(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115775170 A
(43)申请公布日 2023.03.10

(21)申请号 202111040107.5

(22)申请日 2021.09.06

(71)申请人 北京橙心无限科技发展有限公司
地址 100120 北京市朝阳区利泽西街6号院3号楼17层1701内8

(72)发明人 庞勇超

(74)专利代理机构 北京超成律师事务所 11646
专利代理师 王晓菲

(51)Int.Cl.
G06Q 30/0601(2023.01)

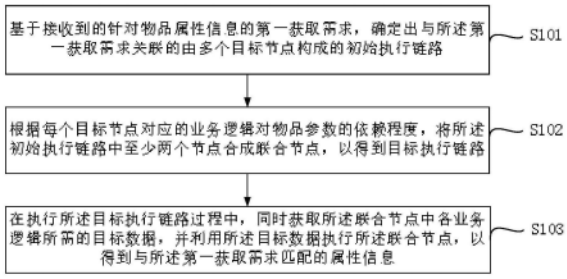
权利要求书2页 说明书18页 附图2页

(54)发明名称

一种物品属性信息的获取方法、装置、电子设备及介质

(57)摘要

本申请涉及数据处理技术领域，尤其涉及一种物品属性信息的获取方法、装置、电子设备及介质，通过基于针对物品属性信息的第一获取需求，可以确定出由多个目标节点构成的初始执行链路，进而，根据目标节点的业务逻辑对物品参数的依赖程度，将初始执行链路中至少两个目标节点合成联合节点，得到目标执行链路，进一步地，在执行目标执行链路过程中，同时获取联合节点中各业务逻辑所需的目标数据，并利用目标数据执行联合节点，以得到与第一获取需求匹配的属性信息。这样，通过两次链路编排来减少整体执行节点的数量，可以缩短链路执行的时间，而且通过提前获取执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。



CN 115775170 A

1.一种物品属性信息的获取方法,其特征在于,所述获取方法包括:

基于接收到的针对物品属性信息的第一获取需求,确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路;

根据每个目标节点对应的业务逻辑对物品参数的依赖程度,将所述初始执行链路中至少两个目标节点合成联合节点,以得到目标执行链路;

在执行所述目标执行链路过程中,同时获取所述联合节点中各业务逻辑所需的目标数据,并利用所述目标数据执行所述联合节点,以得到与所述第一获取需求匹配的属性信息。

2.根据权利要求1所述的获取方法,其特征在于,所述第一获取需求中携带有满足需求的待获取物品的条件信息,以及物品属性信息的属性标识;所述基于接收到的针对物品属性信息的第一获取需求,确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路,包括:

根据所述条件信息和所述属性标识,确定与所述第一获取需求匹配的多个业务逻辑;

确定与所述第一获取需求匹配的每个业务逻辑对应的所述目标节点;

将各个所述目标节点进行执行顺序的初始编排,生成所述初始执行链路。

3.根据权利要求1所述的获取方法,其特征在于,根据以下步骤合成所述联合节点:

确定所述初始执行链路中用于合成所述联合节点的至少两个节点分别对应的业务逻辑;

根据确定出的各业务逻辑的逻辑内容对联合配置模板进行配置,生成所述联合节点。

4.根据权利要求1所述的获取方法,其特征在于,根据以下步骤在执行所述目标执行链路过程中,同时获取所述联合节点中各业务逻辑所需的目标数据:

根据所述联合节点执行对应的各个业务逻辑所需的数据量,开启多个用于获取所述目标数据的目标线程;

通过开启的多个目标线程,同时获取所述联合节点中各业务逻辑所需的目标数据。

5.根据权利要求1所述的获取方法,其特征在于,

在所述获取所述联合节点中各业务逻辑所需的目标数据之后,所述获取方法还包括:

将所述目标数据存储在所述联合节点的缓存区中;

根据以下步骤利用所述目标数据执行所述联合节点:

从所述缓存区中调取所述目标数据,以利用所述目标数据执行所述联合节点中的各个业务逻辑。

6.根据权利要求1所述的获取方法,其特征在于,所述目标节点包括功能节点和脚本节点;所述功能节点的业务逻辑写在聚合系统的文件中,所述脚本节点的业务逻辑写在第三方存储器的文件中。

7.根据权利要求1所述的获取方法,其特征在于,所述目标节点包括脚本节点;在所述利用所述目标数据执行所述联合节点,以得到与所述第一获取需求匹配的属性信息之后,所述获取方法还包括:

接收针对物品属性信息的第二获取需求;所述第二获取需求为对所述第一获取需求进行更新后的需求;

根据所述第二获取需求相对于所述第一获取需求的更新内容,确定需要调整业务逻辑的逻辑内容的脚本节点,并对所述脚本节点进行调整;

加载调整后的所述脚本节点，并对所述目标执行链路的链路配置进行更新；

执行更新后的所述目标执行链路，获取与所述第二获取需求匹配的属性信息。

8.一种物品属性信息的获取装置，其特征在于，所述获取装置包括：

确定模块，用于基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路；

生成模块，用于根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个目标节点合成联合节点，以得到目标执行链路；

获取模块，用于在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息。

9.一种电子设备，其特征在于，包括：处理器、存储器和总线，所述存储器存储有所述处理器可执行的机器可读指令，当电子设备运行时，所述处理器与所述存储器之间通过所述总线进行通信，所述机器可读指令被所述处理器运行时执行如权利要求1至7任一所述的物品属性信息的获取方法的步骤。

10.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器运行时执行如权利要求1至7任一所述的物品属性信息的获取方法的步骤。

一种物品属性信息的获取方法、装置、电子设备及介质

技术领域

[0001] 本申请涉及数据处理技术领域，尤其涉及一种物品属性信息的获取方法、装置、电子设备及介质。

背景技术

[0002] 在互联网商城中，比如互联网社区团购的商品展示页面，用户关注的是商品所展示出来商品的本身售卖属性、活动属性，以及不同的展示形式等，而这些商品的数据属性是从后端的多个数据系统聚合而来的，研究如何保证商品本身属性信息的完整、正确、高效的聚合，有助于应对获取需求的不断变化。

[0003] 通常，商品的数据属性信息维护在不同的后端系统上，如何高速支持线上的需求迭代、减少开发成本、如何进行聚合，根据用户所处的购物模块，定制化的返回商品数据属性信息变成了整个商品属性信息聚合系统业务全链路的重中之重。

[0004] 目前的商品数据属性的聚合方案无法灵活定制化的提供商品数据属性的返回，而且每新增一个获取需求，下游系统都需要进行开发，开发包括上线、测试和联调，这样需要花费较多时间成本，导致下游系统往往无法支持高速的需求迭代，这种模式的技术也导致后端系统编码内容的增多，后端系统的维护成本较大。

发明内容

[0005] 有鉴于此，本申请实施例至少提供一种物品属性信息的获取方法、装置、电子设备及介质，可以缩短链路执行的时间，而且通过提前获取执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。

[0006] 本申请主要包括以下几个方面：

[0007] 第一方面，本申请实施例提供一种物品属性信息的获取方法，所述获取方法包括：

[0008] 基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路；

[0009] 根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个目标节点合成联合节点，以得到目标执行链路；

[0010] 在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息。

[0011] 在一种可能的实施方式中，所述第一获取需求中携带有满足需求的待获取物品的条件信息，以及物品属性信息的属性标识；所述基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路，包括：

[0012] 根据所述条件信息和所述属性标识，确定与所述第一获取需求匹配的多个业务逻辑；

[0013] 确定与所述第一获取需求匹配的每个业务逻辑对应的所述目标节点；

[0014] 将各个所述目标节点进行执行顺序的初始编排，生成所述初始执行链路。

[0015] 在一种可能的实施方式中，根据以下步骤合成所述联合节点：

[0016] 确定所述初始执行链路中用于合成所述联合节点的至少两个节点分别对应的业务逻辑；

[0017] 根据确定出的各业务逻辑的逻辑内容对联合配置模板进行配置，生成所述联合节点。

[0018] 在一种可能的实施方式中，根据以下步骤生成所述目标执行链路：

[0019] 对所述联合节点中的各业务逻辑进行执行顺序的编排；

[0020] 根据未进行合成的各个所述目标节点和编排完成的联合节点，生成所述目标执行链路。

[0021] 在一种可能的实施方式中，根据以下步骤在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据：

