(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115811552 B
(45)授权公告日 2025.02.07

(21)申请号 202211537056.1

(22)申请日 2022.12.02

(65)同一申请的已公布的文献号
申请公布号 CN 115811552 A

(43)申请公布日 2023.03.17

(73)专利权人 珠海金山数字网络科技有限公司
地址 519000 广东省珠海市高新区唐家湾镇前岛环路325号102室、202室、302室、402室,327号102室、202室,329号302室

(72)发明人 叶帆

(74)专利代理机构 北京智信禾专利代理有限公司 11637
专利代理师 李东海

(51)Int.Cl.
H04L 69/04(2022.01)
H03M 7/30(2006.01)

(56)对比文件
CN 111587417 A,2020.08.25
CN 115145763 A,2022.10.04

审查员 侯停停

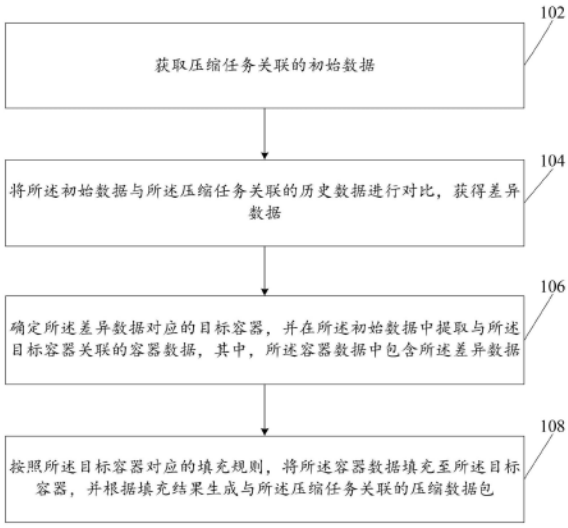
权利要求书3页 说明书23页 附图7页

(54)发明名称

数据压缩方法及装置

(57)摘要

本申请提供一种数据压缩方法及装置,其中所述数据压缩方法包括:获取压缩任务关联的初始数据;将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据;确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据;按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包。实现了将初始数据进行数据压缩,并通过数据压缩减少了空间占用,提升了后续对初始数据的处理效率。



CN 115811552 B

1.一种数据压缩方法,其特征在于,应用于服务端,包括:

获取压缩任务关联的初始数据;

将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据;

确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据;

按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包,其中,所述目标容器是根据所述差异数据从初始容器中确定,各个数据包对应的初始容器根据各个数据包中的目标历史子数据的数据属性信息创建,所述各个数据包中的目标历史子数据是基于与压缩任务关联的预设周期内的周期历史数据集合中周期历史数据的数据类型相同的目标历史数据中的目标历史子数据之间的关联权重对所述目标历史子数据进行分包得到。

2.根据权利要求1所述的数据压缩方法,其特征在于,所述填充规则的生成,包括:

将所述周期历史数据集合中相邻时间点的周期历史数据进行对比,获得各个周期历史数据中周期历史子数据的变化信息;

基于所述变化信息对所述周期历史数据集合对应的周期历史子数据进行关联性统计,获得关联统计信息;

根据所述关联统计信息对所述周期历史数据集合关联的目标历史数据进行分包处理,基于分包处理结果生成初始容器以及所述初始容器对应的填充规则。

3.根据权利要求2所述的数据压缩方法,其特征在于,所述根据所述关联统计信息对所述周期历史数据集合关联的目标历史数据进行分包处理,基于分包处理结果生成初始容器以及所述初始容器对应的填充规则,包括:

根据所述关联统计信息计算所述目标历史数据中目标历史子数据之间的关联权重;

基于所述关联权重,将所述目标历史数据中的目标历史子数据划分为至少一个数据包;

根据各个数据包中目标历史子数据的数据属性信息,生成各个数据包对应的初始容器以及所述初始容器对应的填充规则。

4.根据权利要求3所述的数据压缩方法,其特征在于,所述基于所述关联权重,将所述目标历史数据中的目标历史子数据划分为至少一个数据包,包括:

基于所述关联权重,对所述目标历史数据中目标历史子数据对应的数据节点进行连接,生成至少一个树集合;

根据所述关联统计信息对各个树集合进行评估,获得各个树集合对应的评估数据;

基于所述评估数据,在至少一个树集合中确定所述目标历史数据对应的目标树集合;

将所述目标树集合中各个连接子树的数据节点在所述目标历史数据中对应的节点子数据分别进行封装,生成至少一个数据包。

5.根据权利要求4所述的数据压缩方法,其特征在于,所述至少一个树集合中任意一个树集合对应的评估数据的确定,包括:

根据树集合中各个连接子树中数据节点对应的节点子数据的存储信息,对各个连接子树对应的节点子数据进行拼接处理,生成各个连接子树对应的拼接子数据;

根据所述拼接子数据的拼接存储信息以及所述关联统计信息进行评估计算,获得树集

合对应的评估数据。

6.根据权利要求5所述的数据压缩方法,其特征在于,所述根据树集合中各个连接子树中数据节点对应的节点子数据的存储信息,对各个连接子树对应的节点子数据进行拼接处理,生成各个连接子树对应的拼接子数据,包括:

获取预设对齐长度以及各个连接子树对应的节点子数据的存储位长度;

遍历各个连接子树对应的节点子数据、所述存储位长度以及预设对齐长度,将各个连接子树对应的节点子数据进行位域拼接,生成各个连接子树对应的拼接子数据。

7.根据权利要求4所述的数据压缩方法,其特征在于,所述基于所述关联权重,对所述目标历史数据中目标历史子数据对应的数据节点进行连接,生成至少一个树集合,包括:

基于所述关联权重以及第i节点连接关系,对所述目标历史数据中目标历史子数据对应的数据节点进行第i连接,生成第i初始树集合;

判断所述第i初始树集合中连接子树的数量是否大于预设阈值;

若大于,将所述第i初始树集合作为中间树集合添加至树集合组;确定所述第i初始树集合中数据节点对应的目标连接关系,i自增1,通过所述目标连接关系作为所述第i节点连接关系,并返回执行所述基于所述关联权重以及第i节点连接关系,对所述目标历史数据中目标历史子数据对应的数据节点进行第i连接,生成第i初始树集合;

若等于,将所述树集合组中的中间树集合组成至少一个树集合。

8.根据权利要求3所述的数据压缩方法,其特征在于,任意一个初始容器以及任意一个初始容器对应的填充规则的生成,包括:

根据数据包中目标历史子数据的数据属性信息,确定数据包的包属性信息;

根据所述包属性信息,创建数据包对应的初始容器;

根据数据包中目标历史子数据的存储位长度以及拼接顺序,确定各个目标历史子数据在初始容器中的填充位置信息;

基于所述填充位置信息,创建初始容器对应的填充规则。

9.根据权利要求1所述的数据压缩方法,其特征在于,所述根据填充结果生成与所述压缩任务关联的压缩数据包之后,还包括:

确定所述压缩任务关联的至少两个压缩数据包;

根据预设编码顺序以及至少两个压缩数据包对应的编码标识,对至少两个压缩数据包进行组合,生成组合压缩数据包;

根据预设编码顺序以及所述编码标识,生成所述组合压缩数据包对应的数据包掩码;

基于所述数据包掩码和所述组合压缩数据包,生成压缩数据总包;

将所述压缩数据总包发送至客户端。

10.根据权利要求1所述的数据压缩方法,其特征在于,所述确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,包括:

根据所述初始数据的类型信息,确定所述初始数据对应的容器集合;

基于所述差异数据的数据属性信息在所述容器集合中选择目标容器;

根据所述目标容器的关联数据信息,在所述初始数据中提取所述容器数据。

11.一种数据解压方法,其特征在于,应用于客户端,包括:

获取与解压任务关联的压缩数据包,其中,所述压缩数据包根据权利要求1所述的数据

压缩方法确定；

根据所述压缩数据包对应的目标容器的提取规则，从所述压缩数据包中提取所述目标容器的容器数据；

根据所述容器数据的数据属性信息，确定所述容器数据关联的历史数据；

基于所述容器数据和所述历史数据，生成目标数据。

12.根据权利要求11所述的数据解压方法，其特征在于，所述获取与解压任务关联的压缩数据包，包括：

接收服务端发送的压缩数据总包；

解析所述压缩数据总包，获得数据包掩码以及组合压缩数据压缩包；

基于所述数据包掩码，在所述组合压缩数据压缩包中提取至少一个压缩数据包，其中，至少一个压缩数据包中包括所述压缩数据包。

13.一种数据压缩装置，其特征在于，应用于服务端，包括：

获取数据模块，被配置为获取压缩任务关联的初始数据；

对比模块，被配置为将所述初始数据与所述压缩任务关联的历史数据进行对比，获得差异数据；

提取模块，被配置为确定所述差异数据对应的目标容器，并在所述初始数据中提取与所述目标容器关联的容器数据，其中，所述容器数据中包含所述差异数据；

生成模块，被配置为按照所述目标容器对应的填充规则，将所述容器数据填充至所述目标容器，并根据填充结果生成与所述压缩任务关联的压缩数据包，其中，所述目标容器是根据所述差异数据从初始容器中确定，各个数据包对应的初始容器根据各个数据包中的目标历史子数据的数据属性信息创建，所述各个数据包中的目标历史子数据是基于与压缩任务关联的预设周期内的周期历史数据集合中周期历史数据的数据类型相同的目标历史数据中的目标历史子数据之间的关联权重对所述目标历史子数据进行分包得到。

14.一种数据解压装置，其特征在于，应用于客户端，包括：

获取包模块，被配置为获取与解压任务关联的压缩数据包，其中，所述压缩数据包根据权利要求1所述的数据压缩方法确定；

提取数据模块，被配置为根据所述压缩数据包对应的目标容器的提取规则，从所述压缩数据包中提取所述目标容器的容器数据；

确定数据模块，被配置为根据所述容器数据的数据属性信息，确定所述容器数据关联的历史数据；

生成数据模块，被配置为基于所述容器数据和所述历史数据，生成目标数据。

15.一种计算设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机指令，其特征在于，所述处理器执行所述计算机指令时实现权利要求1-10或者11-12任意一项所述方法的步骤。

16.一种计算机可读存储介质，其存储有计算机指令，其特征在于，所述计算机指令被处理器执行时实现权利要求1-10或者11-12任意一项所述方法的步骤。

17.一种计算机程序产品，其特征在于，包括计算机指令，该计算机指令被处理器执行时实现权利要求1-10或者11-12任意一项所述方法的步骤。

数据压缩方法及装置

技术领域

[0001] 本申请涉及数据处理技术领域,特别涉及一种数据压缩方法。本申请同时涉及一种数据压缩装置,一种数据解压方法,一种数据解压装置,一种计算设备,以及一种计算机可读存储介质。

背景技术

[0002] 随着互联网技术的发展,各个企业每日所需要处理的处理量越来越大,为了能达到较高的数据处理效率,通常会采用数据压缩技术对待处理数据进行压缩。其中,数据压缩是指在不丢失有用信息的前提下,缩减数据量以减少存储空间,提高其传输、存储和处理效率,或按照预设算法对数据进行重新组织,减少数据的冗余和存储的空间的一种技术方法。

[0003] 现有技术中,在进行数据压缩时,经常会出现压缩效果不理想的情况,并影响后续对数据的处理效率。因此,亟需提供一种解决上述问题的方案。

发明内容

[0004] 有鉴于此,本申请实施例提供了一种数据压缩方法,以解决现有技术中存在的技术缺陷。本申请实施例同时提供了一种数据压缩装置,一种数据解压方法,一种数据解压装置,一种计算设备,以及一种计算机可读存储介质。

[0005] 根据本申请实施例的第一方面,提供了一种数据压缩方法,应用于服务端,包括:

[0006] 获取压缩任务关联的初始数据;

[0007] 将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据;

[0008] 确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据;

[0009] 按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包。

[0010] 根据本申请实施例的第二方面,提供了一种数据压缩装置,应用于服务端,包括:

[0011] 获取数据模块,被配置为获取压缩任务关联的初始数据;

[0012] 对比模块,被配置为将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据;

[0013] 提取模块,被配置为确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据;

[0014] 生成模块,被配置为按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包。

[0015] 根据本申请实施例的第三方面,提供了一种数据解压方法,应用于客户端,包括:

[0016] 获取与解压任务关联的压缩数据包;

[0017] 根据所述压缩数据包对应的目标容器的提取规则,从所述压缩数据包中提取所述目标容器的容器数据;

[0018] 根据所述容器数据的属性信息,确定所述容器数据关联的历史数据;

[0019] 基于所述容器数据和所述历史数据,生成目标数据。

[0020] 根据本申请实施例的第四方面,提供了一种数据解压装置,应用于客户端,包括:

[0021] 获取包模块,被配置为获取与解压任务关联的压缩数据包;

[0022] 提取数据模块,被配置为根据所述压缩数据包对应的目标容器的提取规则,从所述压缩数据包中提取所述目标容器的容器数据;

[0023] 确定数据模块,被配置为根据所述容器数据的属性信息,确定所述容器数据关联的历史数据;

[0024] 生成数据模块,被配置为基于所述容器数据和所述历史数据,生成目标数据。

[0025] 根据本申请实施例的第五方面,提供了一种计算设备,包括:

[0026] 存储器和处理器;

[0027] 所述存储器用于存储计算机可执行指令,所述处理器执行所述计算机可执行指令时实现所述数据压缩方法或所述数据解压方法的步骤。

[0028] 根据本申请实施例的第六方面,提供了一种计算机可读存储介质,其存储有计算机可执行指令,该指令被处理器执行时实现所述数据压缩方法或所述数据解压方法的步骤。

[0029] 根据本申请实施例的第七方面,提供了一种芯片,其存储有计算机程序,该计算机程序被芯片执行时实现所述数据压缩方法或所述数据解压方法的步骤。

[0030] 本申请提供的数据压缩方法,通过获取压缩任务关联的初始数据,并将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据,即获得初始数据与历史数据相比发生变化的数据,再确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据,实现了通过差异数据对应的目标容器,提取初始数据中与差异数据强相关的数据。再按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包,以便将提取的数据进行组装生成待处理的压缩数据包,实现了将初始数据进行数据压缩,并通过数据压缩减少了空间占用,提升了后续对初始数据的处理效率。

