

基于用户画像的数据处理方法、装置、设备、介质及程序

技术领域

5 本申请实施例涉及数据处理技术领域，尤其涉及一种基于用户画像的数据处理方法、装置、设备、介质及程序。

背景技术

10 用户画像是一种描述用户、联系用户诉求与产品设计方向的工具，应用于产品设计、精准营销等领域。服务器可以根据用户性别、年龄、页面访问情况、商品交易情况等用户数据，确定用户的行为喜好等用户特征，进而生成用户画像，从而可以根据用户画像即一个或者多个用户特征发掘用户需求，提供给用户更高效、更有针对性的服务。

15 Kappa 架构是一种数据处理方式，不仅可以对数据进行实时处理，还可以基于其消息队列的数据保留功能，实现数据重放能力，进而完成对数据的离线分析或者再次计算。例如，服务器在重新计算用户特征，以生成用户画像时，服务器可以基于 Kappa 架构的数据重放能力对消息队列存储的多个用户数据进行再次计算，在再次计算过程中，服务器可以依次读取多个用户数据中的每个用户数据，当读取到第一个用户数据时，服务器可以对第一个用户数据进行计算，生成第一个用户数据对应的用户特征，并将该用户特征存储在数据表中，当读取到第二个用户数据时，可以对第二个用户数据进行计算，生成第二个用户数据对应的用户特征，然后服务器可以使用第二个用户数据对应的用户特征更新数据表中
20 第一个用户数据对应的用户特征，以此类推，服务器可以完成对多个用户数据的再次计算，得到用户特征，从而根据用户特征生成用户画像。

25 然而，上述计算过程中的数据存储方式会导致用户特征查询不准确，从而导致根据查询到的用户特征生成的用户画像不准确，例如：若在再次计算过程中对用户特征进行查询，显然此时服务器还没有完成对上述多个用户数据的再次计算，那么查询到的用户特征不会是根据上述多个用户数据计算出的用户特征，进而导致查询结果不准确，从而对用户画像的生成和应用都造成一定的影响。

发明内容

30 本申请提供一种基于用户画像的数据处理方法、装置、设备、介质及程序，以解决现有技术中因数据存储导致的用户特征查询不准确，从而导致根据查询到的用户特征生成的用户画像不准确的问题，可以提高用户特征查询的准确性，进而提高生成用户画像的准确性，从而提高用户画像应用的效率和精确性。

35 第一方面，本申请提供一种基于用户画像的数据处理方法，该方法包括：获取多个用户数据以及多个用户数据中最后一个用户数据的生成时间；若从多个用户数据中每读取一个第一用户数据，则生成第一用户数据对应的用户特征；若第一用户数据的生成时间和最后一个用户数据的生成时间一致，则将第一用户数据对应的用户特征存储至目标数据库；若第一用户数据生成时间和最后一个用户数据的生成时间不一致，则不将第一用户数据对应的用户特征存储至目标数据库；基于目标数据库中的用户特征生成用户画像。

40 第二方面，本申请提供一种基于用户画像的数据处理装置，包括：第一获取模块、生成模块、处理模块和第二生成模块，其中，第一获取模块用于获取多个用户数据以及多个用户数据中最后一个用户数据的生成时间；生成模块用于若从多个用户数据中每读取一个第一用户数据，则生成第一用户数据对应的用户特征；处理模块用于若第一用户数据的生

成时间和最后一个用户数据的生成时间一致，则将第一用户数据对应的用户特征存储至目标数据库；若第一用户数据生成时间和最后一个用户数据的生成时间不一致，则不将第一用户数据对应的用户特征存储至目标数据库；第二生成模块用于基于目标数据库中的用户特征生成用户画像。

5 第三方面，提供一种电子设备，包括：处理器和存储器，该存储器用于存储计算机程序，该处理器用于调用并运行该存储器中存储的计算机程序，执行如第一方面或其各实现方式中的方法。

第四方面，提供一种计算机可读存储介质，用于存储计算机程序，计算机程序使得计算机执行如第一方面或其各实现方式中的方法。

10 第五方面，提供一种计算机程序产品，包括计算机程序指令，该计算机程序指令使得计算机执行如第一方面或其各实现方式中的方法。

第六方面，提供一种计算机程序，计算机程序使得计算机执行如第一方面或其各实现方式中的方法。

通过本申请技术方案，服务器可以先获取多个用户数据以及多个用户数据中最后一个用户数据的生成时间，若从多个用户数据中每读取一个第一用户数据，则服务器可以生成第一用户数据对应的用户特征，若第一用户数据的生成时间和最后一个用户数据的生成时间一致，则服务器可以将第一用户数据对应的用户特征存储至目标数据库；若第一用户数据生成时间和最后一个用户数据的生成时间不一致，则服务器可以不将第一用户数据对应的用户特征存储至目标数据库，最后，服务器可以基于目标数据库中的用户特征生成用户画像。在上述过程中，服务器可以通过判断当前读取的用户数据的生成时间是否和最后一个用户数据的生成时间一致，进而可以确定当前读取的用户数据是否是最后一个用户数据，从而可以只将最后一个用户数据对应的用户特征存储在目标数据库中，而不将其他用户数据对应的用户特征存储在目标数据库中，那么在对用户特征进行查询时，例如在服务器读取用户数据时对用户特征进行查询，查询到的结果不会是所有其他用户数据对应的用户特征，只会是最后一个原始对应的用户特征即是根据所有用户数据计算出来的最终用户特征，从而服务器可以根据上述最终用户特征生成正确的用户画像，从而可以解决现有技术中因数据存储导致的用户特征查询不准确，从而导致根据查询到的用户特征生成的用户画像不准确的问题，可以提高用户特征查询的准确性，进而提高生成用户画像的准确性，从而提高用户画像应用的效率和精确性。

30

附图说明

为了更清楚地说明本发明实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，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，还可以根据这些附图获得其他的附图。

35

- 图 1 为本申请实施例提供的一种应用场景图；
- 图 2 为本申请实施例提供的一种基于用户画像的数据处理方法的流程图；
- 图 3 为本申请实施例提供的一种基于用户画像的数据处理的示意图；
- 图 4 为本申请实施例提供的另一种基于用户画像的数据处理的示意图；
- 40 图 5 为本申请实施例提供的再一种基于用户画像的数据处理的示意图；
- 图 6 为本申请实施例提供的又一种基于用户画像的数据处理的示意图；
- 图 7 为本申请实施例提供的又一种基于用户画像的数据处理的示意图；
- 图 8 为本申请实施例提供的又一种基于用户画像的数据处理的示意图；
- 图 9 为本申请实施例提供的又一种基于用户画像的数据处理的示意图；

图 10 为本申请实施例提供的一种基于用户画像的数据处理装置 1000 的示意图；
图 11 是本申请实施例提供的电子设备的示意性框图。

具体实施方式

5 下面将结合本发明实施例中的附图，对本发明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发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例，都属于本发明保护的范围。

需要说明的是，本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等
10 是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖
15 不排除他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或服务器不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、
产品或设备固有的其它步骤或单元。

如上所述，服务器在重新计算用户特征，以生成用户画像时，服务器可以基于 Kappa
架构的数据重放能力对其消息队列存储的多个用户数据进行再次计算，在再次计算过程中，
服务器可以依次读取多个用户数据中的每个用户数据，当读取到第一个用户数据时，服
20 务器可以对第一个用户数据进行计算，生成第一个用户数据对应的用户特征，并将该用户特征存储在数据表中，当读取到第二个用户数据时，可以对第二个用户数据进行计算，生成
第二个用户数据对应的用户特征，然后服务器可以使用第二个用户数据对应的用户特征更
新数据表中第一个用户数据对应的用户特征，以此类推，服务器可以完成对多个用户数据
的再次计算。然而，上述计算过程中的数据存储方式会导致用户特征查询不准确，从而导
25 致根据查询到的用户特征生成的用户画像不准确，例如：若在再次计算过程中对用户特征
进行查询，显然此时服务器还没有完成对多个用户数据的再次计算，那么查询到的用户特征
不会是根据上述多个用户数据计算出的用户特征，从而导致查询结果不准确，进而对用户
画像的生成和应用都造成一定的影响。

