(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108009849 B
(45)授权公告日 2021.12.17

(21)申请号 201711236896.3

(22)申请日 2017.11.30

(65)同一申请的已公布的文献号
申请公布号 CN 108009849 A

(43)申请公布日 2018.05.08

(73)专利权人 北京小度互娱科技有限公司
地址 100193 北京市海淀区西北旺东路10号院东区17号楼303-305室

(72)发明人 段轶轩

(74)专利代理机构 北京英赛嘉华知识产权代理有限责任公司 11204
代理人 王达佐 王艳春

(51)Int.Cl.
G06Q 30/02(2012.01)
G06F 9/54(2006.01)

(56)对比文件
CN 106815254 A,2017.06.09
CN 106022829 A,2016.10.12
CN 106485535 A,2017.03.08
US 9606877 B2,2017.03.28
CN 106204136 A,2016.12.07
明星it.storm滑动时间窗口实现.《CSDN》.2016,

审查员 宋敏

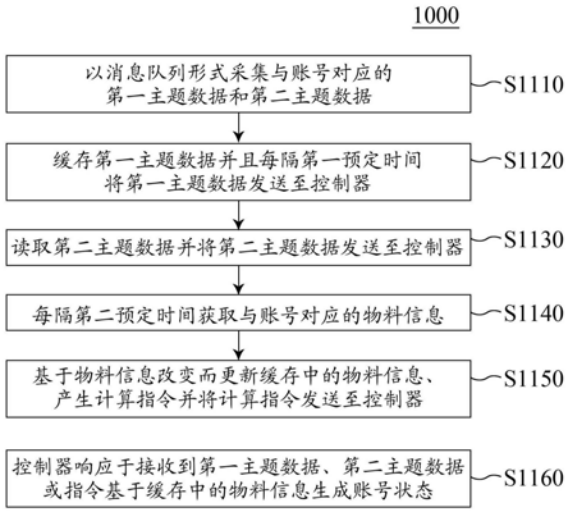
权利要求书3页 说明书10页 附图3页

(54)发明名称

生成账号状态的方法以及生成账号状态的装置

(57)摘要

本申请涉及生成账号状态的方法以及生成账号状态的装置。生成账号状态的方法包括:以消息队列形式采集并解析与账号对应的第一主题数据和第二主题数据;缓存第一主题数据并且每隔第一预定时间将第一主题数据发送至控制器;读取第二主题数据并将第二主题数据发送至控制器;每隔第二预定时间获取与账号对应的物料信息;以及基于物料信息改变而更新缓存中的物料信息并产生计算指令,并将计算指令发送至控制器,其中,控制器响应于接收到第一主题数据、第二主题数据或计算指令基于缓存中的物料信息生成账号状态。



CN 108009849 B

1.一种生成账号状态的方法,包括:

以消息队列形式采集与所述账号对应的第一主题数据和第二主题数据;

缓存所述第一主题数据,对所述第一主题数据预处理以生成预处理数据,并且每隔第一时间将所述预处理数据发送至控制器;

读取所述第二主题数据并将所述第二主题数据发送至所述控制器;

每隔第二预定时间获取与所述账号对应的物料信息;以及

基于所述物料信息改变而更新缓存中的物料信息并产生计算指令,并将所述计算指令发送至所述控制器,

其中,所述控制器响应于接收到所述预处理数据、所述第二主题数据或所述计算指令基于缓存中的物料信息生成账号状态。

2.如权利要求1所述的生成账号状态的方法,包括通过异步交互的方式每隔第三预定时间存储所述账号状态。

3.如权利要求2所述的生成账号状态的方法,由Redis存储系统通过异步交互的方式每隔所述第三预定时间存储所述账号状态。

4.如权利要求2或3所述的生成账号状态的方法,还包括基于所存储的所述账号状态控制所述账号的活动。

5.如权利要求4所述的生成账号状态的方法,还包括将所述账号状态传输至第三方控制端。

6.一种生成账号状态的装置,包括:存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时执行以下步骤:

以消息队列形式采集与所述账号对应的第一主题数据和第二主题数据;

缓存所述第一主题数据,对所述第一主题数据预处理以生成预处理数据,并且每隔第一时间将所述预处理数据发送至控制器;

读取所述第二主题数据并将所述第二主题数据发送至所述控制器;

每隔第二预定时间获取与所述账号对应的物料信息;以及

基于所述物料信息改变而更新缓存中的物料信息并产生计算指令,并将所述计算指令发送至所述控制器,

其中,所述控制器响应于接收到所述预处理数据、所述第二主题数据或所述计算指令生成账号状态。

7.一种生成账号状态的装置,包括:

流式收集模块,以消息队列形式采集与所述账号对应的第一主题数据和第二主题数据;

Storm模块,包括:

ControlBolt组件;

ImpSout组件和CacheBolt组件,所述ImpSout组件从所述流式收集模块读取所述第一主题数据,并将所述第一主题数据发送至所述CacheBolt组件,所述CacheBolt组件对所述第一主题数据进行预处理以生成预处理数据;

FlushSpout组件,所述FlushSpout组件每隔第一预定时间产生发送指令,所述

CacheBolt组件响应于所述发送指令将所述预处理数据发送至所述ControlBolt组件；

ClickSpout组件，所述ClickSpout组件从所述流式收集模块读取所述第二主题数据，并将所述第二主题数据发送至所述ControlBolt组件；

InfoSpout组件，每隔第二预定时间获取与所述账号对应的物料信息；

DbBolt组件，所述DbBolt组件同步由所述InfoSpout组件获取的物料信息，基于所述物料信息改变而更新缓存中的物料信息并产生计算指令，并将所述计算指令发送至所述ControlBolt组件，

其中，所述ControlBolt组件响应于接收到所述预处理数据、所述第二主题数据或所述计算指令基于缓存中的物料信息生成账号状态。

8.如权利要求7所述的生成账号状态的装置，还包括存储模块，所述ControlBolt组件配置为每隔第三预定时间将所述账号状态以异步交互的方式存储在所述存储模块。

9.如权利要求8所述的生成账号状态的装置，所述存储模块为Redis存储系统。

10.如权利要求8所述的生成账号状态的装置，其中，所述流式收集模块包括Kafka消息队列系统、ActiveMQ消息队列系统或RabbitMQ消息队列系统。

11.如权利要求8所述的生成账号状态的装置，其中，所述流式收集模块包括Kafka消息队列系统，所述ImpSout组件和所述ClickSpout组件与所述Kafka消息队列系统的主题的分区数保持一致。

12.如权利要求8-11中任一项所述的生成账号状态的装置，还包括账号管理模块，所述账号管理模块基于所述存储模块中存储的所述账号状态控制所述账号的活动。

13.如权利要求12所述的生成账号状态的装置，所述账号管理模块还配置为将所述账号状态传输至第三方控制端。

14.一种生成账号状态的方法，包括:

