NVMe 存储器和所述主机用 PCIe 总线连接；或者

发送至少两个 RDMA 传输请求给所述主机以获得所述 value，每个 RDMA 传输请求用于请求获得所述 value 的一部分，当任意一个 RDMA 传输请求执行失败，则 NVMe 存储器释放为所述 value 分配的存储空间，其中，所述 NVMe 存储器和所述主机用 Frabic 连接。

22、如权利要求 14 所述的 NVMe 存储器，其中，所述处理器还被配置为：

生成所述 key 与所述第二存储空间的映射关系;

从所述主机接收 NVMe 读命令，所述 NVMe 读命令中携带所述 key;

根据所述 key 在所述映射关系中查询，获得存储所述 value 的所述第二存储空间的位置信息，把所述第二存储空间的位置信息发送给所述主机;

接收所述主机发送的第二传输请求，所述第二传输请求用于请求获得存储在所述第二存储空间的数据。

23、一种数据处理方法，其特征在于，该方法包括:

非易失性存储快速 NVMe 存储器接收 NVMe 写命令，所述 NVMe 写命令的头部携带键 key，所述 NMVe 命令还携带值 value，所述 key 和所述 value 对应，所述 key 和所述 value 属于同一个 KV 对;

所述 NVMe 存储器从所述 NVMe 写命令中获得所述 key 以及所述 value;

所述 NVMe 存储器把所述 value 保存在所述 NVMe 存储器的存储介质中。

24、如权利要求 23 所述的数据处理方法，其中:

所述 NVMe 写命令中进一步携带 KV 数量的字段，所述 KV 数量的字段用于描述所述 NVMe 写命令中 KV 的数量，所述 NVMe 存储器从所述 NVMe 写命令中获得与 KV 数量相同数量的 key，以及获得与 KV 数量相同数量的 value。

25、如权利要求 23 所述的数据处理方法，其中:

所述 NVMe 写命令中进一步携带 KV 格式的字段，其中，所述 KV 格式的字段描述所述 NVMe 写命令中字段的结构，所述 NVMe 存储器按照所述 KV 格式字段所定义的字段内容，从所述 NVMe 写命令中获取各个字段。

26、如权利要求 23 所述的数据处理方法，其中，所述 NVMe 写命令中进一步携带所述 key 的长度，所述 NVMe 存储器从所述写命令中获得所述 key 具体包括:

从所述 key 的预设起始位置，按照所述 key 的长度从所述写命令中获得所述 key。

27、如权利要求 23 所述的方法，其中，所述 NVMe 写命令中进一步携带 value 偏移量以及所述 value 的长度，所述 NVMe 存储器从所述 NVMe 写命令中获得所述 value 具体包括:

所述 NVMe 存储器从所述偏移量指示的位置，按照所述 value 的长度获得所述 value。

28、如权利要求 23 所述的数据处理方法，其中:

该方法进一步包括: 生成所述 key 与所述 value 存储空间的映射关系;

该方法之后，进一步包括:

主机发送 NVMe 读命令给所述 NVMe 存储器，所述 NVMe 读命令中携带所述 key;

所述 NVMe 存储器从所述主机接收所述读命令，从所述 NVMe 读命令中获得所述 key;

所述 NVMe 存储器使用所述 key 从所述映射关系中查找所述 value 存储空间;

所述 NVMe 存储器从使用所述 value 存储空间获得所述 value;

所述 NVMe 存储器生成所述 NVMe 读命令的响应消息发送给所述主机，所述响应消息携带所述 value。

29、一种 NVMe 存储器，包括控制器以及存储介质，所述控制器和所述存储介质连接，所述存储介质用于提供存储空间，其特征在于，所述处理器被配置为执行:

接收主机发送的快速非易失性存储 NVMe 写命令，所述 NVMe 写命令的头部携带键 key，所述 NMVe 命令还携带值 value，所述 key 和所述 value 对应，所述 key 和所述 value 属于同一个 KV 对;

从所述 NVMe 写命令中获得所述 key 以及所述 value;

把所述 value 保存在所述存储介质中。

30、如权利要求 29 所述的 NVMe 存储器数据处理方法，其中:

所述 NVMe 写命令中进一步携带 KV 数量的字段，所述 KV 数量的字段用于描述所述 NVMe 写命令中 KV 的数量，所述处理器从所述 NVMe 写命令中获得与 KV 数量相同数量的 key，以及获得与 KV 数量相同数量的 value。

31、如权利要求 29 所述的 NVMe 存储器数据处理方法，其中：

所述 NVMe 写命令中进一步携带 KV 格式的字段，其中，所述 KV 格式的字段描述所述 NVMe 写命令中字段的结构，所述处理器按照所述 KV 格式字段所定义的字段内容，从所述 NVMe 写命令中获取各个字段。

32、如权利要求 29 所述的 NVMe 存储器数据处理方法，其中，所述 NVMe 写命令中进一步携带所述 key 的长度，从所述写命令中获得所述 key 具体包括：