[0022] 根据所述联合节点执行对应的各个业务逻辑所需的数据量，开启多个用于获取所述目标数据的目标线程；

[0023] 通过开启的多个目标线程，同时获取所述联合节点中各业务逻辑所需的目标数据。

[0024] 在一种可能的实施方式中，在所述获取所述联合节点中各业务逻辑所需的目标数据之后，所述获取方法还包括：

[0025] 将所述目标数据存储在所述联合节点的缓存区中；

[0026] 根据以下步骤利用所述目标数据执行所述联合节点：

[0027] 从所述缓存区中调取所述目标数据，以利用所述目标数据执行所述联合节点中的各个业务逻辑。

[0028] 在一种可能的实施方式中，根据以下步骤执行所述目标执行链路：

[0029] 按照所述联合节点、未被合成的各个所述目标节点分别对应的执行顺序，执行所述目标执行链路中的所述联合节点、未被合成的各个所述目标节点分别对应的业务逻辑。

[0030] 在一种可能的实施方式中，对于执行方式为串行方式的所述目标节点，根据以下步骤执行所述目标执行链路中的目标节点：

[0031] 将多个第一候选物品的物品标识一同输入所述目标节点，以对所述多个第一候选物品进行筛选。

[0032] 在一种可能的实施方式中，对于执行方式为并行方式的所述目标节点，根据以下步骤执行所述目标执行链路中的目标节点：

[0033] 从线程池中为所述目标节点分配一个线程，通过该线程来执行所述目标节对应点的业务逻辑。

[0034] 在一种可能的实施方式中，根据以下步骤执行所述联合节点：

[0035] 将各个第二候选物品的物品参数分别输入所述联合节点中，按照所述联合节点中的各个业务逻辑对应的执行顺序，执行所述联合节点中的各个业务逻辑。

[0036] 在一种可能的实施方式中，针对每个第二候选物品，所述按照所述联合节点中的各个业务逻辑对应的执行顺序，执行所述联合节点中的各个业务逻辑，包括；

[0037] 根据所述第二候选物品的物品参数，判断所述第二候选物品是否满足执行顺序排

在首个的业务逻辑；

[0038] 若满足，执行顺序排在下一个的业务逻辑；

[0039] 若不满足，不再执行下一个业务逻辑。

[0040] 在一种可能的实施方式中，在所述基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路之前，所述获取方法还包括根据以下步骤生成不同节点：

[0041] 确定各个业务所需的业务逻辑，根据不同业务逻辑对节点生成模板进行配置，生成不同节点；其中，所述节点生成模板中包括节点名称、节点地址信息、节点执行路径、节点执行算法。

[0042] 在一种可能的实施方式中，所述目标节点包括功能节点和脚本节点；所述功能节点的业务逻辑写在聚合系统的文件中，所述脚本节点的业务逻辑写在第三方存储器的文件中。

[0043] 在一种可能的实施方式中，所述目标节点包括脚本节点；在所述利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息之后，所述获取方法还包括：

[0044] 接收针对物品属性信息的第二获取需求；所述第二获取需求为对所述第一获取需求进行更新后的需求；

[0045] 根据所述第二获取需求相对于所述第一获取需求的更新内容，确定需要调整业务逻辑的逻辑内容的脚本节点，并对所述脚本节点进行调整；

[0046] 加载调整后的所述脚本节点，并对所述目标执行链路的链路配置进行更新；

[0047] 执行更新后的所述目标执行链路，获取与所述第二获取需求匹配的属性信息。

[0048] 在一种可能的实施方式中，根据以下步骤对所述脚本节点进行调整：

[0049] 根据所述更新内容确定业务需求；

[0050] 将所述脚本节点的业务逻辑修改为与所述业务需求匹配的业务逻辑。

[0051] 第二方面，本申请实施例还提供一种物品属性信息的获取装置，所述获取装置包括：

[0052] 确定模块，用于基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路；

[0053] 生成模块，用于根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个目标节点合成联合节点，以得到目标执行链路；

[0054] 获取模块，用于在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息。

[0055] 第三方面，本申请实施例还提供一种电子设备，包括：处理器、存储器和总线，所述存储器存储有所述处理器可执行的机器可读指令，当电子设备运行时，所述处理器与所述存储器之间通过所述总线进行通信，所述机器可读指令被所述处理器运行时执行上述第一方面或第一方面中任一种可能的实施方式中所述的物品属性信息的获取方法的步骤。

[0056] 第四方面，本申请实施例还提供了一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器运行时执行上述第一方面或第一方

面中任一种可能的实施方式中所述的物品属性信息的获取方法的步骤。

[0057] 本申请实施例提供的一种物品属性信息的获取方法、装置、电子设备及介质，在接收到业务端针对物品属性信息的获取需求时，系统无需开发，直接通过运行可复用的节点构成的执行链路，就可以得到与获取需求匹配的物品属性信息，与现有技术中每新增一个获取需求，下游系统都需要进行开发，需要花费较多时间成本，导致下游系统往往无法支持高速的需求迭代相比，本申请可以大大减少针对新需求的处理成本和维护成本；另外，本申请通过两次编排可以减少整体执行节点的数量，缩短执行链路的长度，可以缩短链路执行的时间，而且通过提前获取到执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。

[0058] 进一步，本申请中的节点包括脚本节点，在接收针对物品属性信息的第一获取需求时，可以直接从第三方脚本组件库中筛选出与第一获取需求匹配的脚本节点。本申请中的脚本节点为groovy文件，支持热加载、热部署，可动态编辑，且存储在聚合系统之外的第三方数据库中。本申请在需求发生改变时，通过更新第三方数据库中的groovy文件，就可以完成整体链路对相关节点的执行修改，因而，无需对聚合系统进行发布上线，与现有技术中只采用功能节点，当需求改变时需要更改聚合系统内部存储的功能节点，在更新功能节点时需要开发上线，才能满足新需求，且在上线过程中无法响应用户的访问请求，带来巨大流量损失相比，本申请可以提高系统级别的需求响应效率。

[0059] 为使本申请的上述目的、特征和优点能更明显易懂，下文特举较佳实施例，并配合所附附图，作详细说明如下。

附图说明

[0060] 为了更清楚地说明本申请实施例的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，应当理解，以下附图仅示出了本申请的某些实施例，因此不应被看作是对范围的限定，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他相关的附图。

[0061] 图1示出了本申请实施例所提供的一种物品属性信息的获取方法的流程图；

[0062] 图2示出了本申请实施例所提供的另一种物品属性信息的获取方法的流程图；

[0063] 图3示出了本申请实施例所提供的一种物品属性信息的获取装置的功能模块图；

[0064] 图4示出了本申请实施例所提供的一种电子设备的结构示意图。

具体实施方式

[0065] 为使本申请实施例的目的、技术方案和优点更加清楚，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，应当理解，本申请中的附图仅起到说明和描述的目的，并不用于限定本申请的保护范围。另外，应当理解，示意性的附图并未按实物比例绘制。本申请中使用的流程图示出了根据本申请的一些实施例实现的操作。应当理解，流程图的操作可以不按顺序实现，没有逻辑的上下文关系的步骤可以反转顺序或者同时实施。此外，本领域技术人员在本申请内容的指引下，可以向流程图添加一个或多个其他操作，也可以从流程图中移除一个或多个操作。

[0066] 另外，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。通常在

此处附图中描述和示出的本申请实施例的组件可以以各种不同的配置来布置和设计。因此，以下对在附图中提供的本申请的实施例的详细描述并非旨在限制要求保护的本申请的范围，而是仅仅表示本申请的选定实施例。基于本申请的实施例，本领域技术人员在没有做出创造性劳动的前提下所获得的全部其他实施例，都属于本申请保护的范围。

[0067] 本申请实施例下述方法、装置、电子设备或计算机可读存储介质可以应用于任何需要进行物品属性信息的获取的场景，比如，互联网社区团购场景，本申请实施例并不对具体的应用场景作限制，任何使用本申请实施例提供的物品属性信息的获取方法及装置的方案均在本申请保护范围内。

[0068] 值得注意的是，在本申请提出之前，现有方案中商品数据属性的聚合系统无法灵活定制化的提供商品数据属性的返回，而且每新增一个获取需求，下游系统都需要进行开发，开发包括上线、测试和联调，这样需要花费较多时间成本，导致下游系统往往无法支持高速的需求迭代，这种模式的技术也导致后端系统编码内容的增多，后端系统的维护成本较大。

