(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112541101 A
(43)申请公布日 2021.03.23

(21)申请号 202011464802.X

(22)申请日 2020.12.11

(71)申请人 武汉旷视金智科技有限公司
地址 430073 湖北省武汉市东湖新技术开发区高新大道999号未来科技城F1栋11层
申请人 北京迈格威科技有限公司

(72)发明人 袁沅祥

(74)专利代理机构 北京超凡宏宇专利代理事务所(特殊普通合伙) 11463
代理人 何少岩

(51)Int.Cl.
G06F 16/903(2019.01)
G06F 16/9032(2019.01)
G06F 16/957(2019.01)

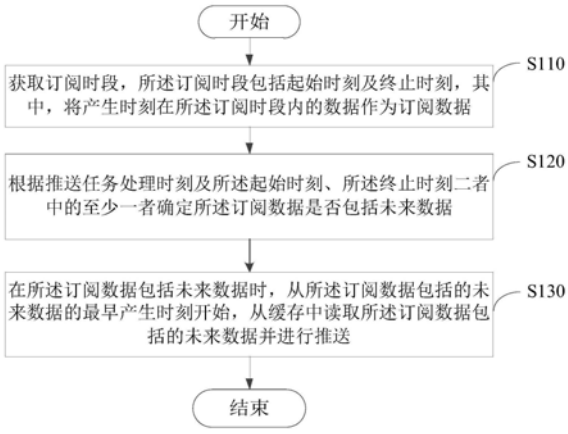
权利要求书3页 说明书16页 附图2页

(54)发明名称

订阅数据的推送方法、装置、电子设备及计算机存储介质

(57)摘要

本发明涉及一种订阅数据的推送方法、装置、电子设备及计算机存储介质，属于数据推送领域。该方法包括：获取订阅时段，订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据；推送任务处理时刻及所述起始时刻，或者根据推送任务处理时刻、起始时刻及所述终止时刻确定订阅数据是否包括未来数据；在订阅数据包括未来数据时，从订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取订阅数据包括的未来数据并进行推送；其中，未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据，且未来数据产生后实时写入缓存。通过该方法，有利于减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。



CN 112541101 A

1.一种订阅数据的推送方法,其特征在于,所述方法包括:

获取订阅时段,所述订阅时段包括起始时刻及终止时刻,其中,将产生时刻在所述订阅时段内的数据作为订阅数据;

根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据;

在所述订阅数据包括未来数据时,从所述订阅数据包括的未来数据的最早产生时刻开始,从缓存中读取所述订阅数据包括的未来数据并进行推送;

其中,未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据,且未来数据产生后实时写入缓存。

2.根据权利要求1所述的方法,其特征在于,所述确定所述订阅数据是否包括未来数据,包括:

在所述推送任务处理时刻早于所述起始时刻时,确定所述订阅数据全部为未来数据;

在所述推送任务处理时刻位于所述起始时刻及所述终止时刻之间时,确定所述订阅数据部分为未来数据,部分为历史数据;

在所述推送任务处理时刻晚于所述终止时刻时,确定所述订阅数据全部为历史数据;

其中,历史数据为数据的产生时刻早于所述推送任务处理时刻的数据。

3.根据权利要求1或2所述的方法,其特征在于,所述方法还包括:

在所述订阅数据内包括历史数据时,从历史数据存储单元中读取所述订阅数据包括的历史数据并进行推送;

其中,历史数据为数据的产生时刻早于所述推送任务处理时刻的数据;所述历史数据存储单元为不同于缓存的存储单元。

4.根据权利要求2或3所述的方法,其特征在于,所述方法还包括:

根据所述推送任务处理时刻将所述订阅时段分割为至少一个子时段;

将各个子时段加入到时段列表;

相应的,所述读取所述订阅数据包括的未来数据并进行推送,包括:

启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送;

相应的,所述读取所述订阅数据包括的历史数据并进行推送,包括:

启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送。

5.根据权利要求4所述的方法,其特征在于,每个子时段都有子起始时刻和子终止时刻,所述根据所述推送任务处理时刻将所述订阅时段分割为至少一个子时段,包括:

若所述推送任务处理时刻未落入所述订阅时段,则将所述订阅时段作为子时段,该子时段的子起始时刻为所述起始时刻,该子时段的子终止时刻为所述终止时刻;

若所述推送任务处理时刻落入所述订阅时段,则将所述订阅时段分割为第一子时段和第二子时段,所述第一子时段的子起始时刻为所述起始时刻,所述第一子时段的子终止时刻为所述推送任务处理时刻,所述第二子时段的子起始时刻为所述推送任务处理时刻,所述第二子时段的子终止时刻为所述终止时刻。

6.根据权利要求5所述的方法,其特征在于,所述启动第一进程和/或第二进程对所述

时段列表中的各个子时段内所产生的数据进行读取和推送，包括：

通过所述第二进程，读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送，直至推送完毕。

7.根据权利要求6所述的方法，其特征在于，

若所述最后一个子时段内所产生的数据的为未来数据，则读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送的步骤从所述最后一个子时段的子起始时刻开始，从所述缓存中读取数据；和/或，

若所述最后一个子时段内所产生的数据的为历史数据，则读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送的步骤从所述推送任务处理时刻开始，从历史数据存储单元中读取数据。

8.根据权利要求6或7所述的方法，其特征在于，所述启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送，还包括：

通过所述第一进程，读取在所述时段列表中的其他子时段内所产生的数据并进行推送，直至推送完毕；

所述其他子时段为所述时段列表中除所述最后一个子时段以外的子时段。

9.根据权利要求8所述的方法，其特征在于，所述读取在所述时段列表中的其他子时段内所产生的数据并进行推送，包括：

从所述推送任务处理时刻开始，从历史数据存储单元中读取在所述其他子时段内所产生的数据并进行推送。

10.根据权利要求5-9任一项所述的方法，其特征在于，所述启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送，包括：

根据待推送子时段确定当前时间切片；

对所述当前时间切片内产生的数据进行读取和推送，并将所述时段列表中待推送子时段的子起始时刻更新为所述当前时间切片的切片终止时刻，直至待推送子时段内的数据被推送完毕。

11.根据权利要求10所述的方法，其特征在于，所述根据待推送子时段确定当前时间切片，包括：

将所述待推送子时段的子起始时刻作为当前时间切片的切片起始时刻，将当前时间切片的切片起始时刻与预设时长粒度之和、所述待推送子时段的子终止时刻中的较小值作为当前时间切片的切片终止时刻。

12.根据权利要求4-11任一项所述的方法，其特征在于，所述方法还包括：

在确定存在异常重启时，获取所述时段列表在重启时刻所包括的子时段；

根据所述重启时刻，对所述时段列表在所述重启时刻所包括的子时段逐一进行分割；

用分割后的子时段和/或孙时段更新所述时段列表；

启动所述第一进程和/或所述第二进程对所述时段列表中的各个子时段和/或孙时段内所产生的数据进行读取和推送。

13.根据权利要求12所述的方法，其特征在于，每个孙时段都有孙起始时刻和孙终止时刻，所述对所述时段列表在所述重启时刻所包括的子时段逐一进行分割，包括：

若所述重启时刻未落入待分割子时段，则所述待分割子时段保持不变；

若所述重启时刻落入待分割子时段，则将所述待分割子时段分割为第一孙时段和第二孙时段，所述第一孙时段的孙起始时刻为所述待分割子时段的子起始时刻，所述第一孙时段的孙终止时刻为所述重启时刻，所述第二孙时段的孙起始时刻为所述重启时刻，所述第二孙时段的孙终止时刻为所述待分割子时段的子终止时刻。