通过流式收集模块以消息队列形式采集与所述账号对应的第一主题数据和第二主题数据；

通过Storm模块中的组件执行的以下操作：

通过ImpSout组件从所述流式收集模块读取所述第一主题数据，并将所述第一主题数据发送至CacheBolt组件，所述CacheBolt组件对所述第一主题数据进行预处理以生成预处理数据；

通过ClickSpout组件从所述流式收集模块读取所述第二主题数据，并将所述第二主题数据发送至ControlBolt组件；

通过FlushSpout组件每隔第一预定时间产生发送指令，所述CacheBolt组件响应于所述发送指令将所述预处理数据发送至所述ControlBolt组件；

通过InfoSpout组件每隔第二预定时间获取与所述账号对应的物料信息；

通过DbBolt组件同步由所述InfoSpout组件获取的物料信息，基于所述物料信息改变而更新缓存中的物料信息并产生计算指令，并将所述计算指令发送至所述ControlBolt组件;以及

响应于接收到所述预处理数据、所述第二主题数据或所述计算指令通过所述ControlBolt组件基于缓存中的物料信息生成账号状态。

15.如权利要求14所述的生成账号状态的方法，还包括:通过异步交互的方式每隔第三

预定时间存储所述账号状态。

16.如权利要求15所述的生成账号状态的方法，其中，由Redis存储系统通过异步交互的方式每隔所述第三预定时间存储所述账号状态。

17.如权利要求15或16所述的生成账号状态的方法，还包括基于所存储的账号状态控制所述账号的活动。

18.如权利要求17所述的生成账号状态的方法，还包括将所存储的账号状态传输至第三方控制端。

生成账号状态的方法以及生成账号状态的装置

技术领域

[0001] 本申请涉及数据处理领域，更具体地，涉及处理计算广告需求方平台(Demand-Side Platform,DSP)上广告主的账号实时状态所需的数据。

背景技术

[0002] 广告需求方平台，简称DSP，其是实时竞价(Real-Time Bidding,RTB)的核心。传统的互联网广告生态链一般最多只有三方，分别是广告主、广告代理商(即广告公司)以及互联网媒体。而在RTB广告交易模式中，原有的广告生态链发生了变化，整个生态链包括广告主、DSP、广告交易平台以及互联网媒体四个主体。广告主将自己的广告需求放到DSP平台上，互联网媒体将自己的广告流量资源放到广告交易平台，DSP通过与广告交易平台的技术对接完成竞价购买。

[0003] 由于广告的实时性，对应于广告主的账号的状态(下文称账号状态)也实时地发生变化。账号状态可包括账号实时余额、账号每日预算余额、账号对应的广告的实时花费等。这样，广告主、DSP、广告交易平台以及互联网媒体中的任何一方可能会需要了解账号状态。例如广告主需要了解账号的实时余额以控制广告活动，DSP可能需要了解账号的实时余额来确定广告主是否能够参与实时竞价以及账号的中短期控制，DSP可能还需要了解账号每日预算余额以控制账号当日的广告活动。

[0004] 由于在广告展示过程中，产生的主要数据分为曝光数据和点击数据，曝光数据的产生速度及数据量远大于点击数据，因此传统的数据处理方式需要处理器在短时间内进行大量的数据处理。现有大多采用单机的方式来生成账号状态，从广告打点接口接收广告的曝光和点击数据，实时计算账号和活动的实时花费，并触发控量逻辑。很显然，单机的解决方案会影响整个广告曝光和点击的每秒吞吐量，并成为业务的瓶颈。而通过HTTP请求与响应的方式进行数据处理对网络的稳定性要求会更加依赖，同时，也会消耗更多的服务器资源，例如：大量的网络连接等。因此需要一种能够在数据量持续快速增加的情况下处理数据以生成账号状态的方法和装置。

发明内容

[0005] 针对RTB中存在的至少一个问题，本申请提出了一种至少能部分地扩展数据处理能力的、生成账号状态的方法和系统。

[0006] 本申请的一方面涉及一种生成账号状态的方法，可包括：以消息队列形式采集与账号对应的第一主题数据和第二主题数据；缓存第一主题数据并且每隔第一预定时间将第一主题数据发送至控制器；读取第二主题数据并将第二主题数据发送至控制器；每隔第二预定时间获取与账号对应的物料信息；以及基于物料信息改变而更新缓存中的物料信息并产生计算指令，并将计算指令发送至控制器，其中，控制器响应于接收到第一主题数据、第二主题数据或指令生成账号状态。

[0007] 在一个实施方式中，生成账号状态的方法例如可包括通过异步交互的方式每隔第

三预定时间存储账号状态。

[0008] 在一个实施方式中，生成账号状态的方法还可包括由Redis存储系统通过异步交互的方式每隔第三预定时间存储账号状态。

[0009] 在一个实施方式中，生成账号状态的方法还可包括基于所存储的账号状态控制账号的活动。

[0010] 在一个实施方式中，生成账号状态的方法还可包括将账号状态传输至第三方控制端。第三方控制端可为广告投放平台。

[0011] 在一个实施方式中，缓存第一主题数据并且每隔第一预定时间将第一主题数据发送至控制器还可包括：对第一主题数据进行预处理以生成预处理数据；以及每隔第一预定时间将预处理数据发送至控制器。

[0012] 在一个实施方式中，控制器还可响应于接收到预处理数据基于缓存中的物料信息生成账号状态。

[0013] 根据本申请的另一方面，提供了一种生成账号状态的装置，本装置包括：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程序，其特征在于，处理器执行计算机程序时执行以下步骤：以消息队列形式采集与账号对应的第一主题数据和第二主题数据；缓存第一主题数据；每隔第一预定时间将第一主题数据发送至控制器；读取第二主题数据并将第二主题数据发送至控制器；每隔第二预定时间获取与账号对应的物料信息；以及基于物料信息改变而更新缓存中的物料信息并产生计算指令，并将计算指令发送至控制器，其中，控制器响应于接收到第一主题数据、第二主题数据或指令生成账号状态。

[0014] 根据本申请的又一方面，提供了生成账号状态的装置，该装置包括：流式收集模块，以消息队列形式采集与账号对应的第一主题数据和第二主题数据；Storm模块，Storm模块包括：ControlBolt组件；ImpSout组件和CacheBolt组件，ImpSout组件从流式收集模块读取第一主题数据并将第一主题数据发送至CacheBolt组件；ClickSpout组件，ClickSpout组件从流式收集模块读取第二主题数据，并将第二主题数据发送至ControlBolt组件；InfoSpout组件，每隔第二预定时间获取与账号对应的物料信息；DbBolt组件，DbBolt组件同步由InfoSpout组件获取的物料信息，基于物料信息改变而更新缓存中的物料信息并产生计算指令，并将计算指令发送至ControlBolt组件；以及FlushSpout组件，FlushSpout每隔第一预定时间产生发送指令，CacheBolt组件响应于发送指令将第一主题数据发送至ControlBolt组件，其中，ControlBolt组件响应于接收到第一主题数据、第二主题数据或计算指令生成账号状态。