[0069] 另外，不是每一个业务方都需要全量的商品数据信息，目前大部分的商品属性信息聚合系统，都有一个共同的问题，就是把全部的商品数据信息都返回给上游，上游系统按需所取商品属性字段。但是这样系统之间的交互方式上，全量的商品属性数据字段较多，在现如今高并发、高流量的模式下，这种系统之间的交互方式的网络开销较大，并且上游系统如果想对结果进行缓存时，下游系统返回的大量的无用数据就变成了缓存开销的成本，影响性能。

[0070] 针对上述问题，本申请实施例在上游系统提出新的业务场景需求时，商品属性信息聚合系统无需开发，只需要通过复用已有的节点，生成执行链路，即可获取满足新需求的物品属性信息，这样，可以大大减少针对新需求的处理成本和维护成本，减小网络开销，而且，通过两次链路编排来减少整体执行节点的数量，可以缩短链路执行的时间，而且通过提前获取执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。

[0071] 为便于对本申请进行理解，下面结合具体实施例对本申请提供的技术方案进行详细说明。

[0072] 图1为本申请实施例所提供的一种物品属性信息的获取方法的流程图。如图1所示，本申请实施例提供的物品属性信息的获取方法，包括以下步骤：

[0073] S101：基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路。

[0074] 这里，执行本申请的物品属性信息的获取方法的执行主体可以为物品属性信息聚合系统中的控制器，这里，物品属性信息聚合系统可以包括多个节点、控制器、执行器和存储器。

[0075] 需要说明的是，对于一个互联网物品展示页面而言，可能有多个不同的页面展示模块，通常，不同页面展示模块展示的内容不同，对应展示的商品属性信息聚合系统也不同，即，不同页面展示模块对应的业务方的获取需求会有所不同，且同一个页面展示模块在不同时间点针对商品属性信息聚合系统的获取需求也会发生变化，即，需求是会随着时间发生变化的。

[0076] 在具体实施中，在接收到一个针对物品属性信息的第一获取需求后，先根据第一获取需求，确定与第一获取需求关联的多个目标节点，进而，对关联的多个目标节点进行链路编排，生成初始执行链路。

[0077] 这里，本申请中的节点都是预先生成好的，且可复用的，在接收到一个针对物品属性信息的获取需求后，会确定出与该获取需求匹配的目标节点。初始执行链路仅是对目标节点的执行顺序和执行方式初步编排，初始执行链路可以理解为是一种链路框架。其中，执行方式包括串行执行和并行执行，各个串行执行的目标节点需按执行顺序依次执行；各个并行执行的目标节点可同时执行。

[0078] 进一步地，第一获取需求中携带有满足需求的待获取物品的条件信息，以及物品属性信息的属性标识；步骤S101中所述基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路，包括以下步骤：根据所述条件信息和所述属性标识，确定与所述第一获取需求匹配的多个业务逻辑；确定与所述第一获取需求匹配的每个业务逻辑对应的所述目标节点；将各个所述目标节点进行执行顺序的初始编排，生成所述初始执行链路。

[0079] 在具体实施中，在接收到一个业务端针对物品属性信息的第一获取需求后，会对第一获取需求进行解析，确定出第一获取需求中描述的关于待获取物品的一些条件信息，以及物品属性信息的属性标识，即，接收到针对物品属性信息的第一获取需求后，要确定出哪些物品符合获取条件，以及要确定出哪些物品属性信息是要获取的对象，进而，可以根据这些条件信息和属性标识，确定与第一获取需求匹配的多个业务逻辑，并确定与第一获取需求匹配的每个业务逻辑对应的目标节点，并将各个目标节点按照执行顺序进行初始编排，可以生成初始执行链路。

[0080] 其中，商品属性信息聚合系统为物品的特性信息、标签信息，属性信息比如名称、价格、产地、重量、储存方式、活动信息等等。待获取物品的条件信息可以为筛选信息、性质信息，条件信息比如库存过滤、黑名单过滤、产地过滤、物品类型过滤等。例如，某一页面展示模块只展示有库存的、不是黑名单的、非肉类的物品，这些限制条件信息即为待获取物品的条件信息。

[0081] 这里，获取需求可以用领域特定语言(Domain Specific Language，DSL)语句进行声明，声明的描述中会带有满足获取需求的待获取物品的条件信息，以及物品属性信息的属性标识。

[0082] 需要说明的是，这里，每个节点对应至少一种业务逻辑，业务逻辑比如是获取物品的某个属性信息的逻辑，也可以是对物品进行筛选的逻辑，也可以是对物品添加某个标签的逻辑等，这样，可以找到与各个条件信息匹配的节点，以及找到获取或添加物品属性信息匹配的节点。

[0083] 需要说明的是，节点是预先建立好且可复用的，当接收到获取需求时，系统无需开发，直接通过运行可复用的节点，就可以得到与获取需求匹配的各个待获取物品的物品属性信息，与现有技术中每新增一个获取需求，下游系统都需要进行开发，需要花费较多时间成本，导致下游系统往往无法支持高速的需求迭代相比，本申请可以大大减少针对新需求的处理成本和维护成本。

[0084] 这里，每个节点具有一定的功能，具体地，将物品信息聚合的需求进行抽象，根据

不同的业务规则进行模块划分，形成多个类别的业务节点，包括但不限于过滤功能、物品属性信息获取功能、物品循环处理功能、业务逻辑处理功能以及脚本功能。不同功能模块内部都包含了多个的节点，每个节点都有自己的职责和任务，不同节点互不依赖，独立存在。

[0085] S102：根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个目标节点合成联合节点，以得到目标执行链路。

[0086] 在具体实施中，根据各个目标节点对应的业务逻辑对物品参数的依赖程度，对各个目标节点进行分类，具体地，将对物品参数依赖程度高的节点归为一类，将对物品参数依赖程度低的节点归为一类，并且将对物品参数的依赖程度高的节点进行合成得到联合节点，并根据未被合成的目标节点和联合节点对初始执行链路进行更新，生成目标执行链路，这个目标执行链路就是最终用来获取第一获取需求匹配的物品属性信息的链路。这里，目标执行链路相比初始执行链路，大大减少了链路中包含的节点数量，使得链路长度大大缩小，这样，由于每个节点在执行对应业务逻辑时都会花费一定时间，所以，目标执行链路的执行时间会比初始执行链路的执行时间大大缩短，可以提高链路执行的效率。

[0087] 这里，对物品参数依赖程度高是说这个对应的节点在执行自己的业务逻辑时，需要依赖物品参数才能执行，比如，库存过滤节点需要物品的库存信息才能执行库存过滤逻辑；对物品参数依赖程度低是说这个对应的节点在执行自己的业务逻辑时，仅需要物品标识就能执行，比如黑名单节点只需要物品的ID就能执行黑名单过滤逻辑。

[0088] 这里，对传统模式下简单链路编排流程进行说明，比如，有5个串行执行的节点，每个节点都要对该物品处理一次，这样，就意味着对于每个物品都要进行5次处理，在线上实际场景中，高并发大量的物品属性信息的聚合请求，任何一个请求都要经过多次处理来满足业务需求，这个系统级别的开销成本是巨大的，可能导致线上耗时严重，导致业务链路执行超时，这种方式是极其耗时的，提高了整体的系统级别响应时间。

[0089] 这里，对于传统模式下简单链路编排出现的问题，即，在传统的节点编排流程中，无法对于节点内部进行编排。一旦遇到了多个处理节点的业务模式，就需要进行多次循环处理(每个物品都需要经过多个节点处理，多个物品的处理相当于多次循环该多个节点进行处理)，这种产生的代价是无法作用在线上的业务的。对于最底层的物品属性信息的聚合服务来说，超过100ms的响应速度都会影响用户的体验。基于此，本申请提出了联合节点来处理复杂的业务逻辑内容。与传统的能力编排模式，只能编排不同节点之间的执行顺序和执行方式，无法深入节点内部的逻辑编排，导致复杂的业务场景出现，就无法编排链路实现相比，本申请将对物品参数的依赖程度高的目标节点进行结合，生成一个联合节点，这里，联合节点对应有被合成的目标节点的业务逻辑，在联合节点执行时，可以根据各个业务逻辑之间的关系，比如，数据相互依赖的情况，可以对联合节点的各个业务逻辑进行动态编排，这样，相比被合成的多个目标节点的执行效率总和而言，联合节点的执行效率更高。其中，联合节点通过遍历一次一个物品的参数能够执行被合成的全部目标节点的内部逻辑。

