

对象处理方法及计算设备

本申请要求于 2023 年 06 月 26 日提交中国专利局、申请号为 202310761011.0、发明名称为“对象处理方法及计算设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请实施例涉及计算机技术领域，尤其涉及一种对象处理方法及计算设备。

背景技术

10 为了方便跨平台存储以及网络传输、需要将对象进行序列化/反序列化处理，序列化处理即是指将对象转化为可传输的字节信息的过程，反序列化即是将字节信息还原为对象的过程。而具有关联关系的一组对象构成了对象图，由于序列化任何对象的时候，如果该对象关联了其它对象，其它对象也会被序列化，因此，实际应用中，涉及对象图的序列化/反序列化。

15 对象图中的每个对象在内存中的数据格式通常包括值域以及指针域，值域存储对象数据，而指针域存储每个对象所关联的子对象的内存地址。在一些情形中，对对象图进行序列化处理时，需要将每个对象的编码数据暂存至缓存中，此时，由于指针域存储子对象的内存地址，需要将内存地址编码为子对象在缓存中的相对偏移量。对象图全部编码之后，将生成的序列化数据发送至接收端，接收端进行反序列化处理以还原对象图的过程中，再将序列化数据中的指针域编码数据，也即相对偏移量解码为反序列之后的内存地址。

20

然而，上述描述的一些情形中的这种处理方式，端到端的延迟较大，影响处理效率。

发明内容

本申请实施例提供一种对象处理方法及计算设备，用以解决相关技术中端到端延迟较大，影响处理效率的技术问题。

25

第一方面，本申请实施例中提供了一种对象处理方法，包括：根据对象图中第一对象关联的第二对象的序列化顺序，确定所述第二对象的序列化编号；将所述第二对象的序列化编号作为所述第一对象的指针域编码数据；按照所述对象图中不同对象的序列化顺序，确定所述对象图中待序列化的至少部分对象；基于所述至少部分对象的值域编码数据及指针域编码数据，生成所述至少部分对象的序列化数据；将所述序列化数据传输至接收端；所述接收端用以结合所述对象图对应的至少一个序列化数据的反序列化结果，获得所述对象图。

30

第二方面，本申请实施例提供了对象处理方法，包括：获取发送端传输的序列化数据；所述序列化数据基于对象图中至少部分对象的值域编码数据以及指针域编码数据生成；所述指针域编码数据为关联子对象的序列化编号；将所述序列化数据进行反序列化处理，以确定所述至少部分对象的内存地址，以及所述至少部分对象的值域编码数据分别对应的值域数据；将所述至少部分对象分别对应的值域数据写入各自的内存地址中；结合所述对象

35

图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入所述对象的内存地址。

第三方面，本申请实施例提供了一种计算设备，包括处理组件以及存储组件；所述存储组件存储一个或多个计算机指令；所述一个或多个计算机指令用以被所述处理组件调用
5 执行，以实现如上述第一方面所述的对象处理方法或者实现如上述第二方面所述对象处理方法。

第四方面，本申请实施例提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时，现如上述第一方面所述的对象处理方法或者实现如上述第二方面所述对象处理方法。

10 本申请的这些方面或其他方面在以下实施例的描述中会更加简明易懂。

附图说明

在附图中，除非另外规定，否则贯穿多个附图相同的附图标记表示相同或相似的部件或元素。这些附图不一定是按照比例绘制的。应该理解，这些附图仅描绘了根据本申请公
15 开的一些实施方式，而不应将其视为是对本申请范围的限制。

为了更清楚地说明本申请实施例或相关技术中的技术方案，下面将对实施例或相关技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据
20 这些附图获得其他的附图。

- 20 图 1 示出了本申请实施例的技术方案应用于其中的一种系统架构图。
- 图 2 示出了本申请提供的一种对象处理方法一个实施例的流程图。
- 图 3 示出了本申请实施例在一个实际应用中表示对象关联关系的一种示意图。
- 图 4A~图 4C 分别示出了本申请实施例在一个实际应用中的预定数据格式示意图。
- 图 4D~图 4F 示出了本申请实施例在一个实际应用中的序列化数据示意图。
- 25 图 4G 示出了本申请实施例在一个实际应用中的元数据关联示意图。
- 图 5 示出了本申请提供的一种对象处理方法又一个实施例的流程图。
- 图 6A 示出了本申请实施例在一个实际应用中的内存数据格式示意图。
- 图 6B 示出了示出了本申请实施例在一个实际应用中的序列化/反序列化的操作示意图。
- 30 图 7 示出了本申请提供的一种对象处理装置一个实施例的结构示意图。
- 图 8 示出了本申请提供的一种对象处理装置又一个实施例的结构示意图。
- 图 9 示出了本申请提供的一种计算设备一个实施例的结构示意图。

具体实施方式

35 为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述。

在本申请的说明书和权利要求书及上述附图中的描述的一些流程中，包含了按照特定

顺序出现的多个操作，但是应该清楚了解，这些操作可以不按照其在本文中出现的顺序来执行或并行执行，操作的序号如 101、102 等，仅仅是用于区分开各个不同的操作，序号本身不代表任何的执行顺序。另外，这些流程可以包括更多或更少的操作，并且这些操作可以按顺序执行或并行执行。需要说明的是，本文中的“第一”、“第二”等描述，是用于区分不同的消息、设备、模块等，不代表先后顺序，也不限定“第一”和“第二”是不同的类型。

本申请实施例的技术方案可以应用于针对对象图进行序列化/反序列化的应用场景，特别是公有云中所涉及的序列化/反序列化的应用场景。

由于序列化/反序列化导致的端到端延迟对用户体验以及处理性能等都非常重要，比如微服务场景下，多个微服务之间往往涉及循环调用，一个微服务发送至另一个微服务的请求，另一个微服务可能需要调用其它微服务进行相应处理之后，才能进行请求的响应，而微服务之间的调用涉及对象传输，在循环调用场景下，由于端到端延迟的存在，会导致请求响应时间更长。因此，如何降低端到端延迟，提高传输效率成为了本领域技术人员需要解决的技术问题。

发明人在实现本申请的过程中发现，目前针对对象图进行序列化处理，将全部对象序列化完成写入缓存中之后，需要遍历全部对象以确定子对象的相对偏移量，再将每个对象的指针域数据编码为该相对偏移量。接收端获得对象图的序列化数据之后，全部对象反序列化之后，需要遍历全部对象，再将每个对象的相对偏移量解码为反序列获得的内存地址，可见这一过程往往耗时较长，从而导致端到端之间的延迟。为了提高降低端到端延迟，提高处理效率，发明人经过一系列研究提出了本申请的技术方案，在本申请实施例中，利用序列化顺序对应的序列化编号作为指针域编码数据，利用序列化顺序关联对象与子对象，相较于采用相对偏移量的方式，无需遍历全部对象以确定子对象的相对偏移量，因此可以减少图遍历操作，降低操作复杂性，从而有助于降低端到端延迟，提高传输效率，且利用序列化顺序关联对象与子对象，部分对象即可以进行序列化/反序列化，而无需等待全部对象处理完成，从而可以进一步的降低端到端延迟，提高传输效率。