[0015] 在一个实施方式中，装置还可包括存储模块，存储模块通过异步交互的方式存储账号状态。存储模块可以为Redis存储系统。

[0016] 在一个实施方式中，流式收集模块可包括Kafka消息队列系统、ActiveMQ消息队列系统或RabbitMQ消息队列系统。

[0017] 在一个实施方式中，流式收集模块可包括Kafka消息队列系统，ImpSout组件和ClickSpout组件与Kafka消息队列系统的主题的分区数保持一致。

[0018] 在一个实施方式中，根据本申请的装置还包括账号管理模块，账号管理模块基于存储模块中存储的账号状态控制账号的活动。账号管理模块还可配置为将账号状态传输至第三方控制端。

[0019] 在一个实施方式中，CacheBolt组件还可配置为对第一主题数据进行预处理以生成预处理数据并且响应于发送指令将预处理数据发送至ControlBolt组件。

[0020] 在一个实施方式中，ControlBolt组件还响应于接收到预处理数据基于缓存中的物料信息生成账号状态。

[0021] 根据本申请的再一方面，提供生成账号状态的装置，包括：以消息队列形式采集与账号对应的第一主题数据和第二主题数据；以及通过Storm模块中的组件执行的以下操作：通过ImpSout组件从流式收集模块读取第一主题数据，并将第一主题数据发送至CacheBolt组件；通过ClickSpout组件从流式收集模块读取第二主题数据，并将第二主题数据发送至ControlBolt组件；通过FlushSpout组件每隔第一预定时间产生发送指令并将发送指令发送至CacheBolt组件，由CacheBolt组件响应于发送指令将第一主题数据发送至ControlBolt组件；通过InfoSpout组件每隔第二预定时间获取与账号对应的物料信息；通过DbBolt组件同步由InfoSpout组件获取的物料信息，基于物料信息改变而更新缓存中的物料信息并产生计算指令，并将计算指令发送至ControlBolt组件；以及通过ControlBolt组件响应于接收到第一主题数据、第二主题数据或计算指令基于缓存中的物料信息生成账号状态。

[0022] 在一个实施方式中，根据本申请的方法还可包括：通过异步交互的方式每隔第三预定时间存储账号状态。可由Redis存储系统通过异步交互的方式每隔第三预定时间存储账号状态。

[0023] 在一个实施方式中，根据本申请的方法还可包括基于所存储的账号状态控制账号的活动。

[0024] 在一个实施方式中，根据本申请的方法还可包括将所存储的账号状态传输至第三方控制端。

[0025] 在一个实施方式中，生成账号状态的方法还可包括：通过CacheBolt组件对第一主题数据进行预处理以生成预处理数据；以及通过CacheBolt组件响应于发送指令将预处理数据发送至ControlBolt组件。

[0026] 在一个实施方式中，生成账号状态的方法还可包括：响应于接收到预处理数据通过ControlBolt组件基于缓存中的物料信息生成账号状态。

附图说明

[0027] 通过参考附图详细描述本发明的示例性实施方式，本发明的上述及其他方面和特征将变得更加明显，在附图中：

[0028] 图1是根据本申请实施方式生成账号状态的方法的流程图；

[0029] 图2是示意性地示出根据本申请实施方式的生成账号状态的装置的框图；

[0030] 图3是示意性地示出根据本申请另一实施方式的生成账号状态的装置的框图；

[0031] 图4示出根据本申请实施方式的Storm模块中的组件的分布式拓扑结构；以及

[0032] 图5示出根据本申请的一个实施方式的生成账号状态的装置在生成账号状态时与外部机构形成的分布式拓扑结构。

具体实施方式

[0033] 现在，将在下文中参照示出各实施方式的附图更充分地描述本发明。然而，本发明能以诸多不同的形式来实现，而不应解释为局限于本文阐述的实施方式。相反，提供这些实施方式，使得本公开将是透彻和完整的，并且将向本领域技术人员充分传达本发明的范围。相同的附图标记始终表示相同的元件。

[0034] 将理解，当元件被称为处于另一元件“上”时，它可直接地处于该另一元件上，或者其间可存在中间元件。相反，当元件被称为直接在另一元件上时，不存在中间元件。

[0035] 将理解，虽然可在本文中使用术语“第一”、“第二”、“第三”等来描述各种元件、组件、区域、层和/或部分，但是这些元件、组件、区域、层和/或部分不应受这些术语限制。这些术语仅用于将一个元件、组件、区域、层或部分与另一元件、组件、区域、层或部分区分开。因此，在没有脱离本文的教导的情况下，下面讨论的第一元件、组件、区域、层或部分可被称为第二元件、组件、区域、层或部分。

[0036] 本文使用的术语仅是出于描述具体实施方式的目的，而并非旨在进行限制。如本文所使用的那样，除非内容清楚地另行指出，否则单数形式“一”、“一个”和“所述”旨在包括复数形式，包括“至少一个”。“或”是指“和/或”。如本文所使用的那样，术语“和/或”包括相关所列项目中的一个或多个的任何和全部组合。还将理解，当措辞“包括”在本说明书中使用时指出所阐述的特征、区域、整体、步骤、操作、元件和/或组件的存在，但是不排除一个或多个其他特征、区域、整体、步骤、操作、元件、组件和/或其群组的存在或添加。

[0037] 此外，在本文中可使用诸如“下”或“底部”以及“上”或“顶部”的相对术语来描述如附图中所示的一个元件与另一元件的关系。将理解，除了附图中所描绘的定向之外，相对术语还旨在涵盖设备的不同定向。例如，如果附图之一中的设备翻转，则描述为在其他元件的“下”侧上的元件于是将定向成在所述其他元件的“上”侧。示例性术语“下”因此可根据附图的具体定向涵盖下和上两个定向。类似地，如果附图之一中的设备翻转，则描述为在其他元件“下方”或“下面”的元件于是将定向在所述其他元件的“上方”。示例性术语“下方”或“下面”因此可涵盖上方和下方两个定向。

[0038] 如本文所使用的，“约”或“近似”包括所阐述的值以及在对于特定值的如由本领域普通技术人员在考虑正在进行的测量和与特定量的测量相关的误差(即，测量系统的局限性)所确定的可接受偏差范围内的平均值。

[0039] 除非另行限定，否则本文所使用的全部术语(包括技术术语和科学术语)具有与由本公开所属领域的普通技术人员通常所理解的含义相同的含义。还将理解，术语，诸如通常使用的词典中所定义的术语，应解释为具有与它们在相关技术的上下文和本公开中的含义相一致的含义，并且将不在理想化或过于正式的意义上进行解释，除非本文明确地限定成这样。

[0040] 图1是根据本申请实施方式生成账号状态的方法的流程图。

[0041] 参照图1，生成账号状态的方法1000包括以消息队列形式采集与账号对应的第一主题数据和第二主题数据(S1110)。由于将第一主题数据和第二主题数据存储在消息队列中，因此可根据数据的产生方式(例如曝光数据因广告被广告受众看到而产生，点击数据因广告被广告受众点击而产生)、每种数据的产生速度(例如单位时间并发的量)等对数据进行划分，但是本申请不限于此。第一主题数据可例如对应于因广告曝光(即广告被广告受众