附图说明

[0031] 图1是本申请一实施例提供的一种数据压缩方法的流程图;

[0032] 图2是本申请一实施例提供的一种数据压缩方法中统计的关联变化频次的示意图;

[0033] 图3是本申请一实施例提供的一种数据压缩方法中的节点子数据的示意图;

[0034] 图4是本申请一实施例提供的一种数据压缩方法中执行过程的示意图;

[0035] 图5是本申请一实施例提供的一种数据解压方法的流程图;

[0036] 图6是本申请一实施例提供的一种应用于游戏场景的数据压缩方法的处理流程图;

[0037] 图7是本申请一实施例提供的一种数据压缩装置的结构示意图;

[0038] 图8是本申请一实施例提供的一种数据解压装置的结构示意图;

[0039] 图9是本申请一实施例提供的一种计算设备的结构框图。

具体实施方式

[0040] 在下面的描述中阐述了很多具体细节以便于充分理解本申请。但是本申请能够以很多不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本申请内涵的情况下做类似推广，因此本申请不受下面公开的具体实施的限制。

[0041] 在本申请一个或多个实施例中使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本申请一个或多个实施例。在本申请一个或多个实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。还应当理解，本申请一个或多个实施例中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0042] 应当理解，尽管在本申请一个或多个实施例中可能采用术语第一、第二等来描述各种信息，但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如，在不脱离本申请一个或多个实施例范围的情况下，第一也可以被称为第二，类似地，第二也可以被称为第一。

[0043] 在本申请中，提供了一种数据压缩方法。本申请实施例同时提供了一种数据压缩装置，一种数据解压方法，一种数据解压装置，一种计算设备，以及一种计算机可读存储介质。在下面的实施例中逐一进行详细说明。

[0044] 在网络游戏中具有大量的需要广播的流量数据，比如基于状态同步的大型多人在线角色扮演游戏，无可避免的会涉及到很多数据需要广播给同一场景里的各个客户端。

[0045] 实际应用中，有一些游戏数据可以根据客户端关注度进行裁剪。当游戏中同一场景下玩家太多，比如游戏攻防战中，涉及两个阵营400对400人(一共800人)在同一场景下的战斗，会产生巨量的状态同步数据(会有N*N的同步量，N为在场玩家人数)。这其中一些数据可以根据玩家关注度做到裁剪，比如释放技能，如果不是直接互相攻击的两人，或者不是同一个战队的玩家，产生的战斗技能数据是可以裁剪掉的。但是有一些游戏数据是不能进行裁剪的。比如玩家移动的数据，因为每一个玩家必须看清在场的其他玩家的位置和将要移动的方向、以及将要做出的动作，因此，这些数据不能被裁剪。

[0046] 现有技术中，当一个区域内的玩家达到比较高的数量，在广播时可以进行降频和细节丢弃。其中，降频指的是广播的频率降低，通常情况下广播是以每秒16帧的精度同步人物移动数据，当人数密度到达附近超过50人以上，就降低到每2帧同步一次人物移动数据。这样做的确可以有效降低流量，但是带来的负面效果也很明显，人物会出现瞬移、动作不连续、看起来像卡顿的情况。丢弃细节指的是对一些不是很重要的变量不再同步，比如关掉移动时的走、跑、跳跃状态，统一显示成跑步状态等，但这样做会导致人物的移动很单调，而且失真。如果玩家在一些复杂地形上(比如一个斜坡上)进行了跳跃，那动作和位置就是严重失真的，会导致属性方瞄准失败。所以还是要尽量保证数据的完整性，而且不要做太大的降频。

[0047] 此外，现有技术中还存在对发送的数据进行压缩场景，比如一个移动数据包里包含大约40个变量，这些变量的变化相互之间其实是有一定关联规律的。但是通常的压缩算法都是基于大量的空值(值为0)，或者有大量的重复数据(比如字符串文本)的场景通过对

数据去除冗余的方式进行压缩的。这种压缩方式对于移动数据包中这些数据的支持并不好。因为,通常移动数据中变量的变量长度各不相同,彼此之间没有太大可比性,并且移动数据中不会有大量的空值,也没有简单的重复规律可循,压缩效果不好。

[0048] 图1出了根据本申请一实施例提供的一种数据压缩方法的流程图,具体包括以下步骤:

[0049] 步骤102:获取压缩任务关联的初始数据。

[0050] 具体的,压缩任务,是指用以对初始数据进行数据压缩的任务。实际应用中,由于初始数据存在被传输、存储和处理的需求,直接对初始数据进行这些处理,需要耗费大量的计算成本以及时间成本。因此,通过对初始数据进行数据压缩可以减少初始数据的存储空间,提高对其进行传输、存储和处理的效率。相应的,初始数据,是指待被压缩的数据。例如:游戏数据、交易数据、用户数据等。本实施例为方便描述,以游戏数据为例进行说明,其他数据的压缩均可参见本实施例相同或相应的描述内容,本实施例在此不作过多赘述。

[0051] 基于此,在确定存在具有压缩需求的初始数据的情况下,即可创建压缩任务对其进行压缩处理。具体实施时,获取初始数据,可以从特定的存储路径获取,也可以从其他数据平台或数据库获取,在此不做限制。

[0052] 例如:获取游戏场景中与压缩任务关联的待传输的属性数据D1。

[0053] 步骤104:将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据。

[0054] 具体的,在上述获取压缩任务关联的初始数据的基础上,考虑到初始数据中可能存在一部分数据已经被处理(比如存储或传输等)过,为了避免对这些数据进行重复处理,可以从初始数据中先识别出未被处理过的数据,即差异数据,以便基于差异数据对初始数据进行压缩。其中,历史数据,是指与初始数据的类型相同的已被处理过的数据。实际应用中,由于通常相邻时间点创建的数据之间的差异较小,因此,为了进一步减少数据处理量,通常将初始数据在相邻时间点对应的上一时间点的数据作为历史数据。此外,还可以确定其他任意时间的历史数据,在此不做限制。相应的,差异数据,是指初始数据中存在的与历史数据相比存在不同的数据。

[0055] 基于此,在获取初始数据之后,将初始数据与历史数据进行对比,即可获得差异数据。具体实施时,可以将初始数据和历史数据中相同名称的数据的数据值进行比对,若不一致,表明该数据为差异数据;若一致,表明该数据不为差异数据。此外,还可以将初始数据中的数据和历史数据中的数据按照数据排列顺序进行一一对比,从而获得差异数据,在此不做限制。

[0056] 沿用上例,在获取属性数据D1的基础上,属性数据D1中包含40个属性变量,服务器上一次传输的属性数据D2中也包含40个属性变量,将二者中相同名称的属性变量的变量值进行一一对比,确定属性数据D1中存在5个属性变量的变量值与属性数据D2不同,则将这5个属性变量作为差异数据。

[0057] 步骤106:确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据。

[0058] 具体的,在获得差异数据之后,考虑到在初始数据的数据数量较多的情况下,如果直接将差异数据作为初始数据的压缩结果,需要掩码数据对该差异数据进行标记,以便可

以将差异数据还原回初始数据。但在数据数量大的情况下，掩码数据需要占用大量存储空间和/或传输带宽。因此，本实施例中预先建立初始数据中数据与各个初始容器之间的对应关系。在获得差异数据后，即可确定差异数据在初始容器中对应的目标容器。

[0059] 其中，初始容器，是指预先建立的用以对数据进行填充的容器。该初始容器可以理解为一个虚拟对象，其存在相应的容器信息，比如容器标识、容器序号、容器大小等信息。目标容器，是指初始容器中与差异数据对应的容器。容器数据，是指允许被填充至目标容器的数据。实际应用中，每个初始容器都对应初始数据的一部分数据。这部分数据彼此存在相似性。因此初始容器也表明对初始数据的一种划分。通常将初始数据中经常一起变化的数据归为一类对应一个初始容器，将不存在关联变化的数据归为一类对应另一个初始容器，此外，还可以将变化频次极低的数据归为一类也对应一个初始容器。

[0060] 以初始数据为移动数据为例，移动数据包内包含40个移动变量，这40个移动变量的变化频次并不相同。其中，一些移动变量通常一起变化，比如角色面向和奔跑的目标点水平坐标，因为角色在跑动过程中转向了，目标点水平坐标也会发生变化；纵向速度和跳跃计数器通常一起发生变化，因为角色跳跃会有纵向速度的改变。还有一些移动变量的变化是相斥的，比如被锁足标志位和坐标变化二者之间就是相斥的。另外，还有一些移动变量之间明显是毫不相关的，比如最大跑速和水中悬浮标志位。此外，还有一些移动变量的变化频次非常低，比如：被锁住移动的标志位只有在角色被定身的时候才会触发，远远低于走动时左右变化的频率。其中，经常在一起变化的移动变量可以称作强关联变量，反过来就是弱关联变量、完全无关变量。因此，根据这些移动变量的变化频次的关联性的强弱关系，可以将移动变量划分为不同的类型（强关联类型、弱关联类型、无关联类型），并将不同类型的移动变量对应不同的初始容器。

[0061] 基于此，可以根据预先建立的数据信息（比如数据标识、数据场景等信息）和初始容器的容器信息（比如容器标识、容器名称等信息）之间的对应关系，在初始容器中确定差异数据对应的目标容器。由于差异数据可能存在多个，而不同的差异数据可能对应不同的初始容器。因此，目标容器的数量也可以为一个或多个。在确定目标容器的基础上，再根据上述对应关系，即可在初始数据中确定目标容器对应的全部数据，即为目标容器关联的容器数据。

[0062] 进一步的，考虑到初始数据中数据的数量和初始容器的数量可能都比较庞大，为了避免在确定二者对应关系的过程耗费大量的时间，可以预先查找初始数据整体对应的容器集合，再从初始容器中确定目标容器关联的容器数据，具体实现如下：

[0063] 根据所述初始数据的类型信息，确定所述初始数据对应的容器集合；基于所述差异数据的数据属性信息在所述容器集合中选择目标容器；根据所述目标容器的关联数据信息，在所述初始数据中提取所述容器数据。

[0064] 具体的，类型信息，是指用以表示初始数据的数据类型的信息，该类型信息可以是类型名称或类型标识等信息。容器集合，是指初始数据对应的初始容器的集合。数据属性信息，是指与数据相关的信息，比如数据名称、数据标识等信息。关联数据信息，是指目标容器所关联的数据相关的信息，该关联数据信息也可以是数据名称或数据标识等信息。

[0065] 基于此，则可以根据预先定义的类型信息和容器集合之间的对应关系以及初始数据的类型信息，确定初始数据对应的容器集合；并根据预先定义的数据属性信息和初始容

器之间的对应关系以及差异数据的数据属性信息,在容器集合中确定差异数据对应的目标容器;再根据目标容器的关联数据信息与初始数据之间的关系,在初始数据中提取容器数据。

[0066] 沿用上例,在上述获得的差异数据为5个属性变量基础上,根据属性数据D1的数据类型:属性类型,确定属性类型对应的容器集合为C。这5个属性变量的变量名称为:d1,d3,d13,d22,d29,根据这5个属性变量的变量名称和初始容器的容器标识之间的对应关系,从容器集合C中确定差异数据中3个属性变量(d1,d3,d29)对应的目标容器为:目标容器c1,差异数据中另外2个属性变量(d13,d22)对应的目标容器为:目标容器c2。再根据目标容器c1关联的变量名称:d1,d3,d5,d16以及d29,在属性数据D1中提取变量名称为:d1,d3,d5,d16以及d29的这5个属性变量(这5个属性变量中包括上述差异数据中的3个属性变量d1,d3,d29)。并根据目标容器c2关联的变量名称:d7,d13,d22,d38,在属性数据D1中提取变量名称为:d7,d13,d22,d38的这4个属性变量(这4个属性变量中包括上述差异数据中的2个属性变量d13,d22)。

[0067] 综上,通过先确定初始数据对应的容器集合,再在容器集合中确定差异数据对应的目标容器,避免了在大量初始容器中筛选目标容器,提升了确定目标容器的效率。

[0068] 步骤108:按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包。

[0069] 具体的,在上述提取目标容器关联的容器数据的基础上,考虑到若直接将容器数据随意填充至目标容器,容器数据仍然会占用较多存储空间。因此,可以将容器数据按照预先设定的规则(即填充规则)进行填充,从而进一步达到数据压缩的目的。其中,填充规则,是指将数据填充至初始容器的规则。该填充规则中可以包括填充数据信息、填充顺序、填充位置、填充优先级和/或填充策略等信息。相应的,填充结果,是指将容器数据填充至目标容器后生成的数据。

[0070] 基于此,将获得的容器数据按照目标容器对应的填充规则进行填充,可以理解为将容器数据按照填充规则进行压缩。具体实施时,填充可以通过包括对数据的拼接、和/或裁剪等操作实现。

[0071] 沿用上例,在上述属性数据D1中提取变量名称为:d1,d3,d5,d16以及d29的这5个属性变量的基础上,将这5个属性变量按照对应的变量长度(即存储位长度)进行裁剪,并将裁剪后的数据按照预设填充顺序进行添加至目标容器c1,再将目标容器c1中的拼接数据进行封装生成数据压缩包。

[0072] 考虑到如果随意设置初始数据和初始容器之间的对应关系,将导致生成的压缩数据包的压缩效果不明显,因此,为了增强初始数据的压缩效果,可以根据数据之间的不同强度的关联性创建初始容器,具体实现包括如下步骤:

[0073] 步骤1082:获取与所述压缩任务关联的预设周期内的周期历史数据集合。

[0074] 具体的,预设周期,是指预先设置的时间周期,该预设周期可以是5小时、1天或3天等。相应的,周期历史数据集合,是指在预设周期内生成或获得的历史数据(即周期历史数据)的集合。该周期历史数据,是指预设周期内的历史数据。