从所述 key 的预设起始位置，按照所述 key 的长度从所述写命令中获得所述 key。

33、如权利要求 29 所述的 NVMe 存储器，其中，所述 NVMe 写命令中进一步携带 value 偏移量以及所述 value 的长度，从所述 NVMe 写命令中获得所述 value 具体包括：

从所述偏移量指示的位置，按照所述 value 的长度获得所述 value。

34、如权利要求 29 所述的 NVMe 存储器数据处理方法，其中，所述处理器还被配置为：

生成所述 key 与所述 value 存储空间的映射关系；

所述 NVMe 存储器从所述主机接收 NVMe 读命令，从所述 NVMe 读命令中获得所述 key；

使用所述 key 从所述映射关系中查找所述 value 存储空间；

从使用所述 value 存储空间获得所述 value；

生成所述 NVMe 读命令的响应消息发送给所述主机，所述响应消息携带所述 value。

35、一种存储系统，包括权利要求 14-22 任意一项所述的 NVMe 存储器，

还包括所述主机，其中：

所述主机用于构造所述 NVMe 写命令，发送所述 NVMe 写命令给所述 NVMe 存储器。

36、一种存储系统，包括主机和权利要求 29-34 任意一项所述的 NVMe 存储器，还包括所述主机，其中：

所述主机用于构造所述 NVMe 写命令，发送所述 NVMe 写命令给所述 NVMe 存储器。

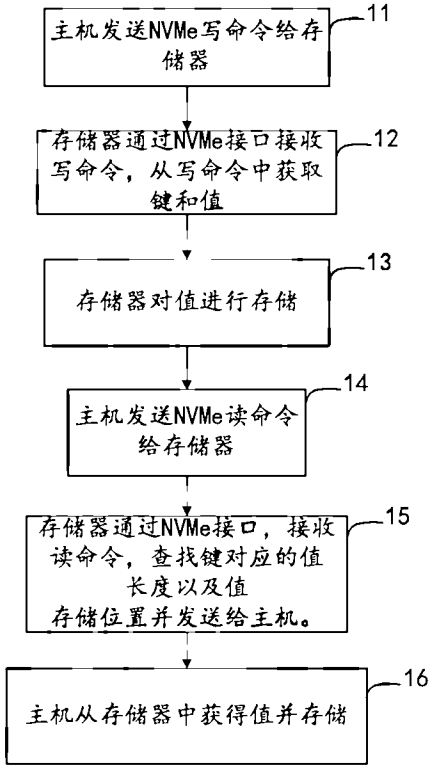

主机发送NVMe写命令给存储器
11
存储器通过NVMe接口接收写命令，从写命令中获取键和值
12
存储器对值进行存储
13
主机发送NVMe读命令给存储器
14
存储器通过NVMe接口，接收读命令，查找键对应的值长度以及值存储位置并发送给主机。
15
主机从存储器中获得值并存储
16

图 1


公共头部
KV格式ID
键1长度
键1
值1长度
值1
元数据1长度
元数据1
键2长度
键2
值2长度
值2
元数据2长度
元数据2

图 2

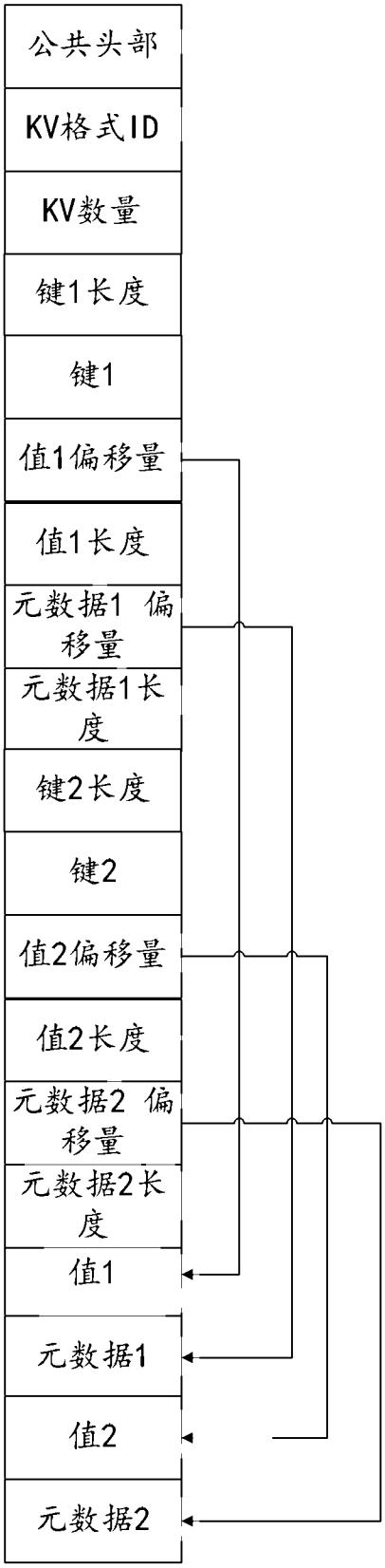

公共头部
KV格式ID
KV数量
键1长度
键1
值1偏移量
值1长度
元数据1 偏移量
元数据1长度
键2长度
键2
值2偏移量
值2长度
元数据2 偏移量
元数据2长度
值1
元数据1
值2
元数据2