看到)而产生的曝光日志,例如广告被曝光一次,随即更新曝光日志。第二主题数据可对应于广告点击(即广告被广告受众点击)而产生的点击日志,例如广告被点击一次,随即更新点击日志。可通过诸如Kafka、ActiveMQ或RabbitMQ的消息队列系统采集第一主题数据和第二主题数据,从而以消息列队形式存储产生的、与第一主题数据或第二主题数据对应的每一条新数据。这样,数据可暂时存放在消息队列中,数据处理方(即,订阅数据方,例如下文将描述的Storm模块)可根据自身的负载处理能力控制处理消息的处理速度,减小数据大量并发时访问的压力。

[0042] 根据本申请实施方式,生成账号状态的方法1000还包括:缓存第一主题数据并且每隔第一预定时间将第一主题数据发送至控制器(S1120)。生成账号状态的方法1000中的步骤S1120还可包括:对第一主题数据进行预处理以生成预处理数据并且每隔第一预定时间间隔将预处理数据发送至控制器。例如,步骤S1120可包括基于缓存的第一主题数据和缓存中的物料信息来生成预处理数据,但本申请的实施方式不限于此。通过步骤S1120,例如可基于第一预定时间内的曝光量和缓存中的物料信息针对广告活动和账号维度进行汇总与计算,以生成缓存金额,例如第一预定时间内曝光量所花费的金额。通过对快速并发数据的缓存与预处理可极大减少控制器的处理量,进而提高数据处理能力。

[0043] 接下来,读取第二主题数据并将第二主题数据发送至控制器(S1130)。如上所述,第二主题数据被存放在消息队列中,因此步骤S1130的第二数据读取过程中,会节省服务器成本并且降低数据丢失的风险。

[0044] 接下来,每隔第二预定时间获取与账号对应的物料信息(S1140)。与账号对应的物料信息可为广告库中投放的物料信息。物料信息可为广告单价,广告中账号、活动、组和创意的映射关系以及广告离线报表分小时花费等,物料信息还可包括广告主在广告投放过程中存入与提取金额的信息。

[0045] 接下来,基于物料信息改变而更新缓存中的物料信息并产生计算指令,并将计算指令发送至控制器(S1150)。缓存中的物料信息可用于生成账号状态,并且还可用于对第一主题数据的预处理。可基于缓存中的物料信息针对第一预定时间内接收的第一主题数据进行汇总与计算。例如,在步骤S1120中,可基于第一预定时间内的曝光量和缓存中的物料信息针对广告活动和账号维度进行汇总与计算,以生成缓存金额,例如第一预定时间内曝光量所花费的金额。通过该操作,可实时地更新缓存中的物料信息,从而可在物料信息改变的情况下对重新生成账号状态,同时更新其他工作节点进行计算所需的物料信息。因此,在物料信息改变时,产生发送至控制器的计算指令以基于更新的物料信息重新生成账号状态。这样进一步保证了数据处理的实时性。

[0046] 控制器响应于接收到第一主题数据、第二主题数据或计算指令基于缓存中的物料信息生成账号状态(S1160)。另外,控制器还可响应于接收到预处理数据基于缓存中的物料信息生成账号状态。

[0047] 当第一主题数据是曝光日志且第二主题数据是点击日志时,由于单位时间内广告曝光次数远大于点击次数,因此单位时间内产生的曝光数据量远大于单位时间内产生的点击数据量。如果按曝光一次,即生成新的账号状态,则数据处理能力会因曝光次数增加而遇到瓶颈。数据处理方则可在横向上对数据进行划分(例如,将数据划分为第一主题数据和第二主题数据)并且在纵向上划分的数据进行区别处理,最终传送至数据结算方以生成账号

状态。

[0048] 因此，方法1000通过以消息队列形式采集数据实现对数据的划分，并且通过不同的工作节点分别读取第一主题数据和第二主题数据，从而第一主题数据和第二主题数据因为被不同的工作节点读取而可分别被不同地处理。例如，通过第一工作节点读取第一主题数据，通过第二工作节点读取第二主题数据，第一工作节点将第一数据发送给其下游的工作节点，第二工作节点将第二数据发送给其下游的工作节点，从而实现数据的分布式处理。

[0049] 由于每隔第一预定时间将缓存的第一主题数据发送至控制器以使得控制器生成账号状态，所以并非第一主题数据更新即重新生成账号状态，而是每隔第一预定时间基于产生的第一主题数据的量来生成账号状态。第一预定时间决定了曝光数据处理的延时性，也是程序能无限扩展处理能力的关键，例如第一预定时间可设置为500ms。针对账号状态相对于第一主题数据(曝光日志)改变，控制器每500ms进行一次计算，且第一主题数据已经被预处理，这样既降低了控制器的计算频率又减少了控制器的计算量。通过这种方式，能够在短时间内并发大量的第一主题数据情况下生成账号状态，同时消耗更少的网络及服务器资源。

[0050] 由于以并行的方式对第一主题数据、第二主题数据和物料信息进行处理，同时通过缓存和/或预处理第一数据来减少控制器负荷，从而可实现数据处理能力的扩展。本发明不限于对三种数据进行处理，并且还可对四种及以上的数据进行并行处理，并且可选择其中的至少一种数据进行缓存和预处理。例如，可针对单位时间并发量多的一种或几种数据进行缓存和预处理。

[0051] 根据本申请的实施方式，方法1000还可包括通过异步交互的方式每隔第三预定时间存储账号状态。由于采用异步交互的方式，因此在不影响控制器处理逻辑的前提下，对账号状态进行定时存储。例如，可由Redis存储系统通过异步交互的方式每隔第三预定时间存储账号状态。由于每隔第三预定时间来存储账号状态，所以可查看由第三预定时间划分的时间节点处的账号状态，便于与账号状态对应数据的回踩和再利用。所存储的账号状态用于程序重启的数据回踩、实时花费数据、账户余额的可视化。

[0052] 根据本申请的实施方式，方法1000还可包括根据存储的账号状态控制账号的活动。例如可根据与账号状态对应的账号余额控制账号状态活动，当账号余额不足以支付广告活动时，可针对该账号终止广告活动；当账号状态在广告主对广告账号投入金额后改变时，可针对账号开启广告活动。

[0053] 根据本申请的实施方式，方法1000还可包括将账号状态传输至第三方控制端。第三方控制端例如可为广告投放平台。

[0054] 图2是示意性地示出根据本申请实施方式的生成账号状态的装置的框图。

[0055] 参照图2，根据本申请示例性实施方式的生成账号状态的装置100包括存储器110和处理器120，存储器110上存储有计算机程序，所述计算机程序可在处理器120上运行，所述处理器120执行存储器110上存储的计算机程序时执行以下步骤：以消息队列形式采集所述账号对应的第一主题数据和第二主题数据；缓存第一主题数据并且每隔第一预定时间将第一主题数据发送至控制器；读取第二主题数据并将所述第二主题数据发送至控制器；每隔第二预定时间获取与账号对应的物料信息；基于物料信息改变而更新缓存中的物料信息并产生计算指令，并将计算指令发送至控制器；以及控制器响应于接收到第一主题数据、第