[0075] 基于此,获取与压缩任务关联的周期历史数据集合的具体获取方式与上述获取初始数据的获取方式类似,参考上述获取初始数据的具体实现方式即可,在此不做赘述。

[0076] 比如:获取1天(预设周期)内的历史属性数据集合(周期历史数据集合)。

[0077] 步骤1084:将所述周期历史数据集合中相邻时间点的周期历史数据进行对比,获得各个周期历史数据中周期历史子数据的变化信息。

[0078] 具体的,相邻时间点,是指周期历史数据对应的生成或获得的时间点在该预设周期内是相邻的,比如:预设周期内存在3个周期历史数据,这3个周期历史数据对应的时间为2点,4点,6点,则2点和4点对应的周期历史数据为相邻时间点的周期历史数据;4点和6点对应的周期历史数据为相邻时间点的周期历史数据。

[0079] 基于此,将相邻时间点的周期历史数据进行对比,通常是将相邻时间点中后一时间点的周期历史数据与前一时间点的周期历史数据进行对比,进而获得每个相邻时间点中后一时间点的周期历史数据中周期历史子数据相比前一时间点的周期历史数据中周期历史子数据的变化信息。其中,周期历史子数据,是指周期历史数据中包含的子数据。比如周期历史数据为5个变量组成的数据,则每个变量都可以称为周期历史子数据。

[0080] 相应的,变化信息,为数据变化相关的信息,该变化信息可以包括:变化的周期历史子数据的数据名称或数据标识、变化次数、没有变化的周期历史子数据的数据名称或数据标识等。比如:6点对应的周期历史数据中包含的40个周期历史子数据,相比上一时间点4点对应的周期历史数据,有3个周期历史子数据(数据标识为:dn4,dn20,dn23)发生了变化。则针对6点对应的40个周期历史子数据获得的变化信息为:dn4和dn20关联变化频次(即一同变化的次数)记录为1,dn4和dn23关联变化频次记录为1,dn20和dn23关联变化频次记录为1。

[0081] 需要说明的是,由于预设周期内中最早时间点的周期历史数据在预设周期内不存在前一时间点的周期历史数据。因此,可以另外获取预设周期外的前一时间点的历史数据用以进行对比,或者直接将最早时间点的周期历史数据中周期历史子数据的变化信息设为空,在此不做限制。

[0082] 步骤1086:基于所述变化信息对所述周期历史数据集合对应的周期历史子数据进行关联性统计,获得关联统计信息。

[0083] 在获得变化信息的基础上,基于获得的变化信息对周期历史子数据进行关联性统计。具体的,关联性统计可以是统计预设周期内周期历史数据中一同发生变化的周期历史子数据的变化次数,和/或统计预设周期内周期历史数据中没有一同发生变化的周期历史子数据的次数等。此外,关联性统计还可以是统计变化的周期历史子数据的数据标识,变化时间等信息。因此,关联统计信息可以包括上述变化次数,数据标识,和/或变化时间等信息。该关联统计信息还可以采用表格或图表的形式进行记录。

[0084] 沿用上例,在上述获取历史属性数据集合的基础上,该历史属性数据集合中包括200个历史属性数据,每个历史属性数据都包括40个历史属性子数据。将这200个历史属性数据中相邻两个时间点的历史属性数据进行对比,获得每个历史属性数据对应的变化信息,其中,历史属性数据D2对应的变化信息为:d4(历史属性子数据)和d20的关联变化频次为1,d4和d23的关联变化频次为1,d20和d23的关联变化频次为1。再对这200个历史属性数据的变化信息进行统计,获得的关联统计信息包括:d1和d20的关联变化频次为110次,d2和d23的关联变化频次为180次,……d20和d23的关联变化频次为20次,该关联统计信息还可以采用表格或图表的形式进行记录,具体的如图2所示。

[0085] 步骤1088:根据所述关联统计信息对所述周期历史数据集合关联的目标历史数据进行分包处理,基于分包处理结果生成初始容器以及所述初始容器对应的填充规则。

[0086] 具体的,目标历史数据,是指待进行分包处理的历史数据。该目标历史数据与周期历史数据集合的关联性可以是二者的数据类型相同,和/或二者的数据结构相同。具体实施时,该目标历史数据也可以是周期历史数据中的任意一个,在此不做限制。

[0087] 相应的,对目标历史数据进行分包处理,是指将目标历史数据中的子数据(即目标历史子数据)划分为不同的数据包。相应的,分包处理结果是指划分目标历史数据后获得的数据包。

[0088] 基于此,根据所述关联统计信息对所述周期历史数据集合关联的目标历史数据进行分包处理的方式是多种多样的,比如,将关联统计信息中关联变化频次划分频次区间,并将各个频次区间内的关联变化频次对应的目标历史子数据划分为一个数据包。或者将相同关联变化频次对应的目标历史子数据划分为一个数据包。

[0089] 进一步的,由于不同分包处理方式对压缩结果的影响差别较大,为了保障分包处理结果可以到达理想的压缩目标。还可以先计算目标历史子数据之间的关联权重,再基于关联权重进行分包处理,具体实现包括如下步骤:

[0090] 步骤10882:确定与所述周期历史数据集合中周期历史数据的数据类型相同的目标历史数据。

[0091] 具体的,数据类型是指周期历史数据所属的类型,该数据类型可以是订单类型、移动类型、攻击类型、皮肤类型、场景类型等,在此不做限制。

[0092] 步骤10884:根据所述关联统计信息计算所述目标历史数据中目标历史子数据之间的关联权重。

[0093] 具体的,关联权重,是指表示目标历史子数据之间的关联关系强弱的数据。该关联权重可以通过预设权重公式进行计算。该预设权重公式,可以是将两个变量一起发生变化的次数除以二者发生变化次数之和的方式进行计算,具体如下所示:

[0094] nWeight=nSameChange/(nAOcc+nBOcc);

[0095] 该预设权重公式还可以如下所示:

[0096] nWeight=nSameChange*min(nASize,nBSize)-min((nAOcc-nSameChange)*nBSize,(nBOcc-nSameChange)*nASize);其中,nSameChange:表示目标历史子数据A以及目标历史子数据B两个目标历史子数据一起变化的次数,nAOcc:表示目标历史子数据A发生变化的次数;nBOcc:表示目标历史子数据B发生变化的次数;nASize:表示目标历史子数据A的占用空间大小;nBSize:表示目标历史子数据B的占用空间大小。该预设权重公式的含义是通过有效通信量减去无效通信量,即两个目标历史子数据一起发生变化时额外产生的收益减去单个目标历史子数据发生变化而另一个目标历史子数据没变化造成的损失。

[0097] 步骤10886:基于所述关联权重,将所述目标历史数据中的目标历史子数据划分为至少一个数据包。

[0098] 具体的,在上述获得关联权重的基础上,可以将关联权重相同的目标历史子数据划分至一个数据包,还可以将关联权重在同一预设区间的目标历史子数据划分至一个数据包,在此不做限制。具体实施时,根据关联统计信息对目标历史数据进行分包处理可以是:将强相关的目标历史子数据进行归类,即将经常一起变化的强相关的目标历史子数据,划

分至一个数据包,将不相关的目标历史子数据,划分至一个数据包,并将频次变化极低的目标历史子数据,划分至一个数据包。

[0099] 沿用上例,在获得关联统计信息的基础上,将关联变化频次大于150次的属性变量划分为数据包A,并将关联变化频次大于50小于等150次的属性变量划分为数据包B,再将关联变化频次小于等于50次的属性变量划分为数据包C,并进一步创建数据包A对应的初始容器c1以及初始容器c1的填充规则r1,数据包B对应的初始容器c以及初始容器c2对应的填充规则r2,并数据包C对应的初始容器c3以及初始容器c3对应的填充规则r3。

[0100] 综上,根据对一段时间内的历史数据进行关联统计,从而按照关联统计信息将关联变化次数的多少对数据进行分包处理,并创建对应的初始容器以及填充规则,以便将关联变化次数相近的数据一起进行填充,保障了创建初始容器以及填充规则的合理性。

[0101] 进一步的,由于不同分包方式对应的压缩的效果不同,为了更加明确哪种分包方式更优,从而采用更优的分包方式进行分包,本申请实施例可以对多种分包方式进行评估,再根据评估结果确定最终的分包方式,具体实现包括如下步骤:

[0102] 步骤10886-2:基于所述关联权重,对所述目标历史数据中目标历史子数据对应的数据节点进行连接,生成至少一个树集合;

[0103] 具体的,数据节点,是将每个目标历史子数据抽象为一个节点所形成的。对这些数据节点进行一种连接,可能生成多个连接子树(每棵连接子树对应一个数据包),这些连接子树组合成为一个树集合;因此,若对这些数据节点进行多种连接,则每种连接方式都对应一个树集合。其中,树集合,是指由至少一个连接子树组成的集合。

[0104] 进一步的,考虑到如果随意生成多种连接方式,生成的连接方式没有规律可言,可能其中很多连接方式与分包的目标是相悖的,造成计算资源的浪费,因此,为了节省计算资源,可以基于关联权重,在每次连接结果的基础上进行连接,具体实现包括如下步骤:

[0105] 基于所述关联权重以及第i节点连接关系,对所述目标历史数据中目标历史子数据对应的数据节点进行第i连接,生成第i初始树集合;

[0106] 判断所述第i初始树集合中连接子树的数量是否大于预设阈值;

[0107] 若大于,表明连接子树的数量较多,还可以进行下一次连接,则将所述第i初始树集合作为中间树集合添加至树集合组;确定所述第i初始树集合中数据节点对应的目标连接关系,i自增1,通过所述目标连接关系作为所述第i节点连接关系,并返回执行所述基于所述关联权重以及第i节点连接关系,对所述目标历史数据中目标历史子数据对应的数据节点进行第i连接,生成第i初始树集合;

[0108] 若等于,表明连接子树的数量不多,不再进行下一次连接,则将所述树集合组中的中间树集合组成至少一个树集合。

[0109] 具体的,第i节点连接关系是指对目标历史子数据对应的数据节点进行第i次连接之前,第i-1次连接生成的数据节点之间的连接关系。在第1次进行连接之前,该第1节点连接关系为空。第i初始树集合,是指在目标历史子数据对应的数据节点之间进行第1次连接,生成的至少一个连接子树所组成的树集合。相应的,预设阈值,是指预先设置的连接子树的数量阈值,当连接子树的数量达到预设阈值时,表示可以停止进行下一次连接。该预设阈值也表示可以接受的数据包数量的最低阈值。该预设阈值的取值可以为1或2等小于节点数量的正整数,在此不做限制。

[0110] 基于此,每次在数据节点之间进行连接,可以连一条边,也可以连两条边,在此不做限制。在每次连接完成的时候,基于数据节点以及这些数据节点之间的连边,可以生成至少一个连接子树。此外,由于每次连接完成,都可能存在有数据节点没有任何连边,在这种情况下将该数据节点单独作为一个连接子树。可选的,本实施例中,可以采用MST(Minimum Spanning Tree)最小生成树算法:比如Prim算法(普里姆算法)或Kruskal算法(克鲁斯卡尔算法)等,根据数据节点之间的关联权重对数据节点进行连接,并且每连接一步作为一次连接,将每次连接生成的连接子树作为当次连接对应的初始树集合。

[0111] 在连接子树的数量大于预设阈值的情况下,表明连接子树的数量较多,还可以进行下一次连接,则将第i初始树集合作为中间树集合添加至树集合组,以此保留第i初始树集合对应的连接结果。并将第i初始树集合中数据节点对应的连接关系(即目标连接关系)作为下一次连接的连接基础(即第i+1节点连接关系),依次类推,不断迭代生成新的第i初始树集合。并将每次生成的第i初始树集合作为中间树集合添加至树集合组进行保留。

[0112] 在连接子树的数量大于预设阈值的情况下,表明连接子树的数量较多,还可以进行下一次连接,则将第i初始树集合作为中间树集合添加至树集合组,以此保留第i初始树集合的连接方法。并将第i初始树集合中数据节点对应的连接关系(即目标连接关系)作为下一次连接的连接基础(即第i+1节点连接关系),依次类推,不断迭代生成新的第i初始树集合。并将新的第i初始树集合作为中间树集合添加至树集合组进行保留。需要说明的是,未添加中间树集合之前,树集合组为空。

[0113] 在连接子树的数量等于预设阈值的情况下,表明连接子树的数量不多,不再进行下一次连接,则将之前连接过程中保留的中间树集合组成至少一个树集合。

[0114] 比如:目标历史属性数据中包括40个目标历史属性子数据,为每个目标历史属性子数据创建对应的数据节点,则生成40个数据节点。采用Kruskal算法将40个数据节点中关联权重最大的两个数据节点之间连接第1条边作为第1次连接,生成39个连接子树,将这39个连接子树组成第1初始树集合。判断39是否大于预设阈值1的判断结果为大于,则将第1初始树集合作为中间树集合添加至树集合组。并将第1初始树集合中数据节点之间的连接关系作为第2节点连接关系。再基于关联权重,在第2节点连接的关系的基础上,采用Kruskal算法对40个数据节点连接第2条边作为第2次连接,生成38个连接子树,将这38个连接子树组成第2初始树集合,以此类推,直至连接子树的数量为1。将添加至树集合组中的中间树集合组成至少一个树集合。

[0115] 综上,基于关联权重,在每次连接的基础上进行下次连接,保障了优先将关联权值大的边的两侧节点关联起来。以便将连接在一起的节点划分至一个数据包中,提升了连接的合理性,并节省了计算资源。

[0116] 步骤10886-4:根据所述关联统计信息对各个树集合进行评估,获得各个树集合对应的评估数据。

[0117] 具体的,根据关联统计信息对各个树集合进行评估,是指根据关联统计信息对各个树集合进行空间占用的评估,也即评估树集合中连接子树划分,是否可以达到在数据处理时,明显缩减待处理的数据量的压缩效果。

[0118] 进一步的,为了准确地评估各个树集合的优劣情况,本实施例中,可以将各个树集合中连接子树对应的目标历史子数据进行拼接,再根据拼接结果对树集合进行评估,针对

任意一个树集合,具体实施包括如下步骤:

[0119] 根据树集合中各个连接子树中数据节点对应的节点子数据的存储信息,对各个连接子树对应的节点子数据进行拼接处理,生成各个连接子树对应的拼接子数据;