[0090] 这里，对本申请利用联合节点编排流程进行说明，传统编排模式不考虑业务执行效率，只为了实现编排模式，导致一个业务如果采用传统的编排模式，会导致执行链路过长，经过本申请的联合配置模板的内部逻辑编排，可以看出来，传统多次执行(几个节点几次执行)对于本申请是1次执行(多个节点合成一个联合节点，执行1次)，对于执行效率来说，本着系统级别毫秒必争的道理，运行速度明显得到了提高。

[0091] 进一步地，下面具体说明生成联合节点的过程，确定所述初始执行链路中用于合成所述联合节点的至少两个节点分别对应的业务逻辑；根据确定出的各业务逻辑的逻辑内容对联合配置模板进行配置，生成所述联合节点。

[0092] 这里，联合节点生成的具体的技术实现上分为初始化与执行两个流程；1)初始化的时候要根据上游系统所配置的DSL语句，解析出要被合成的各目标节点的节点内容，将解析出来的内容与上游系统绑定，形成一个键值对的结构，之后通过全局单例设计模式的方式，全局储存一份在内存中；2)执行的时候，当执行到了一个目标节点对应的业务逻辑，通过静态引入的全局内存，从内存中读取本执行的上游的配置，之后根据动态代理的方式进行执行。

[0093] 进一步地，根据以下步骤生成所述目标执行链路：对所述联合节点中的各业务逻辑进行执行顺序的编排；根据未进行合成的各个所述目标节点和编排完成的联合节点，生成所述目标执行链路。

[0094] 在具体实施中，在生成联合节点后，按照联合节点中各个业务逻辑之间的逻辑关系，以及相互对数据的依赖关系，先对联合节点中的各业务逻辑进行执行顺序的编排，之后，根据未进行合成的各个目标节点和编排完成的联合节点编排后形成目标执行链路，这个目标执行链路就是用来获取与第一获取需求匹配的物品属性信息的。

[0095] 需要说明的是，执行链路对于某个需求来说，它所需要的是物品属性信息，在本申请中，是通过多个节点进行编排，形成的一条可执行的链路。

[0096] 进一步地，根据以下步骤执行所述目标执行链路：按照所述联合节点、未被合成的各个所述目标节点分别对应的执行顺序，执行所述目标执行链路中的所述联合节点、未被合成的各个所述目标节点分别对应的业务逻辑。

[0097] 在具体实施中，在基于第一获取需求，确定出联合节点、未被合成的各个目标节点分别对应的执行顺序之后，可以根据这些联合节点、未被合成的各个目标节点分别对应的执行顺序进行执行链路编排，编排后生成目标执行链路。这里，不同功能的节点互不依赖、独立存在，再通过按照DSL语句进行排列组合的方式进行编排，就形成了一条可执行链路。

[0098] 需要说明的是，本申请通过提出联合节点的概念，重新定义业务中的循环处理的逻辑，通过这样的计，可以在联合节点执行的行为中去继续编排业务逻辑，这是二次编排业务的模式，这种方式与传统方式的区别在于，1)传统模式是一次编排模式，本申请通过2次编排模式，链路编排+业务逻辑编排；2)传统方式编排的仅仅只是执行链路，一旦业务逻辑复杂，整体的业务模式链路过长，执行的循环次数过多，系统的执行开销就巨大、耗时提高，系统并发度减少，无法支持线上业务的实现，两次编排模式可以让整体流程节点进一步减少，优化缩短执行链路。提升执行的效率；3)联合节点内部编排的是功能节点内的执行逻辑，缩短了正常业务的执行节点，通过提出的联合配置模板，专用于解决复杂的业务循环逻辑。通过两次链路编排来减少整体执行节点的数量，可以缩短链路执行的时间，而且通过提前获取执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。

[0099] S103：在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息。

[0100] 在具体实施中，引入了联合节点概念，在加载联合节点时就可以事先知晓即将要执行哪些业务逻辑，因而可以在执行的行为中去继续编排业务逻辑，即，一边从外部数据源获取各业务逻辑所需的数据，一边动态编排各个业务逻辑的执行顺序，而且，也可以提前获取联合节点中各业务逻辑要用到的数据信息并存储在联合节点的缓存区中，这样，在执行联合节点时，可以直接从联合节点中的缓存区进行调取，这样可以加快执行效率。

[0101] 需要说明的是，物品属性信息聚合系统还包括执行器层，执行器层的主要作用就是执行不同的节点组合产生的可执行链路。因为每一个可执行链路都是可以被动态修改的，为了保证需求变更，后端系统能够高效支持上游需求，所以需要一套稳定的执行器来执行，这样随着链路变更，物品属性信息聚合系统就不需要开发上线，只需重新生成可执行链路即可。

[0102] 这里，链路类就是定义了一条可执行链路，它的形成依赖于业务线声明的执行链路，比如某一个业务线的需求中说，需要从多个下游系统获取到不同的商品属性信息聚合系统做一次聚合，具体的执行流程是，先串行执行过滤节点A、过滤节点B、过滤节点C，之后需要并行去调用不同的下游系统，获取到所依赖的数据内容，如并行调用，获取物品活动属性信息，获取商品库存属性信息，获取商品标签属性信息，之后将所有物品实体进行循环遍历，执行某一些业务所需要的逻辑功能节点，最后返回结果。这一条业务线的链路编排中涉及了串行、并行，循环多种执行的方式，在本申请中会通过声明的DSL语句，来声明上面描述的执行链路，经过DSL语句的描述，排列组合下来的链路就是执行链路。

[0103] 进一步地，对于执行方式为串行方式的所述目标节点，根据以下步骤执行所述目标执行链路中的目标节点：将多个第一候选物品的物品标识一同输入所述目标节点，以对所述多个第一候选物品进行筛选。

[0104] 在具体实施中，对于执行方式为串行方式的各个目标节点，按照各个目标节点的执行顺序，将第一候选物品的物品标识先输入第一个目标节点中，通过第一个目标节点对第一候选物品的筛选后，将通过筛选的第一候选物品的物品标识再输入第二个目标节点中，对通过第一次筛选出的第一候选物品进行第二次筛选，直到输入最后一个目标节点，来完成对第一候选物品的筛选。

[0105] 这里，对于执行方式为串行方式的各个目标节点而言，会从上下文中获取到要执行的各个目标节点的执行顺序，之后进行按照顺序依次执行。

[0106] 进一步地，对于执行方式为并行方式的所述目标节点，根据以下步骤执行所述目标执行链路中的目标节点：从线程池中为所述目标节点分配一个线程，通过该线程来执行所述目标节对应点的业务逻辑。

[0107] 在具体实施中，对于属于并行方式的各个目标节点，可以同时运行这些目标节点，具体地，可以从线程池中分别为每个目标节点分配一个线程，通过各个线程来运行各个目标节点的业务逻辑。

[0108] 这里，对于执行方式为并行方式的各个目标节点而言，各个目标节点是并行进行执行的，会从上下文中获取要同时执行的目标节点，为每一个目标节点从池子中分配一个线程来执行，这样可以保证同一时间，可以执行多个目标节点，而不需要进行等待。并行和串行最大的区别就是，串行需要通过循环遍历依次等待，并行通过线程池的方式，实现不需要等待。

[0109] 进一步地，根据以下步骤执行所述联合节点：将各个第二候选物品的物品参数分别输入所述联合节点中，按照所述联合节点中的各个业务逻辑对应的执行顺序，执行所述联合节点中的各个业务逻辑。

[0110] 在具体实施中，在利用串行执行方式的目标节点对第一候选物品进行筛选之后，得到第二候选物品，进而，将第二候选物品的参数集合输入联合节点中，以各个第二候选物品为循环实体，将每个第二候选物品输入联合节点中进行处理，即，每个第二候选物品都需要通过联合节点来处理，可以是对第二候选物品进行筛选，和/或获取第二候选物品的物品属性信息。

[0111] 这里，与未被合成的目标节点不同，联合节点的执行是以物品实体为循环对象，每个物品实体都需要经过联合节点的处理，而未被合成的目标节点通过遍历各个物品的物品标识，对物品进行筛选，未被合成的目标节点的执行方式主要适用于需要对与物品内部的某一个属性做业务逻辑处理。联合节点的设计，主要为了解决单功能节点内部业务逻辑无法进行编排的情况，比如在社区团购场景下，线上的业务往往要求高速的系统响应结果。尽量减少循环处理的次数以提高系统级别的响应效率。