二主题数据或指令生成账号状态。

[0056] 装置100能够以并行的方式对第一主题数据、第二主题数据以及物料信息进行处理，并且可缓存的第一主题数据进行预处理，从而实现数据处理能力的扩展。例如可基于第一预定时间内的曝光量和缓存中的物料信息针对广告活动和账号维度进行汇总与计算，以生成缓存金额，例如第一预定时间内曝光量所花费的金额。

[0057] 处理器120执行存储器110上存储的计算机程序时还可执行以下步骤：对第一主题数据进行预处理以生成预处理数据；以及每隔第一预定时间将预处理数据发送至控制器。控制器还可响应于接收到预处理数据基于缓存中的物料信息生成账号状态。

[0058] 图3是示意性地示出根据本申请另一实施方式的生成账号状态的装置的框图。

[0059] 生成账号状态的装置200包括流式收集模块210和Storm模块220。流式收集模块210以消息队列形式采集账号对应的第一主题数据和第二主题数据。

[0060] Storm模块220可基于用于流式大数据处理的流式处理计算引擎，Storm流式处理计算引擎通过Spout节点和Bolt节点构成的拓扑结构进行数据处理，其中，Spout节点发送消息，负责以tuple元组的形式发送数据流；而Bolt节点负责转换数据流，在Bolt节点中可完成计算、过滤等操作，Bolt节点也可将数据发送给其他Bolt节点。由Spout节点发送的tuple元组是不可变数组，对应着固定的键值对。

[0061] Storm模块220包括InfoSpout组件221、ImpSout组件222、CacheBolt组件223、ClickSpout组件224、ControlBolt组件225、DbBolt组件226和FlushSpout组件227。

[0062] ImpSout组件222从流式收集模块210读取第一主题数据并且将第一主题数据发送至CacheBolt组件223。ControlBolt组件225还可对第一主题数据进行预处理以生成预处理数据。

[0063] ClickSpout组件224从流式收集模块210读取第二主题数据，并将第二主题数据发送至ControlBolt组件225。例如，ImpSout组件222和ClickSpout组件224分别从消息队列实时收集曝光日志和点击日志，解析后发送给各自的下游。由于广告点击量较少，根据本发明实施方式的装置200中未设置对第二主题数据进行缓存处理的下游组件，但是本申请不限于此。例如，可根据需要对某一组件设置多个下游组件。由于第一主题数据和第二主题数据以消息队列形式存储，因此在数据写入与读取过程中节省服务器成本并且降低数据丢失的风险。

[0064] FlushSpout组件227可全局唯一，启动一个定时器，定时(即每隔第一预定时间)向下游的CacheBolt组件223发送将聚合结果发送到下游ControlBolt组件225的指令。FlushSpout组件227可每隔第一预定时间产生发送指令，CacheBolt组件223可响应于发送指令将第一主题数据发送至ControlBolt组件225。例如，CacheBolt组件223可响应于发送指令将预处理数据发送至ControlBolt组件225。第一预定时间决定了曝光数据处理的延时性，也是程序能无限扩展处理能力的关键，可设置为500ms。

[0065] InfoSpout组件221每隔第二预定时间获取与账号对应的物料信息。Infospout组件221可全局唯一，用于从数据库中定时扫描广告库中投放的物料信息，每第二预定时间(例如1分钟)获取物料的变更状态，同步给下游的DbBolt组件226。

[0066] DbBolt组件226同步由InfoSpout组件获取的物料信息，基于物料信息改变而更新缓存中的物料信息并产生计算指令，并将计算指令发送至ControlBolt组件225。可在每一

个工作节点设置一个DbBolt组件226，从而保证物料信息在每个工作节点都有一份完整镜像。DbBolt组件226用于缓存：广告单价、广告账号、活动、组和创意的映射关系；以及广告离线报表分小时花费等，当任何影响账号余额和活动预算的物料信息改变，DbBolt组件226即通知下游的ControlBolt组件225，触发控量计算逻辑，并且可更新缓存中的物料信息。

[0067] 另外，由于DbBolt组件226可实时地更新缓存中的物料信息，从而可在物料信息改变的情况下对重新生成账号状态，同时更新其他工作节点进行计算所需的物料信息。这样进一步保证了数据处理的实时性。

[0068] ControlBolt组件225响应于接收到第一主题数据、第二主题数据或计算指令基于缓存中的物料信息生成账号状态。例如，ControlBolt组件225中可存储有用于生成账号状态的控量计算逻辑，该控量计算逻辑可响应于ControlBolt组件225接收到第一主题数据、第二主题数据和计算指令中的至少之一而触发，从而生成账号状态。

[0069] ControlBolt组件225可响应于接收到预处理数据基于缓存中的物料信息生成账号状态。这样可进一步减少ControlBolt组件225所需处理数据的量。

[0070] 作为示例，ControlBolt组件225例如可基于接收到的曝光日志基于缓存中的当前物料信息按照广告活动和账号维度进行汇总和计算以生成预处理数据。ControlBolt组件225可全局唯一并且缓存当前的物料信息，例如缓存全局唯一的账号和广告活动分小时花费等。ControlBolt组件225可响应于上游传送的消息(例如响应于接收到第一主题数据、第二主题数据或计算指令)基于缓存中的物料信息执行控量计算逻辑。

[0071] 装置200还包括存储模块230(参见图5)，ControlBolt组件225配置为每隔第三预定时间将账号状态以异步交互的方式存储在存储模块230中。存储模块230可为Redis存储系统。通过装置200与存储模块230以异步传输的方式进行交互，可在不影响ControlBolt组件225的处理逻辑的前提下，解决了程序重启的数据回踩以及核心实时花费数据的可视化等问题。由于每隔第三预定时间来存储账号状态，所以可查看由第三预定时间划分的时间节点处的账号状态，便于与账号状态对应的数据的回踩和再利用。

[0072] 根据本申请的一个实施方式，流式收集模块220可包括Kafka消息队列系统、ActiveMQ消息队列系统或RabbitMQ消息队列系统。

[0073] 根据本申请的一个实施方式，流式收集模块210可包括Kafka消息队列系统，在这种情况下，ImpSout组件222和ClickSpout组件224与所述Kafka消息队列系统的主题的分区数保持一致。这样，可进一步提高装置200进行数据处理时的吞吐量。

[0074] 装置200还可包括账号管理模块(未示出)，账号管理模块基于存储模块中存储的账号状态控制账号的活动。账号管理模块还配置为将账号状态传输至第三方控制端。第三方控制端例如可为广告投放平台。