[0120] 根据所述拼接子数据的拼接存储信息以及所述关联统计信息进行评估计算,获得树集合对应的评估数据。

[0121] 具体的,节点子数据,是指数据节点对应的目标历史子数据。存储信息,是指与节点子数据的存储相关的信息。该存储信息可以为节点子数据的存储位长度,和/或对齐长度等信息。拼接子数据,是指将节点子数据进行拼接后生成的数据。该拼接子数据的数量可以为一个,也可以为多个,在此不做限制。拼接存储信息,是指与拼接子数据的存储相关的信息。该拼接存储信息可以为拼接子数据的存储位长度,和/或对齐长度等信息。评估数据,是指对树集合进行评估计算获得的评估结果。该评估数据可以为评估分值的形式,此外,还可以为评估等级等其他形式,在此不做限制。

[0122] 具体实施时,可以采用预设评估函数对树集合进行评估计算。该预设评估函数,在每次生成的树集合中,先计算各个连接子树的总大小乘以该连接子树中变化最多的那个节点子数据的变化次数后,加上掩码长度。再对各个连接子树对应的计算结果求和,具体可以如下所示:

[0123] $$EScore=\sum_{k=1}^{n}(\max(vChange)*mSize+mLen)$$

[0124] 其中,EScore:表示评估分值,k表示当前连接子树的序号,n为连接子树的数量;vChange:表示连接子树中每个数据节点对应的节点子数据的变化次数,mSize:表示每个连接子树中拼接子数据对应的最小空间,mLen:表示掩码长度。具体的,该掩码长度,是指树集合中连接子树对应的掩码的长度,具体的,每个连接子树对应一个掩码位,则掩码的长度等于树集合中连接子树的数量,掩码位为1表示存在该掩码位对应的连接子树,掩码位为0表示不存在该掩码位对应的连接子树。

[0125] 比如,在上述获得至少一个树集合的基础上,针对任意一个树集合,假设树集合S中包括3个连接子树,其中,连接子树1中数据节点对应3个目标历史属性子数据,获取这2个目标历史属性子数据的存储位长度,获取的存储位长度分别为2个比特位,6个比特位。并对连接子树1中的2个目标历史属性子数据进行拼接处理,生成拼接子数据cd1;连接子树2中数据节点对应3个目标历史属性子数据,获取这3个目标历史属性子数据的存储位长度,获取的存储位长度分别为5个比特位,2个比特位,3个比特位。并对连接子树2中的3个目标历史属性子数据进行拼接处理,生成拼接子数据cd2;以此类推,根据连接子树3中数据节点对应35个目标历史属性子数据生成连接子树3对应的拼接子数据cd3。其中,拼接子数据cd1占用1个字节的空间大小(拼接存储信息),拼接子数据cd2占用2个字节的空间大小(拼接存储信息),拼接子数据cd3占用15个字节的空间大小(拼接存储信息),则将上述拼接存储信息以及以上40个目标历史属性子数据的关联统计信息,按照上述预设评估函数进行评估计算,获得树集合S的评估分值为80分。

[0126] 综上,将对树集合中各个连接子树对应的节点子数据先进行拼接,再根据拼接结果以及统计的关联频次对树集合进行整体评估计算,保障了评估的全面性并提高了评估的准确度。

[0127] 进一步的,考虑到对各个连接子树对应的节点子数据进行简单的拼接处理,无法达到明显的数据压缩的效果,因此,本实施例中可以遍历每个节点子数据以及各种对齐长度尝试多种拼接方式,采动态规划的思想获得达到理想的数据压缩效果的拼接方式并进行拼接,具体实现包括如下步骤:

[0128] 获取预设对齐长度以及各个连接子树对应的节点子数据的存储位长度;

[0129] 遍历各个连接子树对应的节点子数据、所述存储位长度以及预设对齐长度,将各个连接子树对应的节点子数据进行位域拼接,生成各个连接子树对应的拼接子数据。

[0130] 具体的,预设对齐长度是指预先设置的用以进行数据拼接的对齐长度,该对齐长度包括:8位、16位、32位、64位等。比如,预设对齐长度为8位,则以8位为长度对节点子数据进行拼接。该预设对齐长度可以为至少一个。存储位长度,是指节点子数据中有效的数据位的长度,比如节点子数据A的有效数据位为1位,则该节点子数据的存储位长度为1。位域拼接,是指在确定节点子数据所占用的二进制位的基础上,将它们进行拼接形成完整的字节。

[0131] 实际应用中,由于计算机存储或同步数据的时候会遇到“位域”问题,一些简单的数据是标志位,表示有无,只需要占用1比特位,比如:是否在水中悬浮(计算机存储最小单位1个字节=8个比特位)。有些数据的长度比计算机最小表示单位还小,比如移动状态枚举,只需要6个比特,这些数据可以通过“位域”拼起来,在传输的时候节省传输带宽。但是计算机又有一个“对齐”的限制,所有拼接后的数据都必须是一个完整的8位、16位、32位或者64位其中之一。并且一个数据在位域状态下是不能跨过对齐的,如果在一个对齐变量上没有填充满,那后面的空位也只能空着。如图3所示的第二个数据byMoveFrameCount之后,有6个比特是空占的,第3个数据是在6个空占比特之后开始的。因此,为了进一步压缩空间占用,还需要根据每个节点子数据的位长,进行顺序调整,尽量在对齐限制下尽可能多的拼接节点子数据。

[0132] 基于此,遍历各个连接子树对应的节点子数据、所述存储位长度以及预设对齐长度,将各个连接子数据对应的节点子数据进行位域拼接,是先将节点子数据转换为位域形式,再依次遍历每个节点子数据、并尝试将节点子数据的存储位长度枚举预设对齐长度与已拼接的节点子数据进行拼接。也即将这些每个节点子数据在对预设对齐长度的限制下进行尝试各种可能的拼接方式,进而选择最优的拼接方式,生成拼接子数据。具体实施时,可以采用动态规划思想,按照状态转移方程进行计算实现上述拼接。

[0133] 具体的,该状态转移方程如下所示:

[0134] $$DP[i][j][k]=\begin{cases} if Align(j)<len(i)\rightarrow\infty \\ if Align(j)\geq len(i)\wedge\exists DP[i-1][j][k-len(i)]\rightarrow min(DP[i][j][k],DP[i-1][j][k-len(i)]+len(i)) \\ \forall_{j_2\neq j}\ if\exists DP[i-1][j_2][k-len(i)]\rightarrow min(DP[i][j][k],DP[i-1][j_2][k-len(i)]+len(i)) \end{cases}$$

[0135] 其中,i表示第几个节点子数据,j表示最末尾的节点子数据使用的对齐长度,k表示最后一个对齐长度的域内所剩余的比特数,Align(j)表示第j个对齐长度的长度,len(i)表示第i个节点子数据的存储位长度。

[0136] 该状态转移方程具体是遍历每个节点子数据,并遍历所有的预设对齐长度,若当前预设对齐长度小于当前节点子数据的存储位长度,则枚举下一预设对齐长度。若当前预设对齐长度大于等于当前节点子数据的存储位长度,则尝试将当前节点子数据与上一节点子数据对应的拼接结果进行拼接,并且在遍历过程中选取拼接长度最小的拼接方式。

[0137] 此外,还可以通过如下步骤实现拼接:获取连接子树对应的第i节点子数据,其中,i为正整数,i从1开始取值;并获取第j对齐长度,将第j对齐长度作为第i节点子数据的对齐长度,其中,j为正整数,j从1开始取值;再判断第j对齐长度是否小于第i节点子数据的存储位长度;若是,j自增1,返回执行获取第j对齐长度;若否,获取第n对齐长度,其中,n为正整数,n从1开始取值,判断第j对齐长度是否等于第n对齐长度;

[0138] 若等于,将第i节点子数据与包含第i-1节点子数据的拼接结果进行拼接,获得第i拼接结果;若不等于,计算第j对齐长度与第i节点子数据的存储位长度的差值(即剩余比特数);

[0139] 若该差值大于等于0,则将第i节点子数据与包含第i-1节点子数据的拼接结果进行拼接;并i自增1,返回执行上述获取连接自数对应的第i节点子数据;若该差值小于0,则n自增1,返回执行获取第n对齐长度。

[0140] 综上,将通过遍历的方式尝试对节点子数据进行位域拼接,以便从中获得最优的拼接效果,进一步压缩了空间占用,提高了压缩的程度。

[0141] 步骤10886-6:基于所述评估数据,在至少一个树集合中确定所述目标历史数据对应的目标树集合。

[0142] 具体的,在上述获得评估数据的基础上,若评估数据为评估分值,可以根据分值高低,在至少一个树集合中确定评估分值最高的树集合为目标树集合。若评估数据为等级,可以根据等级高低,在至少一个树集合中确定评估等级最高的树集合为目标树集合。目标树集合中的各个连接子树表明对目标历史子数据的划分。

[0143] 步骤10886-8:将所述目标树集合中各个连接子树的数据节点在所述目标历史数据中对应的节点子数据分别进行封装,生成至少一个数据包。

[0144] 具体的,在上述确定目标树集合的基础上,再将按照各个连接子树划分的节点子数据按照预先设置的数据协议(比如UDP协议、TCP协议)进行封装,生成各个连接子树对应的数据包。

[0145] 步骤10888:根据各个数据包中目标历史子数据的数据属性信息,生成各个数据包对应的初始容器以及所述初始容器对应的填充规则。

[0146] 具体的,在上述生成至少一个数据包的基础上,为各个数据包创建对应的初始容器以及初始容器对应的填充规则,以便可以使初始容器关联的数据按照填充规则进行填充。

[0147] 进一步的,考虑到所创建的初始容器是用以对其对应的容器数据进行填充,因此,填充规则越清楚则越易于进行填充,因此,本实施例,为了增加填充的便易度,针对任意一个初始容器以及任意一个初始容器对应的填充规则的生成,具体实现如下所示:

[0148] 根据数据包中目标历史子数据的数据属性信息,确定数据包的包属性信息;

[0149] 根据所述包属性信息,创建数据包对应的初始容器;

[0150] 根据数据包中目标历史子数据的存储位长度以及拼接顺序,确定各个目标历史子数据在初始容器中的填充位置信息;

[0151] 基于所述填充位置信息,创建初始容器对应的填充规则。

[0152] 具体的,包属性信息,是指数据包相关的信息,比如:数据包标识、数据包中包含的数据数量等信息。拼接顺序,是指目标历史子数据在数据包中的前后顺序。填充位置信息,

是指可以标识历史子数据在初始容器中位置的信息。该填充位置信息可以包括每个历史子数据在初始容器中对应的填充起始位置以及填充终止位置等。

[0153] 基于此,根据数据包中目标历史子数据的数据属性信息,确定数据包的数据包标识以及数据,再根据数据包的数据包标识以及数据数量,创建数据包对应的初始容器。该初始容器为抽象概念,仅包括容器标识以及容器包含数据数量等基本信息。

[0154] 实际应用中,可以再根据数据包中目标历史子数据的存储位长度以及拼接顺序,确定各个目标历史子数据在初始容器中的填充的起始位置,再将各个目标历史子数据以及对应的填充起始位置作为初始容器的填充规则。

[0155] 比如:根据数据包A中目标历史子属性变量的数据数量3,这3个目标历史属性变量分别为d2,d23以及d30,确定数据包的包属性信息为A3,根据包属性信息A3创建对应的初始容器的容器名称为c-A3,再获取d2,d23以及d30的存储位长度分别为4比特位,5比特位以及12比特位,以及它们之间的拼接顺序为d2,d30,d23,则确定d2的填充起始位置为第1个比特位,d3的填充起始位置为第5个比特位,d23的填充起始位置为第17个比特位,将这三个信息作为填充位置信息,创建初始容器c-A3对应的填充规则。

[0156] 综上,先创建初始容器,再根据目标历史子数据的属性信息确定的填充位置信息,创建初始容器对应的填充规则,保障了填充的准确性,并提高了填充的效率。

[0157] 在上述生成压缩数据包的基础上,考虑到在数据传输场景下,可能存在多个压缩数据包,为了保障传输的完整性,本实施例,将这些压缩数据包封装成一个压缩数据总包再进行传输,具体实现如下所示:

[0158] 确定所述压缩任务关联的至少两个压缩数据包;

[0159] 根据预设编码顺序以及至少两个压缩数据包对应的编码标识,对至少两个压缩数据包进行组合,生成组合压缩数据包;

[0160] 根据预设编码顺序以及所述编码标识,生成所述组合压缩数据包对应的数据包掩码;

[0161] 基于所述数据包掩码和所述组合压缩数据包,生成压缩数据总包;

[0162] 将所述压缩数据总包发送至客户端。

[0163] 具体的,预设编码顺序是指预先设置的压缩数据包的编码顺序。相应的,各个压缩数据包在这预设编码中对应的编号,即为编码标识。

[0164] 基于此,根据预设编码顺序以及编码标识,将至少两个压缩数据包进行组合,是指将压缩数据包按照其编码标识在预设编码顺序中的排列顺序进行组合。

[0165] 比如初始数据对应64个压缩数据包,64个压缩数据包对应1-64的编号。则1-64的前后顺序则为预设编码顺序。在压缩任务关联的压缩数据包为10个,这10个压缩数据包按照预设编码顺序进行组合,生成组合压缩数据包CB。其中,空缺的数据包对应的掩码位为0,这10个压缩数据包对应的掩码位为1,则生成64位的数据包掩码。将组合压缩数据包CD和64位的数据包掩码,生成压缩数据总包TD,将该压缩数据总包TD发送至客户端。

[0166] 综上,将多个压缩数据包进行组合,并将组合结果结合对应的数据包掩码生成压缩数据总包进行数据传输,减少了传输带宽,增加了传输效率。

[0167] 需要说明的是,因为游戏是需要进行维护升级和需求变更的,而游戏数据也会跟着需求进行改变,若每次压缩时都采集数据的关联变化频次,再进行上述数据压缩,非常耗

费计算资源以及时间成本,也会成为需求日常修改需求的阻碍,所以,所述压缩方法,还需要进一步通过采用工具进行自动化,从而提升可维护性。

[0168] 因此,可以维护最原始的未压缩的数据结构,再通过工具自动获取周期内的初始数据,并在实际运行中统计每一个变量变化的频次,生成表格,将这个表格会交个一个工具,工具中会执行上面的MST和动态规划算法,生成数据包方案的“压缩版协议”、以及对应的压缩、解压缩函数体。