[0112] 进一步地，针对每个第二候选物品，所述按照所述联合节点中的各个业务逻辑对应的执行顺序，执行所述联合节点中的各个业务逻辑，包括；根据所述第二候选物品的物品参数，判断所述第二候选物品是否满足执行顺序排在首个的业务逻辑；若满足，执行顺序排在下一个的业务逻辑；若不满足，不再执行下一个业务逻辑。

[0113] 在具体实施中，在通过联合节点对第二候选物品进行处理时，针对每个第二候选物品，根据该第二候选物品的物品参数，判断该第二候选物品是否满足第一个逻辑条件，若满足执行下一个逻辑条件的判断，直至执行完所有的逻辑条件；若不满足结束对第二候选物品的逻辑条件的判断。这里，是对物品是否要执行联合节点的各个业务逻辑而进行的判断，可以减少不必要的计算，进一步提升联合节点的执行效率。

[0114] 需要说明的是，通过两次链路编排来减少整体执行节点的数量，可以缩短链路执行的时间，而且通过提前获取执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。

[0115] 进一步地，根据以下步骤在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据：根据所述联合节点执行对应的各个业务逻辑所需的数据量，开启多个用于获取所述目标数据的目标线程；通过开启的多个目标线程，同时获取所述联合节点中各业务逻辑所需的目标数据。

[0116] 在具体实施中，在执行联合节点的过程中，要获取联合节点中各个业务逻辑所需要的目标数据，这里，目标数据是包括物品的参数信息的数据，在从各个外部数据源获取目标数据时，由于目标数据的数据量较大，为了节省获取的时间，可以将获取目标数据的任务拆分成多个子任务，并为每个子任务分配一个线程来执行，这样，可以大大提升目标数据获取的速度，具体地，可以根据联合节点执行对应的各个业务逻辑所需的数据量的大小，来确定开启目标线程的数量，进而，通过开启的多个目标线程，可以同时来共同获取联合节点中各业务逻辑所需的目标数据。

[0117] 进一步地，在所述获取所述联合节点中各业务逻辑所需的目标数据之后，所述获取方法还包括：将所述目标数据存储在所述联合节点的缓存区中；根据以下步骤利用所述

目标数据执行所述联合节点:从所述缓存区中调取所述目标数据,以利用所述目标数据执行所述联合节点中的各个业务逻辑。

[0118] 在具体实施中,在执行目标执行链路时,向数据源服务器获取联合节点所需的目标数据,这里,可以开启多个线程,将获取任务进行拆分后分给多个线程,进而,可以一次性获取目标数据,并将获取的目标数据放置在联合节点中的缓存区,这样,可以直接从缓存中调取联合节点中各个业务逻辑在执行时所需的数据,并在所有要处理的物品都完成经过联合节点中的全部业务逻辑的处理后,释放缓存区中的目标数据,这样可以加快执行效率。

[0119] 在本申请实施例中,通过基于针对物品属性信息的第一获取需求,可以确定出由多个目标节点构成的初始执行链路,进而,根据目标节点的业务逻辑对物品参数的依赖程度,将初始执行链路中至少两个目标节点合成联合节点,得到目标执行链路,进一步地,在执行目标执行链路过程中,同时获取联合节点中各业务逻辑所需的目标数据,并利用目标数据执行联合节点,以得到与第一获取需求匹配的属性信息。这样,通过两次链路编排来减少整体执行节点的数量,可以缩短链路执行的时间,而且通过提前获取执行链路所需的全部数据,可以大大提升执行的效率,进而提高获取物品属性信息的速度。

[0120] 图2为本申请实施例所提供的一种物品属性信息的获取方法的流程图。如图2所示,本申请实施例提供的物品属性信息的获取方法,包括以下步骤:

[0121] S201:接收针对物品属性信息的第二获取需求;所述第二获取需求为对所述第一获取需求进行更新后的需求。

[0122] S202:根据所述第二获取需求相对于所述第一获取需求的更新内容,确定需要调整业务逻辑的逻辑内容的脚本节点,并对所述脚本节点进行调整。

[0123] S203:加载调整后的所述脚本节点,并对所述目标执行链路的链路配置进行更新。

[0124] S204:执行更新后的所述目标执行链路,获取与所述第二获取需求匹配的属性信息。

[0125] 在具体实施中,当接收到针对物品属性信息的一个更新后的获取需求时,即第二获取需求,第二获取需求为对第一获取需求进行调整后的需求,可以根据第二获取需求相对于第一获取需求的调整内容,确定需要调整业务逻辑的目标脚本节点,并以此对目标脚本节点进行调整,加载调整后的目标脚本节点,这里,脚本节点存储在第三方数据库,在对目标脚本节点进行修改时,不会影响物品属性信息聚合系统的正常运行,进而,直接对目标执行链路的链路配置进行更新,运行更新后的目标执行链路,就可以获取与第二获取需求对应的各个待获取物品的物品属性信息。

[0126] 进一步地,在接收获取需求之前,还包括根据以下步骤生成不同节点:确定各个业务所需的业务逻辑,根据不同业务逻辑对节点生成模板进行配置,生成不同节点;其中,所述节点生成模板中包括节点名称、节点地址信息、节点执行路径、节点执行算法。

[0127] 在具体实施中,节点是预先建立好且可复用的,在最初建立各个节点时,先搭建一个节点生成模板,进而,根据不同的业务逻辑,对节点生成模板进行不同的配置,生成不同节点。

[0128] 需要说明的是,功能节点利用抽象出来的方法通过聚合系统内部的一些模版定义,被声明成的一个可复用的节点。功能节点的使用方是上游业务系统按照自己的特定需求来选择的,通常,功能节点只具备处理某一个业务逻辑能力,比如“过滤下架物品”的功能

节点,这个功能节点的职责就只是过滤掉下架的物品,不会处理其他业务逻辑,其他的业务逻辑也有特定的节点来执行。

[0129] 这里,对聚合系统声明的不同节点的过程进行说明,节点类就是定义了节点的节点生成模板,该节点生成模板中拥有节点的名称、身份标识字段、真实的节点的class类路径和真实的节点引用实体。节点类是一个抽象的模板父类,具体的执行方法交由真实的节点组件生成模板类来实现。节点组件类是一个抽象类,内部声明了一个抽象方法,不同的节点会继承这个抽象类,实现抽象类定义好的执行行为,在这个方法中,不同的节点比如节点A、节点B、节点C,都会编写自己特定的业务逻辑。

[0130] 进一步地,目标节点包括功能节点和脚本节点;所述功能节点的业务逻辑写在聚合系统的文件中,所述脚本节点的业务逻辑写在第三方存储器的文件中。

[0131] 需要说明的是,节点分为功能节点和脚本节点;功能节点是根据不同业务逻辑抽象出来的方法,被范称为某一个功能的功能节点,功能节点存储于聚合系统的内部;脚本节点是可拓展的功能脚本,可以通过、编译、动态加载的方式生成,具体地,是采用grovvy脚本方式生成的,支持热加载、热部署,存储于第三方数据库,比如第三方脚本组件库。

[0132] 这里,脚本节点是采用grovvy脚本方式生成的,存储于第三方数据库,这样,在线上业务的获取需求发生改变时,只需要动态修改脚本节点,之后加载修改后的脚本节点,就能满足新的获取需求,无需对聚合系统进行发布上线,与现有技术中只采用功能节点,当需求改变时需要更改聚合系统内部存储的功能节点,在更新功能节点时需要开发上线,才能满足新需求,且在上线过程中无法响应用户的访问请求,带来巨大流量损失相比,本申请可以提高系统级别的需求响应效率。

[0133] 这里,本申请通过定制化的groovy脚本节点,不用在需求发生变更时,就开发新的功能节点,减少了系统交互之间的网络传输的带宽,并且减少了上游系统所需要缓存的对象大小,以及系统交互之间的成本。

[0134] 另外,在互联网商城,比如在互联网社区团购场景下,不同业务模块的界面获取需求不一样,所以对于返回结果组件要支持动态做配置。在传统的模式下,这就意味着要开发N个返回结果组件进行编排才能达到业务要求效果,并且每一个需求的开发都需要上线发布,这大大的提高了需求的迭代效率和开发成本,并且降低了系统的灵活性,使系统的某些功能在系统中膨胀。对此,本申请除了使用功能节点,还使用了脚本节点,脚本节点可以灵活进行编辑,脚本节点的修改后不需要上线发布聚合系统,可以达到灵活多变的定制化支持不同的需求,并且可以减少聚合系统的开发量,提高需求迭代效率。