为了解决上述技术问题，服务器可以先获取多个用户数据以及多个用户数据中最后
一个用户数据的生成时间，若从多个用户数据中每读取一个第一用户数据，则服务器可以
30 生成第一用户数据对应的用户特征，若第一用户数据的生成时间和最后一个用户数据的生成
时间一致，则服务器可以将第一用户数据对应的用户特征存储至目标数据库；若第一用户
数据生成时间和最后一个用户数据的生成时间不一致，则服务器可以不将第一用户数据对
应的用户特征存储至目标数据库，最后，服务器可以基于目标数据库中的用户特征生成用
户画像。在上述过程中，服务器可以通过判断当前读取的用户数据的生成时间是否和最后
35 一个用户数据的生成时间一致，进而可以确定当前读取的用户数据是否是最后一个用户数
据，从而可以只将最后一个用户数据对应的用户特征存储在目标数据库中，而不将其他用
户数据对应的用户特征存储在目标数据库中，那么在对用户特征进行查询时，例如在服
务器读取用户数据时对用户特征进行查询，查询到的结果不会是其用户数据对应的用户特征，
只会是最后一个原始对应的用户特征即是根据所有用户数据计算出来的最终用户特征，
40 从而服务器可以根据上述最终用户特征生成正确的用户画像，从而可以解决现有技术中因
数据存储导致的用户特征查询不准确，从而导致根据查询到的用户特征生成的用户画像不
准确的问题，可以提高用户特征查询的准确性，进而提高生成用户画像的准确性，从而提
高用户画像应用的效率和精确性。

应理解的是，本申请技术方案可以应用于如下场景，但不限于：

在一些可实现方式中，图 1 为本申请实施例提供的一种应用场景图，如图 1 所示，该应用场景中可以包括终端 110、服务器 120。终端 110 与服务器 120 之间可以通信。

示例性的，服务器 120 可以基于 Kappa 架构的数据重放能力对其消息队列存储的多个用户数据进行再次计算，得到用户特征，并将用户特征存储在数据库中，从而可以基于该数据库中的用户特征生成用户画像，该数据库可以是服务器 120 内部的数据库，也可以是服务器 120 内部的数据库，本申请对此不做限制。例如，服务器 120 可以是数据中台，在需要重新构建用户画像时，如在针对股票开户情况等用户数据的计算口径变化时，数据中台可以重新获取消息队列中的用户数据，并确定新的用户特征，进而重新生成用户画像。终端 110 上可以安装用户特征查询客户端，用户通过访问该用户特征查询客户端，基于自然语言对上述用户特征进行查询，或者，终端 110 上可以不安装有上述用户特征查询客户端，用户通过浏览器，基于自然语言对上述用户特征进行查询，查询时，服务器 120 可以将上述自然语言转换为自然语言对应的结构化查询语言 (Structured Query Language, SQL)，并基于该自然语言对应的 SQL 对上述数据库中存储的用户特征进行查询，将查询结果返回给终端 110。

在一些可实现方式中，终端 110 可以为手机、平板电脑、桌面型、膝上型、手持计算机、笔记本电脑、车载设备、超级移动个人计算机 (Ultra-Mobile Personal Computer, UMPC)、上网本，以及蜂窝电话、个人数字助理 (Personal Digital Assistant, PDA)、增强现实 (Augmented Reality, AR)、虚拟现实 (Virtual Reality, VR) 设备，本申请对此不做限制。服务器 120 可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云计算服务的云服务器，本申请实施例对此不做限制。

应该理解，图 1 中的终端、服务器的数目仅仅是示意性的，实际上，根据实际情况需要可以设置任意数目的终端和服务器，本申请对此不做限制。

在介绍了本申请实施例的应用场景之后，下面将对本申请技术方案进行详细阐述：

图 2 为本申请实施例提供的一种基于用户画像的数据处理方法的流程图，该方法可以由如图 1 所示的服务器 120 执行，本申请对此不做限制，如图 2 所示，该方法可以包括如下步骤：

S210：获取多个用户数据以及多个用户数据中最后一个用户数据的生成时间；

S220：若从多个用户数据中每读取一个第一用户数据，则生成第一用户数据对应的用户特征；

S230：判断第一用户数据的生成时间和最后一个用户数据的生成时间是否一致，若第一用户数据的生成时间和最后一个用户数据的生成时间一致，则执行 S240，若第一用户数据生成时间和最后一个用户数据的生成时间不一致，则执行 S250；

S240：将第一用户数据对应的用户特征存储至目标数据库；

S250：不将第一原始数据对应的用户特征存储至目标数据库；

S260：基于目标数据库中的用户特征生成用户画像。

应理解的是，上述步骤是服务器在数据重放时执行的步骤，例如，上述步骤可以是服务器基于 Kappa 架构的数据重放能力对多个用户数据进行再次计算，以生成用户画像时执行的步骤。需要说明的是，Kappa 架构包括消息队列、流处理集群、数据表，其中，消息队列具有数据保留功能，可以存储用户数据，在数据重放时，流处理集群可以读取消息队列中的用户数据。由于卡夫卡 (Kafka) 是一种具有历史数据保存、历史数据重放功能的消息系统，所以可以选择卡夫卡作为消息队列。流处理集群可以对从消息队列中读取的用户数据进行计算，得到用户数据对应的用户特征。由于 Flink 是一种支持流批一体处理方式的计算框架，所以可以选择 Flink 作为流处理集群。数据表可以用来存储流处理集群计算得到的用户数据对应的用户特征。

示例性的，本申请在此先以服务器基于 Kappa 架构的数据重放能力对多个用户数据进

行再次计算，以生成用户画像时为例，对上述步骤进行简单介绍。首先，服务器可以从其他数据库获取多个用户数据，并将该多个用户数据存储于消息队列中，在服务器需要基于 Kappa 架构的数据重放能力对其消息队列存储的多个用户数据进行再次计算时，服务器可以获取上述多个用户数据中最后一个用户数据的生成时间，如果数据的版本或者生成时间比存储的版本小，将该过程数据过滤或者删除，可以保证正确地写入的数据。具体的，服务器可以基于流处理集群依次读取消息队列中存储的多个用户数据，当读取到第一个用户数据时，流处理集群可以对第一个用户数据进行计算，生成第一个用户数据对应的用户特征，并判断第一个用户数据的生成时间和最后一个用户数据的生成时间不一致，则服务器不会将第一个用户数据对应的用户特征存储在数据表即上述目标数据库中，接着，服务器可以基于流处理集群读取消息队列中存储的第二个用户数据，流处理集群可以对第二个用户数据进行计算，生成第二个用户数据对应的用户特征，并判断第二个用户数据的生成时间和最后一个用户数据的生成时间是否不一致，若一致，则服务器可以确定该第二个用户数据是上述最后一个用户数据，从而服务器可以确定该第二个用户数据对应的用户特征是最后一个用户数据对应的用户特征，那么服务器可以将上述用户特征存储在数据表中；若不一致，则服务器可以确定该第二个用户数据不是上述最后一个用户数据，那么服务器可以不将第二个用户数据对应的用户特征存储在数据表中，类似的，对于其他用户数据，服务器可以执行和上述第二个用户数据类似的步骤。这样一来，在终端对用户特征进行查询时，查询结果只会是最后一个用户数据即最新的用户数据对应的用户特征，而不会是其他用户数据对应的用户特征，那么查询结果就是准确的，从而可以解决现有技术中因数据存储导致的用户特征查询不准确，从而导致根据查询到的用户特征生成的用户画像不准确的问题，可以提高用户特征查询的准确性，进而提高用户画像应用的效率和精确性。

需要说明的是，如图 3 所示，服务器在构建用户画像时，可以先根据用户性别、年龄、页面访问情况、商品交易情况等用户数据，确定用户的行为喜好等用户特征，进而构建用户画像，从而可以根据用户画像即一个或多个用户特征发掘用户需求，提供给用户更高效、更有针对性的服务，其中，用户特征可以存储在目标数据库中，在根据用户特征发掘用户需求时，可以通过用户画像接口服务查询对应的用户特征，以将查询到的用户特征应用到下游业务中，如产品设计、精准营销等领域。

下面实施例中将以服务器为数据中台为例来对本申请技术方案进行说明。需要说明的是，在下面的实施例中，都是基于数据合规来获取、计算用户数据的，且获取的用户数据都是经过用户授权的，同时也对用户数据、用户特征等进行了加密和保护。