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

图 1 示出了本申请实施例的技术方案可以应用于其中的一种系统架构图，该系统架构可以包括发送端 101 以及接收端 102。

实际应用中，发送端 101 以及接收端 102 可以为客户端、服务端或者服务端中的应用程序等，图 1 中虽以服务端形式表示发送端或接收端，可以理解的是，本申请并不限于此。

发送端 101 以及接收端 102 之间可以通过网络建立连接。网络为发送端 101 与接收端

102 之间提供了通信链路的介质。网络可以包括各种连接类型，例如有线、无线通信链路或者光纤电缆等等。

发送端 101 以及接收端 102 之间进行交互通信过程中涉及对象图的网络传输，将对象图进行网络传输，即涉及序列化/反序列化处理。

5 发送端 101 或接收端 102 实现为客户端的情况下，客户端可以为浏览器、APP（Application，应用程序）、或网页应用如 H5（HyperText Markup Language5，超文本标记语言第 5 版）应用、或轻应用（也被称为小程序，一种轻量级应用程序）或云应用等，客户端可以部署在电子设备中，需要依赖设备运行或者设备中的某些 app 而运行等。

10 发送端 101 或接收端 102 实现为服务端的情况下，服务端可以实现成多个服务器组成的分布式服务器集群，也可以实现成单个服务器。服务器也可以为分布式系统的服务器，或者是结合了区块链的服务器。服务器也可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、内容分发网络(Content Delivery Network, CDN)、以及大数据和人工智能平台等基础云计算服务的云服务器，或者是带人工智能技术的智能云计算服务器或智能云主机。

15 应该理解，图 1 中的发送端和接收端的数目仅仅是示意性的。根据实现需要，可以具有任意数目的发送端和接收端。

需要说明的是，本申请实施例中可能会涉及到对用户数据的使用，在实际应用中，可以在符合所在国的适用法律法规要求的情况下（例如，用户明确同意，对用户切实通知，等），在适用法律法规允许的范围内在本文描述的方案中使用用户特定的个人数据。

20 以下对本申请实施例的技术方案的实现细节进行详细阐述。

图 2 为本申请实施例提供的一种对象处理方法一个实施例的流程图，本实施例的技术方案由发送端执行，该方法可以包括以下几个步骤 201 至 205。

201：根据对象图中第一对象关联的第二对象的序列化顺序，确定第二对象的序列化编号。

25 对象图可以包括具有关联关系的多个对象，一个对象关联（或者指向）另一个对象的情况下，为了方便区分，本申请实施例中，另一个对象也被称为该对象的子对象。

在针对对象图进行序列化处理时，可以从对象图的第一个对象开始，依次对各个对象进行序列化处理，也即进行编码处理，因此，根据对象图中多个对象的关联关系，可以确定每个对象的序列化顺序。

30 该第一对象可以是指对象图中当前编码的任意一个对象，第二对象也即为第一对象关联的任一个子对象。

根据第一对象的序列化顺序，可以确定对应的序列化编号，序列化顺序不同，序列化编号亦不同。序列化编号例如可以采用整型字符表示，例如从 0 或者 1 开始，表示第一个进行序列化处理的对象。当然，序列化编号也可以其它类型的字符表示，也不局限于从 0

或 1 开始，本申请对此不进行限定。

为了方便理解，假设对象图中包括 8 个对象：A、B、C、D、E、F、G、H。假设对象 A 为第一个序列化的对象，该 8 个对象的关联关系可以如图 3 中所示，图 3 中的箭头指向即表示了一个对象所引用的其它对象。

5 对象 A 为第一个序列化对象，对象 A 对应的序列化编号为 0。

对象 A 关联两个子对象：对象 B 以及对象 D，根据对象 B 和对象 D 在对象 A 的代码调用中的出现顺序或者数据类型，可以确定对象 B 为下一个序列化的对象，对象 D 为对象 B 之后的下一个序列化的对象，因此对象 B 的序列化编号为 1，对象 D 的序列化编号为 2。

对象 B 关联 1 个对象：对象 F，对象 F 的序列化编号为 3。

10 依次类推，对象 D 关联 1 个对象：对象 E，则对象 E 的序列化编号即为 4。

对象 F 关联 1 个对象：对象 D，由于对象 D 已经进行序列化处理，序列化编号为 2。

对象 E 关联两个对象：对象 B 以及对象 G，由于对象 B 已经进行了序列化处理，其序列化编号为 1，对象 G 的序列化编号即为 5。

15 对象 G 关联两个对象：对象 B 和对象 H，对象 B 已经进行了序列化处理，其序列化编号为 1，对象 H 的序列化编号即为 6。

对象 H 关联两个对象：对象 G 和对象 C，对象 G 已经进行了序列化处理，其序列化编号为 5，对象 C 的序列化编号即为 7。

对象 C 关联 3 个对象：对象 A、对象 G 以及对象 H，对象 A 的序列化编号为 0，对象 G 的序列化编号为 5，对象 H 的序列化编号为 6。

20 结合上述描述，该八个对象的序列化顺序为：A→B→D→F→E→G→H→C。

需要说明的是，每个对象的序列化顺序可以在对象图的序列化过程中逐一确定，可以无需预先确定，上述仅是为了方便理解而进行的描述。

202：将第二对象的序列化编号作为第一对象的指针域编码数据。

25 每个对象在内存中数据格式中通常由对象头部信息和实例数据构成，实例数据可以包括至少一个值域以及至少一个指针域，指针域数据即为子对象的内存地址。进行序列化处理时，除了对值域数据进行编码处理之外，还需要对指针域数据进行编码处理。

30 每个对象在内存的中对象头部信息例如通常可以包括 Mark Word（对象自身运行时数据）以及 Klass Pointer（类型指针）等；每个对象的对象头部信息，以及每个对象的实例数据所包括的至少一个值域以及至少一个指针域分别对应的位置以及类型等，可以作为元数据而预先注册并保存在不同端中。其中，每个对象的实例数据所包括的至少一个值域以及至少一个指针域分别对应的位置以及类型可以采用位图形式表示，例如 00110，其中，0 代表该位置对应的是值域，1 代表该位置对应的是指针域。

本申请实施例中，可以利用序列化顺序来关联对象与其子对象。因此，可以将第一对象中的第二对象对应的指针域数据修改为第二对象的序列化编号，作为指针域编码数据。

仍以图 3 对应的八个对象为例, 对象 A 中对应对象 B 的指针域数据即编码为序列化编号 1, 对应对象 D 的指针域数据即编码为序列化编号 2, 也即对象 A 的指针域编码数据为 1 和 2, 依次类推; 对象 B 的指针域编码数据为 3; 对象 D 的指针域编码数据为 4; 对象 F 的指针域编码数据为 2; 对象 E 的指针域编码数据为 1 和 5; 对象 G 的指针域编码数包括 1 和 6; 对象 H 的指针域编码数据包括 5 和 7; 对象 C 的指针域编码数据包括 0、5 和 6。