[0135] 需要说明的是,本申请中的功能节点对应的是稳定不变的业务逻辑,脚本节点对应的是灵活多变的业务逻辑,所以,通常,无需对功能节点进行修改,修改的都是脚本节点。如果只有功能节点,且功能节点存储于聚合系统的本地,当变更时,就需要新开发功能节点或更改功能节点,在更新功能节点时,即,更新聚合系统,使得聚合系统需要进行上线、下线,而上线下线过程中,聚合系统都是无法响应用户的访问请求的,这样的方式严重损失了访问流量;对此,本申请中的节点还包括脚本节点,在需求发生变更,脚本节点更新时,由于脚本节点存储于第三方数据库,不会影响聚合系统的正常运行,当有新需求时,秩序加载更新后的脚本节点即可,所以聚合系统无需上线下线环节。

[0136] 其中,所述执行方式包括串行方式、并行方式和循环方式。这里,不同节点的执行

方式由于其业务逻辑的不同，会有不同的执行方式，有的节点是按照先后顺序依次执行的，上一个节点的输出会作为下一个节点的输入，这样的节点的执行方式为串行方式；有的节点的业务逻辑互不影响，为了提升执行效率，可以将这些节点一同执行，将这些节点的输出进行汇总，比如用于属性信息获取的节点，这样的节点的执行方式为并行方式；还有的节点要多次循环进行执行，比如用于过滤库存的节点，对每个物品进行过滤时都需要使用，这样的节点的执行方式为循环方式。需要说明的是，本申请支持串行方式、并行方式和循环方式这三种执行方式来进行链路编排，可以覆盖到全部业务线场景的需求。

[0137] 进一步地，根据以下步骤对所述脚本节点进行调整：根据所述更新内容确定业务需求；将所述脚本节点的业务逻辑修改为与所述业务需求匹配的业务逻辑；其中，不同业务逻辑对应的所述目标脚本节点返回的物品属性信息不同。

[0138] 这里，本申请实施例中所有的基础能力节点是物品属性信息聚合系统的基础能力，对于需求的提出，物品属性信息聚合系统不再需要特意开发，支持线上需求，只需要新增一个对应需求的DSL流程编排配置语句，把所要执行的链路节点声明好，就可以快速支持线上业务的迭代。如果后续随着业务的发展，这个需求要进行新的迭代，需要对于整体内容流程进行修改，可能会不需要对物品进行比如“下架过滤处理”或者不需要对物品进行“判断当前的物品是否是黑名单物品，如果存在进行过滤”的能力执行节点，采用本申请中的方案，物品属性信息聚合系统无需开发，只需要重新生成一条新的操作执行链路，即可完成针对新需求的物品属性信息的获取和聚合。

[0139] 这里，具体对脚本节点进行详细说明，脚本节点是groovy脚本编辑的，它的引入让整个物品属性信息聚合系统更加灵活，不在局限于组件与组件之间的能力编排。线上有这么一个实际情况，要给不同的业务方返回不同的结果对象。对于本申请，比如购物车场景下，只需要返回物品的库存，用户加购数量，以及物品图片和价钱；但是首页中，需要返回物品的全部属性信息；但是在用户的下单中，只需要返回物品的图片，名称以及价钱。这种场景对于传统的编排能力来说，每一个场景返回情况，都需要开发成一个组件，编排进整体执行链路中，才能达到效果。这样意味着每新增一个上游业务方，就需要新增一个执行脚本，组件能力的复用产生了局限性。这种模式对于开发成本来说是巨大的，如果某一天线上业务进行了迭代，仅仅只需要多新增一个返回结果字段，在传统模式下，仍需要开发上线，才能解决，一个系统的发布上线，所带来的流量损失是巨大的，这个问题如果不解决，频繁的发布上线，会产生极差劲的用户体验。在互联网商城的聚合服务场景下，一个系统要能够承载上万每秒访问量，具有上百台机器，对于非核心的模型组件，如果仍然采用传统模式开发组件，进行编排上线，就会导致技术跟不上业务。

[0140] 这里，本申请采用了grovvy脚本的方式，支持热加载、热部署，采用grovvy脚本方式生成的可执行节点，可以不用对系统进行发布上线，通过更新系统缓存中的节点文件，就可以完成整体链路对相关节点的执行修改。增强了整体编排流程的组件灵活多变性；对于业务组件这种十分灵活，要不停变化的组件，采用grovvy脚本的方式来提供支持，提高系统级别的需求响应效率；采用grovvy脚本生成的组件，不再会因为哪怕业务中需要一个字段属性的变动，仍需要让系统进行发布上线的流程。也就是说，在本申请中，只需要通过修改执行链路的配置，即可配合上游系统完成业务迭代的需求。

[0141] 这里，本申请提供的物品属性信息聚合系统，将采用组件化的思维方式，首先将获

取物品某一个数据属性信息的能力抽象成不同的节点，这些节点的职责就是向不同的物品属性维护系统获取基础信息。根据获取需求，编排这些节点，通过节点的排列组合，定制化的形成一条或者多条可执行的获取物品属性信息的执行链路，每一条执行链路的返回结果对于用户所能看到的数据信息是不同的。并且上游系统可以通过groovy可拓展功能的脚本节点，控制具体业务所需要的物品属性信息，之后物品属性信息聚合系统会通过数据聚合引擎，来执行这个执行链路，获取全部的物品数据属性信息，这样系统内部可以减少代码的维护成本。可以通过编排的方式，复用不同的节点，高效稳定的支持业务的迭代。通过groovy可拓展功能的脚本节点，各个业务需求实现不同的结果返回内容，减少网络流转开销。并且通过DSL语句编排的执行链路在本申请物品属性信息聚合统中是无需进行上线开发的，通过第三方配置如Apollo或者redis，可以实现动态加载更新执行链路的方式，减少对于需求的迭代成本，和后端系统开发成。

[0142] 一示例中，针对物品属性信息的聚合系统中为解决多变的物品属性信息聚合需求，而提出的一个链路流程编排，这里，物品属性信息聚合系统中会包含多个可复用的节点，所有的节点都可以进行逻辑编排，随意排列组合形成一条执行链路。这种方式，不再像其他大部分系统，某一个需求的提出，就需要物品聚合系统进行开发上线，某一个已经存在的需求，但是现在有变动，又需要物品属性信息聚合系统的开发上线。本申请通过这种节点编排的方式，可以有三种执行方式，串行、并行、循环执行。这三种执行方式的能力支持，可以覆盖目前市场业务的全量需求编排，并且会通过动态加载，及时更新业务的编排链路。图1中所表示的含义，首先，依次执行“过滤节点A”—“过滤节点B”—“过滤节点C”—“过滤节点D”之后，会根据业务线的要求，从“缓存中获取商品”和“非缓存中获取商品”选择一个节点执行，之后执行“后置处理节点”之后并行的调用“业务逻辑节点A”“业务逻辑节点B”“业务逻辑节点C”“业务逻辑节点D”，之后在执行联合配置模板。联合配置模板内部中，会以物品实体为循环基础，每一个物品都会依次执行，“循环过滤节点E”、“循环过滤节点F”、“循环过滤节点G”、“业务逻辑节点E”、“业务逻辑节点F”、“业务逻辑节点G”，最后执行为业务线所定制化的groovy脚本节点，处理业务线所需要的字段内容。

[0143] 另一示例中，在一个业务模块展示的物品，需要对物品进行“下架过滤处理”、“判断当前的物品是否存在黑名单物品，如果存在进行过滤”，并行的去“获取物品的活动属性”、“标签属性”，循环对物品进行“冻品过滤”、“活动属性的价格覆盖”、最终返回的“物品属性要包含，活动数据属性、库存属性、用户加购属性、标签属性”。

[0144] 针对上面示例，在本申请中，会对这个场景所提出的业务需求结合物品属性信息聚合系统中的节点进行排列组合，生成一个可执行的链路。当线上有对应的业务请求来获取物品的属性数据时，物品聚合会先获取对应的DSL配置内容生成的执行链路，之后根据执行链路的DSL语句来进行接续。流程如下。首先解析到了串行流程中，分别执行了“下架过滤处理”、“判断当前的物品是否存在黑名单物品，如果存在进行过滤”，之后解析到了并行执行逻辑，从线程池中获取到同时分配给“获取物品的活动属性”、“标签属性”的能力执行节点，之后把并行获取到的数据进行整合。最后解析到需要循环执行“冻品过滤”、“活动属性的价格覆盖”这两个能力执行节点，在循环解析中，会以物品的集合为基础进行循环遍历，分别对物品进行判断当前循环到的物品是否需要冻品过滤、当前循环到的物品，活动数据属性信息是什么，进行重新覆盖最新的活动数据，最后解析到要求的返回值groovy脚本，从