图 3


公共头部
KV格式ID
键1长度
键1
值1长度
值1
元数据1长度
元数据1
键2长度
键2
值2长度
值2
元数据2长度
元数据2

图 4

| 公共头部 |
| --- |
| KV格式ID |
| 键1长度 |
| 键1 |
| 值1长度 |
| 元数据1长度 |
| 值1 |
| 元数据1 |
| 值2长度 |
| 键2 |
| 值2长度 |
| 元数据2长度 |
| 值2 |
| 元数据2 |

图 5

| 键 | 值存储空间的起始地址 | 值长度 | 元数据存储空间的起始地址 | 元数据长度 |
| --- | --- | --- | --- | --- |
| 键 1 | 地址 1 | 长度 1 | 地址 2 | 长度 2 |
| 键 2 | 地址 3 | 长度 3 | 地址 4 | 长度 4 |
| 键 3 | 地址 5 | 长度 5 | 地址 6 | 长度 6 |
| …… | …… | …… | …… | …… |

图 6

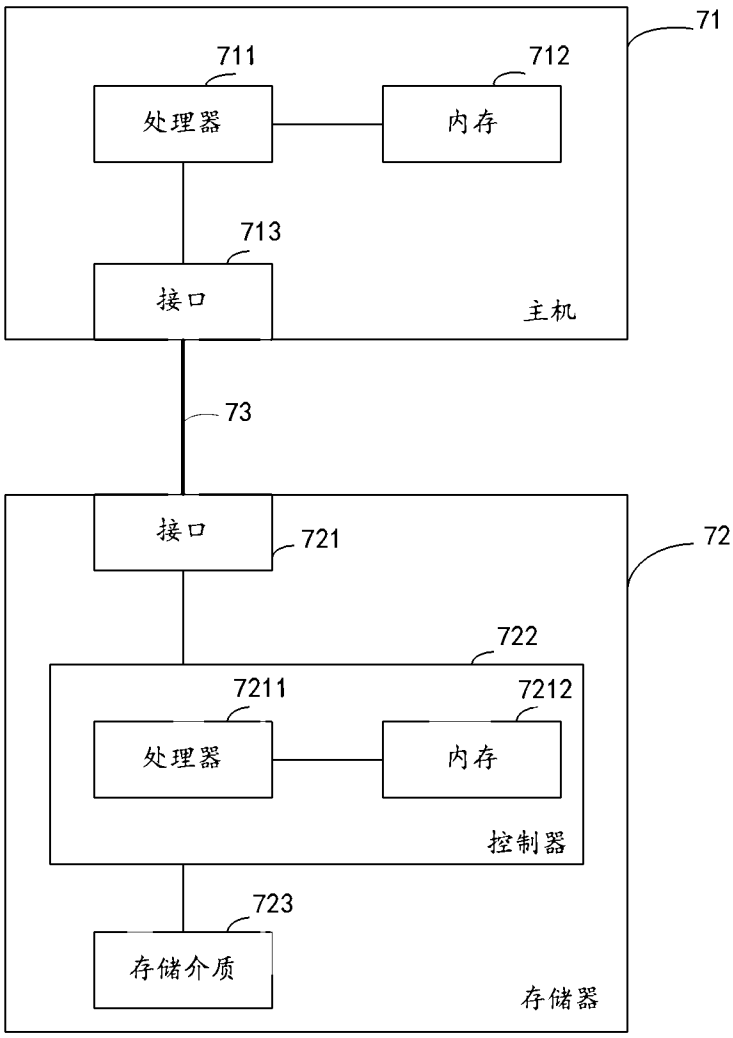

71
711
处理器
712
内存
713
接口
主机
73
接口
721
72
722
7211
处理器
7212
内存
控制器
723
存储介质
存储器

图 7

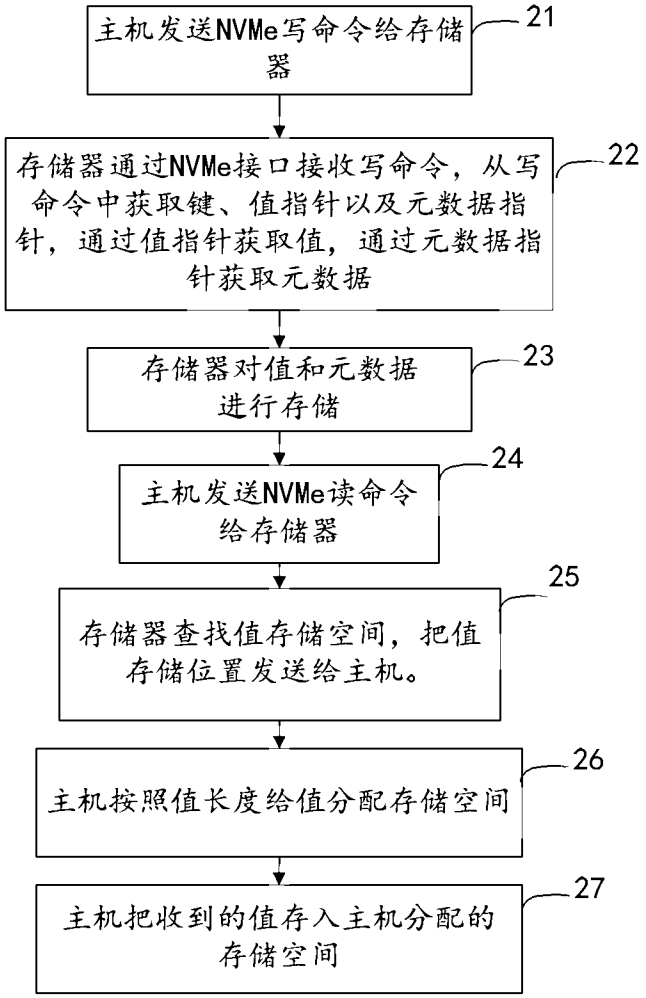

主机发送NVMe写命令给存储器
21
存储器通过NVMe接口接收写命令，从写命令中获取键、值指针以及元数据指针，通过值指针获取值，通过元数据指针获取元数据
22
存储器对值和元数据进行存储
23
主机发送NVMe读命令给存储器
24
存储器查找值存储空间，把值存储位置发送给主机。
25
主机按照值长度给值分配存储空间
26
主机把收到的值存入主机分配的存储空间
27