[0169] 游戏服务器会在发送协议前生成原始的协议数据结构,然后交给工具生成的配套压缩函数,将他做和上一次同步比较差异后,找出变化的变量,转成压缩版协议发送客户端收到压缩版协议后,会通过解压缩函数,结合上一次收到的数据,重新转换拼装成原始协议数据结构,(压缩协议没有同步的变量,即为之前已经收到过的变量),然后将这个原始的协议数据结构交给对应的协议处理函数。这样做的好处是只需要维护老的发送和接收协议函数以及协议数据结构,将压缩算法的优化和日常维护解耦,全部交给工具进行自动化处理。客户端将整套方案生成和服务器一样的程序源码,编译进去。因为实际运行的时候变量同步频次和关联性不会经常变,所以可以隔一段时间用工具优化一次,不需要做成实时的,这样整个流程的复杂度也不是很高,还能兼顾性能。

[0170] 具体的,如图4中(a)所示,可以包括工具自动优化流程以及原始制作流程。在原始制作流程中,在对协议变量进行增删修改后,由人工做对应修改,并发送修改,接收函数,对接收的函数进行编译,完成维护。而在工具自动优化流程中,在对协议变量进行增删修改后,运行数据采集,获得预设周期内的历史协议变量,并通过频次分析器进行分析,生成频次关联表。再通过子包(数据包)方案生成工具直接生成代码,通过代码优化协议(子包方案)/压缩/解压函数。并对优化后的函数进行编译完成维护。

[0171] 进一步的,图4中(a)的两种方法的实际应用过程如图4中(b)所示:发起广播后,将广播的数据拼装成原始协议包,在开启优化的情况下,将该原始协议包和上一次广播比较差量(差异数据),并在差量的基础上压缩(优化协议拼接子包),并将生成的压缩结果代替原始协议数据包进行发送(即劫持过程)。原始协议数据包通过网络进行传输,由客户端接收。客户端发送是优化协议,则(解压缩)按照掩码解开子包,并查询子包对应的本地已有数据,二者重新组装,得到原始协议包格式,将该原始协议包格式作为待处理的原始协议(即转嫁过程)。若客户端接收的为原始协议数据包,即不是优化协议,则走原始流程,直接处理原始协议数据。

[0172] 综上所述,本申请施例提供的数据压缩方法,通过获取压缩任务关联的初始数据;将所述初始数据与所述压缩任务关联的历史数据进行对比,获得差异数据;确定所述差异数据对应的目标容器,并在所述初始数据中提取与所述目标容器关联的容器数据,其中,所述容器数据中包含所述差异数据;按照所述目标容器对应的填充规则,将所述容器数据填充至所述目标容器,并根据填充结果生成与所述压缩任务关联的压缩数据包。实现了将初始数据进行数据压缩,减少了初始数据的空间占用,提升了对初始数据的后续处理效率。

[0173] 与上述方法实施例相对应,本说明书还提供了数据解压方法实施例,图5示出了本说明书一个实施例提供的一种数据解压方法的流程图,应用于客户端,具体包括以下步骤:

[0174] 步骤502,获取与解压任务关联的压缩数据包。

[0175] 具体的,解压任务是指对压缩数据包进行解压的任务。该解压任务可以理解为上

述方法实施例中压缩任务的逆过程。

[0176] 步骤504:根据所述压缩数据包对应的目标容器的提取规则,从所述压缩数据包中提取所述目标容器的容器数据。

[0177] 具体的,提取规则,是指在目标容器中提取容器数据的规则,该提取规则和上述方法实施例中的填充规则类似,也可以具有容器数据相关的提取信息,比如提取顺序,提取数据信息,提取名称,提取位置信息,提取策略等信息,在此不做限制。

[0178] 步骤506:根据所述容器数据的数据属性信息,确定所述容器数据关联的历史数据。

[0179] 具体的,根据容器数据所属的初始数据的数据类型,可以获取容器数据所属的初始数据对应的上一时间点的历史数据。

[0180] 步骤508:基于所述容器数据和所述历史数据,生成目标数据。

[0181] 具体的,所述目标数据可以理解为上述方法实施例中的初始数据。通过在历史数据中剔除容器数据对应的数据,再将剔除后的历史数据和容器数据进行组合,即可生成目标数据。

[0182] 可选地,考虑到和服务端进行数据传输的过程中,所传输的通常不是单一的数据压缩包,因此,需要先对接收的压缩数据总包进行解析,获得至少一个压缩数据包,本实施例,具体实现如下:

[0183] 接收服务端发送的压缩数据总包;

[0184] 解析所述压缩数据总包,获得数据包掩码以及组合压缩数据压缩包;

[0185] 基于所述数据包掩码,在所述组合压缩数据总包中提取至少一个压缩数据包,其中,至少一个压缩数据包中包括所述压缩数据包。

[0186] 具体的,解析压缩数据总包,可以理解为将压缩数据总包中的数据包掩码以及组合压缩数据包进行拆解。

[0187] 比如:接收服务端发送的压缩数据总包TD,对该压缩数据总包进行解析,获得64位的数据包掩码以及组合压缩数据包CD。根据64位的数据包掩码,在组合数据包CD中提取掩码位为1的10个压缩数据包,并将这10个压缩数据包中的数据分别进行解压,获得每个压缩数据包对应的容器数据。再根据容器数据的数据类型T获取该数据类型T对应的上一相邻时间点的历史数据,再通过容器数据替换历史数据中相同数据名称的数据,则可以还原目标数据。

[0188] 综上,通过接收服务端发送的压缩数据总包,并对压缩数据总包进行解析,获得各个压缩数据包,实现了在数据压缩总包的基础上对压缩数据包进行接收,并通过解析该压缩数据总包一次性获取多个数据压缩包。

[0189] 综上所述,本申请施例提供的数据解压方法,通过获取与解压任务关联的压缩数据包;根据所述压缩数据包对应的目标容器的提取规则,从所述压缩数据包中提取所述目标容器的容器数据,实现了对压缩数据包中数据的还原,再根据所述容器数据的属性信息,确定所述容器数据关联的历史数据;基于所述容器数据和所述历史数据,生成目标数据。实现了通过将容器数据与历史数据进行合并,对目标数据进行还原。

[0190] 上述为本实施例的一种数据解压方法的示意性方案。需要说明的是,该数据解压方法的技术方案与上述的数据压缩方法的技术方案属于同一构思,数据解压方法的技术方

案未详细描述的细节内容,均可以参见上述数据压缩方法的技术方案的描述。

[0191] 下述结合附图6以本申请提供的数据压缩方法对游戏场景的应用为例,对所述数据压缩方法进行进一步说明。其中,图6示出了本申请一实施例提供的一种应用于游戏场景的数据压缩方法的处理流程图,具体包括以下步骤:

[0192] 步骤602:获取与压缩任务关联的预设周期内的周期历史数据集合。

[0193] 具体的,获取1周内的周期历史移动数据集合。

[0194] 步骤604:将周期历史数据集合中相邻时间点的周期历史数据进行对比,获得各个周期历史数据中周期历史子数据的变化信息。

[0195] 具体的,周期历史移动数据集合中包括100个周期历史移动数据,将这100个周期历史移动数据中相邻时间点的周期历史移动数据进行对比,对这1周内相邻时间点中后一时间点的周期历史移动数据中各个周期历史移动子数据两两一起产生变化的情况进行记录。

[0196] 步骤606:基于变化信息对周期历史数据集合对应的周期历史子数据进行关联性统计,获得关联统计信息。

[0197] 具体的,根据记录的变化信息,对100个周期历史移动数据中30周期移动历史子数据中两两一起的变化次数进行统计,获得的关联变化频次包括:周期移动历史子数据md1和md5的关联变化频次为60次,md2和d33的关联变化频次为30次,……md8和md40的关联变化频次为5次。进一步,还可以统计每个周期移动历史子数据的单独变化频次,比如:md1的单独变化频次为80次,md5的单独变化频次为65次,

[0198] 步骤608:确定与周期历史数据集合中周期历史数据的数据类型相同的目标历史数据。

[0199] 具体的,确定周期历史移动数据中任意一个作为目标历史数据。

[0200] 步骤610:根据关联统计信息计算目标历史数据中目标历史子数据之间的关联权重。

[0201] 具体的,根据上述统计的关联变化频次,以及上述方法实施例的预设权重公式,计算每两个目标历史移动子数据之间的关联权重,其中,md1和md5的计算过程如下:

[0202] nWeight=60*min(5kb,8kb)-min((65-60)*8kb,(80-60)*5kb)=260。其中,5kb为md1占用的空间大小,8kb为md5占用的空间大小。

[0203] 步骤612:基于关联权重,对目标历史数据中目标历史子数据对应的数据节点进行连接,生成至少一个树集合。

[0204] 具体的,可以基于关联权重从大到小的顺序对30个目标历史移动子数据对应的数据节点进行多次连接,将每连接一次生成的连接子树作为一个树集合。并且,每次在上一次连接的基础上进行下一次连接。则第1次连接生成29个连接子树,这29个连接子数据作为一个树集合,并在这29个连接子树的基础上进行第2次连接,生成28个连接子树,将这28个连接子树也作为一个树集合,以此类推,最终全部数据节点连成一个连接子树时,停止连接。该连接过程可以采用MST算法实现,最终可以获得29个树集合。

[0205] 步骤614:获取预设对齐长度以及各个连接子树对应的节点子数据的存储位长度。

[0206] 具体的,获取4种预设对齐长度分别为:8位、16位、32位、64位,并获取每个节点子数据的存储位长,其中md1的存储位长为5个比特位,md5的存储位长为10个比特位,md7的存

储位长为3个比特位。

[0207] 步骤616:遍历各个连接子树对应的节点子数据、存储位长度以及预设对齐长度,将各个连接子树对应的节点子数据进行位域拼接,生成各个连接子树对应的拼接子数据。

[0208] 具体的,假设连接子树T1中包括3个节点子数据,分别为md1、md5以及md7。通过执行上述状态转移方程,遍历这3个节点子数据、预设对齐长度,以及每种预设对齐长度和存储位长度的差值,尝试各种可能的连接方式,并在连接方式中,取得最佳的连接方式进行位域拼接,生成连接子数据T1对应的拼接子数据。

[0209] 步骤618:根据拼接子数据的拼接存储信息以及关联统计信息进行评估计算,获得树集合对应的评估数据。

[0210] 具体的,对29个树集合,每个树集合按照上述方法实施例中的预设评估函数进行评估计算,获得每个树集合对应的评估分值。假设树集合TS1对应的评估分值为75。

[0211] 步骤620:基于评估数据,在至少一个树集合中确定目标历史数据对应的目标树集合。

[0212] 具体的,假设树集合TS1对应的评估分值为75为最高评估分值,则在29个树集合中,确定目标历史移动数据对应的目标树集合为ST1。

[0213] 步骤622:将目标树集合中各个连接子树的数据节点在目标历史数据中对应的节点子数据分别进行封装,生成至少一个数据包。

[0214] 具体的,假设树集合ST1中包含3个连接子树,对每个连接子树中的节点子数据分别进行封装,生成每个连接子树对应的数据包。

[0215] 步骤624:根据各个数据包中目标历史子数据的数据属性信息,生成各个数据包对应的初始容器以及初始容器对应的填充规则。

[0216] 具体的,根据3个数据包中目标历史移动子数据的数据信息,生成每个数据包对应的初始容器以及这3个初始容器每个初始容器对应的填充规则。

[0217] 步骤626:获取压缩任务关联的初始数据。

[0218] 具体的,获取压缩任务关联的移动数据IMD。

[0219] 步骤628:将初始数据与压缩任务关联的历史数据进行对比,获得差异数据。

[0220] 具体的,将移动数据IMD与压缩任务关联的历史移动数据HMD进行对比,获得二者之间的差异数据,该差异数据中包括2个移动变量,分别为md1以及md5。

[0221] 步骤630:确定差异数据对应的目标容器,并在初始数据中提取与目标容器关联的容器数据,其中,容器数据中包含差异数据。

[0222] 具体的,根据数据属性信息和容器信息的对应关系,确定md1和md5对应的目标容器都为TC1。则在移动数据IMD中提取TC1关联的容器移动数据。

[0223] 步骤632:按照目标容器对应的填充规则,将容器数据填充至目标容器,并根据填充结果生成与压缩任务关联的压缩数据包。

[0224] 具体的,将容器移动数据按照目标容器的填充位置信息填充至目标容器,生成移动数据IMD对应的压缩数据包。

[0225] 进一步,在存在多个压缩数据包的情况下,可以将多个压缩数据包生成压缩数据总包发送至客户端,完成移动数据的传输。

[0226] 综上所述,本申请施例提供的数据压缩方法,通过获取压缩任务关联的初始数据,

并将所述初始数据与所述压缩任务关联的历史数据进行对比，获得差异数据，即获得初始数据与历史数据相比发生变化的数据，再确定所述差异数据对应的目标容器，并在所述初始数据中提取与所述目标容器关联的容器数据，其中，所述容器数据中包含所述差异数据，实现了通过差异数据对应的目标容器，提取初始数据中与差异数据强相关的数据。再按照所述目标容器对应的填充规则，将所述容器数据填充至所述目标容器，并根据填充结果生成与所述压缩任务关联的压缩数据包，以便将提取的数据进行组装生成待处理的压缩数据包，实现了将初始数据进行数据压缩，并通过数据压缩减少了空间占用，提升了后续对初始数据的处理效率。

[0227] 与上述数据压缩方法实施例相对应，本申请还提供了数据压缩装置实施例，图7示出了本申请一实施例提供的一种数据压缩装置的结构示意图。如图7所示，该装置包括：