在一些可实现方式中，假设数据中台需要确定的用户特征是：股票开户用户数量，确定该用户特征需要用到的用户数据是：用户的股票开户情况，数据中台可以基于 Kappa 架构对上述用户数据进行计算，确定用户特征，其中，Kappa 架构的消息队列是卡夫卡，流处理集群是 Flink，存储用户特征的目标数据库是数据表。在数据中台需要根据用户数据重新计算用户特征时，例如，数据中台之前在确定股票开户用户数量时，计算口径是：统计用户数据中开户状态是已开户的用户数据的数量，但是，数据中台现在确定股票开户用户数量时，计算口径变化为：统计用户数据中开户状态是已开户和审核中的用户数据的数量，那么数据中台需要重新确定股票开户用户数量这一用户特征，具体过程为：首先，数据中台可以从其他数据库如记录用户股票开户情况等用户数据的业务源数据库中获取关于用户股票开户情况的多个用户数据，并将其存储在卡夫卡中，假设数据中台获取到的关于用户股票开户情况的多个用户数据有三个，第一个用户数据是用户 1 在 2022 年 6 月 30 日 13:22 时对股票 1 的开户状态是已开户，第二个用户数据是用户 2 在 2022 年 6 月 30 日 13:23 时对股票 1 的开户状态是审核中，第三个用户数据是用户 3 在 2022 年 6 月 30 日 13:30 时对股票 2 的开户状态是未开户，接着，数据中台可以获取该多个用户数据中最后一个用户数据的生成时间即第三个用户数据的生成时间：2022 年 6 月 30 日 13:30，然后，数据中台可以

基于 Flink 从卡夫卡中分别读取上述三个用户数据，当读取到第一个用户数据时，Flink 可以确定第一个用户数据对应的用户特征即股票开户用户数量是 1，并且可以判断第一个用户数据的生成时间是 2022 年 6 月 30 日 13:22，这和最后一个用户数据的生成时间 2022 年 6 月 30 日 13:30 不一致，则数据中台不会将第一个用户数据对应的用户特征存储在数据表中，接着，数据中台可以基于 Flink 从卡夫卡中读取第二个用户数据，Flink 可以确定第二个用户数据对应的用户特征是 2，并且可以判断第二个用户数据的生成时间是 2022 年 6 月 30 日 13:23，这和最后一个用户数据的生成时间 2022 年 6 月 30 日 13:30 不一致，则数据中台不会将第二个用户数据对应的用户特征存储在数据表中，最后，数据中台可以基于 Flink 从卡夫卡中读取第三个用户数据，Flink 可以确定第三个用户数据对应的用户特征是 2，并且可以判断第三个用户数据的生成时间是 2022 年 6 月 30 日 13:30，这和最后一个用户数据的生成时间 2022 年 6 月 30 日 13:30 一致，则数据中台可以将第三个用户数据对应的用户特征存储在数据表中，从而数据中台可以确定出用户特征：股票开户用户数量为 2。那么对于在上述再次计算用户特征的过程中接收到的查询用户特征的查询请求，返回的查询结果只会是根据最后一个用户数据计算出的用户特征即股票开户用户数量为 2，而由于根据其他用户数据计算出的用户特征并没有存储在数据表中，所以查询结果就不会是根据其他用户数据计算出的用户特征如股票开户用户数量为 1，因此可以提高用户特征查询的准确性，进而可以提高生成用户画像的准确性和效率。

在一些可实现方式中，如图 4 所示，数据中台在构建用户画像时，可以先通过 Binlog 同步或者 Agent 上报等方式从其他数据库如业务数据源中获取用户数据，该用户数据可以用户股票开户情况、用户登录情况等用户数据，并将获取到的用户数据放在数据运营层（Operation Data Store，ODS）层，其中，ODS 层是数据模型中最接近数据源中的原始数据即用户数据的一层，一般可以将数据源中的原始数据原封不动的接入。Binlog 同步是数据中台从上述其他数据库中获取用户数据并存储在 ODS 层的一种方式，Agent 上报是上述其他数据库主动发送用户数据给 ODS 层的一种方式，另外，Binlog 同步包括 Changelog（更新日志），数据中台可以通过更新日志的方式实时捕捉其他数据库中的变更数据，并将变更数据同步到 ODS 层，例如，其他数据库中记录了一个用户数据：用户 2 对股票 1 的开户状态为审核中，数据中台可以通过 Binlog 同步从其他数据库中获取到这个用户数据并存储在 ODS 层，那么此时 ODS 层存储的用户数据为用户 2 对股票 1 的开户状态为审核中，之后，在其他数据库中记录的上述用户数据变更为：用户 2 对股票 1 的开户状态为已开户时，数据中台可以通过更新日志的方式实时捕捉上述变更数据即“已开户”，那么数据中台可以将该变更数据同步到 ODS 层，即将 ODS 层中存储的用户数据用户 2 对股票 1 的开户状态为审核中的“审核中”变更为“已开户”，如此一来，可以减少数据传输开销，提高传输效率。之后，数据中台可以在计算层对 ODS 层的用户数据进行计算，确定出用户特征，计算方法可以是流计算方法，也可以是批量计算方法，具体计算过程本申请将在下面的实施例中进行详细介绍，在此先不赘述。接着，数据中台可以将用户特征存储在服务数据层（Data Warehouse Service，DWS）层，进而可以生成用户画像，其中，DWS 层可以用来存储计算结果，如针对 ODS 层中的用户数据的计算结果。可以理解的是，在上述数据中台基于 Kappa 架构对上述用户数据进行计算中，卡夫卡可以作为 ODS 层，数据表可以作为 DWS 层。

在一些可实现方式中，服务器从多个用户数据中每读取一个第一用户数据，则生成第一用户数据对应的用户特征时，可以先按照计算口径和应用场景，选择第一用户数据对应的用户特征的生成方法，生成方法是流生成方法或批量生成方法，然后，服务器可以基于上述第一用户数据对应的用户特征的生成方法生成第一用户数据对应的用户特征。其中，数据中台可以基于 Kappa 架构完成流生成方法即流计算方法，可以通过 Airflow 定时触发批量生成方法即批量计算方法，当然也可以通过其他方法定时触发批生成方法，本申请对此不做限制。其中，Airflow 是一种任务调度工具，可以设置计算任务如批量计算任务的触

发时间和计算任务的执行时长如批量计算任务的执行时长。需要说明的是，服务器可以基于 Kappa 架构完成流生成方法，还可以基于 Lambda 架构完成流生成方法，本申请对此不做限制。其中，Lambda 架构是一种的数据处理架构，包括实时处理即流计算和离线处理即批量计算两个模块，所以维护成本较高，而 Kappa 架构没有离线处理模块即批量计算模块，因此基于 Kappa 架构完成流生成方法可以降低维护成本。

应理解的是，批量生成方法即批量计算方法是一种批量、高时延、主动发起的计算方法。批量计算方法必须先定义计算作业逻辑，并提交到流计算系统，且该计算作业逻辑在整个运行期间是不可更改的。批量计算方法计算的数据一定需要预先加载到计算系统，后续计算系统才会在数据加载完成后进行计算。不同于批量计算方法，流生成方法即流计算方法更加强调计算数据流和低时延。流计算方法可以将大量数据平摊到每个时间点上，连续地进行小批量的进行传输，数据持续流动，计算完之后就丢弃该数据。流计算方法计算后的结果可以立刻投递到在线系统，做到实时化展现。

示例性的，假设需要确定的用户特征 1 是：用户 1 在 2022 年 5 月 30 日 13:30 前 10 天内对页面 1 的浏览次数；用户特征 1 的应用场景 1 是：根据用户特征 1 预测用户 1 在 2022 年 5 月 30 日 13:30 后 10 天内对页面 1 的浏览次数，以判断用户 1 在 2022 年 5 月 30 日 13:30 后 10 天内对页面 1 的真实浏览次数是否和上述预测的浏览次数一致；确定用户特征 1 的计算口径 1 是：获取用户 1 在 2022 年 5 月 30 日 13:30 前 10 天内对所有页面的浏览情况数据，然后统计该数据中对页面 1 的浏览次数，根据上述用户特征 1 对应的应用场景 1 和计算口径 1 可以确定出对确定用户特征 1 的实时性要求不高，所以数据中台可以选择用户特征 1 的生成方法是批生成方法即批计算方法，例如，数据中台可以通过 Airflow 设置生成用户特征 1 的开始时间如 2022 年 6 月 30 日 13:30，然后在该时间时，获取用户 1 在 2022 年 5 月 30 日 13:30 前 10 天内对所有页面的浏览情况数据，然后对该数据采用批计算方法，统计该数据中对页面 1 的浏览次数，从而确定出用户特征 1。