203: 按照对象图中不同对象的序列化顺序, 确定对象图中待序列化的至少部分对象。

204: 基于至少部分对象的值域编码数据及指针域编码数据, 生成至少部分对象的序列化数据。

10 在一些实施方式中, 可以是将对对象图中的全部对象作为待序列化的对象。从而可以基于全部对象的值域编码数据以及指针域编码数据而生成序列化数据。

在另一些实施方式中, 由于本申请实施例利用序列化顺序关联对象与子对象, 因此可以无需等待对象中全部对象编码完成, 可以基于部分对象的值域编码数据以及指针域编码数据, 即可以生成一段序列化数据。

因此, 一个对象图可以对应至少一个序列化数据。

15 其中, 每个对象的值域编码数据可以通过对值域数据进行编码获得, 以降低数据传输量等, 当然, 值域数据也可以直接作为值域编码数据等, 本申请对此不进行限定。

其中, 序列化数据中还可以包括基于每个对象的对象头部信息, 而生成的头部编码信息。每个对象的头部编码信息可以关联每个对象的元数据, 可以具体是与每个对象的元数据指针关联, 而元数据指针可以指向保存的每个对象的元数据, 每个对象的元数据中即可
20 以包括对象头部信息以及值域、指针域分别对应的位置和类型等。

该头部编码信息可以采用任意与元数据指针关联的标识信息。头部编码信息与对应的元数据指针可以保存在类型表, 类型表可以预先在不同端中注册保存等, 从而序列化数据中可以只包括每个对象的头部编码信息即可。

25 此外, 序列化数据中还可以包括至少部分对象对应的至少一个图属性信息, 至少一个图属性信息例如可以包括图大小以及对象数量中的一个或多个等, 至少一个图属性信息可以用于指示接收端分配对应的内存空间等, 其中, 图大小即是指所占内存大小。

因此, 在一实施例中, 可以是基于至少部分对象的值域编码数据、指针域编码数据、头部编码数据以及至少一个图属性信息, 生成序列化数据。该序列化数据的数据格式可以有多种实现方式。在下文实施例中会详情进行介绍。

30 205: 将序列化数据传输至接收端。

序列化数据生成之后即可以传输至接收端。其中, 可以每生成一个序列化数据即传输至接收端, 以减少缓存占用等。

其中, 接收端可以对对象图对应的至少一个序列化数据的反序列化结果, 获得对象图。

上述序列化数据中, 值域编码数据可以反序列化为值域数据, 指针编码数据可以用来

建立内存地址、偏移地址以及指针编码数据的映射关系，而至少一个序列化数据分别对应的映射关系可以查找指针编码数据映射的内存地址，并将指针编码数据映射的内存地址作为指针编码数据反序列化的指针域数据，具体的反序列化过程会在下文实施例中进行详情介绍。

5 本实施例中，利用序列化顺序对应的序列化编号作为指针域编码数据，利用序列化顺序关联对象与子对象，相较于采用相对偏移量的方式，可以减少图遍历操作，降低操作复杂性，从而有助于降低端到端延迟，提高传输效率，且利用序列化顺序关联对象与子对象，可以将对象图进行分段处理，至少部分对象即可以进行序列化/反序列化，而无需等待全部对象处理完成，从而可以进一步的降低端到端延迟，提高传输效率。

10 由上文描述可知，可以基于部分对象的值域编码数据以及指针域编码数据，生成序列化数据。而进行序列化处理时，需要将每个对象的编码结果暂存在缓存中。因此，在一实施方式中，按照对象图中不同对象的序列化顺序，确定待序列化的至少部分对象可以包括：按照对象图中不同对象的序列化顺序，将第一对象的值域编码数据及指针域编码数据写入第一缓存中；在第一缓存达到预定缓存大小或者写入的对象数量达到预定数量，将已写入
15 第一缓存中的对象作为待序列化的至少部分对象。

也即对每个对象进行编码时，可以将第一对象的值域编码数据及指针域编码数据写入第一缓存，此外还可以将第一对象的头部编码信息写入第一缓存中。

第一缓存即接收端中用于进行序列化处理的缓存（cache）。

若第一缓存达到了预定缓存大小或者写入的对象数量达到预定数量，此时即可以将第一缓存中已写入的所有对象，作为待序列化的至少部分对象而用以生成一段序列化数据。
20 该预定缓存大小或预定数量可以结合发送端或者接收端的最大缓存大小来确定等，以避免数据溢出问题等；或者可以结合实际延迟需求来确定等，本申请对此不进行限定。

在该实施例中，已编码对象所占的缓存大小达到预定缓存大小或者对象数量达到预定数量，即可以生成序列化数据，该序列化数据可以及时发送至接收端，从而第一缓存可以
25 继续用来序列化剩余对象。每个对象是按照序列化顺序而进行编码，并将编码结果写入第一缓存中。编码结果即可以包括值域编码数据、指针域编码数据、此外还可以包括头部编码数据等。

作为另一实施方式，该方法还可以包括：按照对象图中不同对象的序列化顺序，结合预定缓存大小或预定对象数量，将对象图进行分段处理，获得多个子对象图。

30 也即可以首先将对象图按照预定缓存大小获得预定对象数量，而将对象图划分多个子对象，例如图3所示示意图对应的包含八个对象的对象图，可以按照序列化顺序而划分为两个子对象图：A→B→D→F以及E→G→H→C。按照对象序列化顺序，也可以确定该多个子对象图之间的序列化顺序。

示例性的，上述根据对象图中第一对象关联的第二对象的序列化顺序，确定第二对象

的序列化编号可以包括：根据第一子对象图中的第一对象关联的第二对象的序列化顺序，确定第二对象的序列化编号；上述按照对象图中不同对象的序列化顺序，确定待序列化的至少部分对象可以包括：将第一子对象图中包含的至少一个对象作为待序列化的至少部分对象。

5 也即是第一子对象图中包括的所有对象全部作为待序列化的至少部分对象。

其中，该第一子对象图可以是指任意一个子对象图。该第一子对象图的确定可以有多种实现方式：在一种实现方式中，可以是响应于针对前一个第二子对象图的数据传输操作，确定待序列化的下一个第一子对象图。

该数据传输操作也即是指序列化数据传输至接收端的处理操作。

10 也即在前一个子对象图生成的序列化数据向接收端传输之后，按照序列化顺序，下一个子对象图也即为第一子对象图。

在又一种实现方式中，可以是将多个子对象图进行并行处理，从而任一个子对象图即可以作为第一子对象图。在并行处理的情况下，可以预先确定对象图中不同对象的序列化顺序。

15 在又一种实现方式中，响应于前一个第二对象图被确定为待序列化子对象图的确定时长大于预定时长，确定待序列化的下一个第一子对象图。

也即前一个子对象图被确定为待序列化的子对象图之后，经过预定时长，即可以对下一个子对象图继续进行处理。在该实现情况下，也可以预先确定对象图中不同对象的序列化顺序。

20 此外，本申请也不局限于上述几种实现方式，例如还可以是响应于针对前一个第二子对象图中的最后一个对象的编码完成操作，确定待序列化的下一个第一子对象图等。

结合上文描述可知，可以预先设定序列化数据的预定数据格式，该预定数据格式可以有多种实现方式，在该预定数据格式中可以包括每个对象对应的头部位置、指针域位置、值域位置、属性位置等。