[0228] 获取数据模块702，被配置为获取压缩任务关联的初始数据；

[0229] 对比模块704，被配置为将所述初始数据与所述压缩任务关联的历史数据进行对比，获得差异数据；

[0230] 提取模块706，被配置为确定所述差异数据对应的目标容器，并在所述初始数据中提取与所述目标容器关联的容器数据，其中，所述容器数据中包含所述差异数据；

[0231] 生成模块708，被配置为按照所述目标容器对应的填充规则，将所述容器数据填充至所述目标容器，并根据填充结果生成与所述压缩任务关联的压缩数据包。

[0232] 可选地，所述填充规则的生成，运行如下模块：

[0233] 获取集合模块，被配置为获取与所述压缩任务关联的预设周期内的周期历史数据集合；

[0234] 数据对比模块，被配置为将所述周期历史数据集合中相邻时间点的周期历史数据进行对比，获得各个周期历史数据中周期历史子数据的变化信息；

[0235] 统计模块，被配置为基于所述变化信息对所述周期历史数据集合对应的周期历史子数据进行关联性统计，获得关联统计信息；

[0236] 生成容器模块，被配置为根据所述关联统计信息对所述周期历史数据集合关联的目标历史数据进行分包处理，基于分包处理结果生成初始容器以及所述初始容器对应的填充规则。

[0237] 可选地，所述生成容器模块，包括：

[0238] 确定数据子模块，被配置为确定与所述周期历史数据集合中周期历史数据的数据类型相同的目标历史数据；

[0239] 计算权重子模块，被配置为根据所述关联统计信息计算所述目标历史数据中目标历史子数据之间的关联权重；

[0240] 划分子模块，被配置为基于所述关联权重，将所述目标历史数据中的目标历史子数据划分为至少一个数据包；

[0241] 生成规则子模块，被配置为根据各个数据包中目标历史子数据的数据属性信息，生成各个数据包对应的初始容器以及所述初始容器对应的填充规则。

[0242] 可选地，所述划分子模块，包括：

[0243] 生成树集合单元，被配置为基于所述关联权重，对所述目标历史数据中目标历史子数据对应的数据节点进行连接，生成至少一个树集合；

[0244] 评估单元,被配置为根据所述关联统计信息对各个树集合进行评估,获得各个树集合对应的评估数据;

[0245] 确定集合单元,被配置为基于所述评估数据,在至少一个树集合中确定所述目标历史数据对应的目标树集合;

[0246] 封装单元,被配置为将所述目标树集合中各个连接子树的数据节点在所述目标历史数据中对应的节点子数据分别进行封装,生成至少一个数据包。

[0247] 可选地,所述划分子模块,还被配置为:

[0248] 根据树集合中各个连接子树中数据节点对应的节点子数据的存储信息,对各个连接子树对应的节点子数据进行拼接处理,生成各个连接子树对应的拼接子数据;

[0249] 根据所述拼接子数据的拼接存储信息以及所述关联统计信息进行评估计算,获得树集合对应的评估数据。

[0250] 可选地,所述划分子模块,还被配置为:

[0251] 获取预设对齐长度以及各个连接子树对应的节点子数据的存储位长度;

[0252] 遍历所述存储位长度以及预设对齐长度,将各个连接子树对应的节点子数据进行位域拼接,生成各个连接子树对应的拼接子数据。

[0253] 可选地,所述生成树集合单元,还被配置为:

[0254] 基于所述关联权重以及第i节点连接关系,对所述目标历史数据中目标历史子数据对应的数据节点进行第i连接,生成第i初始树集合;

[0255] 判断所述第i初始树集合中连接子树的数量是否大于预设阈值;

[0256] 若大于,将所述第i初始树集合作为中间树集合添加至树集合组;确定所述第i初始树集合中数据节点对应的目标连接关系,i自增1,通过所述目标连接关系作为所述第i节点连接关系,并返回执行所述基于所述关联权重以及第i节点连接关系,对所述目标历史数据中目标历史子数据对应的数据节点进行第i连接,生成第i初始树集合;

[0257] 若等于,将所述树集合组中的中间树集合组成至少一个树集合。

[0258] 可选地,所述生成规则子模块,还被配置为:

[0259] 根据数据包中目标历史子数据的数据属性信息,确定数据包的包属性信息;

[0260] 根据所述包属性信息,创建数据包对应的初始容器;

[0261] 根据数据包中目标历史子数据的存储位长度以及拼接顺序,确定各个目标历史子数据在初始容器中的填充位置信息;

[0262] 基于所述填充位置信息,创建初始容器对应的填充规则。

[0263] 可选地,所述数据压缩装置,还包括:

[0264] 确定所述压缩任务关联的至少两个压缩数据包;

[0265] 根据预设编码顺序以及至少两个压缩数据包对应的编码标识,对至少两个压缩数据包进行组合,生成组合压缩数据包;

[0266] 根据预设编码顺序以及所述编码标识,生成所述组合压缩数据包对应的数据包掩码;

[0267] 基于所述数据包掩码和所述组合压缩数据包,生成压缩数据总包;

[0268] 将所述压缩数据总包发送至客户端。

[0269] 可选地,所述提取模块706,包括:

[0270] 确定容器集合子模块，被配置为根据所述初始数据的类型信息，确定所述初始数据对应的容器集合；

[0271] 选择子模块，被配置为基于所述差异数据的数据属性信息在所述容器集合中选择目标容器；

[0272] 提取数据子模块，被配置为根据所述目标容器的关联数据信息，在所述初始数据中提取所述容器数据。

[0273] 综上所述，本申请施例提供的数据压缩装置，通过获取压缩任务关联的初始数据，并将所述初始数据与所述压缩任务关联的历史数据进行对比，获得差异数据，即获得初始数据与历史数据相比发生变化的数据，再确定所述差异数据对应的目标容器，并在所述初始数据中提取与所述目标容器关联的容器数据，其中，所述容器数据中包含所述差异数据，实现了通过差异数据对应的目标容器，提取初始数据中与差异数据强相关的数据。再按照所述目标容器对应的填充规则，将所述容器数据填充至所述目标容器，并根据填充结果生成与所述压缩任务关联的压缩数据包，以便将提取的数据进行组装生成待处理的压缩数据包，实现了将初始数据进行数据压缩，并通过数据压缩减少了空间占用，提升了后续对初始数据的处理效率。

[0274] 上述为本实施例的一种数据压缩装置的示意性方案。需要说明的是，该数据压缩装置的技术方案与上述的数据压缩方法的技术方案属于同一构思，数据压缩装置的技术方案未详细描述的细节内容，均可以参见上述数据压缩方法的技术方案的描述。此外，装置实施例中的各组成部分应当理解为实现该程序流程各步骤或该方法各步骤所必须建立的功能模块，各个功能模块并非实际的功能分割或者分离限定。由这样一组功能模块限定的装置权利要求应当理解为主要通过说明书记载的计算机程序实现该解决方案的功能模块构架，而不应当理解为主要通过硬件方式实现该解决方案的实体装置。

[0275] 与上述数据解压方法实施例相对应，本申请还提供了数据解压装置实施例，图8示出了本申请一实施例提供的一种数据解压装置的结构示意图，应用于客户端，如图8所示，该装置包括：

[0276] 获取包模块802，被配置为获取与解压任务关联的压缩数据包；

[0277] 提取数据模块804，被配置为根据所述压缩数据包对应的目标容器的提取规则，从所述压缩数据包中提取所述目标容器的容器数据；

[0278] 确定数据模块806，被配置为根据所述容器数据的数据属性信息，确定所述容器数据关联的历史数据；

[0279] 生成数据模块808，被配置为基于所述容器数据和所述历史数据，生成目标数据。

[0280] 可选地，所述获取包模块802，包括：

[0281] 接收子模块，被配置为接收服务端发送的压缩数据总包；

[0282] 解析子模块，被配置为解析所述压缩数据总包，获得数据包掩码以及组合压缩数据压缩包；

[0283] 提取包子模块，被配置为基于所述数据包掩码，在所述组合压缩数据总包中提取至少一个压缩数据包，其中，至少一个压缩数据包中包括所述压缩数据包。

[0284] 综上所述，本申请施例提供的数据解压装置，通过获取与解压任务关联的压缩数据包；根据所述压缩数据包对应的目标容器的提取规则，从所述压缩数据包中提取所述目

标容器的容器数据,实现了对压缩数据包中数据的还原,再根据所述容器数据的属性信息,确定所述容器数据关联的历史数据;基于所述容器数据和所述历史数据,生成目标数据。实现了通过将容器数据与历史数据进行合并,对目标数据进行还原。

[0285] 上述为本实施例的一种数据解压装置的示意性方案。需要说明的是,该数据解压装置的技术方案与上述的数据解压方法的技术方案属于同一构思,数据解压装置的技术方案未详细描述的细节内容,均可以参见上述数据解压方法的技术方案的描述。此外,装置实施例中的各组成部分应当理解为实现该程序流程各步骤或该方法各步骤所必须建立的功能模块,各个功能模块并非实际的功能分割或者分离限定。由这样一组功能模块限定的装置权利要求应当理解为主要通过说明书记载的计算机程序实现该解决方案的功能模块构架,而不应当理解为主要通过硬件方式实现该解决方案的实体装置。

[0286] 图9示出了根据本申请一实施例提供的一种计算设备900的结构框图。该计算设备900的部件包括但不限于存储器910和处理器920。处理器920与存储器910通过总线930相连接,数据库950用于保存数据。

[0287] 计算设备900还包括接入设备940,接入设备940使得计算设备900能够经由一个或多个网络960通信。这些网络的示例包括公用交换电话网(PSTN,Public Switched Telephone Network)、局域网(LAN,LocalAreaNetwork)、广域网(WAN,WideAreaNetwork)、个域网(PAN,PersonalAreaNetwork)或诸如因特网的通信网络的组合。接入设备940可以包括有线或无线的任何类型的网络接口(例如,网络接口卡(NIC,network interface controller))中的一个或多个,诸如IEEE802.11无线局域网(WLAN,Wireless LocalAreaNetwork)无线接口、全球微波互联接入(Wi-MAX,Worldwide Interoperability for MicrowaveAccess)接口、以太网接口、通用串行总线(USB,Universal Serial Bus)接口、蜂窝网络接口、蓝牙接口、近场通信(NFC,Near Field Communication)接口,等等。

[0288] 在本申请的一个实施例中,计算设备900的上述部件以及图9中未示出的其他部件也可以彼此相连接,例如通过总线。应当理解,图9所示的计算设备结构框图仅仅是出于示例的目的,而不是对本申请范围的限制。本领域技术人员可以根据需要,增添或替换其他部件。

[0289] 计算设备900可以是任何类型的静止或移动计算设备,包括移动计算机或移动计算设备(例如,平板计算机、个人数字助理、膝上型计算机、笔记本计算机、上网本等)、移动电话(例如,智能手机)、可佩戴的计算设备(例如,智能手表、智能眼镜等)或其他类型的移动设备,或者诸如台式计算机或个人计算机(PC,Personal Computer)的静止计算设备。计算设备900还可以是移动式或静止式的服务器。

[0290] 其中,处理器920用于执行上述数据压缩方法或数据解压方法的计算机可执行指令。

[0291] 上述为本实施例的一种计算设备的示意性方案。需要说明的是,该计算设备的技术方案与上述的数据压缩方法或数据解压方法的技术方案属于同一构思,计算设备的技术方案未详细描述的细节内容,均可以参见上述数据压缩方法或数据解压方法的技术方案的描述。

[0292] 本申请一实施例还提供一种计算机可读存储介质,其存储有计算机指令,该指令被处理器执行时以用于上述数据压缩方法或数据解压方法。

[0293] 上述为本实施例的一种计算机可读存储介质的示意性方案。需要说明的是,该存储介质的技术方案与上述的数据压缩方法或数据解压方法的技术方案属于同一构思,存储介质的技术方案未详细描述的细节内容,均可以参见上述数据压缩方法或数据解压方法的技术方案的描述。

[0294] 所述计算机指令包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,RandomAccess Memory)、电载波信号、电信信号以及软件分发介质等。需要说明的是,所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读介质不包括电载波信号和电信信号。

[0295] 本申请一实施例还提供一种芯片,其存储有计算机程序,该计算机程序被芯片执行时实现上述数据压缩方法或数据解压方法的步骤。

[0296] 需要说明的是,对于前述的各方法实施例,为了简便描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本申请并不受所描述的动作顺序的限制,因为依据本申请,某些步骤可以采用其它顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定都是本申请所必须的。

[0297] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中没有详述的部分,可以参见其它实施例的相关描述。

[0298] 以上公开的本申请优选实施例只是用于帮助阐述本申请。可选实施例并没有详尽叙述所有的细节,也不限制该发明仅为所述的具体实施方式。显然,根据本申请的内容,可作很多的修改和变化。本申请选取并具体描述这些实施例,是为了更好地解释本申请的原理和实际应用,从而使所属技术领域技术人员能很好地理解和利用本申请。本申请仅受权利要求书及其全部范围和等效物的限制。