示例性的，假设需要确定的用户特征 2 是：用户 2 是否在近三天登录过应用 1；用户特征 2 的应用场景 2 是：根据用户特征 2 判断用户 2 是否为活跃用户，若判断用户 2 为活跃用户，则向用户 2 实时推送消息 1，若判断用户 2 不是活跃用户，则不向用户 2 推送消息 1；用户特征 2 的计算口径 2 是：获取用户 2 最近一次登录应用 1 的数据，判断该数据的生成时间是否在近三天内，若在近三天内，则确定用户特征 2 为“1”，若不在近三天内，则确定用户特征 2 为“0”，根据上述用户特征 2 对应的应用场景 2 和计算口径 2 可以确定出对确定用户特征 2 的实时性要求较高，所以数据中台可以选择用户特征 2 的生成方法是流生成方法即流计算方法。例如，假设数据中台需要在 2022 年 6 月 30 日 13:30 判断是否要向用户 2 推送消息 1，则数据中台可以实时获取用户 2 最近一次登录应用 1 的数据：用户 2 在 2022 年 6 月 30 日 12:30 登录应用 1，可以判断该数据的生成时间在三天内，那么可以确定用户特征为“1”，即可以判断用户 2 为活跃用户，从而可以实时向用户 2 推送消息 1。

在一些可实现方式中，如图 4 所示，本实施例中的目标数据库可以包括但不限于远程字典服务（Remote Dictionary Server，Redis）数据库、ElasticSearch 数据库以及云端数据库等。上述数据中台在将用户特征存储在 DWS 层（目标数据库）时，可以将用户特征分别存储在 Redis 数据库、ElasticSearch 数据库或者云端服务器等数据库中。

应理解的是，Redis 数据库采用的是 key-value 存储方式，即每一条记录只包含一个用于查询数据的 Key 以及与之对应的存储数据的 value，所以对实时性要求较高、查询数据量较小的查询接口如联机事务处理（On-Line Transaction Processing，OLTP）一般会选择 Redis 数据库作为查询引擎；ElasticSearch 数据库可以实现高性能的复杂聚合查询，所以对实时性要求不高、查询数据量较大的查询接口如联机分析处理 OLAP（OLAP，On-Line Analytical Processing）一般会选择 ElasticSearch 数据库作为查询引擎。而云端存储适用于数据量较大、

数据涵盖范围广、实时性也要求较高的数据。具体的，在确定一个数据包的存储方式时，本实施例中先获取数据包的数据参数，数据参数可以包括数据名称对应的查询频率Dat_fre、数据包对应的数据量Dat_voe、数据名称对应的数据优先级Dat_pro；之后基于数据参数确定数据包对应的属性参数Dat_pre为：

$$\text{Dat_pre} = \begin{cases} \alpha \cdot \text{Dat_pro} \cdot (\text{Dat_fre} + \gamma \cdot \log_2 \text{Dat_voe}); & \text{Dat_fre} \geq \text{Dat_mon} \\ \epsilon \cdot \text{Dat_pro} \cdot \text{Dat_fre} + \gamma \cdot \log_2 \text{Dat_voe}; & \text{Dat_fre} < \text{Dat_mon} \end{cases}$$

- 5 其中， α 、 γ 、 ϵ 表示根据历史数据训练得到属性因子，Dat_mon表示预设的频率阈值，通过频率阈值衡量查询频率的高低，以针对性的确定对应的属性参数确定方式。本实施例中将数据的查询频率、优先级以及数据量考虑进行属性参数的计算中，以基于上述参数来衡量数据包的存储方式。之后基于属性参数来确定数据包对应的存储位置，具体可以预设各存储方式对应的参数阈值，根据各存储方式对应的参数阈值确定数据包对应的存储位置。
- 10 通过上述方式保证对数据存储的个性化存储，进而提高数据存储和数据调用的效率，降低数据存储的成本。

- 15 在将用户特征分别存储在 Redis 数据和 ElasticSearch 数据库以及云端服务器等存储介质之后，数据中台可以基于查询请求选择确定该查询请求对应的查询接口，从而使得不同的查询接口在基于合适的数据库进行查询时，可以保证不同的查询接口对应的数据库都存储有用户特征，以提高查询效率。其中，上述数据中台可以基于查询请求选择确定该查询请求对应的查询接口的具体实现方式将在下面的实施例中进行详细描述，本申请在此先不赘述。

- 20 在一些可实现方式中，如图 5 所示，终端可以向服务器发送第一用户特征查询请求，服务器接收到第一用户特征查询请求后，可以响应于第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征。若目标数据库中存储有目标用户特征，则服务器可以将目标用户特征发送给终端；若目标数据库中未存储目标用户特征，且目标用户特征由目标数据库中的多个子用户特征构成，则服务器可以将目标用户特征分解为上述多个子用户特征，并基于上述多个子用户特征进行数据查询，最后，服务器可以根据查询到的子用户特征确定目标用户特征，并将目标用户特征发送给终端，也就是说，服务器还可以实现使用多个用户特征进行组合判断。如此一来，服务器可以不用生成目标用户特征，目标数据库只需要存储子用户特征，也即目标数据库可以只存储原子特征，在服务器收到第一查询请求时，服务器可以根据子用户特征来确定目标用户特征，从而可以减少服务器的计算成本，减少目标数据库的存储成本，也即可以减少用户画像数据的冗余和重复的开发成本。
- 25 而且在上述过程中，服务器可以基于对多个子用户特征的查询实现用户特征的组合判断，即服务器可以实现使用多个用户特征进行组合判断，提高了服务器的处理能力，提高了用户体验。

- 30 示例性的，假设终端需要查询的目标用户特征是：近三天内有登录应用 1 且对股票 1 的开户状态为已开户的用户，目标数据库没有存储上述目标用户特征，目标数据库存储有子用户特征 1 和子用户特征 2，分别是：近三天内有登录应用 1 的用户、对股票 1 的开户状态为已开户的用户，在服务器接收到终端发送的第一用户特征查询请求后，服务器可以在目标数据库中查找上述目标用户特征，确定目标数据库中未存储该目标用户特征，而服务器可以确定该目标用户特征可以由子用户特征 1 和子用户特征 2 的交集构成，那么服务器可以在目标数据库中分别查找子用户特征 1 和子用户特征 2，确定子用户特征 1 为“用户 1、用户 2、用户 3”、子用户特征 2 为“用户 1、用户 2”，从而服务器可以确定目标用户特征为“用户 1、用户 2”，然后，服务器可以将确定的目标用户特征发送给终端。其中，服务器在确定目标数据库中未存储目标用户特征，确定目标用户特征可以由目标数据库中存储的多个子用户特征构成后，可以基于 AST 树在目标数据库中查询该多个子用户特征并根据该多个子用户特征确定目标用户特征，上述 AST 树是由关于上述多个子用户特征、多个子

用户特征构成目标用户特征时的构成关系的代码转化来的。

5 在一些可实现方式中，服务器在接收到终端发送的第一用户特征查询请求后，可以响应于该第一用户特征查询请求，将第一用户特征查询请求转换为含义相同的第二用户特征查询请求，然后，服务器可以响应于第二用户特征查询请求，在目标数据库中查找是否存储有目标用户特征。如此一来，在服务器没有在目标数据库中查找到第一用户特征查询请求对应的用户特征时，可以根据和第一用户特征查询请求含义相同的第二用户特征查询请求在目标数据库查找对应的用户特征，在目标数据库中存储有第二用户特征查询请求对应的用户特征时，那么服务器可以将该第二用户特征查询请求对应的用户特征发送给终端，从而可以提高查询效率，提高用户画像的生成效率。

10 示例性的，假设目标数据库未存储用户特征 1：用户 1 是否在近三天有登录应用 1，目标数据库存储有用用户特征 2：用户 1 最近一个登录应用 1 的时间在近三天内，终端向服务器发送的第一用户特征查询请求是用来在目标数据库中查找用户特征 1 的，服务器在接收到第一用户特征查询请求后，可以响应于该第一用户特征查询请求，将第一用户特征查询请求转换为含义相同的第二用户特征查询请求，该第二用户特征查询请求用以在目标数据库
15 中查找用户特征 2，然后，服务器可以响应于第二用户特征查询请求，在目标数据库查找用户特征 2，并将查询到的用户特征 2 发送给终端。其中，服务器可以预先存储第二用户特征查询请求和与第二用户特征查询请求含义相同的第一用户特征查询请求的对应关系，在将第一用户特征查询请求转换为第二用户特征查询请求时，可以根据上述存储的对应关系来进行转换，本申请对此不做限制。当然，服务器在接收到第一用户特征查询请求后，
20 可以响应于该第一用户特征查询请求，在目标数据库中查询第一用户特征查询请求对应的用户特征 1，在没有查找到用户特征 1 后，再将第一用户特征查询请求转换为第二用户特征查询请求，然后响应于第二用户特征查询请求，在目标数据库中查找是否存储有用用户特征 2，本申请对此不做限制。