25 因此，一些实施例中，基于至少部分对象的值域编码数据及指针域编码数据，生成至少部分对象的序列化数据可以包括：将至少部分对象分别对应的值域编码数据，写入预定数据格式中分别对应的值域位置；将至少部分对象分别对应的指针域编码数据，写入预定数据格式中分别对应的指针域位置；将至少部分对象分别对应的头部编码信息，写入预定数据格式中分别对应的头部位置；头部编码信息用以关联元数据；将至少部分对象对应的至少一个图属性信息，写入预定数据格式中的属性位置。

30 其中，预定数据格式可以有多种实现方式，每个对象的头部位置和值域位置可以依次排列而组合在一起；当然，每个对象的头部位置、值域位置以及指针域位置也可以组合在一起，值域位置和指针域位置可以和在内存中的布局方式保持一致，当然，也可以先排列值域位置再排列指针域位置而进行组合等。

而属性位置可以排列在预定数据格式中的最前面，之后可以按照序列化顺序依次排列每个对象；当然，属性位置也可以排列在预定数据格式的最后面等。

5 为了便于理解，图 4A、图 4B 和图 4C 示出了三种预定数据格式，以 ABC 三个对象为例，其中，H 表示头部位置、P 表示指针域位置、V 表示值域位置，T 表示数值位置。当然，本申请并不限于此。

10 此外，作为又一实施方式，一些实施例中，基于至少部分对象的值域编码数据及指针域编码数据，生成至少部分对象的序列化数据包括：将每个对象分别对应的值域编码数据以及头部编码数据进行组合，获得值域编码数组；将至少部分对象分别对应的指针域编码数据进行组合，获得指针域编码数组；将至少部分对象对应的至少一个图属性信息进行组合，获得属性编码数组；将值域编码数组、指针域编码数组以及属性编码数组作为序列化数据。

15 其中，每个对象的头部编码数据和值域编码数据可以排列组合而得到每个对象的目标编码数据，之后可以按照序列化顺序而将至少部分对象分别对应的目标编码数据而进行排列组合，形成值域编码数组。同样，对于指针域编码数据、以及至少一个图属性信息也可以分别组合，而获得数组。序列化数据也可以由三个数组构成，并传输至接收端等。

为了便于理解，仍以图 3 对应的对象图为例，图 4D 示出了该对象图对应的序列化数据的一种实现方式，包括值域编码数据、指针域数据以及属性编码数组。图中，classID 即表示头部编码信息，V 表示值域编码数据。

20 此外，对象图划分为 A→B→D→F，以及 E→G→H→C 的情况下，图 4E 示出了子对象 A→B→D→F 的序列化数据，包括对应的值域编码数据、指针域数据以及属性编码数组。以及图 4F 示出了子对象图 E→G→H→C 对应的序列化数据，包括对应的值域编码数据、指针域数据以及属性编码数组。

25 此外，由前文描述可知，序列化数据中可以包括基于每个对象的对象头部信息，而生成的头部编码信息（classID）。每个对象的头部编码信息可以关联每个对象的元数据，可以具体是与每个对象的元数据指针关联，而元数据指针可以指向保存的每个对象的元数据，每个对象的元数据中即可以包括对象头部信息以及值域、指针域分别对应的位置和类型等。如图 4G 所示，示出了头部编码信息与元数据的关联示意图。

30 序列化数据传输至接收端之后，接收端即可以进行反序列化处理，以还原对象图，如图 5 所示，为本申请实施例提供的一种对象处理方法又一个实施例的流程图，本实施例技术方案由接收端执行，从反序列化处理角度对本申请技术方案进行介绍，该方法可以包括以下几个步骤 501 至 504。

501：获取发送端传输的序列化数据。

其中，序列化数据可以是基于对象图中至少部分对象的值域编码数据以及指针域编码数据生成；指针域编码数据为所关联子对象的序列化编号。

该序列化数据的具体生成方式可以详见图 2 所示实施例中，此处不再赘述。

502：将序列化数据进行反序列化处理，以确定至少部分对象的内存地址，以及至少部分对象的值域编码数据分别对应的值域数据。

503：将至少部分对象分别对应的值域数据写入各自的内存地址中。

5 由上文描述可知，对象图可以对应生成至少一个序列化数据，接收端每接收一个序列化数据，即可对其进行反序列化处理，可以无需等待全部的序列化数据。

接收端进行反序列化处理，需要在内存中分配合适空间以保存对象，因此，可以确定该至少部分对象分别在在内存中的内存地址，此外还可以获得值域编码数据对应的值域数据。

10 此外，由于序列化数据中还可以包括每个对象的头部编码信息、每个对象的指针编码数据以及至少一个图属性信息，通过反序列化处理也可以获得头部编码信息、指针编码数据以及至少一个图属性信息。

根据头部编码信息可以获取每个对象的元数据，从而根据每个对象的元数据可以获知每个对象的值域以及指针域的位置和类型等，以及对象头部信息等。

15 从而，可以是根据每个对象的元数据，将每个对象的值域数据写入其对应内存地址中的值域位置处。同时，也可以将对象头部信息写入内地地址中的对象头部位置等。

504：结合对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址。

由于指针编码数据采用序列化编号实现，反序列化处理时，每个对象的指针域编码数据需要修改为其关联的子对象在内存中的内存地址，而在对象图分段序列化的场景下，由于一个序列化数据中的一个对象关联的子对象可能在下一个序列化数据中，下一个序列化数据可能还未传输过来，因此，子对象的内存地址可能还无法确定，因此，每个对象在内存地址中的指针域位置可以为空，当然也可以填充指针域编码数据。可以等待全部序列化数据接收完成，再结合该对象图的全部序列化数据，即可以确定任一个将任一对象的指针编码数据关联的子对象的内存地址，并将其作为指针域数据写入该的对象的内存地址，具体可以是写入内存地址中的指针域位置。而值域位置、指针域位置可以根据元数据而确定。

本实施例中，利用序列化顺序对应的序列化编号作为指针域编码数据，利用序列化顺序关联对象与子对象，相较于采用相对偏移量的方式，可以减少图遍历操作，降低操作复杂性，从而有助于降低端到端延迟，提高传输效率，且利用序列化顺序关联对象与子对象，可以将对象图进行分段处理，至少部分对象即可以进行序列化/反序列化，而无需等待全部对象处理完成，从而可以进一步的降低端到端延迟，提高传输效率。