14.一种订阅数据的推送装置，其特征在于，所述装置包括：

获取模块，用于获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据；

判断模块，用于根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据；

推送模块，用于在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送；

其中，未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据，且未来数据产生后实时写入缓存。

15.一种电子设备，其特征在于，包括：存储器和处理器，所述存储器和所述处理器连接；

所述存储器用于存储程序；

所述处理器调用存储于所述存储器中的程序，以执行如权利要求1-13中任一项所述的方法。

16.一种计算机存储介质，其特征在于，其上存储有计算机程序，所述计算机程序被计算机运行时执行如权利要求1-13中任一项所述的方法。

订阅数据的推送方法、装置、电子设备及计算机存储介质

技术领域

[0001] 本申请属于数据推送领域，具体涉及一种订阅数据的推送方法、装置、电子设备及计算机存储介质。

背景技术

[0002] 上级数据库可以向下级数据库订阅内容(例如可以包括视频、图像、信息等)，此时，下级数据库的subservice(简称sub，订阅推送服务)负责推送订阅数据到上级数据库。

[0003] 在现有技术中，为了保证上级数据库获取到订阅内容的速度，sub启动后，获取自身所负责的并处于订阅中的订阅任务，然后以固定的时间间隔不断轮询自身所处的电子设备的历史数据存储单元，以从历史数据存储单元中获取与订阅任务对应的订阅数据并推送给上级数据库。在此过程中，即使与订阅任务对应的订阅数据还未被保存在历史数据存储单元中，sub也会等待该数据保存至历史数据存储单元后，再以固定的时间间隔不断轮询历史数据存储单元的方式从历史数据存储单元中读取该数据并进行推送。

[0004] 因此，现有技术对历史数据存储单元查询的查询较为频繁，使得历史数据存储单元的查询压力较大。

发明内容

[0005] 有鉴于此，本申请的目的在于提供一种订阅数据的推送方法、装置、电子设备及计算机存储介质，有利于减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。

[0006] 本申请的实施例是这样实现的：

[0007] 第一方面，本申请实施例提供一种订阅数据的推送方法，所述方法包括：获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据；根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据；在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送；其中，未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据，且未来数据产生后实时写入缓存。相较于现有技术中的等待未来数据保存到历史数据存储单元后再从历史数据存储单元读取未来数据的方案，采用本方案至少可以减少sub在读取未来数据时对历史数据存储单元的查询，从而可以减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。

[0008] 结合第一方面实施例，在一种可能的实施方式中，所述确定所述订阅数据是否包括未来数据，包括：在所述推送任务处理时刻早于所述起始时刻时，确定所述订阅数据全部为未来数据；在所述推送任务处理时刻位于所述起始时刻及所述终止时刻之间时，确定所述订阅数据部分为未来数据，部分为历史数据；在所述推送任务处理时刻晚于所述终止时刻时，确定所述订阅数据全部为历史数据；其中，历史数据为数据的产生时刻早于所述推送

任务处理时刻的数据。

[0009] 结合第一方面实施例，在一种可能的实施方式中，所述方法还包括：在所述订阅数据内包括历史数据时，从历史数据存储单元中读取所述订阅数据包括的历史数据并进行推送；其中，历史数据为数据的产生时刻早于所述推送任务处理时刻的数据；所述历史数据存储单元为不同于缓存的存储单元。

[0010] 结合第一方面实施例，在一种可能的实施方式中，所述方法还包括：根据所述推送任务处理时刻将所述订阅时段分割为至少一个子时段；将各个子时段加入到时段列表；相应的，所述读取所述订阅数据包括的未来数据并进行推送，包括：启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送；相应的，所述读取所述订阅数据包括的历史数据并进行推送，包括：启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送。

[0011] 结合第一方面实施例，在一种可能的实施方式中，每个子时段都有子起始时刻和子终止时刻，所述根据所述推送任务处理时刻将所述订阅时段分割为至少一个子时段，包括：若所述推送任务处理时刻未落入所述订阅时段，则将所述订阅时段作为子时段，该子时段的子起始时刻为所述起始时刻，该子时段的子终止时刻为所述终止时刻；若所述推送任务处理时刻落入所述订阅时段，则将所述订阅时段分割为第一子时段和第二子时段，所述第一子时段的子起始时刻为所述起始时刻，所述第一子时段的子终止时刻为所述推送任务处理时刻，所述第二子时段的子起始时刻为所述推送任务处理时刻，所述第二子时段的子终止时刻为所述终止时刻。

[0012] 结合第一方面实施例，在一种可能的实施方式中，所述启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送，包括：通过所述第二进程，读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送，直至推送完毕。

[0013] 结合第一方面实施例，在一种可能的实施方式中，若所述最后一个子时段内所产生的数据的为未来数据，则读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送的步骤从所述最后一个子时段的子起始时刻开始，从所述缓存中读取数据；和/或，若所述最后一个子时段内所产生的数据的为历史数据，则读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送的步骤从所述推送任务处理时刻开始，从所述历史数据存储单元中读取数据。

[0014] 结合第一方面实施例，在一种可能的实施方式中，所述启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送，还包括：通过所述第一进程，读取在所述时段列表中的其他子时段内所产生的数据并进行推送，直至推送完毕；所述其他子时段为所述时段列表中除所述最后一个子时段以外的子时段。

[0015] 结合第一方面实施例，在一种可能的实施方式中，所述读取在所述时段列表中的其他子时段内所产生的数据并进行推送，包括：从所述推送任务处理时刻开始，从所述历史数据存储单元中读取在所述其他子时段内所产生的数据并进行推送。

[0016] 结合第一方面实施例，在一种可能的实施方式中，所述启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送，包括：根据待推送子时段确定当前时间切片；对所述当前时间切片内产生的数据进行读取和推送，并将所述时

段列表中待推送子时段的子起始时刻更新为所述当前时间切片的切片终止时刻，直至待推送子时段内的数据被推送完毕。

[0017] 结合第一方面实施例，在一种可能的实施方式中，所述根据待推送子时段确定当前时间切片，包括：将所述待推送子时段的子起始时刻作为当前时间切片的切片起始时刻，将当前时间切片的切片起始时刻与预设时长粒度之和、所述待推送子时段的子终止时刻中的较小值作为当前时间切片的切片终止时刻。

[0018] 结合第一方面实施例，在一种可能的实施方式中，所述方法还包括：在确定存在异常重启时，获取所述时段列表在重启时刻所包括的子时段；根据所述重启时刻，对所述时段列表在所述重启时刻所包括的子时段逐一进行分割；用分割后的子时段和/或孙时段更新所述时段列表；启动所述第一进程和/或所述第二进程对所述时段列表中的各个子时段和/或孙时段内所产生的数据进行读取和推送。

[0019] 结合第一方面实施例，在一种可能的实施方式中，每个孙时段都有孙起始时刻和孙终止时刻，所述对所述时段列表在所述重启时刻所包括的子时段逐一进行分割，包括：若所述重启时刻未落入待分割子时段，则所述待分割子时段保持不变；若所述重启时刻落入待分割子时段，则将所述待分割子时段分割为第一孙时段和第二孙时段，所述第一孙时段的孙起始时刻为所述待分割子时段的子起始时刻，所述第一孙时段的孙终止时刻为所述重启时刻，所述第二孙时段的孙起始时刻为所述重启时刻，所述第二孙时段的孙终止时刻为所述待分割子时段的子终止时刻。