图 8

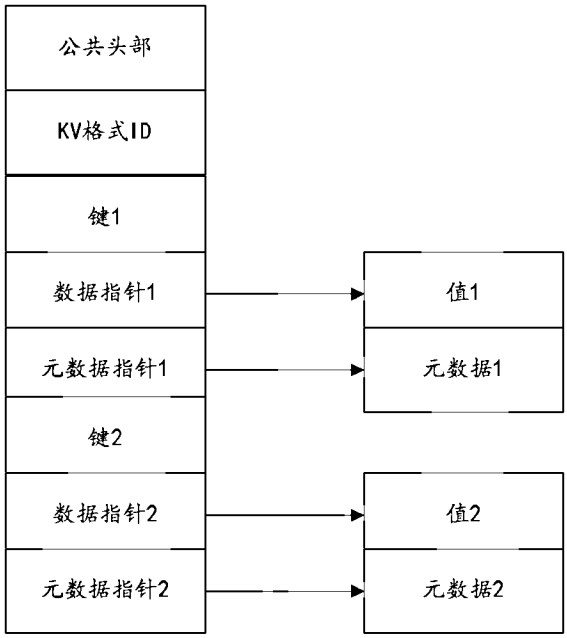



图 9

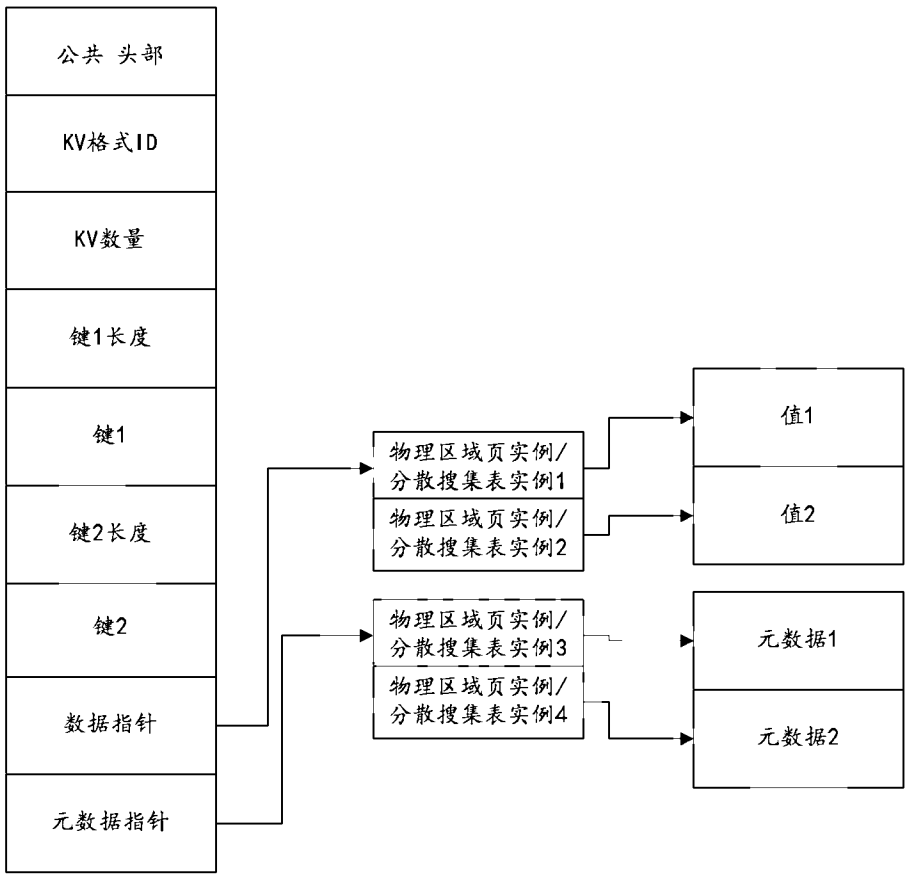

公共 头部
KV格式ID
KV数量
键1长度
键1
键2长度
键2
数据指针
元数据指针
物理区域页实例/
分散搜集表实例1
物理区域页实例/
分散搜集表实例2
物理区域页实例/
分散搜集表实例3
物理区域页实例/
分散搜集表实例4
值1
值2
元数据1