30 一些实施例中，由于在对象图分段序列化的场景下，由于一个序列化数据中的一个对象关联的子对象可能在下一个序列化数据中，下一个序列化数据可能还未传输过来，因此，子对象的内存地址可能还无法确定。对于每个序列化数据可以记录指针编码数据、内存地

址以及偏移地址的映射关系；因此，该方法还可以包括：确定至少部分对象分别对应的指针编码数据、内存地址以及偏移地址的映射关系。

上述结合对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址包括：根据至少一个序列化数据分别对应的映射关系，确定任一对象的指针编码数据关联的子对象的内存地址；将子对象的内存地址作为指针域数据写入对象的内存地址。

其中，由于指针域编码数据对应序列化编号，每个序列化编号可以标识一个对象，因此，映射关系可以包括每个对象的序列化编号与其内存地址的第一对应关系，以及每个对象的内存地址，与其指针域编码数据，以及指针域编码数据指向的子对象相对该内存地址的偏移地址之间的第二对应关系。

从而结合不同序列化数据的第二对应关系，可以计算获得第一对应关系中每个对象的内存地址。对于一个序列化数据对应的偏移地址可能结合下一个序列化数据而确定，此时偏移地址可以为空，下一个序列化数据反序列化处理之后，即可以确定该偏移地址，进而可以计算获得内存地址。

为了便于理解，一些实施例中，确定至少部分对象分别对应的内存地址可以包括：针对至少部分对象中的第一对象，根据第一对象的内存地址，以及第一对象关联的第二对象相对第一对象的偏移地址，确定第二对象的内存地址。

其中，第一对象可以是指至少部分对象中的任一个对象，为了方便描述为命名为第一对象。第二对象即为第一对象的子对象。

上述确定至少部分对象分别对应的指针编码数据、对象地址以及偏移地址的映射关系可以包括：确定至少部分对象分别对应的序列化编号；根据第一对象的指针编码数据对应的序列化编号，确定指针编码数据、内存地址及偏移地址之间的映射关系。

其中，至少部分对象的序列化顺序及反序列化顺序相同，序列化数据由至少部分对象的编码数据按照各自的序列化顺序而生成，进行反序列化处理即可以按照同样的顺序进行，因此可以根据反序列顺序，来确定至少部分对象分别对应的序列化编号。

第一对象的序列化编号确定的情况下，序列化编号与第一对象的内存地址即可以建立第一对应关系。第一对象的指针编码数据指向的第二对象在内存中相对第一对象的偏移地址可以确定，第二对象的内存地址即等于第一对象的内存地址+第二对象的偏移地址，故此，可以建立指针编码数据对应的序列化编号、内存地址与偏移地址之间的第二对应关系。

为了方便记录和查找，上述第一对应关系以及第二对应关系可以记录在不同映射表中，因此，一些实施例中，根据第一对象的指针编码数据对应的序列化编号，确定指针编码数据、内存地址及偏移地址之间的映射关系包括：将第一对象的内存地址对应序列化编号保存至地址映射表中；将第一对象的指针编码数据对应的序列化编号、与第一对象的对象地址以及第二对象的偏移地址对应保存指针映射表中。

第一对象的指针编码数据对应的序列化编号即指向第二对象。

在一实施例中，可以预先根据至少部分对象对应的至少一个图属性信息中的对象数量而创建对应大小的地址映射表，根据指针域编码数据的数量而创建对应大小的指针映射表。

5 上述根据至少一个序列化数据分别对应的映射关系，确定任一对象的指针编码数据关联的子对象的内存地址包括：查找至少一个序列化数据分别对应的地址映射表以及指针映射表，确定任一对象的指针编码数据关联的子对象的内存地址。

10 一些实施例中，上述将序列化数据进行反序列化处理，以确定至少部分对象的内存地址，以及至少部分对象的值域编码数据分别对应的值域数据可以包括：将序列化数据进行反序列化处理，获得至少一个图属性信息、序列化数据包含的至少部分对象的值域编码数据分别对应的值域数据以及头部编码信息；按照至少一个图属性信息，为至少部分对象分配内存空间；查找至少部分对象的头部编码信息索引的元数据；基于元数据确定至少部分对象分别对应的内存数据格式。

15 则上述将至少部分对象分别对应的值域数据写入各自的内存地址中可以包括：按照内存数据格式，将至少部分对象分别对应的值域数据写入各自的内存地址的值域位置。

 上述结合对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址可以包括：结合对象图对应的至少一个序列化数据，按照内存数据格式，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址中指针域位置。

20 此外，头部编码信息索引的对象头部信息写入内存数据格式中的对象头部位置等。

 在一实施例中，可以是按照至少一个图属性信息中的图大小，为至少部分对象分配内存空间，之后可以按照至少部分对象分别对应的内存数据格式，以及至少部分对象的序列化顺序，确定在内存空间中至少部分对象分别对应的内存地址。

25 为了便于理解，仍以图 3 所示示意图对应的对象图为例，假设预定数量为 4，则按照对象图中不同对象的序列化顺序，在第一缓存中写入 A→B→D→F 之后，即可以基于 A→B→D→F 的值域编码数据和指针域编码数据而生成第一个序列化数据，如图 4E 中所示，之后，可以继续对 E→G→H→C 进行序列化处理，而生成图 4F 所示的第二个序列化数据。

30 接收端接收到第一个序列化数据之后，可以即对其进行反序列化处理，并创建地址映射表以及指针映射表，并将每个对象的内存地址对应序列化编号保存至地址映射表，将每个对象的指针编码数据对应的序列化编号、每个对象的内存地址以及每个对象的指针编码数据指向的子对象的偏移地址对应保存至指针映射表。

 接收端接收到第二个序列数据之后，可以执行相同操作以获得地址映射表以及指针映射表。

 下表 1 示出了第一个序列化数据对应的地址映射表的一种实现方式：

0	Addr-A
1	Addr-B
2	Addr-D
3	Addr-F

其中，Addr-A 表示对象 A 的内存地址，Addr-B 表示对象 B 的内存地址，Addr-D 表示对象 D 的内存地址，Addr-F 表示对象 F 的内存地址。

下表 2 示出了第一个序列化数据的指针映射表的一种实现方式：

Addr-A+offsetA1	1
Addr-A+offsetA2	2
Addr-B+offset B1	3
Addr-D+offset D1	4
Addr-F+offset F1	3

其中，offsetA1 表示对象 A 的第一个子对象（也即对象 B）的偏移地址，offsetA2 表示对象 A 的第二个子对象（也即对象 D）的偏移地址，offsetB1 表示对象 B 的第一个子对象的偏移地址；offseD1 表示对象 D 的第一个子对象的偏移地址；offseF1 表示对象 F 的第一个子对象的偏移地址。

下表 3 示出了第二个序列化数据对应的地址映射表的一种实现方式：

4	Addr-E
5	Addr-G
6	Addr-H
7	Addr-C

其中，Addr-E 表示对象 E 的内存地址，Addr-G 表示对象 G 的内存地址，Addr-H 表示对象 H 的内存地址，Addr-C 表示对象 C 的内存地址。