[0020] 第二方面，本申请实施例提供一种订阅数据的装置，所述装置包括：获取模块、判断模块以及推送模块。获取模块，用于获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据；判断模块，用于根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据；推送模块，用于在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送；其中，未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据，且未来数据产生后实时写入缓存。

[0021] 结合第二方面实施例，在一种可能的实施方式中，所述判断模块，用于在所述推送任务处理时刻早于所述起始时刻时，确定所述订阅数据全部为未来数据；在所述推送任务处理时刻位于所述起始时刻及所述终止时刻之间时，确定所述订阅数据部分为未来数据，部分为历史数据；在所述推送任务处理时刻晚于所述终止时刻时，确定所述订阅数据全部为历史数据；其中，历史数据为数据的产生时刻早于所述推送任务处理时刻的数据。

[0022] 结合第二方面实施例，在一种可能的实施方式中，所述推送模块，还用于在所述订阅数据内包括历史数据时，从历史数据存储单元中读取所述订阅数据包括的历史数据并进行推送；其中，历史数据为数据的产生时刻早于所述推送任务处理时刻的数据；所述历史数据存储单元为不同于缓存的存储单元。

[0023] 结合第二方面实施例，在一种可能的实施方式中，所述装置还包括分割模块，用于根据所述推送任务处理时刻将所述订阅时段分割为至少一个子时段；将各个子时段加入到时段列表；相应的，所述推送模块，用于启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送；还用于启动第一进程和/或第二进程对所述

时段列表中的各个子时段内所产生的数据进行读取和推送。

[0024] 结合第二方面实施例，在一种可能的实施方式中，每个子时段都有子起始时刻和子终止时刻，所述分割模块，用于若所述推送任务处理时刻未落入所述订阅时段，则将所述订阅时段作为子时段，该子时段的子起始时刻为所述起始时刻，该子时段的子终止时刻为所述终止时刻；若所述推送任务处理时刻落入所述订阅时段，则将所述订阅时段分割为第一子时段和第二子时段，所述第一子时段的子起始时刻为所述起始时刻，所述第一子时段的子终止时刻为所述推送任务处理时刻，所述第二子时段的子起始时刻为所述推送任务处理时刻，所述第二子时段的子终止时刻为所述终止时刻。

[0025] 结合第二方面实施例，在一种可能的实施方式中，所述推送模块，用于通过所述第二进程，读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送，直至推送完毕。

[0026] 结合第二方面实施例，在一种可能的实施方式中，若所述最后一个子时段内所产生的数据的为未来数据，则所述推送模块读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送时，是从所述最后一个子时段的子起始时刻开始，从所述缓存中读取数据；和/或，若所述最后一个子时段内所产生的数据的为历史数据，则所述推送模块读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送时，是从所述推送任务处理时刻开始，从所述历史数据存储单元中读取数据。

[0027] 结合第二方面实施例，在一种可能的实施方式中，所述推送模块，用于通过所述第一进程，读取在所述时段列表中的其他子时段内所产生的数据并进行推送，直至推送完毕；所述其他子时段为所述时段列表中除所述最后一个子时段以外的子时段。

[0028] 结合第二方面实施例，在一种可能的实施方式中，所述推送模块，用于从所述推送任务处理时刻开始，从所述历史数据存储单元中读取在所述其他子时段内所产生的数据并进行推送。

[0029] 结合第二方面实施例，在一种可能的实施方式中，所述推送模块，用于根据待推送子时段确定当前时间切片；对所述当前时间切片内产生的数据进行读取和推送，并将所述时段列表中待推送子时段的子起始时刻更新为所述当前时间切片的切片终止时刻，直至待推送子时段内的数据被推送完毕。

[0030] 结合第二方面实施例，在一种可能的实施方式中，所述推送模块，用于将所述待推送子时段的子起始时刻作为当前时间切片的切片起始时刻，将当前时间切片的切片起始时刻与预设时长粒度之和、所述待推送子时段的子终止时刻中的较小值作为当前时间切片的切片终止时刻。

[0031] 结合第二方面实施例，在一种可能的实施方式中，所述装置还包括异常处理模块，用于在确定存在异常重启时，获取所述时段列表在重启时刻所包括的子时段；根据所述重启时刻，对所述时段列表在所述重启时刻所包括的子时段逐一进行分割；用分割后的子时段和/或孙时段更新所述时段列表；启动所述第一进程和/或所述第二进程对所述时段列表中的各个子时段和/或孙时段内所产生的数据进行读取和推送。

[0032] 结合第二方面实施例，在一种可能的实施方式中，每个孙时段都有孙起始时刻和孙终止时刻，所述异常处理模块，用于若所述重启时刻未落入待分割子时段，则所述待分割子时段保持不变；还用于若所述重启时刻落入待分割子时段，则将所述待分割子时段分割

为第一孙时段和第二孙时段，所述第一孙时段的孙起始时刻为所述待分割子时段的子起始时刻，所述第一孙时段的孙终止时刻为所述重启时刻，所述第二孙时段的孙起始时刻为所述重启时刻，所述第二孙时段的孙终止时刻为所述待分割子时段的子终止时刻。

[0033] 第三方面，本申请实施例还提供一种电子设备包括：存储器和处理器，所述存储器和所述处理器连接；所述存储器用于存储程序；所述处理器调用存储于所述存储器中的程序，以执行上述第一方面实施例和/或结合第一方面实施例的任一种可能的实施方式提供的方法。

[0034] 第四方面，本申请实施例还提供一种非易失性计算机可读取存储介质(以下简称计算机存储介质)，其上存储有计算机程序，所述计算机程序被计算机运行时执行上述第一方面实施例和/或结合第一方面实施例的任一种可能的实施方式提供的方法。

[0035] 本申请的其他特征和优点将在随后的说明书阐述，并且，部分地从说明书中变得显而易见，或者通过实施本申请实施例而了解。本申请的目的和其他优点可通过在所写的说明书以及附图中所特别指出的结构来实现和获得。

附图说明

[0036] 为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。通过附图所示，本申请的上述及其它目的、特征和优势将更加清晰。在全部附图中相同的附图标记指示相同的部分。并未刻意按实际尺寸等比例缩放绘制附图，重点在于示出本申请的主旨。

[0037] 图1示出本申请实施例提供的订阅数据的推送方法的流程图。

[0038] 图2示出本申请实施例提供的订阅数据的推送装置的结构框图。

[0039] 图3示出本申请实施例提供的一种电子设备的结构示意图。

[0040] 标号：100-电子设备；110-处理器；120-存储器；400-订阅数据的推送装置；410-获取模块；420-判断模块；430-推送模块。

具体实施方式

[0041] 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描述。

[0042] 应注意到：相似的标号和字母在下面的附图中表示类似项，因此，一旦某一项在一个附图中被定义，则在随后的附图中不需要对其进行进一步定义和解释。同时，在本申请的描述中诸如“第一”、“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0043] 再者，本申请中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存

在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。

[0044] 此外，针对现有技术中下级数据库的sub向上级数据库推送与订阅任务相对的订阅数据时所存在的缺陷(对历史数据存储单元查询的查询较为频繁，使得历史数据存储单元的查询压力较大)是申请人在经过实践并仔细研究后得出的结果，因此，上述缺陷的发现过程以及在下文中本申请实施例针对上述缺陷所提出的解决方案，都应该被认定为申请人对本申请所做出的贡献。