[0075] 图4示出根据本申请实施方式的Storm模块中的组件的分布式拓扑结构。图5示出根据本申请的一个实施方式的生成账号状态的装置在生成账号状态时与外部机构形成的分布式拓扑结构。

[0076] 生成账号状态的方法包括：以消息队列形式采集与账号对应的第一主题数据和第二主题数据；以及通过Storm模块220中的组件执行的操作。

[0077] 参照图4中示出的拓扑结构。通过Storm模块220中的组件执行的操作包括：通过ImpSout组件222从流式收集模块210读取第一主题数据并将第一主题数据发送至

CacheBolt组件223;通过ClickSpout组件224从流式收集模块读取第二主题数据,并将第二主题数据发送至ControlBolt组件225。例如,ImpSout组件222和ClickSpout组件224分别从消息队列实时收集曝光日志和点击日志,解析后发送给各自的下游。由于第一主题数据和第二主题数据以消息队列形式存储,因此在数据写入与读取过程中节省服务器成本并且降低数据丢失的风险。

[0078] 通过Storm模块220中的组件执行的操作还可包括:通过FlushSpout组件227每隔第一预定时间产生发送指令,CacheBolt组件223响应于发送指令将第一主题数据发送至ControlBolt组件225。其中,CacheBolt组件223还可对第一主题数据进行预处理以生成预处理数据,并且响应于送指令将预处理数据ControlBolt组件225。例如,可基于第一预定时间内的曝光量进行汇总与计算,以生成缓存金额(例如第一预定时间内曝光量所花费的金额)。通过对快速并发数据的缓存与预处理可极大减少控制器的处理量,进而提高数据处理能力。

[0079] 通过Storm模块220中的组件执行的操作还可包括:通过InfoSpout221组件每隔第二预定时间获取与账号对应的物料信息。与账号对应的物料信息可为广告库中投放的物料信息。物料信息可为广告单价,广告中账号、活动、组和创意的映射关系以及广告离线报表分小时花费等,物料信息还可包括广告主在广告投放过程中存入与提取金额的信息。

[0080] 通过Storm模块220中的组件执行的操作还可包括:通过DbBolt组件226同步由InfoSpout组件221获取的物料信息,基于物料信息改变而更新缓存中的物料信息,并将计算指令发送至ControlBolt组件225。通过该操作,可实时地更新缓存中的物料信息,从而可在物料信息改变的情况下对重新生成账号状态,同时更新其他工作节点进行计算所需的物料信息。这样进一步保证了数据处理的实时性。

[0081] 通过Storm模块220中的组件执行的操作还可包括:通过ControlBolt组件225响应于接收到第一主题数据、第二主题数据或计算指令基于缓存中的物料信息生成账号状态。

[0082] 通过Storm模块220中的组件执行的操作还可包括:通过CacheBolt组件223对第一主题数据进行预处理以生成预处理数据;以及通过CacheBolt组件223响应于发送指令将预处理数据发送至ControlBolt组件225。例如,可基于缓存的第一主题数据和缓存中的物料信息来生成预处理数据,但本申请的实施方式不限于此。预处理数据可以是例如基于第一预定时间内的曝光量和缓存中的物料信息针对广告活动和账号维度进行汇总与计算而生成的缓存金额,例如第一预定时间内曝光量所花费的金额。

[0083] 根据本申请的一个实施方式,生成账号状态的方法还可包括:响应于接收到预处理数据通过ControlBolt组件225基于缓存中的物料信息生成账号状态。这样可进一步减少ControlBolt组件225所需处理的数据量。通过对快速并发数据的缓存与预处理可极大减少控制器的处理量,进而提高数据处理能力。

[0084] 通过图4中示出的拓扑结构实现的生成账号状态的方法通过横向扩展工作节点数以对不同数据进行并行处理以及通过纵向增加工作节点数以对一部分数据进行缓存与预处理,减少了用于最终生成账号状态的ControlBolt组件225单位时间需要进行数据处理的量。

[0085] 本方法还包括通过异步交互的方式每隔第三预定时间存储账号状态。例如可由Redis存储系统通过异步交互的方式每隔第三预定时间存储账号状态。通过异步交互的方

式，可在不影响通过ControlBolt组件225生成账号状态前提下，每隔预定时间存储账号状态。存储的账号状态可用于程序重启的数据回踩以及核心实时花费数据的可视化等。由于每隔第三预定时间来存储账号状态，所以可查看由第三预定时间划分的时间节点处的账号状态，便于与账号状态对应的数据的回踩和再利用。

[0086] 生成账号状态的方法还可包括基于存储模块中存储的账号状态控制账号的活动。例如可根据与账号状态对应的账号余额控制账号状态活动，当账号余额不足以支付广告活动时，可针对该账号终止广告活动；当广告主针对广告账号投入金额后，可针对账号开启广告活动。

[0087] 账号状态的方法还可包括将存储模块230中账号状态传输至第三方控制端。第三方控制端可基于账号状态执行程序重启的数据回踩、实时花费数据、账户余额的可视化。第三方控制端可为广告供应平台。

[0088] 图5中示出的分布式拓扑结构示出的外部机构可包括数据库以及Kafka消息队列系统。其中，数据库可为MySQL数据库管理系统(以下简称为“mysql”)。

[0089] InfoSpout组件221从mysql定时或每隔第二预定时间间隔(例如每隔1分钟)扫描广告库中投放的物料信息。ImpSout组件222和ClickSpout组件224与Kafka消息队列系统的topic的分区数保持一致，从消息队列实时收集曝光日志(例如，第一主题数据)和点击日志(例如第二主题数据)。除此之外，图5中所示的拓扑结构与图4相同，因此将省略重复的描述。

[0090] 参照图4和图5描述的Storm模块中的拓扑结构仅是示例性的，并且Storm模块中的拓扑结构可根据数据类型的多少、数据产生的速度以及计算节点处需要完成计算等而在横向增加读取数据的工作节点以及在纵向增加对数据进行缓存与预处理的工作节点，从而实现数据处理能力的扩展。

[0091] 根据本申请一方面的生成账号状态的方法及装置通过横向扩展工作节点数以对不同数据进行并行处理以及通过纵向增加工作节点数以对一部分数据进行缓存与预处理，减少了用于最终生成账号状态的控制器单位时间需要进行数据处理的量。这样可实现数据处理能力在空间和时间维度上的扩展。

[0092] 根据本申请另一方面的生成账号状态方法及装置由于借助于Storm流式计算框架，在提高数据处理能力的前提下，进一步提高生成账号状态的方法及装置可靠性和容错特性。另外以消息队列的方式进行数据采集，会节省服务器成本、提升服务整体的吞吐量以及减少数据丢失的风险。

[0093] 然已经参考本申请的示例性实施方式具体示出和描述了本申请，但是将由本领域普通技术人员理解的是，在不脱离本申请的精神和范围的情况下，可在本发明中做出形式和细节上的各种改变，本申请的精神和范围如由所附权利要求限定。示例性实施方式应仅在描述性的意义上进行考虑，而不是出于限制的目的。