物品实体中，分别提取业务线所需要的字段如：物品的基础信息，活动属性、标签属性和库存属性，最后组装成业务所需要的结果进行返回。

[0145] 在本申请实施例中，在上游系统提出新的业务场景需求时，商品属性信息聚合系统无需开发，只需要通过复用已有的节点，生成执行链路，即可获取满足新需求的物品属性信息，这样，可以大大减少针对新需求的处理成本和维护成本，进一步地，通过两次编排可以让执行链路中的节点组件数量进一步减少，进而缩短执行链路的长度，可以提升获取物品属性信息的执行效率。

[0146] 基于同一申请构思，本申请实施例中还提供了与上述实施例提供的物品属性信息的获取方法对应的物品属性信息的获取装置，由于本申请实施例中的装置解决问题的原理与本申请上述实施例的物品属性信息的获取方法相似，因此装置的实施可以参见方法的实施，重复之处不再赘述。

[0147] 如图3所示，图3为本申请实施例提供的一种物品属性信息的获取装置300的功能模块图，物品属性信息的获取装置300包括确定模块310、生成模块320和获取模块330，其中，确定模块310，用于基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路；生成模块320，用于根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个目标节点合成联合节点，以得到目标执行链路；获取模块330，用于在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息。

[0148] 在一种可能的实施方式中，如图3所示，所述第一获取需求中携带有满足需求的待获取物品的条件信息，以及物品属性信息的属性标识；所述确定模块310用于根据以下步骤生成所述初始执行链路：

[0149] 根据所述条件信息和所述属性标识，确定与所述第一获取需求匹配的多个业务逻辑；

[0150] 确定与所述第一获取需求匹配的每个业务逻辑对应的所述目标节点；

[0151] 将各个所述目标节点进行执行顺序的初始编排，生成所述初始执行链路。

[0152] 在一种可能的实施方式中，如图3所示，所述生成模块320，用于根据以下步骤合成所述联合节点：

[0153] 确定所述初始执行链路中用于合成所述联合节点的至少两个节点分别对应的业务逻辑；

[0154] 根据确定出的各业务逻辑的逻辑内容对联合配置模板进行配置，生成所述联合节点。

[0155] 在一种可能的实施方式中，如图3所示，所述生成模块320，用于根据以下步骤生成所述目标执行链路：

[0156] 对所述联合节点中的各业务逻辑进行执行顺序的编排；

[0157] 根据未进行合成的各个所述目标节点和编排完成的联合节点，生成所述目标执行链路。

[0158] 在一种可能的实施方式中，如图3所示，所述获取模块330，用于根据以下步骤在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据：

[0159] 根据所述联合节点执行对应的各个业务逻辑所需的数据量，开启多个用于获取所述目标数据的目标线程；

[0160] 通过开启的多个目标线程，同时获取所述联合节点中各业务逻辑所需的目标数据。

[0161] 在一种可能的实施方式中，如图3所示，所述获取模块330还用于：

[0162] 将所述目标数据存储在所述联合节点的缓存区中；

[0163] 根据以下步骤利用所述目标数据执行所述联合节点：

[0164] 从所述缓存区中调取所述目标数据，以利用所述目标数据执行所述联合节点中的各个业务逻辑。

[0165] 在一种可能的实施方式中，如图3所示，所述获取模块330，还用于根据以下步骤执行所述目标执行链路：

[0166] 按照所述联合节点、未被合成的各个所述目标节点分别对应的执行顺序，执行所述目标执行链路中的所述联合节点、未被合成的各个所述目标节点分别对应的业务逻辑。

[0167] 在一种可能的实施方式中，如图3所示，对于执行方式为串行方式的所述目标节点，所述获取模块330，还用于根据以下步骤执行所述目标执行链路中的目标节点：

[0168] 将多个第一候选物品的物品标识一同输入所述目标节点，以对所述多个第一候选物品进行筛选。

[0169] 在一种可能的实施方式中，如图3所示，对于执行方式为并行方式的所述目标节点，所述获取模块330，根据以下步骤执行所述目标执行链路中的目标节点：

[0170] 从线程池中为所述目标节点分配一个线程，通过该线程来执行所述目标节对应点的业务逻辑。

[0171] 在一种可能的实施方式中，如图3所示，所述获取模块330，根据以下步骤执行所述联合节点：

[0172] 将各个第二候选物品的物品参数分别输入所述联合节点中，按照所述联合节点中的各个业务逻辑对应的执行顺序，执行所述联合节点中的各个业务逻辑。

[0173] 在一种可能的实施方式中，如图3所示，针对每个第二候选物品，所述获取模块330，具体用于根据以下步骤执行所述联合节点中的各个业务逻辑；

[0174] 根据所述第二候选物品的物品参数，判断所述第二候选物品是否满足执行顺序排在首个的业务逻辑；

[0175] 若满足，执行顺序排在下一个的业务逻辑；

[0176] 若不满足，不再执行下一个业务逻辑。

[0177] 在一种可能的实施方式中，如图3所示，所述生成模块320还包括根据以下步骤生成不同节点：

[0178] 确定各个业务所需的业务逻辑，根据不同业务逻辑对节点生成模板进行配置，生成不同节点；其中，所述节点生成模板中包括节点名称、节点地址信息、节点执行路径、节点执行算法。

[0179] 在一种可能的实施方式中，所述目标节点包括功能节点和脚本节点；所述功能节点的业务逻辑写在聚合系统的文件中，所述脚本节点的业务逻辑写在第三方存储器的文件中。

[0180] 在一种可能的实施方式中，所述目标节点包括脚本节点；所述获取装置300还包括执行模块；所述执行模块包括：

[0181] 接收单元，用于接收针对物品属性信息的第二获取需求；所述第二获取需求为对所述第一获取需求进行更新后的需求；

[0182] 确定单元，用于根据所述第二获取需求相对于所述第一获取需求的更新内容，确定需要调整业务逻辑的逻辑内容的脚本节点，并对所述脚本节点进行调整；

[0183] 更新单元，用于加载调整后的所述脚本节点，并对所述目标执行链路的链路配置进行更新；

[0184] 执行单元，用于执行更新后的所述目标执行链路，获取与所述第二获取需求匹配的属性信息。

[0185] 在一种可能的实施方式中，所述确定单元，具体用于根据以下步骤对所述脚本节点进行调整：

[0186] 根据所述更新内容确定业务需求；

[0187] 将所述脚本节点的业务逻辑修改为与所述业务需求匹配的业务逻辑。

[0188] 基于同一申请构思，参见图4所示，为本申请实施例提供的一种电子设备400的结构示意图，包括：处理器410、存储器420和总线430，所述存储器420存储有所述处理器410可执行的机器可读指令，当电子设备400运行时，所述处理器410与所述存储器420之间通过所述总线430进行通信，所述机器可读指令被所述处理器410运行时执行如上述实施例中任一所述的物品属性信息的获取方法的步骤。

[0189] 具体地，所述机器可读指令被所述处理器410执行时可以执行如下处理：

[0190] 基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路；

[0191] 根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个目标节点合成联合节点，以得到目标执行链路；

[0192] 在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息。

[0193] 基于同一申请构思，本申请实施例还提供了一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器运行时执行上述实施例提供的物品属性信息的获取方法的步骤。

[0194] 具体地，所述存储介质能够为通用的存储介质，如移动磁盘、硬盘等，所述存储介质上的计算机程序被运行时，能够执行上述物品属性信息的获取方法，通过两次链路编排来减少整体执行节点的数量，可以缩短链路执行的时间，而且通过提前获取执行链路所需的全部数据，可以大大提升执行的效率，进而提高获取物品属性信息的速度。

[0195] 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，上述描述的系统和装置的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。在本申请所提供的几个实施例中，应所述理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，又例如，多个单元或组件可以结合或者可以

集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些通信接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0196] 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单元来实现本实施例方案的目的。

[0197] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。

[0198] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个处理器可执行的非易失的计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者所述技术方案的部分可以以软件产品的形式体现出来,所述计算机软件产品存储在一个存储介质中,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)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0199] 以上仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以权利要求的保护范围为准。