[0045] 为了解决上述问题，本申请实施例提供一种订阅数据的推送方法、装置、电子设备及计算机存储介质，有利于减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。

[0046] 该技术可采用相应的软件、硬件以及软硬结合的方式实现。以下对本申请实施例进行详细介绍。

[0047] 首先，对本申请实施例所涉及到的概念进行介绍。

[0048] 订阅时段：指的是某个时间范围，例如[a,b](a<＝b)、[a,b)(a<b)。其中，a称之为订阅时段的起始时刻，b称之为订阅时段的终止时刻。一般情况下，订阅任务所需获取的订阅数据是从由一个时刻到另一个时刻所组成的订阅时段所包括的时间范围内的数据，即订阅任务所需获取的订阅数据是针对一个订阅时段而言。

[0049] 时间分辨率：指的是将数据存储到历史数据存储单元时所记录的时间的最小时长粒度。例如，在存储数据时，记录的时间精确到秒数，则时间分辨率为1秒；如果记录的时间精确到分钟数，则时间分辨率为1分钟。

[0050] 预设时长粒度(时间增量)inc：表征sub从历史数据存储单元或缓存中分次(分时间切片)获取一个订阅时段内的订阅数据时，在常规情况下，一次(一个时间切片)所获取到的订阅数据的最早产生时间与最晚产生时间之间的时间宽度。例如订阅时段是[10,20)，inc＝5，第一个时间切片为[10,15)，第二个时间切片为[15,20)。订阅时段收敛：在本申请实施例中，对于订阅时段[a,b)，总是使得sub从起始时刻a开始，分多次从订阅时段[a,b)内获取订阅数据。每次获取订阅数据时，对应于一个当前时间切片，且每个当前时间切片均存在一个切片起始时刻以及一个切片终止时刻。在当前时间切片内的订阅数据推送完毕后，更新订阅时段[a,b)中a的值为当前时间切片的切片终止时刻，然后再更新当前时间切片，并重复上述操作。随着sub推送任务的进行，订阅时段[a,b)中a的值被不断更新变大，订阅时段所包括的时间范围越来越小，直至a≥b时，[a,b)为0。至此之后，[a,b)处于收敛状态。

[0051] 其中，最初的当前时间切片(P0)的切片起始时刻A为订阅时段[a,b)的起始时刻a，最初的当前时间切片(P0)的切片终止时刻为其切片起始时刻A与inc之和，即A+inc，且A+inc需要小于b，若A+inc大于等于b，则最初的当前时间切片(P0)的切片终止时刻为b。

[0052] 在最初的当前时间切片(P0)内的订阅数据被推送完毕后，下一个当前时间切片(P1)的切片起始时刻A为上一个当前时间切片(P0)的切片终止时刻A+inc，下一个的当前时间切片(P1)的切片终止时刻为本时间切片(P1)的切片起始时刻和预设时长粒度inc之和与b中的较小值，即A+inc+inc与b中的较小值。

[0053] 下面以订阅时段[a,b)的收敛过程为例，对订阅时段收敛的过程进行介绍。

[0054] 假设存在时间分辨率p、预设时长粒度inc(inc的值>＝p的值)，当前时间切片为[A,A+inc)。在不存在异常的情况下，在sub第一次启动时，令A＝a，当前时间切片P0为[A,A+

inc)，即[a,a+inc)，sub获取[a,a+inc)时间切片内的订阅数据，并将[a,a+inc)时间切片内的订阅数据推送给上级数据库。在推送成功之后，更新订阅时段[a,b)中a的值为当前时间切片P0的切片终止时刻a+inc，更新当前时间切片P1，将A的值修改为更新后的订阅时段的起始时刻，即上一次当前时间切片的切片终止时刻a+inc，那么当前时间切片P0为[a+inc,a+inc+inc]。sub获取[a,a+2inc)时间切片内的订阅数据，并将[a,a+2inc)时间切片内的订阅数据推送给上级数据库。然后重复上述过程，直至a≥b时，[a,b)收敛。通过上述举例可知，在整个获取数据并推送订阅数据的过程中，A为动态变量。

[0055] 值得指出的是，下一个当前时间切片的切片终止时刻为本时间切片的切片起始时刻和预设时长粒度inc之和与b中的较小值，即A+2inc与b中的较小值，这意味着sub每次获取的订阅数据对应的当前时间切片所涵盖的时间范围不能超越与最原始的订阅任务对应的订阅时段所涵盖的时间范围，即A+inc<＝b。若A+inc>b时，需要将A+inc从b的值所在的时间点截断，并将该次的当前时间切片的切片终止时刻从A+inc修改为b。

[0056] 例如，假设时间分辨率p＝1s，预设时长粒度inc＝10，订阅任务所需的数据对应的时段为[10,36)。针对这种情况，a＝10,b＝36。

[0057] 按照上述流程，sub启动后，A＝a＝10，当前时间切片[A,A+inc)为[10,20)，第一次获取并推送[10,20)之间的订阅数据。当第二次获取订阅数据时，将A修改为20，当前时间切片[A,A+inc)为[20,30)，第二次获取并推送[20,30)之间的订阅数据。当第三次获取订阅数据时，将A修改为30，当前时间切片[A,A+inc)为[30,40)，时间跨度超越[10,36)，因此，将当前时间切片[A,A+inc)修改为[A,b)，即修改为[30,36)，所以，第三次获取并推送[30,36)之间的订阅数据。当第四次获取订阅数据时，将A修改为36，此时36＝b，[10,36]收敛，说明[10,36)这个订阅时段的订阅数据完成推送。

[0058] 下面将针对本申请实施例所提供的推送方法进行介绍。

[0059] 本申请实施例提供一种应用于电子设备的sub的推送方法，用于将电子设备的本地数据库内的与订阅任务所需要的订阅数据推送给上级数据库，其中，上级数据库预先向本地数据库发起订阅任务，订阅任务中包括与订阅任务所需获取的订阅数据对应的订阅时段。

[0060] 下面将结合图1对其所包含的步骤进行说明。

[0061] 步骤S110:获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据。

[0062] 步骤S120:根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据。

[0063] 步骤S130:在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送。

[0064] 上文提及，上级数据库发起的订阅任务所需获取的订阅数据一般是针对一个订阅时段而言。假设与订阅数据对应的订阅时段为[a,b)。其中，a为起始时刻，b为终止时刻。

[0065] 此外，在本申请实施例中，引入缓存的概念。当产生新的数据时，实时产生的新的数据暂时保存在缓存中，然后再由缓存保存到历史数据存储单元中。

[0066] 在本申请实施例中，可以根据推送任务处理时刻now与起始时刻a以及终止时刻b之间的时间先后顺序，确定订阅数据所涵盖的数据的状态。

[0067] 其中，数据的状态可以分为未来数据以及历史数据。

[0068] 未来数据为数据的产生时刻晚于推送任务处理时刻now的数据，即未来数据的产生时刻在推送任务处理时刻now之后，说明未来数据即将产生并暂时存放在缓存内。

[0069] 历史数据为数据的产生时刻早于推送任务处理时刻now的数据，即历史数据的产生时刻在推送任务处理时刻now之前，且历史数据已经保存在历史数据存储单元内。历史数据存储单元为存储了历史数据的存储单元，可以理解的是，历史数据存储单元并不一定专用于历史数据的存储，还可以存储其他数据。历史数据存储单元例如是ElasticSearch(全文搜索引擎，简称ES)。