下表 4 示出了第二个序列化户数赌赢的指针映射表的一种实现方式：

Addr-E+ offset E1	1
Addr-E+ offset E2	5
Addr-G+ offset G1	5
Addr-G+ offset G2	7
Addr-H+ offset H1	1
Addr-H+ offset H2	5
Addr-H+ offset H3	0
Addr-C+ offset C1	5
Addr-C+ offset C2	6

其中，offsetE1 表示对象 A 的第一个子对象的偏移地址，offsetE2 表示对象 E 的第二个子对象的偏移地址；offsetG1 表示对象 G 的第一个子对象的偏移地址，offsetG2 表示对象 G 的第二个子对象的偏移地址；offsetH1 表示对象 H 的第一个子对象的偏移地址，offsetH2 表示对象 H 的第二个子对象的偏移地址；offsetH3 表示对象 H 的第三个子对象的偏移地址；offsetC1 表示对象 C 的第一个子对象的偏移地址，offsetC2 表示对象 C 的第二个子对象的偏移地址。

结合表 1~表 4 可知，对象 A 为第一个反序列化的对象，其对应的序列化编号可以为 0，其首先写入内存空间可以获知其对应的内存地址 (Addr-A)。对象 B 为对象 A 的子对象，其为对象 A 之后的下一个反序列化对象，对象 B 的序列化编号为 1，写入内存空间中，对象 B 相对对象 A 的偏移地址 (offsetA1) 可以确定。对象 B 的内存地址 (Addr-B) 等于对象 A 的内存地址 (Addr-A) 加上对象 B 的偏移地址 (offsetA1)。

而对象 E 的序列化编号 4，与对象 D 的内存地址 (Addr-D) 以及对象 D 的子对象的偏移地址 (offsetD1) 相对应，序列化编号 4 即标识该子对象的序列化顺序，对象 D 的内存地址 (Addr-D) 加上对象 D 的子对象的偏移地址 (offsetD1)，即为序列化编号 4 对应子对象的内存地址 (Addr-E)。

由于第一个序列化数据中不包括序列化编号 4 对应的对象 E，对象 E 的内存地址 (Addr-E) 以及偏移地址 (offsetD1) 无法确定，因此可以结合第二个序列化数据的反序列结果，确定对象 E 的内存地址 (Addr-D) 和偏移地址 (offsetD1)，进而可以填入表 2 以及表 3 中等。

最后，可以通过查询表 1~表 4，可以修改每个对象的指针域编码数据为对应序列化编号所对应的内存地址，并将内存地址写入每个对象对应的内存地址中的指针域位置，从而最终获得反序列化之后的对象图。

如图 6A 示出了反序列之后的对象图中一个对象的内存数据格式，该对象中用于指示值域以及指针域位置和类型的元数据假设采用位图“01100”表示，其中，0 代表该位置对应的是值域，1 代表该位置对应的是指针域，对应的内存数据格式即图 6 所示，在内存数据格式中，H 即代表对象头部位置，V 代表值域位置，P 代表指针域位置等。指针域位置写入的指针域数据即为关联子对象的内存地址，与发送端中的内存数据格式可以相同。

此外，如图 6B 示出了对象图分段序列化/反序列化的操作示意图，结合图 6B 可知，发送端基于部分对象的即可以进行序列化处理 60，从而生成一个序列化数据，每生成一个序列化数据即可以传输至接收端，接收端每接收到一个序列化数据，即可对其进行反序列化处理 61，而无需等待全部的序列化数据。最后，再结合接收到的全部序列化数据，对指针域数据进行更新 62，即可以还原获得对象图。

通过本申请实施例的技术方案，采用对象序列化/反序列化顺序来代替传统序列化编码中的相对偏移量来关联对象及其子对象，消除了反序列化必须等待序列化完全完成，并完

整接收到整个序列化缓冲才能开始反序列化的限制，实现了序列化、网络传输和反序列化的重叠实现，并消除了序列化过程中修改相对偏移的需求，从而降低了序列化/反序列化的端到端延迟，提高了处理效率，保证了用户体验。

本申请实施例中，发送端对于对象图中的任一个对象，也即第一对象，根据第一对象
5 关联的第二对象的序列化顺序，确定所述第二对象的序列化编号；将所述第二对象的序列化编号作为所述第一对象的指针域编码数据；按照所述对象图中不同对象的序列化顺序，确定待序列化的至少部分对象；基于所述至少部分对象的值域编码数据及指针域编码数据，生成所述至少部分对象的序列化数据；将所述序列化数据传输至接收端；所述接收端用以结合所述对象图对应的至少一个序列化数据的反序列化结果，获得所述对象图；通过
10 利用序列化顺序对应的序列化编号作为指针域编码数据，利用序列化顺序关联对象与子对象，相较于采用相对偏移量的方式，无需遍历全部对象以确定子对象的相对偏移量，因此可以减少图遍历操作，降低操作复杂性，从而有助于降低端到端延迟，提高传输效率，且利用序列化顺序关联对象与子对象，部分对象即可以进行序列化/反序列化，而无需等待全部对象处理完成，从而可以进一步降低端到端延迟，提高处理效率。

图 7 为本申请实施例提供的一种对象处理装置一个实施例的结构示意图，该装置可以包括：编号确定模块 701，用于根据对象图中第一对象关联的第二对象的序列化顺序，确定第二对象的序列化编号；指针编码模块 702，用于将第二对象的序列化编号作为第一对象的指针域编码数据；对象确定模块 703，用于按照对象图中不同对象的序列化顺序，确定对象图中待序列化的至少部分对象；数据生成模块 704，用于基于至少部分对象的值域
20 编码数据及指针域编码数据，生成至少部分对象的序列化数据；数据传输模块 705，用于将序列化数据传输至接收端；接收端用以结合对象图对应的至少一个序列化数据的反序列化结果，获得对象图。

一些实施例中，对象确定模块可以具体是按照对象图中不同对象的序列化顺序，将第一对象的值域编码数据及指针域编码数据写入第一缓存中；在第一缓存达到预定缓存大小
25 或者写入的对象数量达到预定数量，将已写入第一缓存中的对象作为待序列化的至少部分对象。

一些实施例中，该装置还可以包括：对象划分模块，用于按照对象图中不同对象的序列化顺序，结合预定缓存大小或预定对象数量，将对象图进行分段处理，获得多个子对象图；指针编码模块可以具体是根据第一子对象图中的第一对象关联的第二对象的序列化顺序，确定第二对象的序列化编号；对象确定模块可以具体是将第一子对象图中包含的至少
30 一个对象作为待序列化的至少部分对象。

一些实施例中，数据生成模块具体是：将至少部分对象分别对应的值域编码数据，写入预定数据格式中分别对应的值域位置；将至少部分对象分别对应的指针域编码数据，写入预定数据格式中分别对应的指针域位置；将至少部分对象分别对应的头部编码信息，写入
35 预定数据格式中分别对应的头部位置；头部编码信息用以关联元数据；将至少部分对象