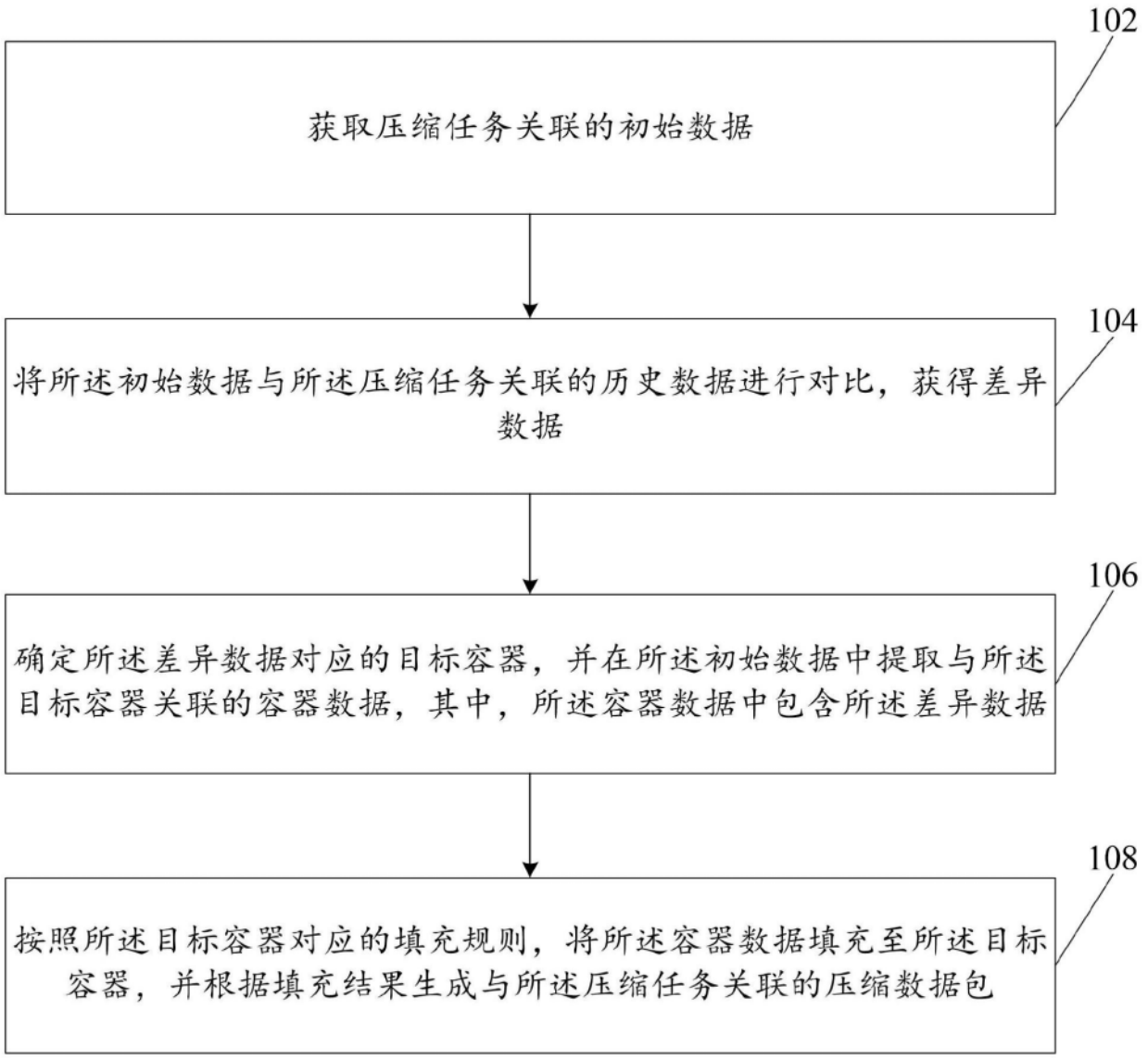

102
获取压缩任务关联的初始数据
104
将所述初始数据与所述压缩任务关联的历史数据进行对比，获得差异数据
106
确定所述差异数据对应的目标容器，并在所述初始数据中提取与所述目标容器关联的容器数据，其中，所述容器数据中包含所述差异数据
108
按照所述目标容器对应的填充规则，将所述容器数据填充至所述目标容器，并根据填充结果生成与所述压缩任务关联的压缩数据包

图1

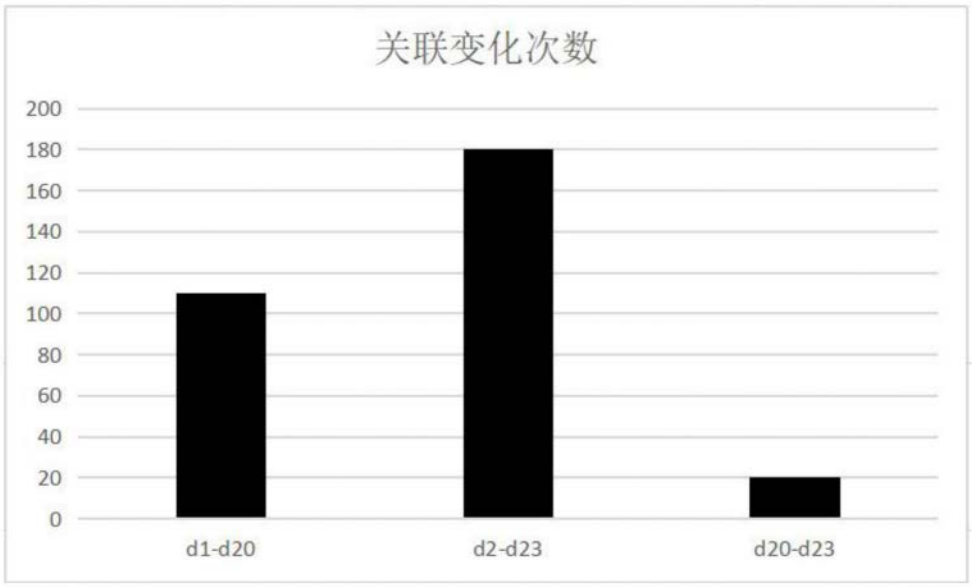


图2

```
struct_block2 {
        int8_t byPitchDirection; //occ:1
// ----
        int8_t byMoveFrameCount; //occ:1
// ----
        int32_t SyncMoveSpeed_nPassiveVelocityZ :12; //occ:1
        int32_t SyncMoveSpeed_uPassiveDirectionXY :8; //occ:1
        int32_t SyncMoveSpeed_uOnTowerFlag :1; //occ:1
        int32_t SyncMoveSpeed_uMinRunSpeed :7; //occ:1
        int32_t SyncMoveSpeed_uDriftFlag :1; //occ:1
        int32_t SyncMoveSpeed_uWaterFloatFlag :1; //occ:1
};
```

图3

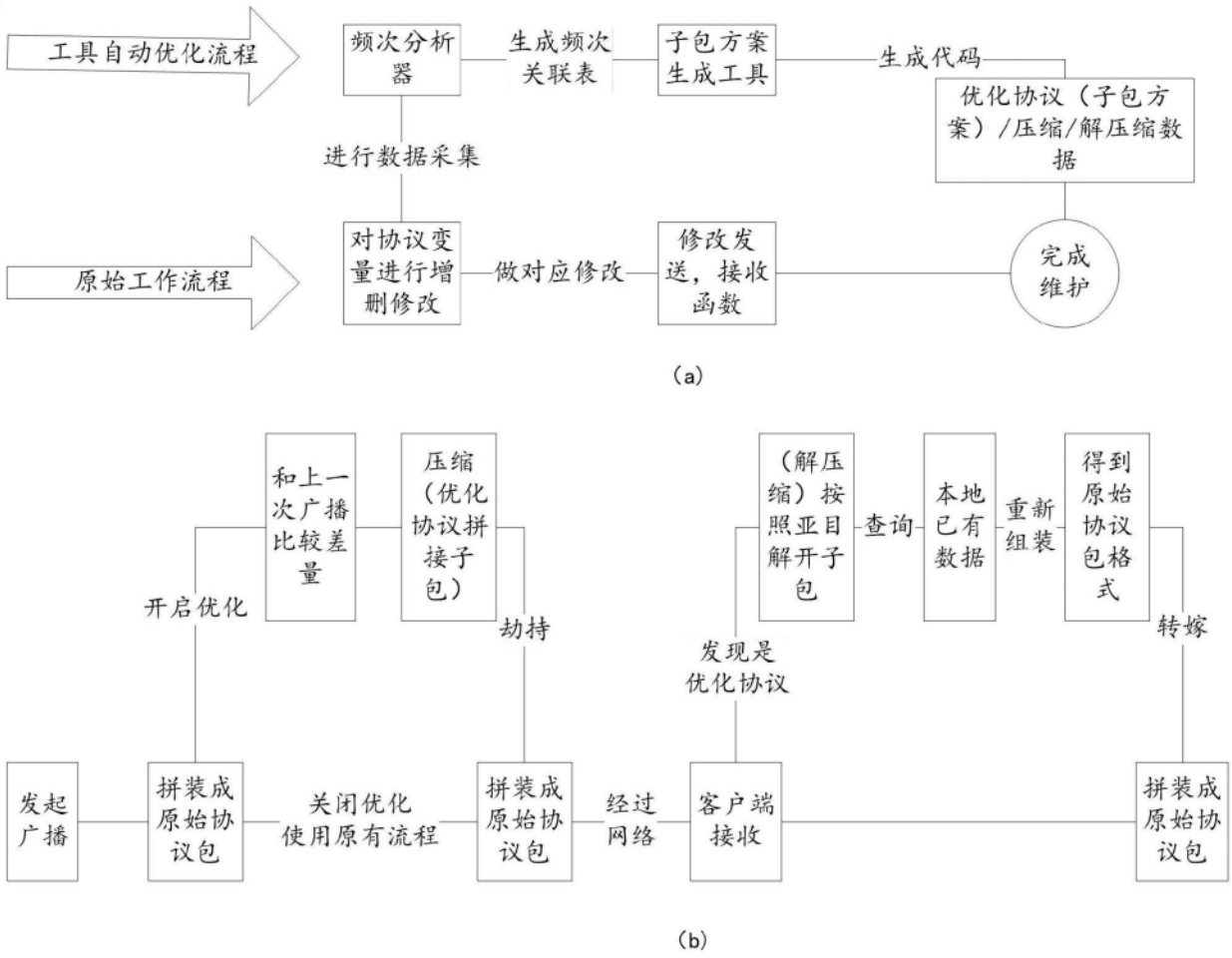

工具自动优化流程
频次分析器
生成频次关联表
子包方案生成工具
生成代码
优化协议（子包方案）/压缩/解压缩数据
进行数据采集
原始工作流程
对协议变量进行增删修改
做对应修改
修改发送，接收函数
完成维护
(a)
和上一次广播比较差量
压缩（优化协议拼接子包）
（解压缩）按照亚目解开子包
查询
本地已有数据
重新组装
得到原始协议包格式
开启优化
劫持
发现是优化协议
转嫁
发起广播
拼装成原始协议包
关闭优化使用原有流程
拼装成原始协议包
经过网络
客户端接收
拼装成原始协议包
(b)

图4

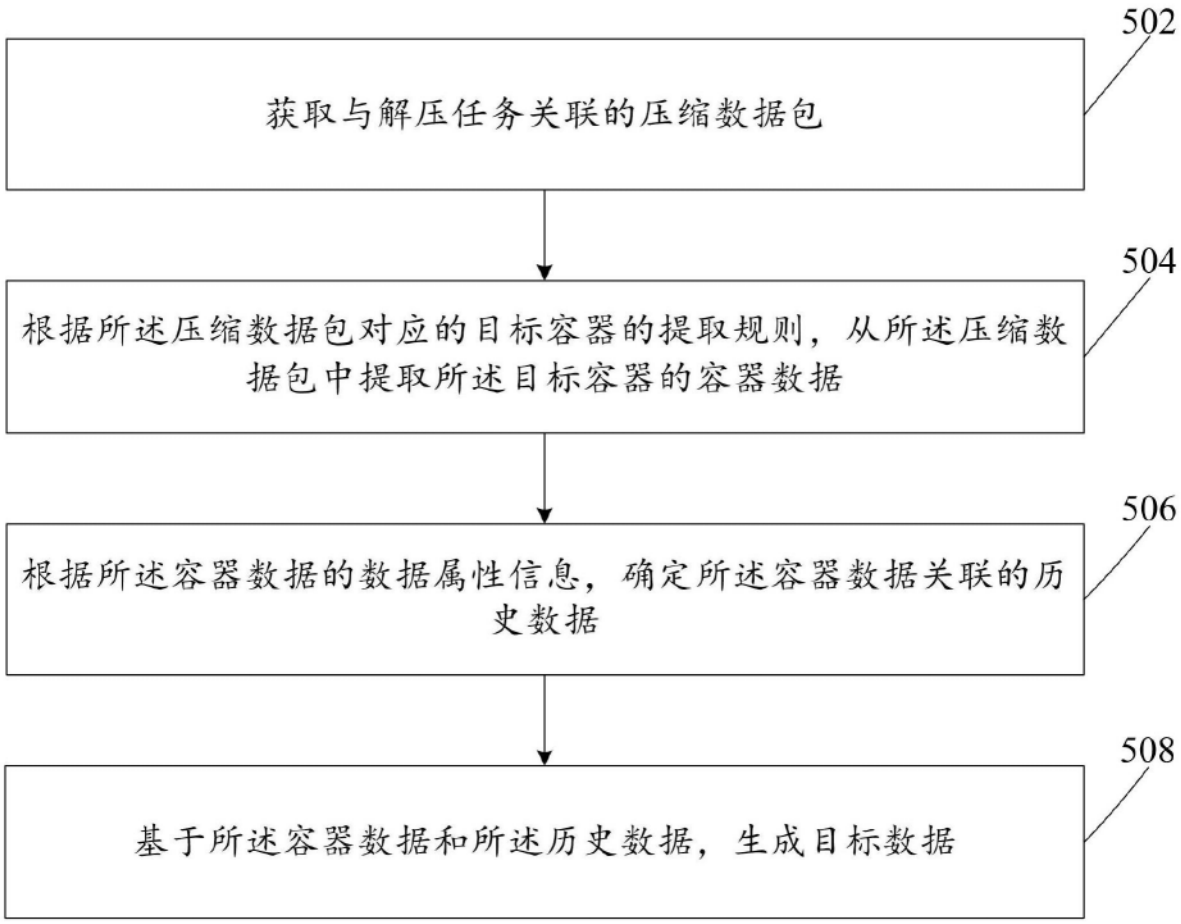

502
获取与解压任务关联的压缩数据包
504
根据所述压缩数据包对应的目标容器的提取规则，从所述压缩数据包中提取所述目标容器的容器数据
506
根据所述容器数据的数据属性信息，确定所述容器数据关联的历史数据
508
基于所述容器数据和所述历史数据，生成目标数据