25 在一些可实现方式中，服务器在响应于上述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征之前，还可以对第一用户特征查询请求的发送方进行权限校验，在对该发送方进行权限校验通过时，可以响应于第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征，以提高数据查询的安全性，提高生成用户画像的安全性。

30 示例性的，在服务器响应于上述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征之前，服务器可以获取发送方的标识，然后，服务器可以根据发送方的标识确定发送方的权限范围，若发送方的权限范围包括查询目标用户特征的权限，则服务器可以确定对发送方进行权限校验通过；若发送方的权限范围不包括查询目标用户特征的权限，则服务器可以确定对发送方进行权限校验未通过。其中，服务器可以预先存储有发送方的标识和发送方的权限范围的对应关系，该发送方的权限范围包括发送方可以在目标
35 数据库中查找的用户特征。例如：假设第一用户特征查询请求用于在目标数据库中查询用户特征 1，服务器预先存储有业务方 1 的标识业务方 1 和业务方 1 的权限范围 1 的对应关系，假设权限范围 1 包括：用户特征 1、用户特征 2，服务器在接收到终端发送的第一用户特征查询请求后，可以先确定第一用户特征查询请求的发送方的标识为发送方 1，然后服务器可以在上述预先存储的对应关系中查找到发送方 1 的权限范围包括用户特征 1，那么
40 服务器可以确定对发送方 1 进行权限校验通过。

45 在一些可实现方式中，服务器在响应于上述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征之前，还可以确定第一用户特征查询请求对应的查询接口，然后，服务器可以基于第一用户特征查询请求对应的查询接口，响应于所述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征。例如：服务器在确定第一用户特征查询请求对应的查询接口时，可以先判断第一用户特征查询请求是否包括用户标识，

若第一用户特征查询请求包括用户标识,则可以确定第一用户特征查询请求对应的查询接口为 OLTP 接口;若第一用户特征查询请求不包括用户标识,则可以确定第一用户特征查询请求对应的查询接口为 OLAP 接口。可以理解的是,在第一用户特征查询请求包括用户标识时,一般可以确定第一用户特征查询请求是用以查询关于该用户标识对应的用户的用户特征的,那么可以确定查询结果包括的数据量较小,结合上述对 OLTP 接口和 OLAP 接口的描述,因此可以选择查询接口为 OLTP 接口,本实施例中基于 OLTP 接口可以支持秒级查询用户特征值,特征值的聚合计数,秒级元数据查询及返回。同理,在第一用户特征查询请求不包括用户标识时,一般可以确定第一用户特征查询请求是用以查询较为复杂的用户特征的,如近三天内有登录应用 1 的所有用户,那么可以选择查询接口为 OLAP 接口。如此一来,服务器可以根据用户特征查询请求,选择合适的查询接口,以提高数据查询效率和数据查询可靠性,进而提高生成用户画像的可靠性和效率。

示例性的,假设第一用户特征查询请求用以在目标数据中查询用户特征 1: 用户 1 的年龄,第一用户特征查询请求包括用户 1 的标识用户 1,那么服务器在接收到第一用户特征查询请求后,可以确定该第一用户特征查询请求包括用户标识用户 1,那么服务器可以选择该第一用户特征查询请求对应的查询接口为 OLTP 接口。通过上述方式可以实现支持通过画像锚定多类型人群数据,实现多样化投放需求,例如设定实时人群标签、查找例行或者静态用户群等等。同时可以支持更灵活的锚定人群方式,网关层能转译更多类型的 SQL 语句。例如支持动态范围的实时数据查询,业务能更灵活的在前端使用画像底层特征数据。

示例性的,服务器可以通过 OLTP 接口判断用户是否满足某种条件,然后对满足该条件的用户下发对应的广告弹窗。例如,假设第一用户特征查询请求用以在目标数据中查询用户特征 2: 用户 2 是否开户,第一用户特征查询请求包括用户 2 的标识用户 2,那么服务器在接收到第一用户特征查询请求后,可以确定该第一用户特征查询请求包括用户标识用户 2,那么服务器可以选择该第一用户特征查询请求对应的查询接口为 OLTP 接口,并且服务器可以在查询到用户特征 2 是用户 2 已开户时,向终端返回正确(true),在查询到用户特征 2 是用户 2 未开户时,向终端返回错误(false)。另外,服务器可以在确认用户特征 2 是用户 2 未开户时,向用户 2 进行开户弹窗广告推荐。

示例性的,在新股开盘的时候,本实施例中通过获取用户授权的自选股信息或者关注信息进行查询匹配,针对自选股列表中含有该自选股的客户,拉取“用户自选股列表 = xxxx”的全量用户人群,进行新股开盘的信息推送。

示例性的,服务器可以通过 OLAP 接口确定出符合某个条件的用户,对该用户进行相应的消息推送。例如,假设第一用户特征查询请求用以在目标数据中查询用户特征 3: 活跃用户,活跃用户指近三天登录过的用户,那么服务器在接收到第一用户特征查询请求后,可以确定该第一用户特征查询请求没有包括用户标识,那么服务器可以选择该第一用户特征查询请求对应的查询接口为 OLAP 接口,然后服务器可以通过 OLAP 接口确定出所有的活跃用户,并对其进行消息推送。

示例性的,本实施例中的用户特征还可以包括用户屏蔽用户列表、拉黑用户列表、自选信息、特别关注股票列表等等,当用户进入牛牛圈,系统可以根据前两个特征过滤推荐的内容,使用后两个特征用于推荐相关的帖子给用户。

示例性的,服务器可以通过 OLTP 接口查询某个用户的多个用户特征,然后结合该多个用户特征,分析该用户的兴趣偏好,给用户推荐感兴趣的资讯文章,其中用户特征可以包括关注信息、持仓信息等等。例如,假设第一用户特征查询请求 1 用以在目标数据中查询用户特征 4: 用户 4 是否关注或者持仓股票 1,第一用户特征查询请求 2 用以在目标数据中查询用户特征 5: 用户 4 是否关注股票 2,那么服务器在接收到第一用户特征查询请求 1 和第一用户特征查询请求 2 后,可以确定都包括用户标识: 用户 4,那么服务器可以选择上述两个第一用户特征查询请求对应的查询接口都为 OLTP 接口,然后,服务器可以查询

到用户特征 4 是用户 4 关注股票 1, 用户特征 5 是用户 4 没有关注股票 2, 那么服务器可以分析出用户 4 的一个兴趣偏好为: 喜欢股票 1, 不喜欢股票 2, 则服务器可以向用户 4 推荐和股票 1 有关的资讯文章、公告、相关帖子、新闻以及论坛等内容。

5 示例性的, 服务器可以通过 OLTP 接口判断用户是否满足某种条件, 然后对满足该条件的用户下发对应的奖励。例如, 假设第一用户特征查询请求用以在目标数据中查询用户特征 6: 用户 6 是否入金, 第一用户特征查询请求包括用户 6 的标识用户 6, 那么服务器在接收到第一用户特征查询请求后, 可以确定该第一用户特征查询请求包括用户标识用户 6, 那么服务器可以选择该第一用户特征查询请求对应的查询接口为 OLTP 接口, 并且服务器可以在查询到用户特征 6 是用户 6 已入金时, 向用户 6 下发入金奖励。

10 示例性的, 不同用户对应的股票市场权限不一样, 比如高资产用户有更高的股票市场行情浏览权限, 未开户的用户只有指定的行情浏览权限, 所以服务器可以通过确定用户特征: 用户的资产是否达到目标条件, 来控制用户的股票市场行情浏览权限。例如, 假设第一用户特征查询请求用以在目标数据中查询用户特征 7: 用户 7 的资产是否为 1 万, 那么服务器在接收到第一用户特征查询请求后, 可以确定该第一用户特征查询请求包括用户标识用户 7, 那么服务器可以选择该第一用户特征查询请求对应的查询接口为 OLTP 接口, 并且服务器可以在查询到用户特征 7 是用户 7 已的资产已达到 1 万, 并向用户 7 开通关于股票 7 的行情情况的权限。

15 在一些可实现方式中, 服务器可以判断目标数据库中是否存在异常的用户特征, 若目标数据库中不存在异常的用户特征, 则服务器可以生成提示信息, 并推送该提示信息, 以提示用户目标数据库中不存在异常的用户特征; 若目标数据库中未存在异常的用户特征, 则服务器可以不生成提示信息, 从而可以保证目标数据库存储用户特征的准确性, 提高数据查询结果的准确性, 进而提高生成用户画像的准确性和效率。