[0070] 可选的，在推送任务处理时刻now早于[a,b)的起始时刻a时，表征订阅数据内的数据全为未来数据；在推送任务处理时刻now晚于[a,b)的终止时刻b时，表征订阅数据内的数据全为历史数据；在推送任务处理时刻now位于起始时刻a以及终止时刻b之间时，表征订阅数据内的数据部分为历史数据，部分为未来数据。

[0071] 在本申请实施例中，对于订阅数据中所包含的历史数据以及未来数据的获取方式不同。

[0072] 其中，针对历史数据部分，sub通过上文中提及的使得订阅时段[a,b)收敛的方式，根据多个时间切片，分多次轮询历史数据存储单元以获取历史数据。

[0073] 针对未来数据部分，sub通过上文中提及的使得订阅时段[a,b)收敛的方式，根据多个时间切片，分多次从缓存中实时读取未来数据。

[0074] 下面将针对订阅数据内的数据全为未来数据、全为历史数据、部分为历史数据且部分为未来数据三种情况对获取订阅数据并进行推送的过程进行介绍。

[0075] 当订阅数据内的数据全为未来数据(即a>＝now)时，sub等待未来数据的最早产生时刻a的到来，并从未来数据的最早产生时刻(a时刻)开始从缓存中实时读取未来数据并进行推送，直到订阅时段[a,b)收敛。

[0076] 当订阅数据内的数据全为历史数据(即b<＝now)时，sub从推送任务处理时刻now开始，陆续获取并推送[a,b)时段内的历史数据，直到时段[a,b)收敛。

[0077] 当订阅数据内的数据部分为历史数据，部分为未来数据(a<now<b)时，为了减少sub对历史数据存储单元的查询次数，在一种可选的实施方式中，sub可以趁[now,b)时段对应的订阅数据(这部分数据为未来数据)被暂存在缓存时，先从now时刻开始从缓存中读取并推送作为未来数据(即在[now,b)这段时间内产生的数据)的订阅数据，待[now,b)这个子时段收敛后，sub再从历史数据存储单元中读取并推送作为历史数据(即与子时段[a,now)对应的数据)的订阅数据。

[0078] 也就是说，在本申请实施例中，当在订阅时段[a,b)内所产生的订阅数据中存在未来数据时，sub都是直接读取缓存获取这部分未来数据并进行推送。相较于现有技术中的等待未来数据保存到历史数据存储单元后再从历史数据存储单元读取未来数据的方案，在本申请实施例中，sub在读取未来数据时无需查询历史数据存储单元，从而至少可以减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。

[0079] 历史数据存储单元作为数据进出的通道，当其查询压力较大时，会对数据的读取以及写入造成影响，因此，减少对历史数据存储单元的查询次数，减轻历史数据存储单元的查询压力，有助于数据进出历史数据存储单元，从而提升数据推送的速度。

[0080] 值得指出的是，在上述实施例中，sub都是基于一个进程，例如第一进程或第二进程在获取并推送订阅数据。

[0081] 在另一种可选的实施方式中，为了提高订阅数据的推送效率，sub可以同时启动两个进程(例如第一进程以及第二进程)来获取并推送订阅数据。

[0082] 其中，第一进程负责获取并推送订阅时段[a,b)所产生的历史数据，第二进程负责获取并推送订阅时段[a,b)所产生的未来数据；或者由第二进程负责获取并推送订阅时段[a,b)所产生的历史数据，第一进程负责获取并推送订阅时段[a,b)所产生的未来数据。

[0083] 在这种实施方式下，可以根据推送任务处理时刻now，将订阅时段[a,b)分割为至少一个子时段。其中，每个子时段内所产生的数据的数据类型相同，即同一个子时段内所产生的数据，要么是历史数据，要么是未来数据。

[0084] 具体的，每个子时段都包括对应的子起始时刻以及子终止时刻。sub在对订阅时段[a,b)进行分割时，以推送任务处理时刻now为界线，根据now、a以及b之间的大小顺序，对订阅时段[a,b)进行分割，得到至少一个子时段。

[0085] 其中，当a>＝now或b<＝now时，说明now未落入订阅时段[a,b)所涵盖的时间范围内，此时，对订阅时段[a,b)分割所得到的子时段为一个，即订阅时段[a,b)本身，相应的，得到的子时段的子起始时刻为a，子终止时刻为b。

[0086] 当a<now<b时，说明now落入订阅时段[a,b)所涵盖的时间范围内，此时将以now为界线，将订阅时段[a,b)分割为第一子时段[a,now)以及第二子时段[now,b)。

[0087] 第一子时段[a,now)的子起始时刻为订阅时段[a,b)的起始时刻a，第一子时段[a,b)的子终止时刻为推送任务处理时刻now，第二子时段[now,b)的子起始时刻为推送任务处理时刻now，第二子时段[now,b)的子终止时刻为订阅时段[a,b)的终止时刻b。

[0088] 在得到至少一个子时段后，将各个子时段加入到时段列表L中，由第一进程和/或第二进程按照其对应的处理对象的类型，去获取并推送时段列表L中各个子时段内所产生的数据。

[0089] 在一个具体实施方式中，假设第一进程负责获取并推送历史数据，第二进程负责获取并推送未来数据。

[0090] 当订阅时段[a,b)经过分割，得到的子时段依旧为[a,b)时，L＝{[a,b)}。若[a,b)内所产生的数据全为历史数据，那么由第一进程负责对子时段[a,b)内所产生的历史数据进行获取并推送，此时第二进程处于空闲状态；若[a,b)内所产生的数据全为未来数据，那么由第而进程负责对子时段[a,b)内所产生的未来数据进行获取并推送，此时第一进程处于空闲状态。

[0091] 当订阅时段[a,b)经过分割，得到的子时段分别为[a,now)、[now,b)时，L＝{[a,now),[now,b)}。此时，由第一进程负责对子时段[a,now)内所产生的历史数据进行获取(查询历史数据存储单元)并推送，由第二进程负责对子时段[now,b)内所产生的未来数据进行获取(查询缓存)并推送。

[0092] 两个进程互不干扰，且当这两个子时段均收敛时，即完成对订阅时段[a,b)内的订阅数据的推送。由于两个进程可以同时工作，因此，可以提高数据推送效率。

[0093] 此外，在一种可选的实施方式中，sub在启动时，可以在不区分订阅数据是否包含未来数据的前提下，直接启动相应进程来对订阅数据进行处理，过程如下。例如，可以通过

第二进程读取并推送时段列表L中的最后一个子时段内所产生的订阅数据，通过第一进程读取时段列表L中除最后一个子时段外的其他子时段内所产生的订阅数据。

[0094] sub依旧按照上述对订阅时段[a,b)进行分割的方式，将订阅时段分割为至少一个子时段，并将得到的子时段加入到时段列表L。其中，当经过分割得到多个子时段时，将按照各个子时段所涵盖的时间段的先后顺序，将各个子时段依次加入时段列表L内。因此，对于a>=now或b<=now的情况，L={[a,b)}，对于a<now<b时的情况，[a,now)排在[now,b)之前，L={[a,now),[now,b)}。