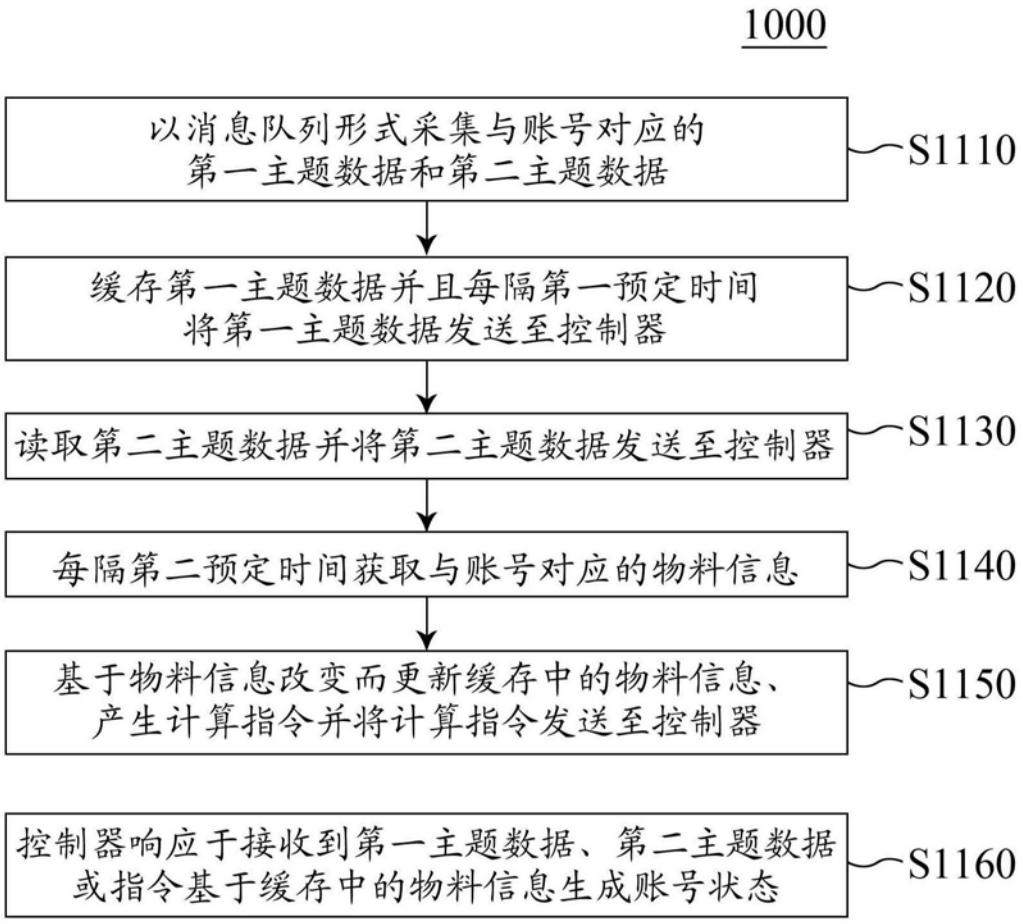

1000
以消息队列形式采集与账号对应的第一主题数据和第二主题数据
S1110
缓存第一主题数据并且每隔第一预定时间将第一主题数据发送至控制器
S1120
读取第二主题数据并将第二主题数据发送至控制器
S1130
每隔第二预定时间获取与账号对应的物料信息
S1140
基于物料信息改变而更新缓存中的物料信息、产生计算指令并将计算指令发送至控制器
S1150
控制器响应于接收到第一主题数据、第二主题数据或指令基于缓存中的物料信息生成账号状态
S1160