20 示例性的, 服务器可以建立一个画像监控模块, 来对不同的用户特征或者用户画像建立特征模型, 以判断目标数据库中是否存在异常的用户特征, 即进行监控告警。例如, 服务器在判断目标数据库中是否存在异常的用户特征时, 可以获取目标数据中在任一时刻存储的第一用户特征以及在该任一时刻之前的预设时长内的至少一个第二用户特征, 然后, 服务器可以对上述至少一个第二用户特征进行统计, 得到第一用户特征的分布范围, 若第一用户特征不在上述分布范围内, 则服务器可以确定目标数据库中不存在异常的用户特征; 若第一用户特征在所述分布范围内, 则服务器可以确定目标数据库中不存在异常的用户特征。例如: 服务器可以获取目标数据库中在 2022 年 6 月 30 日 24:00 存储的第一用户特征: 2022 年 6 月 30 日登录应用 1 的用户数量, 该第一用户特征为 a, 获取 2022 年 6 月 30 日 24:00 之前的两天内的两个第二用户特征: 第二用户特征 1 和第二用户特征 2, 分别为: 2022 年 6 月 29 日登录应用 1 的用户数量、2022 年 6 月 28 日登录应用 1 的用户数量, 该第二用户特征 1 和第二用户特征 2 分别为 b、c, 然后, 服务器可以计算第二用户特征 1 和第二用户特征 2 的平均值为 $(b+c)/2=d$, 接着, 服务器可以将 $(d-e, d+e)$ 确定为第一用户特征的分布范围, 若第一用户特征 a 在分布范围 $(d-e, d+e)$ 中, 则可以确定第一用户特征为正常数据, 即可以确定目标数据库中不存在异常的用户特征; 若第一用户特征 a 不在分布范围 $(d-e, d+e)$ 中, 则可以确定第一用户特征为异常数据, 即可以确定目标数据库中不存在异常的用户特征, 其中, a、b、c、d、e 为正整数。

40 可以理解的是, 由于用户画像生产流程的链路较长, 而业务数据迁徙或者数据源变更很容易导致获取的用户数据不准确, 从而导致用户画像不准确, 所以需要用户对用户特征或者用户画像进行监控告警。例如, 上述用户数据可以是服务器从其他数据库如记录用户股票开户情况、用户登录情况等用户数据的业务源数据库中获取的, 那么在上述其他数据库中存储的用户数据发生变更时, 如用户数据的存储位置发生了变更, 服务器获取到到用户数据就会不准确, 进而导致生成的用户特征不准确, 从而导致数据查询结果不准确, 因此通

过上述判断目标数据库中是否存在异常的用户特征，可以保证目标数据库存储用户特征的准确性，优化了特征数据的计算变更速度，更快速根据业务数据进行实时流计算，目前 90% 以上的特征支持秒级实时更新，以提高数据查询结果的准确性，进而提高生成用户画像的准确性和效率。

5 在一些可实现方式中，如图 1 所示，假设终端 110 上安装应用 1，用户通过访问应用 1，基于自然语言对目标用户特征进行查询，在用户在应用 1 上输入用以查询目标用户特征的自然语言后，服务器可以将该自然语言转换为 SQL，并基于该 SQL 在目标数据库中查询上述目标用户特征。其中，服务器可以预先存储自然语言和 SQL 的对应关系，然后基于该对应关系，将自然语言转换为 SQL。例如，如表 1 所示，假设服务器预先存储有自然语言和 SQL 的部分对应关系如表 1 所示，其中，如表 1 中的第二行所示，自然语言“年龄”和 SQL“SQL1”对应。

表 1

自然语言	SQL	自然语言	SQL
年龄	SQL1	且	SQL4
=	SQL2	社交天数	SQL5
20	SQL3	30	SQL6

示例性的，如图 6 所示，假设用户在应用 1 上输入的自然语言 1 为“年龄=20”，那么终端可以将自然语言 1 发送给服务器，服务器接收到自然语言 1 后，可以基于上述表 1 将自然语言 1 转换为对应的 SQL“SQL1, SQL2, SQL3”。

15 示例性的，如图 7 所示，自然语言 1“年龄=20”后有一个加号按钮，当用户点击了该加号按钮后，可以继续输入自然语言 2“社交天数=20”，且输入的自然语言 2 和自然语言 1 的关系如图 4 所示的“且”所示，表示用户希望查询“年龄=20 且社交天数=20 的用户”，然后，服务器可以将用户输入在自然语言转换为对应的 SQL“SQL1, SQL2, SQL3, SQL4, SQL5, SQL2, SQL3”。

应理解的是，本申请对自然语言和 SQL 的对应关系仅是示意性的，将该自然语言转换为对应的 SQL 也是示意性的。

25 在一些可实现方式中，如图 8 所示，服务器可以包括：权限校验模块、画像服务模块、目标数据库，终端可以向服务器发送第一用户特征查询请求，以用于查询目标用户特征，服务器接收到第一用户特征查询请求后，可以先基于上述权限校验模块对第一用户特征查询请求的发送方进行权限校验，在对该发送方进行权限校验通过时，可以响应于第一用户特征查询请求，基于上述画像服务模块确定第一用户特征查询请求对应的查询接口，然后基于第一用户特征查询请求对应的查询接口，响应于所述第一用户特征查询请求，在目标数据库中查找目标用户特征，若目标数据库中存储有目标用户特征，则服务器可以将目标用户特征发送给终端。

30 在一些可实现方式中，如图 9 所示，服务器可以包括：参数校验模块、SQL 解析模块、路由模块、目标数据库，终端可以向服务器发送第一用户特征查询请求，以用于查询目标用户特征，第一用户特征查询请求可以是自然语言，服务器接收到第一用户特征查询请求后，可以先基于上述参数校验模块对第一用户特征查询请求的发送方进行权限校验，在对该发送方进行权限校验通过时，可以响应于第一用户特征查询请求，基于上述 SQL 解析模块将自然语言形式的第一用户特征查询请求解析为 SQL 形式的第一用户特征查询请求，接着，服务器可以基于上述路由模块确定第一用户特征查询请求对应的查询接口，然后，服务器可以基于第一用户特征查询请求对应的查询接口，响应于所述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征，若目标数据库中存储有目标用户特征，则服务器可以将目标用户特征发送给终端。

综上所述，上述实施例提供的技术方案至少带来以下有益效果：通过本申请技术方案，

服务器可以先获取多个用户数据以及多个用户数据中最后一个用户数据的生成时间，若从多个用户数据中每读取一个第一用户数据，则服务器可以生成第一用户数据对应的用户特征，若第一用户数据的生成时间和最后一个用户数据的生成时间一致，则服务器可以将第一用户数据对应的用户特征存储至目标数据库；若第一用户数据生成时间和最后一个用户数据的生成时间不一致，则服务器可以不将第一用户数据对应的用户特征存储至目标数据库，最后，服务器可以基于目标数据库中的用户特征生成用户画像。在上述过程中，服务器可以通过判断当前读取的用户数据的生成时间是否和最后一个用户数据的生成时间一致，进而可以确定当前读取的用户数据是否是最后一个用户数据，从而可以只将最后一个用户数据对应的用户特征存储在目标数据库中，而不将其他用户数据对应的用户特征存储在目标数据库中，那么在对用户特征进行查询时，例如在服务器读取用户数据时对用户特征进行查询，查询到的结果不会其他用户数据对应的用户特征，只会是最后一个原始对应的用户特征即是根据所有用户数据计算出来的最终用户特征，从而服务器可以根据上述最终用户特征生成正确的用户画像，从而可以解决现有技术中因数据存储导致的用户特征查询不准确，从而导致根据查询到的用户特征生成的用户画像不准确的问题，可以提高用户特征查询的准确性，进而提高用户画像应用的效率和精确性。

进一步的，服务器接收到第一用户特征查询请求后，可以响应于第一用户特征查询请求，可以在目标数据库中查找是否存储有目标用户特征。若目标数据库中未存储目标用户特征，且目标用户特征由目标数据库中的多个子用户特征构成，则服务器可以将目标用户特征分解为上述多个子用户特征，并基于上述多个子用户特征进行数据查询。如此一来，服务器可以不用生成目标用户特征，目标数据库只需要存储子用户特征，在服务器收到第一查询请求时，服务器可以根据子用户特征来确定目标用户特征，从而可以减少服务器的计算成本，减少目标数据库的存储成本。