[0095] 在得到时段列表L后，当时段列表L中包含2个以上的子时段时，sub同时启动第一进程及第二进程，然后通过第二进程读取时段列表L中的最后一个子时段内所产生的订阅数据并进行推送，直至推送完毕，通过第一进程读取时段列表L中除最后一个子时段外的其他子时段内所产生的订阅数据并进行推送，直至推送完毕。

[0096] 对应于a>=now或者b<=now的情况，时段列表L中仅包含一个子时段，sub通过第二进程读取并推送L中最后一个子时段[a,b)(由于只有[a,b)，因此，[a,b)即为最后一个子时段)所对应的订阅数据。此时，第一进程处于空闲等待状态。

[0097] 值得指出的是，当a>now时，由于订阅数据还未产生，因此，第二进程需要等待a时刻到来后，才能在a时刻从缓存读取并推送对应的数据。a<now时，由于订阅数据已经产生，因此，第二进程可以从now时刻开始，从历史数据存储单元读取并推送对应的数据。

[0098] 对应于a<now<b时的情况，sub通过第二进程读取并推送L中最后一个子时段[now,b)所对应的订阅数据，通过第一进程读取并推送L中其他子时段[a,now)所对应的订阅数据。

[0099] 在此过程中，不管a、b与now之间的大小关系为哪种情况，对于第一进程以及第二进程而言，当其所需读取的订阅数据为历史数据时，从历史数据存储单元中获取对应的数据，当其所需要读取的订阅数据为未来数据时，从缓存中获取对应的数据。

[0100] 参照前文介绍订阅时段[a,b)收敛时的相关描述，在对订阅时段[a,b)内的订阅数据进行推送时，总是使得sub从起始时刻a开始，分多次从订阅时段[a,b)内获取订阅数据。每次获取订阅数据时，对应于一个当前时间切片，且每个当前时间切片均存在一个切片起始时刻以及一个切片终止时刻。在当前时间切片内的订阅数据推送完毕后，更新订阅时段[a,b)中a的值为当前时间切片的切片终止时刻，然后再更新当前时间切片，并重复上述操作，直至订阅时段[a,b)收敛。

[0101] 当针对订阅时段[a,b)进行分割得到至少一个子时段时，在一种可选的实施方式中，针对每个子时段，也可以按照上述类似的方式，对每个子时段内的订阅数据进行推送，具体方式可以参照以下过程：

[0102] 各个进程在处理其负责处理的待推送子时段时，也是先确定当前时间切片，然后对当前时间切片内产生的数据进行读取和推送，当前时间切片内的数据推送完毕后，将时段列表中待推送子时段的子起始时刻更新为当前时间切片的切片终止时刻，直至待推送子时段内的数据被推送完毕。此时，将待推送子时段从时段列表L中删除。

[0103] 其中，将待推送子时段的子起始时刻作为当前时间切片的切片起始时刻，将当前时间切片的切片起始时刻与预设时长粒度inc之和、待推送子时段的子终止时刻中的较小值作为当前时间切片的切片终止时刻。

[0104] 针对最开始的当前时间切片P0,其切片起始时刻为待推送子时段的子起始时刻,其切片终止时刻为当前时间切片P0的切片起始时刻与预设时长粒度inc之和与待推送子时段的子终止时刻中的较小值。

[0105] 假设切片起始时刻与inc之和小于待推送子时段的子终止时刻,表明当前时间切片P0内的数据推送完毕后,待推送子时段中的数据并未推送完毕。在这种情况下,当前时间切片P0内的数据推送完毕后,更新待推送子时段的子起始时刻为当前时间切片P0的切片终止时刻,更新下一个当前时间切片P1的切片起始时刻为当前的待推送子时段的子起始时刻(即前一个当前时间切片P0的切片终止时刻),更新下一个当前时间切片P1的切片终止时刻为P的切片起始时刻与inc之和与待推送子时段的子终止时刻中的较小值。

[0106] 假设切片起始时刻与inc之和大于等于待推送子时段的子终止时刻,表明当前时间切片P0内的数据推送完毕后,待推送子时段中的数据就推送完毕。

[0107] 从上述过程可知,当第一进程和/或第二进程随着推送任务处理时刻now的推移,对其所负责处理的子时段内的订阅数据进行获取并推送时,sub可以根据第一进程和/或第二进程的处理进度,实时更新L中的每个子时段所包括的子起始时刻。

[0108] 例如,与订阅数据对应的订阅时段为[10,100),时间分辨率为1s,预设时长粒度为1,推送任务处理时刻now为20,因此,a=10,b=100,now=20,p=1,inc=1。

[0109] 当sub在推送任务处理时刻v启动时,根据now、a以及b之间的大小关系,将[10,100)分割为[10,20)以及[20,100),并得到L={[10,20),[20,100)}。

[0110] 启动第一进程以及第二进程,假设由第二进程负责处理子时段[20,100)对应的订阅数据,由第一进程负责处理子时段[10,20)对应的订阅数据。

[0111] 假设时间经过5S后,第一进程从历史数据存储单元获取并推送了8S的订阅数据,第二进程从缓存获取并推送了5S的订阅数据,此时,实时更新L中的每个子时段所包括的子起始时刻:将[10,20)的子起始时刻从10更新为18,将[20,100)的子起始时刻从20更新为25。至此,更新后的L为L={[18,20),[25,100)}。

[0112] 值得指出的是,时间的流逝速度与进程从历史数据存储单元或者从缓存内读取数据并进行推送的速度不存在相关性,因此,在此处经过5S后,第二进程可能从历史数据存储单元内推送出8S的订阅数据。

[0113] 以上对于sub推送订阅数据的介绍均是在sub处于正常状态的情况下进行的。当sub发生异常时,sub可能会停止推送服务。当sub再次重启后,之前被确定为是未来数据的订阅数据可能会随着时间的流逝已被保存在历史数据存储单元内,因此,依旧按照之前的方式对订阅数据进行推送会存在数据丢失的问题。

[0114] 为了解决上述问题,当sub在确定本次启动是由于异常而导致的异常重启时,先获取与重启对应的重启时刻now'。

[0115] 其中,当sub启动时,可以先查询L。若L为空,可以表征本次启动为第一次启动,需要按照上述流程对订阅时段进行分割,并加入到L,完成对L的初始化,并进行后续推送。此外,若L为空时,还可以表征订阅数据已经被推送完。

[0116] 若L不为空,则表征本次启动为发生异常后的启动,为了避免数据丢失,需要对L内已有的子时段进行分割。

[0117] 其中,可以以重启时刻now'为界线,重新对当前的L内存在的子时段逐一进行分

割，得到孙时段。至于对子时段进行分割的原理，与前文中对订阅时段进行分割的原理类似。

[0118] 其中，针对每个待分割的子时段，若重启时刻now'未落入待分割子时段，则待分割子时段保持不变。

[0119] 若重启时刻now'落入待分割子时段内，则将待分割子时段分割为第一孙时段和第二孙时段。其中，第一孙时段的孙起始时刻为待分割子时段的子起始时刻，第一孙时段的孙终止时刻为重启时刻now'；第二孙时段的孙起始时刻为重启时刻now'，第二孙时段的孙终止时刻为待分割子时段的子终止时刻。

[0120] 在对L内的各个子时段进行分割后，用分割后的子时段和/或孙时段更新时段列表L。然后sub通过第二进程读取并推送更新后的L中位于最后的时段(该时段可能是子时段，也可能是孙时段)所对应的订阅数据，并通过第一进程读取并推送更新后的L中除最后的子时段或孙时段之外的其他子时段或孙时段所对应的订阅数据。