元数据2

图 10

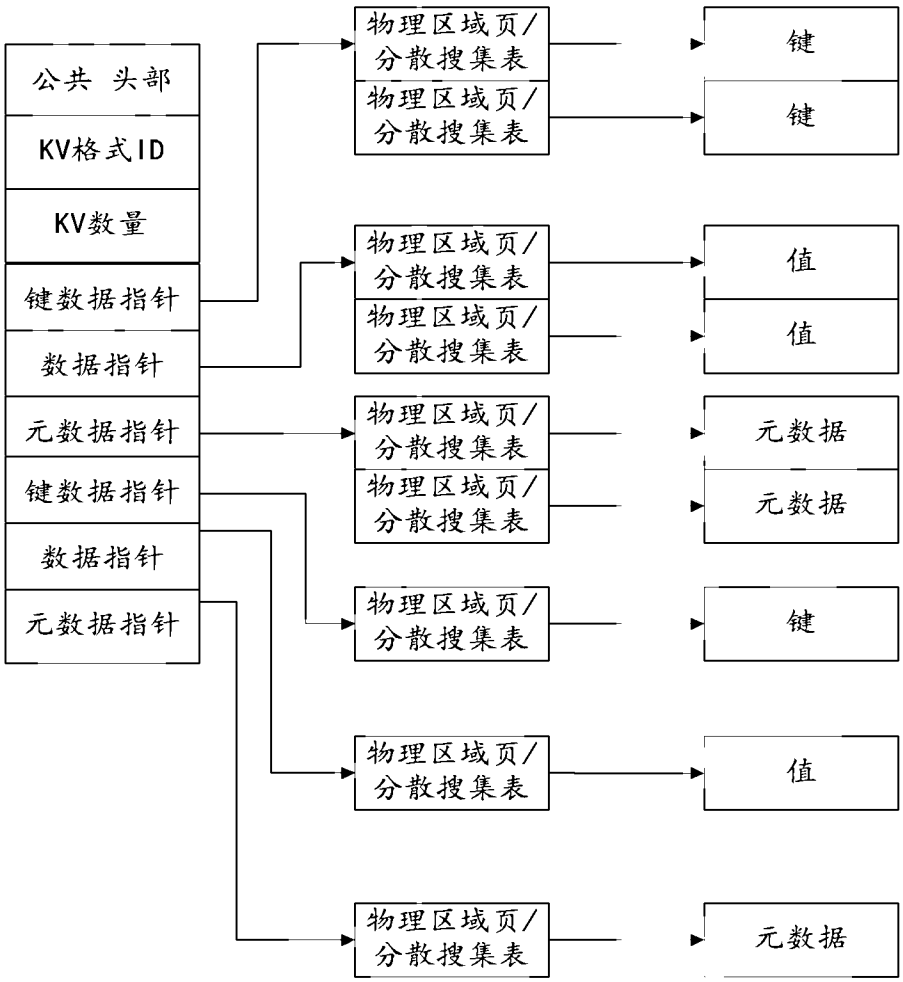

公共 头部
KV格式ID
KV数量
键数据指针
数据指针
元数据指针
键数据指针
数据指针
元数据指针
物理区域页/分散搜集表
物理区域页/分散搜集表
物理区域页/分散搜集表
物理区域页/分散搜集表
物理区域页/分散搜集表
物理区域页/分散搜集表
物理区域页/分散搜集表
物理区域页/分散搜集表
键
键
值
值
元数据
元数据
键
值
元数据