更进一步的，服务器在接收到终端发送的第一用户特征查询请求后，可以响应于第一用户特征查询请求，将第一用户特征查询请求转换为含义相同的第二用户特征查询请求，然后，服务器可以响应于第二用户特征查询请求，在目标数据库中查找是否存储有目标用户特征。如此一来，在服务器没有在目标数据库中查找到第一用户特征查询请求对应的用户特征时，可以根据和第一用户特征查询请求含义相同的第二用户特征查询请求在目标数据库查找对应的用户特征，在目标数据库中存储有第二用户特征查询请求对应的用户特征时，服务器可以将该第二用户特征查询请求对应的用户特征发送给终端，从而可以提高查询效率，提高生成用户画像的效率。

再进一步的，服务器在响应于第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征之前，还可以对第一用户特征查询请求的发送方进行权限校验，在该发送方进行权限校验通过时，可以响应于第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征，以提高数据查询的安全性，提高生成用户画像的安全性。

又进一步的，服务器在响应于上述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征之前，还可以确定第一用户特征查询请求对应的查询接口，然后，服务器可以基于第一用户特征查询请求对应的查询接口，响应于所述第一用户特征查询请求，在目标数据库中查找是否存储有目标用户特征。如此一来，服务器可以根据用户特征查询请求，选择合适的查询接口，以提高数据查询效率和数据查询可靠性，提高生成用户画像的效率和可靠性。

又进一步的，服务器可以判断目标数据库中是否存在异常的用户特征，若目标数据库中不存在异常的用户特征，则服务器可以生成提示信息，并推送该提示信息，以提示用户目标数据库中不存在异常的用户特征；若目标数据库中未存在异常的用户特征，则服务器可以不生成提示信息，从而可以保证目标数据库存储用户特征的准确性，以提高数据查询结果的准确性，提高生成用户画像的准确性。

图 10 为本申请实施例提供的一种基于用户画像的数据处理装置 1000 的示意图, 如图 10 所示, 该装置 1000 包括:

第一获取模块 1001, 用于获取多个用户数据以及多个用户数据中最后一个用户数据的生成时间;

5 第一生成模块 1002, 用于若从多个用户数据中每读取一个第一用户数据, 则生成第一用户数据对应的用户特征;

处理模块 1003, 用于: 若第一用户数据的生成时间和最后一个用户数据的生成时间一致, 则将第一用户数据对应的用户特征存储至目标数据库; 若第一用户数据生成时间和最后一个用户数据的生成时间不一致, 则不将第一用户数据对应的用户特征存储至目标数据库;

10 第二生成模块 1004, 用于基于所述目标数据库中的用户特征生成用户画像。

在一些可实现方式中, 装置 1000 还包括: 第二获取模块 1005、查找模块 1006、分解模块 1007、查询模块 1008, 其中, 第二获取模块 1005 用于获取第一用户特征查询请求; 查找模块 1006 用于响应于第一用户特征查询请求, 在目标数据库中查找是否存储有目标用户特征; 分解模块 1007 用于若目标数据库中未存储目标用户特征, 且目标用户特征由目标数据库中的多个子用户特征构成, 则将目标用户特征分解为多个子用户特征; 查询模块 1008 用于基于多个子用户特征进行数据查询。

在一些可实现方式中, 查找模块 1006 具体用于响应于第一用户特征查询请求, 将第一用户特征查询请求转换为含义相同的第二用户特征查询请求; 响应于第二用户特征查询请求, 在目标数据库中查找是否存储有所目标用户特征。

20 在一些可实现方式中, 装置 1000 还包括: 校验模块 1009, 其中, 校验模块 1009 用于对第一用户特征查询请求的发送方进行权限校验; 查找模块 1006 具体用于在对发送方进行权限校验通过时, 响应于第一用户特征查询请求, 在目标数据库中查找是否存储有目标用户特征。

25 在一些可实现方式中, 校验模块 1009 具体用于获取发送方的标识; 根据发送方的标识确定发送方的权限范围; 若发送方的权限范围包括查询目标用户特征的权限, 则确定对发送方进行权限校验通过; 若发送方的权限范围不包括查询目标用户特征的权限, 则确定对发送方进行权限校验未通过。

30 在一些可实现方式中, 装置 1000 还包括: 确定模块 1010, 其中, 确定模块 1010 用于确定第一用户特征查询请求对应的查询接口; 查找模块 1006 具体用于基于第一用户特征查询请求对应的查询接口, 响应于第一用户特征查询请求, 在目标数据库中查找是否存储有目标用户特征。

35 在一些可实现方式中, 确定模块 1010 具体用于若第一用户特征查询请求包括用户标识时, 则确定第一用户特征查询请求对应的查询接口为 OLTP 接口; 若第一用户特征查询请求不包括用户标识时, 则确定第一用户特征查询请求对应的查询接口为 OLAP 接口。

在一些可实现方式中, 装置 1000 还包括: 判断模块 1011、第三生成模块 1012、推送模块 1013, 其中, 判断模块 1011 用于判断目标数据库中是否存在异常的用户特征; 第三生成模块 1012 用于若目标数据库中不存在异常的用户特征, 则生成提示信息; 推送模块 1013 用于推送提示信息, 以提示用户目标数据库中不存在异常的用户特征。

40 在一些可实现方式中, 判断模块 1011 具体用于获取目标数据库中在任一时刻存储的第一用户特征以及在任一时刻之前的预设时长内的至少一个第二用户特征; 对至少一个第二用户特征进行统计, 得到第一用户特征的分布范围; 若第一用户特征不在分布范围内, 则确定目标数据库中不存在异常的用户特征; 若第一用户特征在分布范围内, 则确定目标数据库中不存在异常的用户特征。

45 在一些可实现方式中, 第一生成模块 1003 具体用于若从多个用户数据中每读取一个第

一用户数据,则按照计算口径和应用场景,选择第一用户数据对应的用户特征的生成方法,生成方法是流生成方法或批量生成方法;基于第一用户数据对应的用户特征的生成方法生成第一用户数据对应的用户特征。

应理解的是,装置实施例与方法实施例可以相互对应,类似的描述可以参照方法实施例。为避免重复,此处不再赘述。具体地,图 10 所示的装置 1000 可以执行上述方法实施例,并且装置 1000 中的各个模块的前述和其它操作和/或功能分别为了实现上述各个方法中的相应流程,为了简洁,在此不再赘述。

上文中结合附图从功能模块的角度描述了本申请实施例的装置 1000。应理解,该功能模块可以通过硬件形式实现,也可以通过软件形式的指令实现,还可以通过硬件和软件模块组合实现。具体地,本申请实施例中的方法实施例的各步骤可以通过处理器中的硬件的集成逻辑电路和/或软件形式的指令完成,结合本申请实施例公开的方法的步骤可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。可选地,软件模块可以位于随机存储器,闪存、只读存储器、可编程只读存储器、电可擦写可编程存储器、寄存器等本领域的成熟的存储介质中。该存储介质位于存储器,处理器读取存储器中的信息,结合其硬件完成上述方法实施例中的步骤。

图 11 是本申请实施例提供的电子设备的示意性框图。

如图 11 所示,该电子设备可包括:

存储器 1110 和处理器 1120,该存储器 1110 用于存储计算机程序,并将该程序代码传输给该处理器 1120。换言之,该处理器 1120 可以从存储器 1110 中调用并运行计算机程序,以实现本申请实施例中的方法。

例如,该处理器 1120 可用于根据该计算机程序中的指令执行上述方法实施例。

在本申请的一些实施例中,该处理器 1120 可以包括但不限于:

通用处理器、数字信号处理器 (Digital Signal Processor, DSP)、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现场可编程门阵列 (Field Programmable Gate Array, FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等等。

在本申请的一些实施例中,该存储器 1110 包括但不限于:

易失性存储器和/或非易失性存储器。其中,非易失性存储器可以是只读存储器 (Read-Only Memory, ROM)、可编程只读存储器 (Programmable ROM, PROM)、可擦除可编程只读存储器 (Erasable PROM, EPROM)、电可擦除可编程只读存储器 (Electrically EPROM, EEPROM) 或闪存。易失性存储器可以是随机存取存储器 (Random Access Memory, RAM),其用作外部高速缓存。通过示例性但不是限制性说明,许多形式的 RAM 可用,例如静态随机存取存储器 (Static RAM, SRAM)、动态随机存取存储器 (Dynamic RAM, DRAM)、同步动态随机存取存储器 (Synchronous DRAM, SDRAM)、双倍数据速率同步动态随机存取存储器 (Double Data Rate SDRAM, DDR SDRAM)、增强型同步动态随机存取存储器 (Enhanced SDRAM, ESDRAM)、同步连接动态随机存取存储器 (synch link DRAM, SLDRAM) 和直接内存总线随机存取存储器 (Direct Rambus RAM, DR RAM)。