[0121] 依旧以上文中的例子进行说明，即与订阅数据对应的订阅时段为[10,100)，a＝10，b＝100，now＝20，p＝1，inc＝1。

[0122] 前文提及，在时间经过5S后，更新后的L为L＝{[18,20)，[25,100)}。此时，sub发生异常，且经过10S后重启。

[0123] sub重启后，获取重启时刻，即重启时刻now'＝20+5(经过5S的正常推送)+10＝35，获取当前的L＝{[18,20)，[25,100)}，并判断L不为空。此时，确定为异常重启，sub将根据now'、当前的L所包括的每个子时段的子起始时刻与子终止时刻(对应于子时段[18,20)，分别为18以及20，对应于子时段[25,100)，分别为25以及100)，重新对当前的L内存在的各个子时段逐一进行分割，得到更新后的子时段以及孙时段[18,20)、[25,35)、[35,100)，以及得到更新后的L＝{[18,20)、[25,35)、[35,100)}。

[0124] sub启动第一进程以及第二进程，通过第二进程获取并推送位于当前的L中最后的孙时段[35,100)所对应的订阅数据，通过第一进程获取并推送当前的L中其他子时段[18,20)、其他孙时段[25,35)所对应的订阅数据。

[0125] 在此过程中，不管各个子时段的子起始时刻、子终止时刻与now'之间的大小关系为哪种情况，不管各个孙时段的孙起始时刻、孙终止时刻与now'之间的大小关系为哪种情况，对于第一进程以及第二进程而言，当其所需读取的订阅数据为历史数据时，从历史数据存储单元中获取对应的数据，当其所需要读取的订阅数据为未来数据时，从缓存中获取对应的数据。

[0126] 在上述举例中，本应该由第二进程从缓存中读取的与[25,35)子时段对应的订阅数据，由于异常启动，这部分的订阅数据从缓存保存到了历史数据存储单元，以避免这部分数据丢失。因此，在sub重启后，需要从历史数据存储单元读取该部分数据。

[0127] 当然，在第一进程与第二进程获取并推送各自对应的子时段和/或孙时段所对应的订阅数据的过程中，依旧可以实时更新当前的L中的每个子时段、孙时段所包括的子起始时刻或孙起始时刻。

[0128] 本方案在sub发生重启的情况后仍能继续推送，具有鲁棒性。

[0129] 本申请实施例所提供的一种订阅数据的推送方法，该方法先获取与订阅数据对应的时段，时段包括起始时刻及终止时刻；在根据推送任务处理时刻与起始时刻，或者在根据

推送任务处理时刻、起始时刻及终止时刻确定订阅数据内包括未来数据时，从未来数据的最早产生时刻开始，从缓存中实时读取未来数据并进行推送。相较于现有技术中的等待未来数据保存到历史数据存储单元后再从历史数据存储单元读取未来数据的方案，本方案中的sub在读取未来数据时无需查询历史数据存储单元，从而可以减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。

[0130] 对应于图1，请参看图2，本申请实施例还提供一种订阅数据的推送装置400，订阅数据的推送装置400可以包括：获取模块410、判断模块420以及推送模块430。

[0131] 其中，获取模块410，用于获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据；

[0132] 判断模块420，用于根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据；

[0133] 推送模块430，用于在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送；

[0134] 其中，未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据，且未来数据产生后实时写入缓存。

[0135] 在一种可能的实施方式中，所述判断模块420，用于在所述推送任务处理时刻早于所述起始时刻时，确定所述订阅数据全部为未来数据；在所述推送任务处理时刻位于所述起始时刻及所述终止时刻之间时，确定所述订阅数据部分为未来数据，部分为历史数据；在所述推送任务处理时刻晚于所述终止时刻时，确定所述订阅数据全部为历史数据；其中，历史数据为数据的产生时刻早于所述推送任务处理时刻的数据。

[0136] 在一种可能的实施方式中，所述推送模块430，还用于在所述订阅数据内包括历史数据时，从历史数据存储单元中读取所述订阅数据包括的历史数据并进行推送；其中，历史数据为数据的产生时刻早于所述推送任务处理时刻的数据；所述历史数据存储单元为不同于缓存的存储单元。

[0137] 在一种可能的实施方式中，所述装置还包括分割模块，用于根据所述推送任务处理时刻将所述订阅时段分割为至少一个子时段；将各个子时段加入到时段列表；相应的，所述推送模块430，用于启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送；还用于启动第一进程和/或第二进程对所述时段列表中的各个子时段内所产生的数据进行读取和推送。

[0138] 在一种可能的实施方式中，每个子时段都有子起始时刻和子终止时刻，所述分割模块，用于若所述推送任务处理时刻未落入所述订阅时段，则将所述订阅时段作为子时段，该子时段的子起始时刻为所述起始时刻，该子时段的子终止时刻为所述终止时刻；若所述推送任务处理时刻落入所述订阅时段，则将所述订阅时段分割为第一子时段和第二子时段，所述第一子时段的子起始时刻为所述起始时刻，所述第一子时段的子终止时刻为所述推送任务处理时刻，所述第二子时段的子起始时刻为所述推送任务处理时刻，所述第二子时段的子终止时刻为所述终止时刻。

[0139] 在一种可能的实施方式中，所述推送模块430，用于通过所述第二进程，读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送，直至推送完毕。

[0140] 在一种可能的实施方式中，若所述最后一个子时段内所产生的数据的为未来数

据，则所述推送模块读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送时，是从所述最后一个子时段的子起始时刻开始，从所述缓存中读取数据；和/或，若所述最后一个子时段内所产生的数据的为历史数据，则所述推送模块读取在所述时段列表中的最后一个子时段内所产生的数据并进行推送时，是从所述推送任务处理时刻开始，从所述历史数据存储单元中读取数据。

[0141] 在一种可能的实施方式中，所述推送模块430，用于通过所述第一进程，读取在所述时段列表中的其他子时段内所产生的数据并进行推送，直至推送完毕；所述其他子时段为所述时段列表中除所述最后一个子时段以外的子时段。

[0142] 在一种可能的实施方式中，所述推送模块430，用于从所述推送任务处理时刻开始，从所述历史数据存储单元中读取在所述其他子时段内所产生的数据并进行推送。

[0143] 在一种可能的实施方式中，所述推送模块430，用于根据待推送子时段确定当前时间切片；对所述当前时间切片内产生的数据进行读取和推送，并将所述时段列表中待推送子时段的子起始时刻更新为所述当前时间切片的切片终止时刻，直至待推送子时段内的数据被推送完毕。

[0144] 在一种可能的实施方式中，所述推送模块430，用于将所述待推送子时段的子起始时刻作为当前时间切片的切片起始时刻，将当前时间切片的切片起始时刻与预设时长粒度之和、所述待推送子时段的子终止时刻中的较小值作为当前时间切片的切片终止时刻。

[0145] 在一种可能的实施方式中，所述装置还包括异常处理模块，用于在确定存在异常重启时，获取所述时段列表在重启时刻所包括的子时段；根据所述重启时刻，对所述时段列表在所述重启时刻所包括的子时段逐一进行分割；用分割后的子时段和/或孙时段更新所述时段列表；启动所述第一进程和/或所述第二进程对所述时段列表中的各个子时段和/或孙时段内所产生的数据进行读取和推送。