图 11

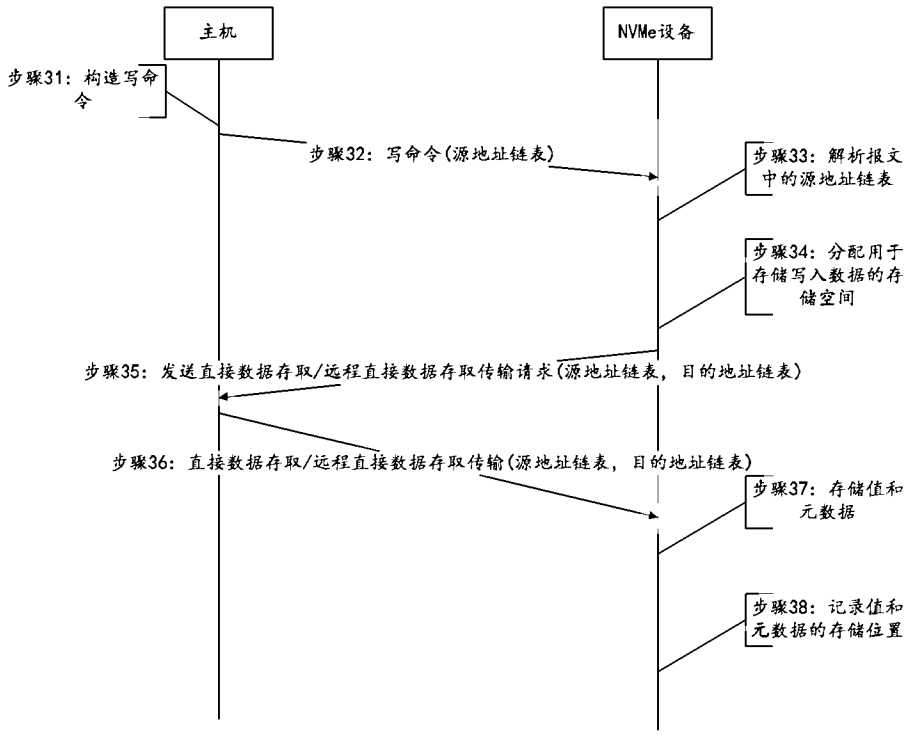

主机
NVMe设备
步骤31：构造写命令
步骤32：写命令(源地址链表)
步骤33：解析报文中的源地址链表
步骤34：分配用于存储写入数据的存储空间
步骤35：发送直接数据存取/远程直接数据存取传输请求(源地址链表，目的地址链表)
步骤36：直接数据存取/远程直接数据存取传输(源地址链表，目的地址链表)
步骤37：存储值和元数据
步骤38：记录值和元数据的存储位置

图 12

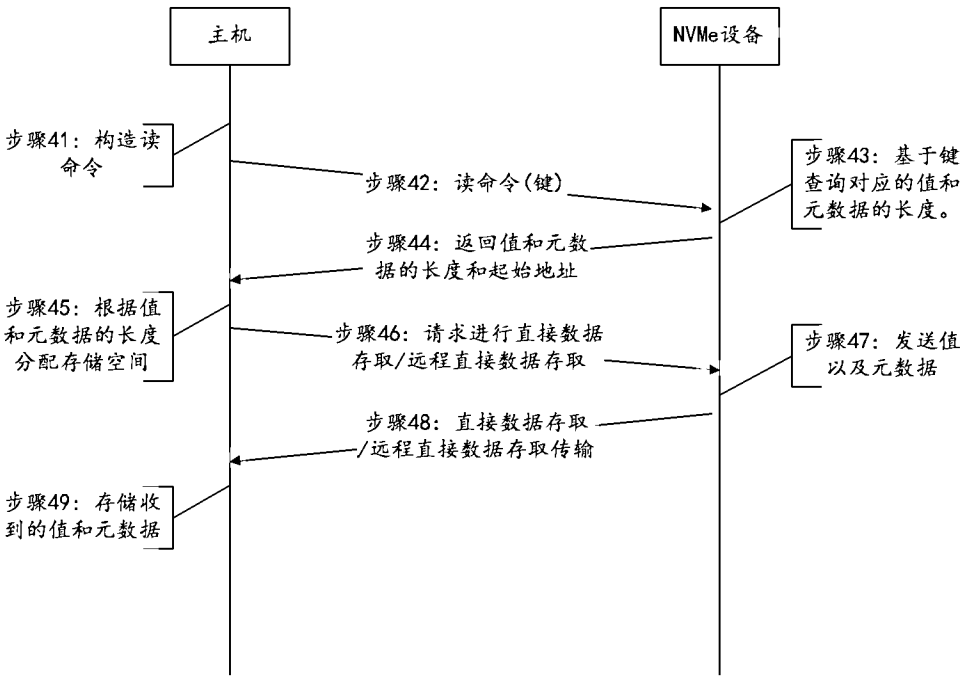

主机
NVMe设备
步骤41：构造读命令
步骤42：读命令(键)
步骤43：基于键查询对应的值和元数据的长度。
步骤44：返回值和元数据的长度和起始地址
步骤45：根据值和元数据的长度分配存储空间
步骤46：请求进行直接数据存取/远程直接数据存取
步骤47：发送值以及元数据
步骤48：直接数据存取/远程直接数据存取传输
步骤49：存储收到的值和元数据