在本申请的一些实施例中,该计算机程序可以被分割成一个或多个模块,该一个或者多个模块被存储在该存储器 1110 中,并由该处理器 1120 执行,以完成本申请提供的方法。该一个或多个模块可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述该计算机程序在该电子设备中的执行过程。

如图 11 所示,该电子设备还可包括:

收发器 1130,该收发器 1130 可连接至该处理器 1120 或存储器 1110。

其中,处理器 1120 可以控制该收发器 1130 与其他设备进行通信,具体地,可以向其他设备发送信息或数据,或接收其他设备发送的信息或数据。收发器 1130 可以包括发射机

和接收机。收发器 1130 还可以进一步包括天线，天线的数量可以为一个或多个。

应当理解，该电子设备中的各个组件通过总线系统相连，其中，总线系统除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。

本申请还提供了一种计算机存储介质，其上存储有计算机程序，该计算机程序被计算机执行时使得该计算机能够执行上述方法实施例的方法。或者说，本申请实施例还提供一种包含指令的计算机程序产品，该指令被计算机执行时使得计算机执行上述方法实施例的方法。

当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。该计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行该计算机程序指令时，全部或部分地产生按照本申请实施例该的流程或功能。该计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。该计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，该计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如同轴电缆、光纤、数字用户线（digital subscriber line, DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。该计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。该可用介质可以是磁性介质（例如，软盘、硬盘、磁带）、光介质（例如数字视频光盘（digital video disc, DVD））、或者半导体介质（例如固态硬盘（solid state disk, SSD））等。

本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的模块及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，该模块的划分，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，例如多个模块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接口，装置或模块的间接耦合或通信连接，可以是电性，机械或其它的形式。

作为分离部件说明的模块可以是或者也可以不是物理上分开的，作为模块显示的部件可以是或者也可以不是物理模块，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。例如，在本申请各个实施例中的各功能模块可以集成在一个处理模块中，也可以是各个模块单独物理存在，也可以两个或两个以上模块集成在一个模块中。

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该权利要求的保护范围为准。

权利要求书

- 1、一种基于用户画像的数据处理方法，其特征在于，包括：
获取多个用户数据以及所述多个用户数据中最后一个用户数据的生成时间；
若从所述多个用户数据中每读取一个第一用户数据，则生成所述第一用户数据对应的
5 用户特征；
若所述第一用户数据的生成时间和所述最后一个用户数据的生成时间一致，则将所述
第一用户数据对应的用户特征存储至目标数据库；
若所述第一用户数据生成时间和所述最后一个用户数据的生成时间不一致，则不将所
述第一用户数据对应的用户特征存储至所述目标数据库；
10 基于所述目标数据库中的用户特征生成用户画像。
- 2、根据权利要求1所述的方法，其特征在于，还包括：
获取第一用户特征查询请求；
响应于所述第一用户特征查询请求，在所述目标数据库中查找是否存储有目标用户特
征；
15 若所述目标数据库中未存储所述目标用户特征，且所述目标用户特征由所述目标数据
库中的多个子用户特征构成，则将所述目标用户特征分解为所述多个子用户特征；
基于所述多个子用户特征进行用户特征查询。
- 3、根据权利要求2所述的方法，其特征在于，所述响应于所述第一用户特征查询请求，
在所述目标数据库中查找是否存储有目标用户特征，包括：
20 响应于所述第一用户特征查询请求，将所述第一用户特征查询请求转换为含义相同的
第二用户特征查询请求；
响应于所述第二用户特征查询请求，在所述目标数据库中查找是否存储有所述目标用
户特征。
- 4、根据权利要求2或3所述的方法，其特征在于，所述响应于所述第一用户特征查询
25 请求，在所述目标数据库中查找是否存储有目标用户特征之前，还包括：
对所述第一用户特征查询请求的发送方进行权限校验；
所述响应于所述第一用户特征查询请求，在所述目标数据库中查找是否存储有目标用
户特征，包括：
在对所述发送方进行权限校验通过时，响应于所述第一用户特征查询请求，在所述目
30 标数据库中查找是否存储有所述目标用户特征。
- 5、根据权利要求4所述的方法，其特征在于，所述对所述第一用户特征查询请求的发
送方进行权限校验，包括：
获取所述发送方的标识；
根据所述发送方的标识确定所述发送方的权限范围；
35 若所述发送方的权限范围包括查询所述目标用户特征的权限，则确定对所述发送方进
行权限校验通过；
若所述发送方的权限范围不包括查询所述目标用户特征的权限，则确定对所述发送方
进行权限校验未通过。
- 6、根据权利要求2或3所述的方法，其特征在于，所述响应于所述第一用户特征查询
40 请求，在所述目标数据库中查找是否存储有目标用户特征之前，还包括：
确定所述第一用户特征查询请求对应的查询接口；
所述响应于所述第一用户特征查询请求，在所述目标数据库中查找是否存储有目标用
户特征，包括：
基于所述第一用户特征查询请求对应的查询接口，响应于所述第一用户特征查询请求，
45 在所述目标数据库中查找是否存储有目标用户特征。

- 7、根据权利要求6所述的方法，其特征在于，所述确定所述第一用户特征查询请求对应的查询接口，包括：
- 若所述第一用户特征查询请求包括用户标识时，则确定所述第一用户特征查询请求对应的查询接口为联机事务处理 OLTP 接口；
- 5 若所述第一用户特征查询请求不包括用户标识时，则确定所述第一用户特征查询请求对应的查询接口为联机分析处理 OLAP 接口。
- 8、根据权利要求1-3任一项所述的方法，其特征在于，还包括：
- 判断所述目标数据库中是否存在异常的用户特征；
- 若所述目标数据库中不存在异常的用户特征，则生成提示信息；
- 10 推送所述提示信息，以提示用户所述目标数据库中不存在异常的用户特征。
- 9、根据权利要求8所述的方法，其特征在于，所述判断所述目标数据库中是否存在异常的用户特征，包括：
- 获取所述目标数据库中在任一时刻存储的第一用户特征以及在所述任一时刻之前的预设时长内的至少一个第二用户特征；
- 15 对所述至少一个第二用户特征进行统计，得到所述第一用户特征的分布范围；
- 若所述第一用户特征不在所述分布范围内，则确定所述目标数据库中不存在异常的用户特征；
- 若所述第一用户特征在所述分布范围内，则确定所述目标数据库中不存在异常的用户特征。
- 20 10、根据权利要求1-3任一项所述的方法，其特征在于，所述若从所述多个用户数据中每读取一个第一用户数据，则生成所述第一用户数据对应的用户特征，包括：
- 若从所述多个用户数据中每读取一个第一用户数据，则按照计算口径和应用场景，选择所述第一用户数据对应的用户特征的生成方法，所述生成方法是流生成方法或批量生成方法；
- 25 基于所述第一用户数据对应的用户特征的生成方法生成所述第一用户数据对应的用户特征。
- 11、一种基于用户画像的数据处理装置，其特征在于，包括：
- 第一获取模块，用于获取多个用户数据以及所述多个用户数据中最后一个用户数据的生成时间；
- 30 第一生成模块，用于若从所述多个用户数据中每读取一个第一用户数据，则生成所述第一用户数据对应的用户特征；
- 处理模块，用于：
- 若所述第一用户数据的生成时间和所述最后一个用户数据的生成时间一致，则将所述第一用户数据对应的用户特征存储至目标数据库；
- 35 若所述第一用户数据生成时间和所述最后一个用户数据的生成时间不一致，则不将所述第一用户数据对应的用户特征存储至所述目标数据库；
- 第二生成模块，用于基于所述目标数据库中的用户特征生成用户画像。
- 12、一种电子设备，其特征在于，包括：
- 处理器和存储器，所述存储器用于存储计算机程序，所述处理器用于调用并运行所述
- 40 存储器中存储的计算机程序，以执行权利要求1-10中任一项所述的方法。
- 13、一种计算机可读存储介质，其特征在于，用于存储计算机程序，所述计算机程序使得计算机执行如权利要求1-10中任一项所述的方法。
- 14、一种包含指令的计算机程序产品，其特征在于，当所述计算机程序产品在电子设备上运行时，使得所述电子设备执行权利要求1-10中任一项所述的方法。