图5

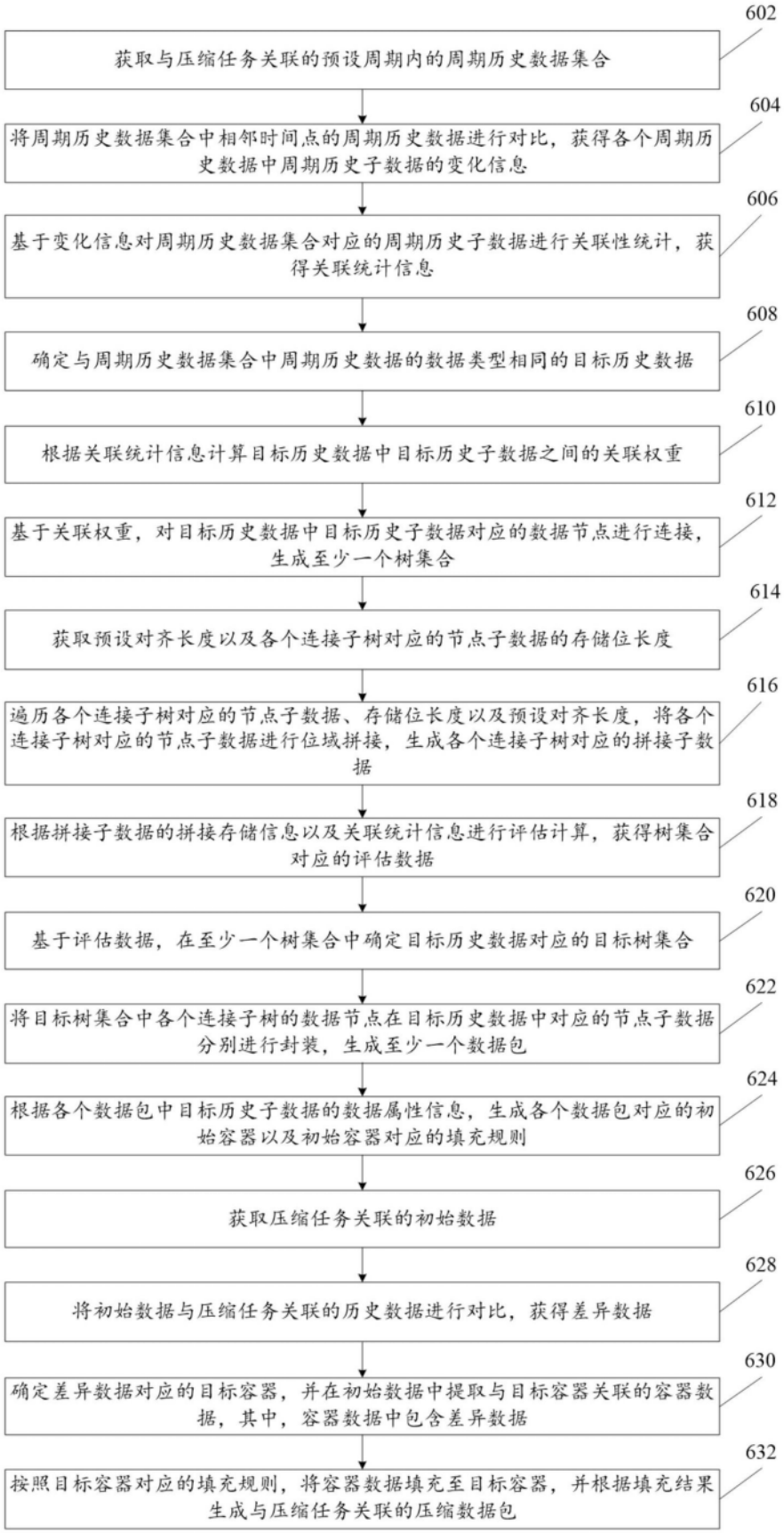

602
获取与压缩任务关联的预设周期内的周期历史数据集合
604
将周期历史数据集合中相邻时间点的周期历史数据进行对比，获得各个周期历史数据中周期历史子数据的变化信息
606
基于变化信息对周期历史数据集合对应的周期历史子数据进行关联性统计，获得关联统计信息
608
确定与周期历史数据集合中周期历史数据的数据类型相同的目标历史数据
610
根据关联统计信息计算目标历史数据中目标历史子数据之间的关联权重
612
基于关联权重，对目标历史数据中目标历史子数据对应的数据节点进行连接，生成至少一个树集合
614
获取预设对齐长度以及各个连接子树对应的节点子数据的存储位长度
616
遍历各个连接子树对应的节点子数据、存储位长度以及预设对齐长度，将各个连接子树对应的节点子数据进行位域拼接，生成各个连接子树对应的拼接子数据
618
根据拼接子数据的拼接存储信息以及关联统计信息进行评估计算，获得树集合对应的评估数据
620
基于评估数据，在至少一个树集合中确定目标历史数据对应的目标树集合
622
将目标树集合中各个连接子树的数据节点在目标历史数据中对应的节点子数据分别进行封装，生成至少一个数据包
624
根据各个数据包中目标历史子数据的数据属性信息，生成各个数据包对应的初始容器以及初始容器对应的填充规则
626
获取压缩任务关联的初始数据
628
将初始数据与压缩任务关联的历史数据进行对比，获得差异数据
630
确定差异数据对应的目标容器，并在初始数据中提取与目标容器关联的容器数据，其中，容器数据中包含差异数据
632
按照目标容器对应的填充规则，将容器数据填充至目标容器，并根据填充结果生成与压缩任务关联的压缩数据包

图6

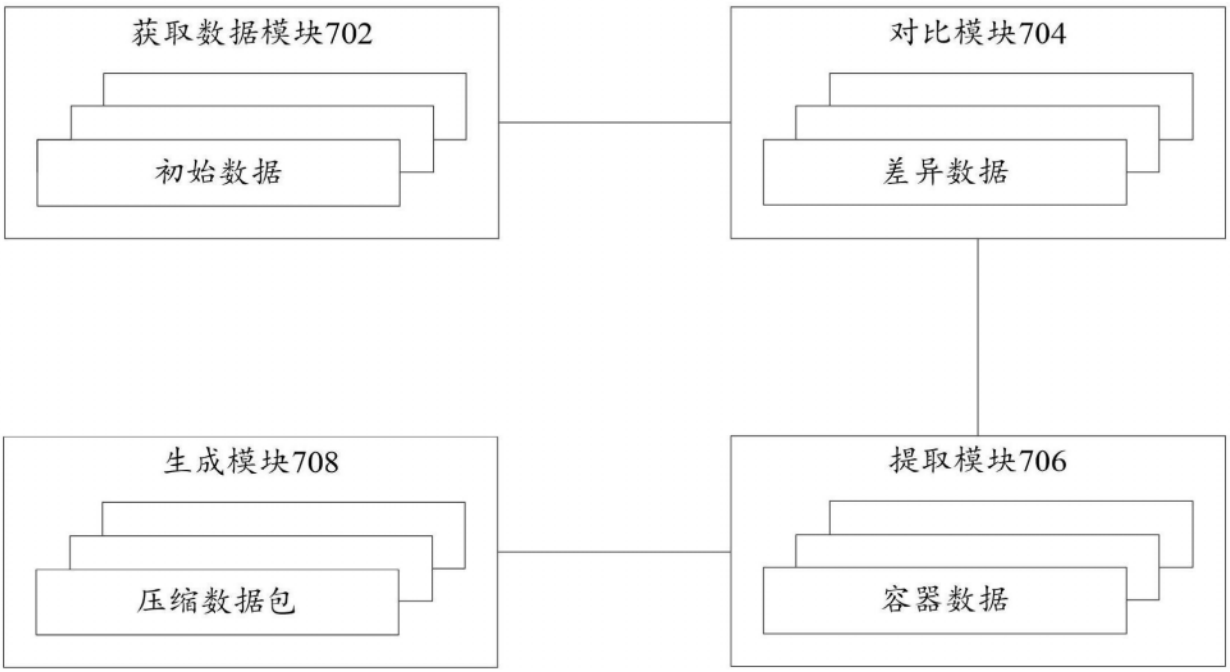

获取数据模块702
初始数据
对比模块704
差异数据
生成模块708
压缩数据包
提取模块706
容器数据

图7

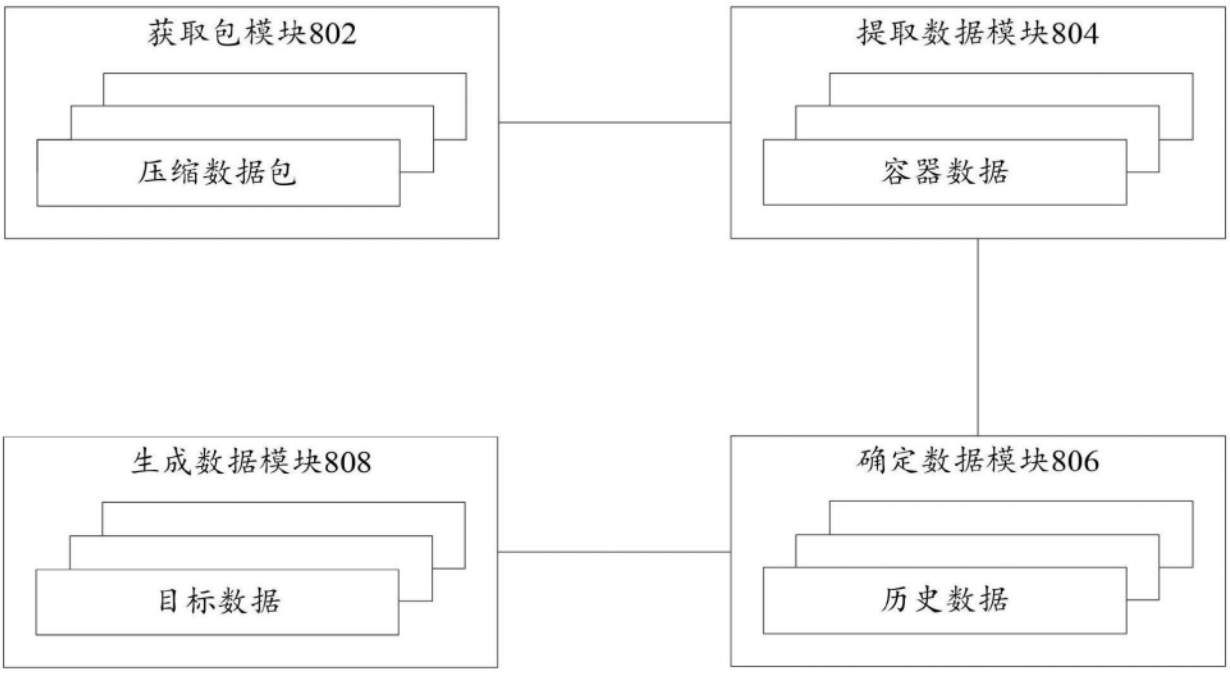

获取包模块802
压缩数据包
提取数据模块804
容器数据
生成数据模块808
目标数据
确定数据模块806
历史数据

图8

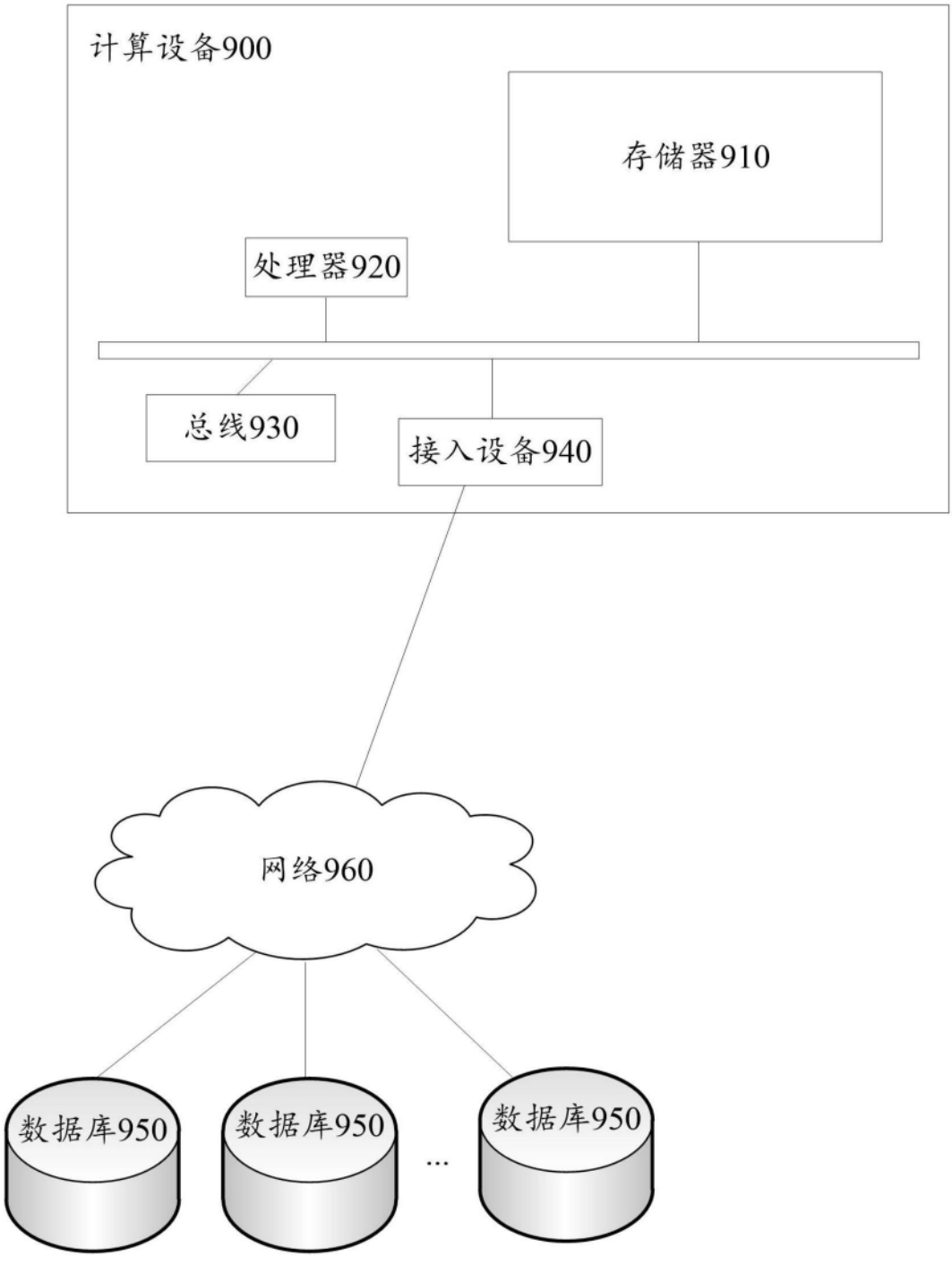

计算设备900
存储器910
处理器920
总线930
接入设备940
网络960
数据库950
数据库950
...
数据库950

图9