对应的至少一个图属性信息，写入预定数据格式中的属性位置。

一些实施例中，数据生成模块可以具体是：将至少部分对象分别对应的值域编码数据以及头部编码信息进行组合，获得值域编码数组；将至少部分对象分别对应的指针域编码数据进行组合，获得指针域编码数组；将至少部分对象对应的至少一个图属性信息进行组合，获得属性编码数组；将值域编码数组、指针域编码数组以及属性编码数组作为序列化数据。

一些实施例中，该装置还可以包括：对象图确定模块，用于响应于针对前一个第二子对象图的数据传输操作，确定待序列化的下一个第一子对象图；或者，将多个子对象图进行并行处理，将任一个子对象图作为第一子对象图；或者，响应于前一个第二对象图被确定为待序列化子对象图的确定时长大于预定时长，确定待序列化的下一个第一子对象图。

图7所述的对象处理装置可以执行图2所示实施例所述的对象处理方法，其实现原理和技术效果不再赘述。对于上述实施例中的对象处理装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

图8为本申请实施例提供的一种对象处理装置又一个实施例的结构示意图，该装置可以包括：数据获取模块801，用于获取发送端传输的序列化数据；序列化数据基于对象图中至少部分对象的值域编码数据以及指针域编码数据生成；指针域编码数据为关联子对象的序列化编号；反序列化模块802，用于将序列化数据进行反序列化处理，以确定至少部分对象的内存地址，以及至少部分对象的值域编码数据分别对应的值域数据；数据写入模块803，用于将至少部分对象分别对应的值域数据写入各自的内存地址中；以及结合对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址。

一些实施例中，该装置还可以包括：关系记录模块，用于确定至少部分对象分别对应的指针编码数据、内存地址以及偏移地址的映射关系；数据写入模块可以具体是根据至少一个序列化数据分别对应的映射关系，确定任一对象的指针编码数据关联的子对象的内存地址；将子对象的内存地址作为指针域数据写入对象的内存地址。

一些实施例中，反序列化模块确定至少部分对象分别对应的内存地址可以包括：针对至少部分对象中的第一对象，根据第一对象的内存地址，以及第一对象关联的第二对象相对第一对象的偏移地址，确定第二对象的内存地址；关系记录模块可以具体是：确定至少部分对象分别对应的序列化编号；根据第一对象的指针编码数据对应的序列化编号，确定指针编码数据、内存地址及偏移地址之间的映射关系。

一些实施例中，关系记录模块根据第一对象的指针编码数据对应的序列化编号，确定指针编码数据、内存地址及偏移地址之间的映射关系包括：将第一对象的内存地址对应序列化编号保存至地址映射表中；将第一对象的指针编码数据对应的序列化编号、与第一对象的对象地址以及第二对象的偏移地址对应保存指针映射表中；数据写入模块根据至少一

个序列化数据分别对应的映射关系，确定任一对象的指针编码数据关联的子对象的内存地址包括：查找至少一个序列化数据分别对应的地址映射表以及指针映射表，确定任一对象的指针编码数据关联的子对象的内存地址。

一些实施例中，反序列化模块具体是按照至少一个图属性信息，为至少部分对象分配内存空间；查找至少部分对象的头部编码信息索引的元数据；基于元数据确定至少部分对象分别对应的内存数据格式；数据写入模块将至少部分对象分别对应的值域数据写入各自的内存地址中包括：按照内存数据格式，将至少部分对象分别对应的值域数据写入各自的内存地址的值域位置；数据写入模块结合对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址包括：结合对象图对应的至少一个序列化数据，按照内存数据格式，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入对象的内存地址中指针域位置。

图8所述的对象处理装置可以执行图5所示实施例所述的对象处理方法，其实现原理和技术效果不再赘述。对于上述实施例中的对象处理装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

本申请实施例还提供了一种计算设备，如图9所示，该计算设备可以包括存储组件901以及处理组件902；存储组件901存储一条或多条计算机指令，其中，一条或多条计算机指令供处理组件902调用执行，以实现如图2所示的对象处理方法或者如图5所示的对象处理方法。

当然，计算设备必然还可以包括其他部件，例如输入/输出接口、显示组件、通信组件等。

输入/输出接口为处理组件和外围接口模块之间提供接口，上述外围接口模块可以是输出设备、输入设备等。通信组件被配置为便于计算设备和其他设备之间有线或无线方式的通信等。

其中，处理组件可以包括一个或多个处理器来执行计算机指令，以完成上述的方法中的全部或部分步骤。当然处理组件也可以为一个或多个应用专用集成电路（ASIC）、数字信号处理器（DSP）、数字信号处理设备（DSPD）、可编程逻辑器件（PLD）、现场可编程门阵列（FPGA）、控制器、微控制器、微处理器或其他电子元件实现，用于执行上述方法。

存储组件被配置为存储各种类型的数据以支持在终端的操作。存储组件可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。

需要说明的是，上述计算设备可以为物理设备或者云计算平台提供的弹性计算主机等。其可以实现成多个服务器或终端设备组成的分布式集群，也可以实现成单个服务器或

单个终端设备。

上述计算设备也可以具体实现为电子设备，电子设备可以是指用户使用的，具有用户所需计算、上网、通信等功能的设备，例如可以是手机、平板电脑、个人电脑、穿戴设备等。

5 本申请实施例还提供了一种计算机可读存储介质，存储有计算机程序，计算机程序被计算机执行时可以实现上述图 2 所示实施例的对象处理方法或者上述图 5 所示实施例的对象处理方法。该计算机可读介质可以是上述实施例中描述的电子设备中所包含的；也可以是单独存在，而未装配入该电子设备中。

10 本申请实施例还提供了一种计算机程序产品，其包括承载在计算机可读存储介质上的计算机程序，计算机程序被计算机执行时可以实现如上述如图 2 所示实施例的对象处理方法或者上述图 5 所示实施例的对象处理方法。在这样的实施例中，计算机程序可以从网络上被下载和安装，和/或从可拆卸介质被安装。在该计算机程序被处理器执行时，执行本申请的系统中限定的各种功能。

15 需要说明的是，本申请实施例中可能会涉及到对用户数据的使用，在实际应用中，可以在符合所在国的适用法律法规要求的情况下（例如，用户明确同意，对用户切实通知，等），在适用法律法规允许的范围内在本文描述的方案中使用用户特定的个人数据。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

20 以上所描述的装置实施例仅仅是示意性的，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的劳动的情况下，即可以理解并实施。

25 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到各实施方式可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件。基于这样的理解，上述技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品可以存储在计算机可读存储介质中，如 ROM/RAM、磁碟、光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行各个实施例或者实施例的某些部分所述的方法。