图 13

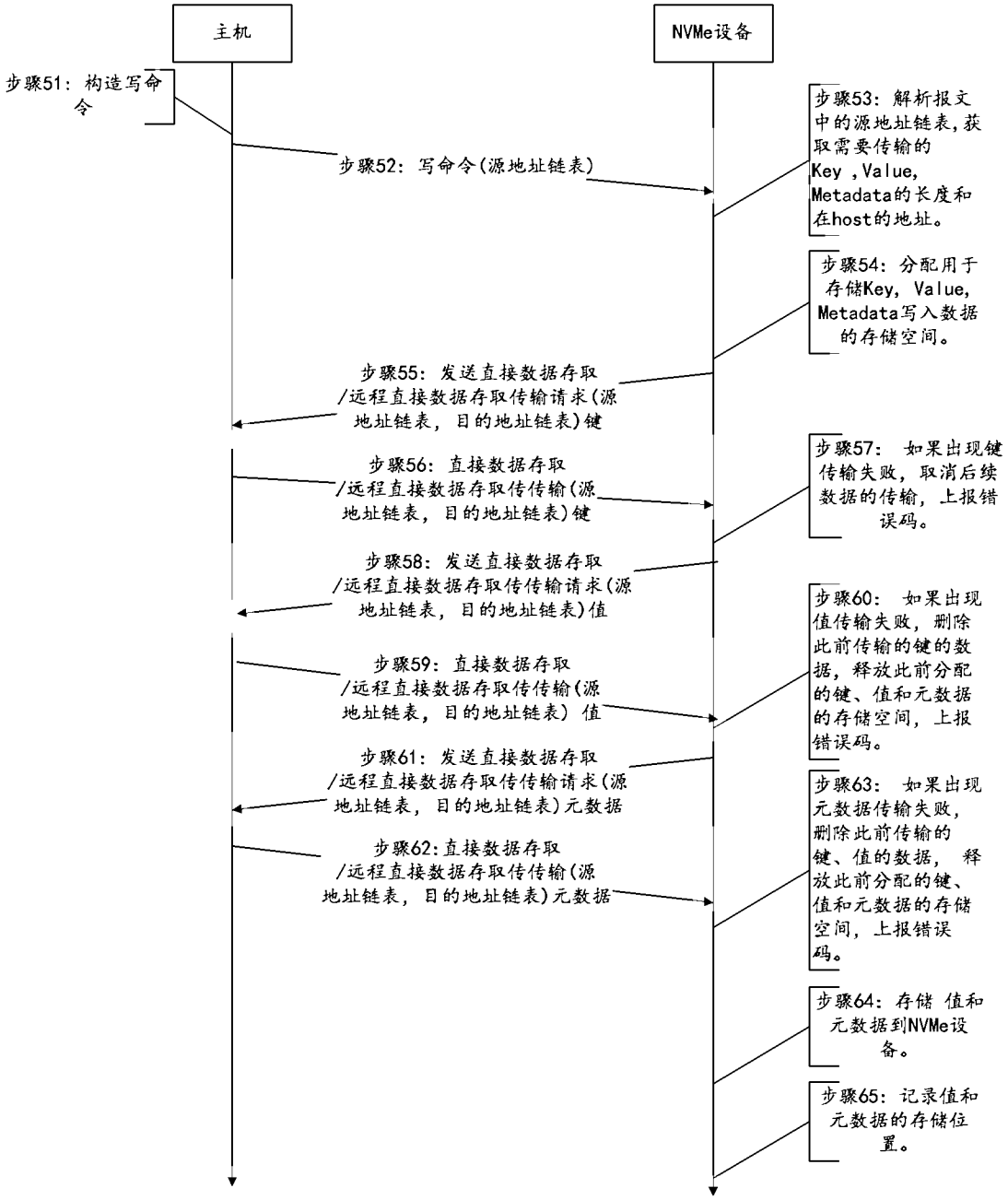

主机
NVMe设备
步骤51：构造写命令
步骤52：写命令(源地址链表)
步骤53：解析报文中的源地址链表,获取需要传输的Key ,Value, Metadata的长度和在host的地址。
步骤54：分配用于存储Key, Value, Metadata写入数据的存储空间。
步骤55：发送直接数据存取/远程直接数据存取传输请求(源地址链表，目的地址链表)键
步骤56：直接数据存取/远程直接数据存取传传输(源地址链表，目的地址链表)键
步骤57： 如果出现键传输失败，取消后续数据的传输，上报错误码。
步骤58：发送直接数据存取/远程直接数据存取传传输请求(源地址链表，目的地址链表)值
步骤59：直接数据存取/远程直接数据存取传传输(源地址链表，目的地址链表) 值
步骤60： 如果出现值传输失败，删除此前传输的键的数据，释放此前分配的键、值和元数据的存储空间，上报错误码。
步骤61：发送直接数据存取/远程直接数据存取传传输请求(源地址链表，目的地址链表)元数据
步骤62:直接数据存取/远程直接数据存取传传输(源地址链表，目的地址链表)元数据
步骤63： 如果出现元数据传输失败，删除此前传输的键、值的数据， 释放此前分配的键、值和元数据的存储空间，上报错误码。
步骤64：存储 值和元数据到NVMe设备。
步骤65：记录值和元数据的存储位置。