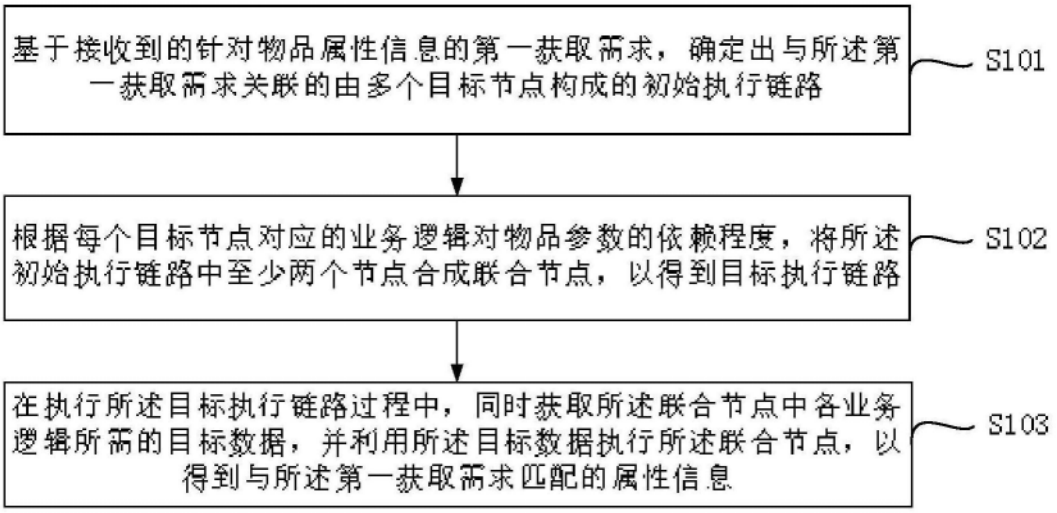

基于接收到的针对物品属性信息的第一获取需求，确定出与所述第一获取需求关联的由多个目标节点构成的初始执行链路
S101
根据每个目标节点对应的业务逻辑对物品参数的依赖程度，将所述初始执行链路中至少两个节点合成联合节点，以得到目标执行链路
S102
在执行所述目标执行链路过程中，同时获取所述联合节点中各业务逻辑所需的目标数据，并利用所述目标数据执行所述联合节点，以得到与所述第一获取需求匹配的属性信息
S103

图1

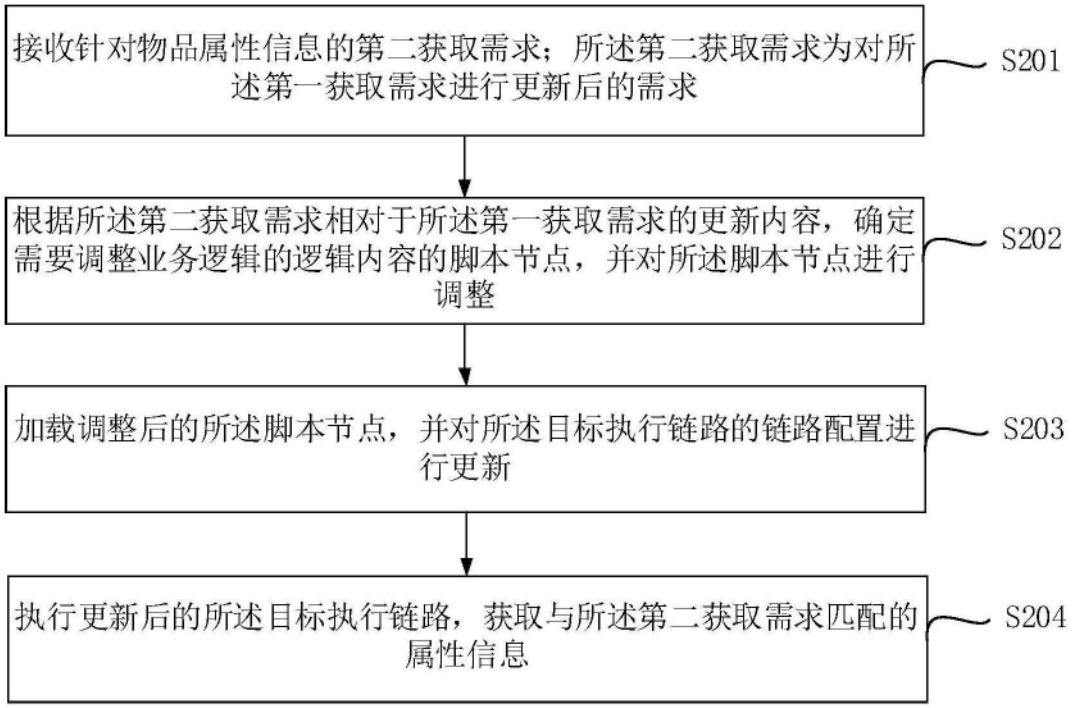

接收针对物品属性信息的第二获取需求；所述第二获取需求为对所述第一获取需求进行更新后的需求
S201
根据所述第二获取需求相对于所述第一获取需求的更新内容，确定需要调整业务逻辑的逻辑内容的脚本节点，并对所述脚本节点进行调整
S202
加载调整后的所述脚本节点，并对所述目标执行链路的链路配置进行更新
S203
执行更新后的所述目标执行链路，获取与所述第二获取需求匹配的属性信息
S204

图2

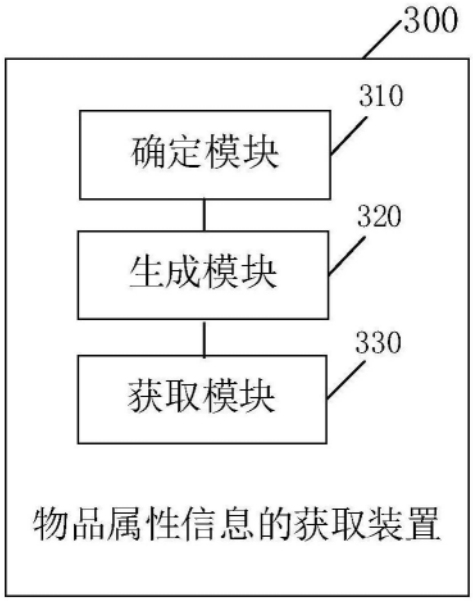

300
310
确定模块
320
生成模块
330
获取模块
物品属性信息的获取装置

图3

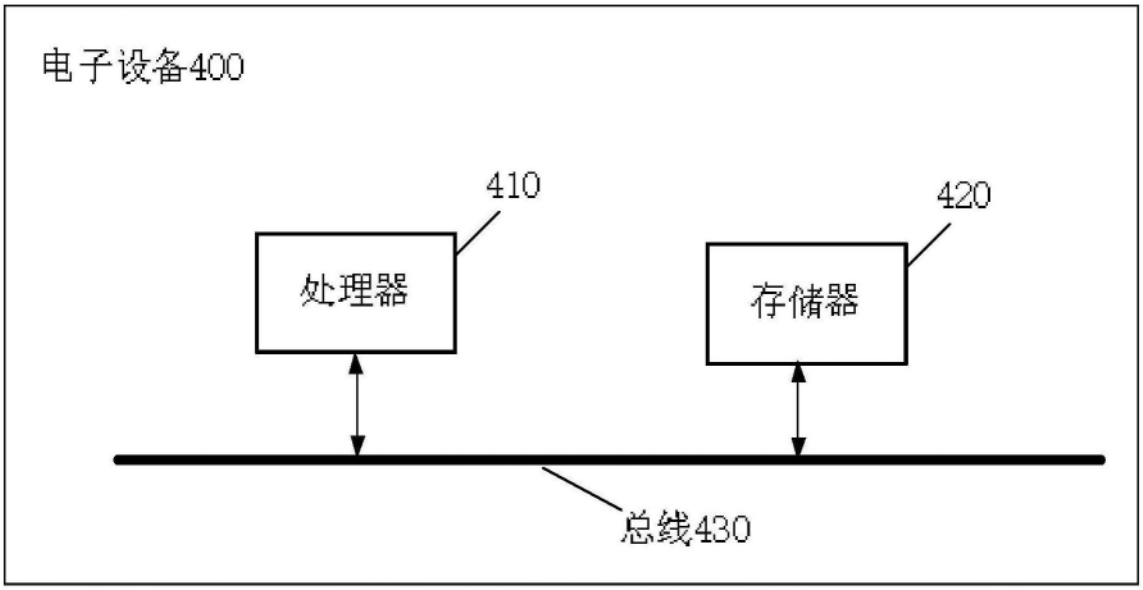

电子设备400
410
处理器
420
存储器
总线430

图4