图1

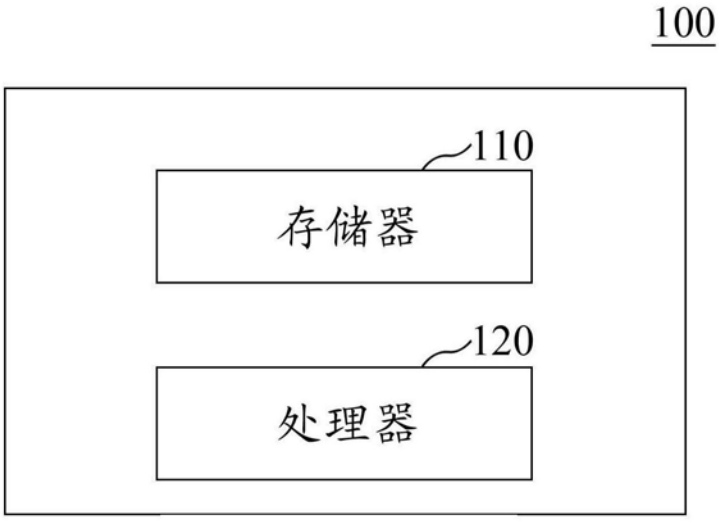

100
110
存储器
120
处理器

图2

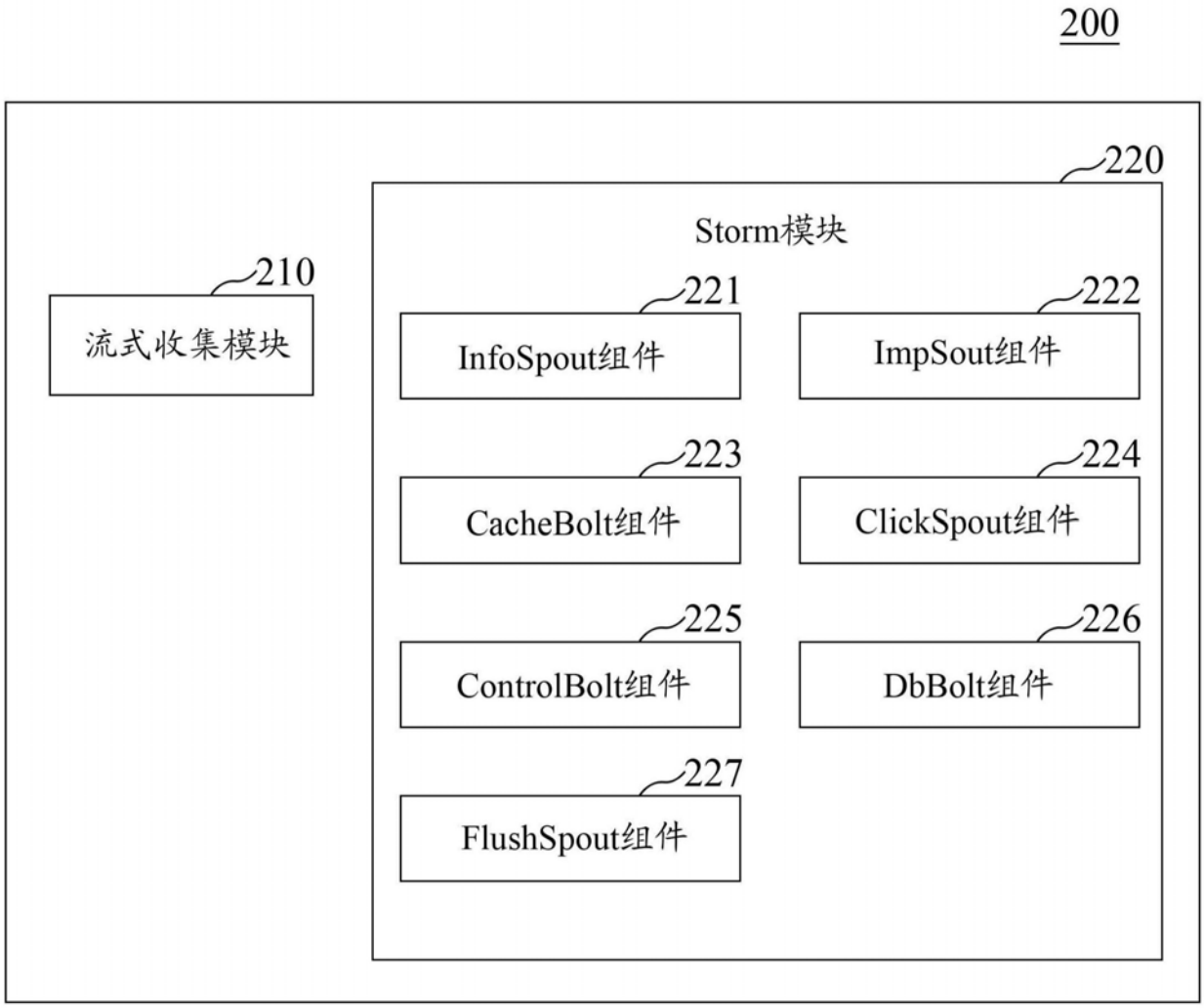

200
220
Storm模块
210
流式收集模块
221
InfoSpout组件
222
ImpSout组件
223
CacheBolt组件
224
ClickSpout组件
225
ControlBolt组件
226
DbBolt组件
227
FlushSpout组件

图3

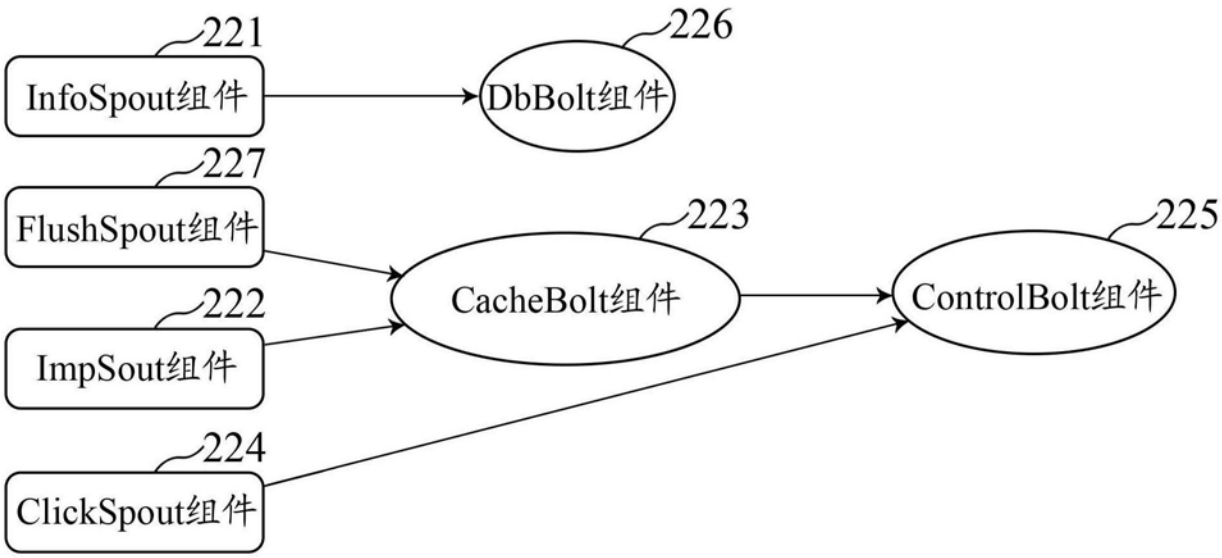

221
InfoSpout组件
226
DbBolt组件
227
FlushSpout组件
223
CacheBolt组件
225
ControlBolt组件
222
ImpSout组件
224
ClickSpout组件

图4

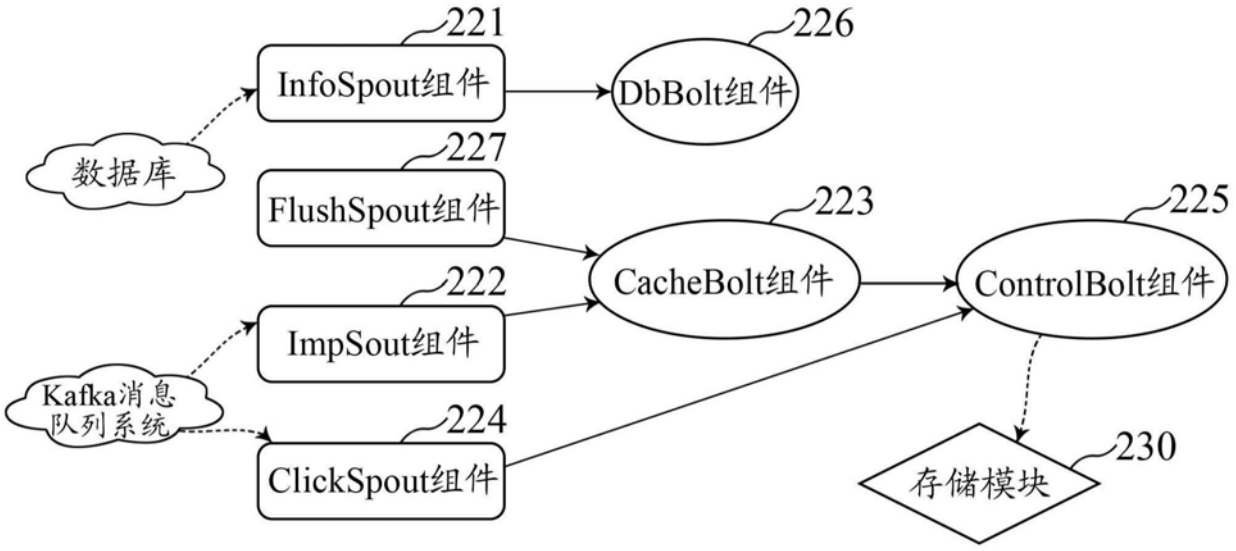

221
InfoSpout组件
226
DbBolt组件
数据库
227
FlushSpout组件
223
CacheBolt组件
225
ControlBolt组件
222
ImpSout组件
Kafka消息
队列系统
224
ClickSpout组件
230
存储模块

图5