图 14

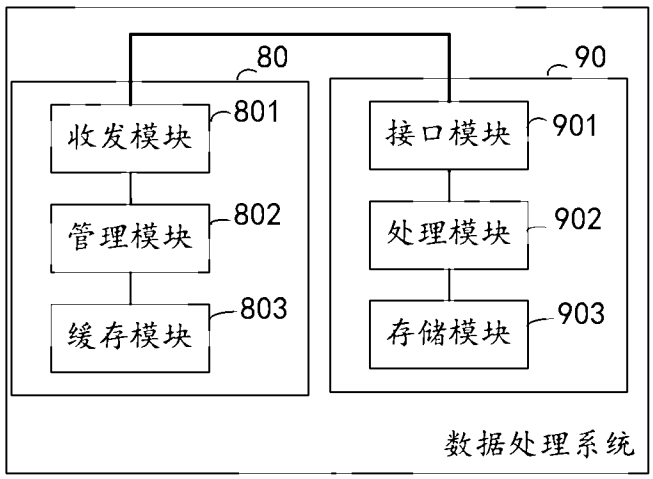

80
90
收发模块
801
管理模块
802
缓存模块
803
接口模块
901
处理模块
902
存储模块
903
数据处理系统

图 15

# INTERNATIONAL SEARCH REPORT

International application No.
**PCT/CN2016/103268**

**A. CLASSIFICATION OF SUBJECT MATTER**

G06F 3/06 (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)

G06F 3/06

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, CNKI, DWPI, SIPOABS: key value, HUAWEI; QIU, Xin; XU, Huifeng; GUO, Haitao; LIU, Hongguang; LIU, Huawei; TAN, Chunyi; Gissin; key, value, KV, K/V, Non-Volatile Memory, NVMe, writ+, read+, pointer, address, length

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | CN 103973810 A (HUAWEI TECHNOLOGIES CO., LTD.), 06 August 2014 (06.08.2014), the whole document | 1-36 |
| A | CN 104461380 A (HUAWEI TECHNOLOGIES CO., LTD.), 25 March 2015 (25.03.2015), the whole document | 1-36 |
| A | US 2013086311 A1 (HUANG, M. et al.), 04 April 2013 (04.04.2013), the whole document | 1-36 |

☐ 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 | Date of mailing of the international search report |
|---|---|
| 24 January 2017 (24.01.2017) | **06 February 2017 (06.02.2017)** |
| Name and mailing address of the ISA/CN:<br>State Intellectual Property Office of the P. R. China<br>No. 6, Xitucheng Road, Jimenqiao<br>Haidian District, Beijing 100088, China<br>Facsimile No.: (86-10) 62019451 | Authorized officer<br>**WANG, Kun**<br>Telephone No.: (86-10) **62412187** |

# INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
**PCT/CN2016/103268**

| Patent Documents referred in the Report | Publication Date | Patent Family | Publication Date |
|---|---|---|---|
| CN 103973810 A | 06 August 2014 | None | |
| CN 104461380 A | 25 March 2015 | None | |
| US 2013086311 A1 | 04 April 2013 | None | |

# 国际检索报告

国际申请号
PCT/CN2016/103268

A. 主题的分类

G06F 3/06(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)

G06F 3/06

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,CNKI,DWPI,SIPOABS 华为，邱鑫，许慧锋，郭海涛，刘洪广，刘华伟，谭春毅，吉辛，键值，非易失性存储，写，读，指针，地址，长度，HUAWEI，QIU XIN，XU HUIFENG，GUO HAITAO，LIU HONGGUANG，LIU HUAWEI，TAN CHUNYI，GISSIN，key，value，KV，K/V，Non-Volatile Memory，NVMe，writ+，read+，pointer，address，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 |
|---|---|---|
| A | CN 103973810 A (华为技术有限公司) 2014年 8月 6日 (2014-08-06)<br>全文 | 1-36 |
| A | CN 104461380 A (华为技术有限公司) 2015年 3月 25日 (2015-03-25)<br>全文 | 1-36 |
| A | US 2013086311 A1 (HUANG M 等) 2013年 4月 4日 (2013-04-04)<br>全文 | 1-36 |

☐ 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 | 国际检索报告邮寄日期 |
|---|---|
| 2017年 1月 24日 | 2017年 2月 6日 |
| ISA/CN的名称和邮寄地址<br>中华人民共和国国家知识产权局(ISA/CN)<br>中国北京市海淀区蓟门桥西土城路6号 100088<br>传真号 (86-10)62019451 | 受权官员<br>王昆<br>电话号码 (86-10)62412187 |

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2016/103268

| 检索报告引用的专利文件 | 公布日<br>(年/月/日) | 同族专利 | 公布日<br>(年/月/日) |
|---|---|---|---|
| CN 103973810 A | 2014年 8月 6日 | 无 | |
| CN 104461380 A | 2015年 3月 25日 | 无 | |
| US 2013086311 A1 | 2013年 4月 4日 | 无 | |