30 最后应说明的是：以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

权 利 要 求

1. 一种对象处理方法，包括：
- 根据对象图中第一对象关联的第二对象的序列化顺序，确定所述第二对象的序列化编号；
- 5 将所述第二对象的序列化编号作为所述第一对象的指针域编码数据；
- 按照所述对象图中不同对象的序列化顺序，确定所述对象图中待序列化的至少部分对象；
- 基于所述至少部分对象的值域编码数据及指针域编码数据，生成所述至少部分对象的序列化数据；
- 10 将所述序列化数据传输至接收端；所述接收端用以结合所述对象图对应的至少一个序列化数据的反序列化结果，获得所述对象图。
2. 根据权利要求1所述的方法，其中，所述按照所述对象图中不同对象的序列化顺序，确定待序列化的至少部分对象包括：
- 按照所述对象图中不同对象的序列化顺序，将所述第一对象的值域编码数据及指针域
- 15 编码数据写入第一缓存中；
- 在所述第一缓存达到预定缓存大小或者写入的对象数量达到预定数量，将已写入所述第一缓存中的对象作为待序列化的至少部分对象。
3. 根据权利要求1所述的方法，其中，还包括：
- 按照所述对象图中不同对象的序列化顺序，结合预定缓存大小或预定对象数量，将所
- 20 述对象图进行分段处理，获得多个子对象图；
- 所述根据对象图中第一对象关联的第二对象的序列化顺序，确定所述第二对象的序列化编号包括：
- 根据第一子对象图中的第一对象关联的第二对象的序列化顺序，确定所述第二对象的序列化编号；
- 25 所述按照所述对象图中不同对象的序列化顺序，确定待序列化的至少部分对象包括：
- 将所述第一子对象图中包含的至少一个对象作为待序列化的至少部分对象。
4. 根据权利要求1所述的方法，其中，所述基于所述至少部分对象的值域编码数据及指针域编码数据，生成所述至少部分对象的序列化数据包括：
- 将所述至少部分对象分别对应的值域编码数据，写入预定数据格式中分别对应的值域
- 30 位置；
- 将所述至少部分对象分别对应的指针域编码数据，写入所述预定数据格式中分别对应的指针域位置；
- 将所述至少部分对象分别对应的头部编码信息，写入所述预定数据格式中分别对应的头部位置；所述头部编码信息用以关联元数据；
- 35 将所述至少部分对象对应的至少一个图属性信息，写入所述预定数据格式中的属性位置。
5. 根据权利要求1所述的方法，其中，所述基于所述至少部分对象的值域编码数据及指针域编码数据，生成所述至少部分对象的序列化数据包括：

将所述至少部分对象分别对应的值域编码数据以及头部编码信息进行组合，获得值域编码数组；

将所述至少部分对象分别对应的指针域编码数据进行组合，获得指针域编码数组；

将所述至少部分对象对应的至少一个图属性信息进行组合，获得属性编码数组；

5 将所述值域编码数组、所述指针域编码数组以及所述属性编码数组作为序列化数据。

6. 根据权利要求 3 所述的方法，其中，还包括：

响应于针对前一个第二子对象图的数据传输操作，确定待序列化的下一个第一子对象图；

或者，将所述多个子对象图进行并行处理，将任一个子对象图作为第一子对象图；

10 或者，响应于前一个第二对象图被确定为待序列化子对象图的确定时长大于预定时长，确定待序列化的下一个第一子对象图。

7. 一种对象处理方法，包括：

获取发送端传输的序列化数据；所述序列化数据基于对象图中至少部分对象的值域编码数据以及指针域编码数据生成；所述指针域编码数据为关联子对象的序列化编号；

15 将所述序列化数据进行反序列化处理，以确定所述至少部分对象的内存地址，以及所述至少部分对象的值域编码数据分别对应的值域数据；

将所述至少部分对象分别对应的值域数据写入各自的内存地址中；

结合所述对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入所述对象的内存地址。

20 8. 根据权利要求 7 所述的方法，其中，还包括：

确定所述至少部分对象分别对应的指针编码数据、内存地址以及偏移地址的映射关系；

所述结合所述对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入所述对象的内存地址包括：

25 根据所述至少一个序列化数据分别对应的映射关系，确定任一对象的指针编码数据关联的子对象的内存地址；

将所述子对象的内存地址作为指针域数据写入所述对象的内存地址。

9. 根据权利要求 8 所述的方法，其中，所述确定至少部分对象分别对应的内存地址包括：

30 针对所述至少部分对象中的第一对象，根据所述第一对象的内存地址，以及所述第一对象关联的第二对象相对所述第一对象的偏移地址，确定所述第二对象的内存地址；

所述确定所述至少部分对象分别对应的指针编码数据、对象地址以及偏移地址的映射关系包括：

确定所述至少部分对象分别对应的序列化编号；

35 根据所述第一对象的指针编码数据对应的序列化编号，确定所述指针编码数据、所述内存地址及所述偏移地址之间的映射关系。

10. 根据权利要求 9 所述的方法，其中，所述根据所述第一对象的指针编码数据对应的序列化编号，确定所述指针编码数据、所述内存地址及所述偏移地址之间的映射关系包

括：

将所述第一对象的内存地址对应序列化编号保存至地址映射表中；

将所述第一对象的指针编码数据对应的序列化编号、与所述第一对象的对象地址以及所述第二对象的偏移地址对应保存指针映射表中；

5 所述根据所述至少一个序列化数据分别对应的映射关系，确定任一对象的指针编码数据关联的子对象的内存地址包括：

查找所述至少一个序列化数据分别对应的地址映射表以及指针映射表，确定任一对象的指针编码数据关联的子对象的内存地址。

10 11. 根据权利要求 8 所述的方法，其中，所述将所述序列化数据进行反序列化处理，以确定所述至少部分对象的内存地址，以及所述至少部分对象的值域编码数据分别对应的值域数据包括：

将所述序列化数据进行反序列化处理，获得至少一个图属性信息、所述序列化数据包含的至少部分对象的值域编码数据分别对应的值域数据以及头部编码信息；

按照所述至少一个图属性信息，为所述至少部分对象分配内存空间；

15 查找所述至少部分对象的头部编码信息索引的元数据；

基于所述元数据确定所述至少部分对象分别对应的内存数据格式；

所述将所述至少部分对象分别对应的值域数据写入各自的内存地址中包括：

按照所述内存数据格式，将所述至少部分对象分别对应的值域数据写入各自的内存地址的值域位置；

20 所述结合所述对象图对应的至少一个序列化数据，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入所述对象的内存地址包括：

结合所述对象图对应的至少一个序列化数据，按照所述内存数据格式，将任一对象的指针编码数据关联的子对象的内存地址作为指针域数据写入所述对象的内存地址中指针域位置。

25 12. 一种计算设备，包括处理组件以及存储组件；所述存储组件存储一个或多个计算机指令；所述一个或多个计算机指令用以被所述处理组件调用执行，以实现如权利要求 1~6 任一项所述的对象处理方法或者实现如权利要求 7~11 任一项所述对象处理方法。

30 13. 一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时，实现如权利要求 1~6 任一项所述的对象处理方法或者实现如权利要求 7~11 任一项所述对象处理方法。