[0146] 在一种可能的实施方式中，每个孙时段都有孙起始时刻和孙终止时刻，所述异常处理模块，用于若所述重启时刻未落入待分割子时段，则所述待分割子时段保持不变；还用于若所述重启时刻落入待分割子时段，则将所述待分割子时段分割为第一孙时段和第二孙时段，所述第一孙时段的孙起始时刻为所述待分割子时段的子起始时刻，所述第一孙时段的孙终止时刻为所述重启时刻，所述第二孙时段的孙起始时刻为所述重启时刻，所述第二孙时段的孙终止时刻为所述待分割子时段的子终止时刻。

[0147] 本申请实施例所提供的订阅数据的推送装置400，其实现原理及产生的技术效果和前述方法实施例相同，为简要描述，装置实施例部分未提及之处，可参考前述方法实施例中相应内容。

[0148] 此外，本申请实施例还提供一种计算机存储介质，该计算机存储介质上存储有计算机程序，该计算机程序被计算机运行时，执行如上述的推送方法所包含的步骤。

[0149] 此外，请参看图3，本申请实施例还提供一种用于实现上述推送方法以及推送装置的电子设备100，包括处理器110以及存储器120。

[0150] 应当注意，图3所示的电子设备100的组件和结构只是示例性的，而非限制性的，根据需要，电子设备100也可以具有其他组件和结构。

[0151] 处理器110、存储器120以及其他可能出现于电子设备100的组件相互之间直接或间接地电性连接，以实现数据的传输或交互。例如，处理器110、存储器120以及其他可能出

现的组件相互之间可通过一条或多条通讯总线或信号线实现电性连接。

[0152] 存储器120中的部分存储空间作为本申请实施例中所提及到的缓存。此外，存储器120中的另一部分存储空间还用于存储数据库、sub以及程序，例如存储有前文出现的推送方法对应的程序或者前文出现的推送装置。可选的，当存储器120内存储有推送装置时，推送装置包括至少一个可以以软件或固件(firmware)的形式存储于存储器120中的软件功能模块。

[0153] 可选的，推送装置所包括软件功能模块也可以固化在电子设备100的操作系统(operating system，OS)中。

[0154] 处理器110用于执行存储器120中存储的可执行模块，例如推送装置包括的软件功能模块或计算机程序。当处理器110在接收到执行指令后，可以执行计算机程序，例如执行：获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据；根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据；在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送；其中，未来数据为数据的产生时刻晚于所述推送任务处理时刻的数据，且未来数据产生后实时写入缓存。

[0155] 当然，本申请任一实施例所揭示的方法都可以应用于处理器110中，或者由处理器110实现。

[0156] 综上所述，本发明实施例提出的订阅数据的推送方法、装置、电子设备及计算机存储介质，该方法先获取与订阅数据对应的时段，时段包括起始时刻及终止时刻；在根据推送任务处理时刻与起始时刻，或者在根据推送任务处理时刻、起始时刻及终止时刻确定订阅数据内包括未来数据时，从未来数据的最早产生时刻开始，从缓存中实时读取未来数据并进行推送。相较于现有技术中的等待未来数据保存到历史数据存储单元后再从历史数据存储单元读取未来数据的方案，本方案中的sub在读取未来数据时无需查询历史数据存储单元，从而可以减少对历史数据存储单元的查询次数，从而减轻历史数据存储单元的查询压力。

[0157] 需要说明的是，本说明书中的各个实施例均采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似的部分互相参见即可。

[0158] 在本申请所提供的几个实施例中，应该理解到，所揭露的装置和方法，也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，附图中的流程图和框图显示了根据本申请的多个实施例的装置、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现方式中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0159] 另外，在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部

分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0160] 所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机存储介质中。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该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品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,笔记本电脑,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(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0161] 以上所述,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。

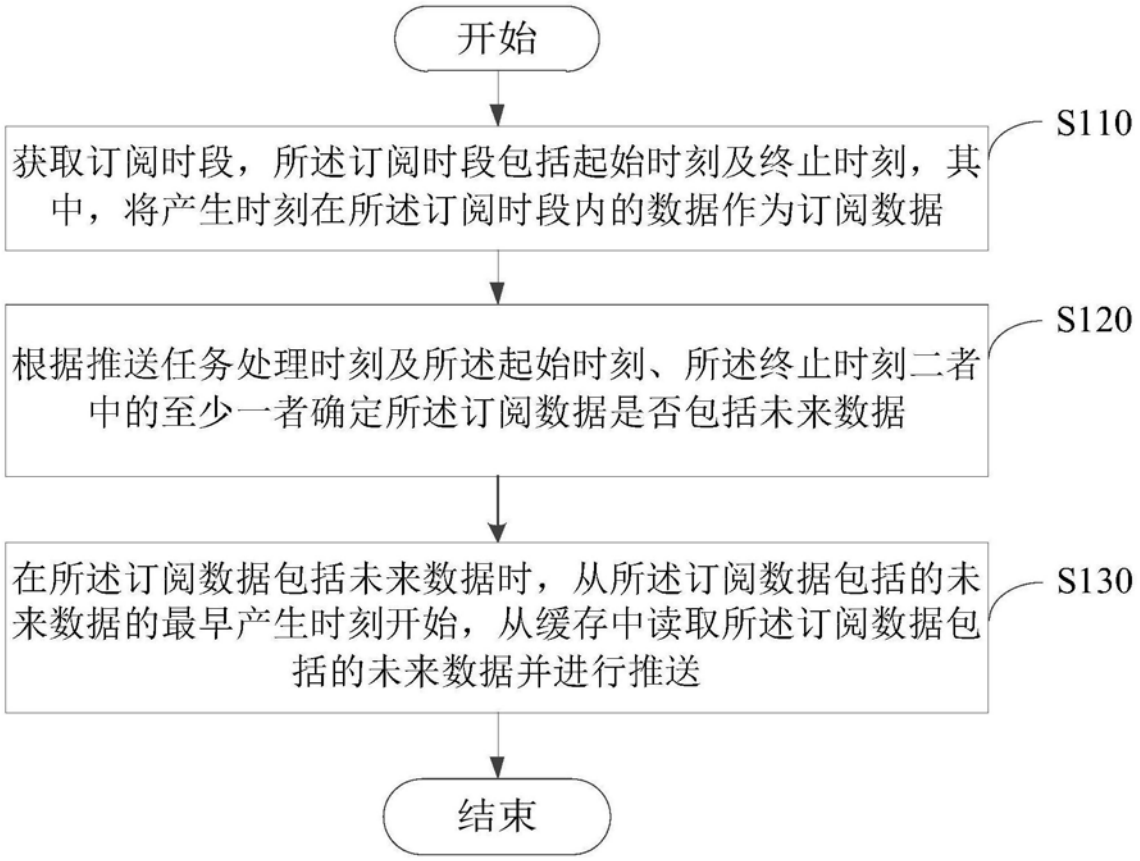

开始
获取订阅时段，所述订阅时段包括起始时刻及终止时刻，其中，将产生时刻在所述订阅时段内的数据作为订阅数据
S110
根据推送任务处理时刻及所述起始时刻、所述终止时刻二者中的至少一者确定所述订阅数据是否包括未来数据
S120
在所述订阅数据包括未来数据时，从所述订阅数据包括的未来数据的最早产生时刻开始，从缓存中读取所述订阅数据包括的未来数据并进行推送
S130
结束

图1

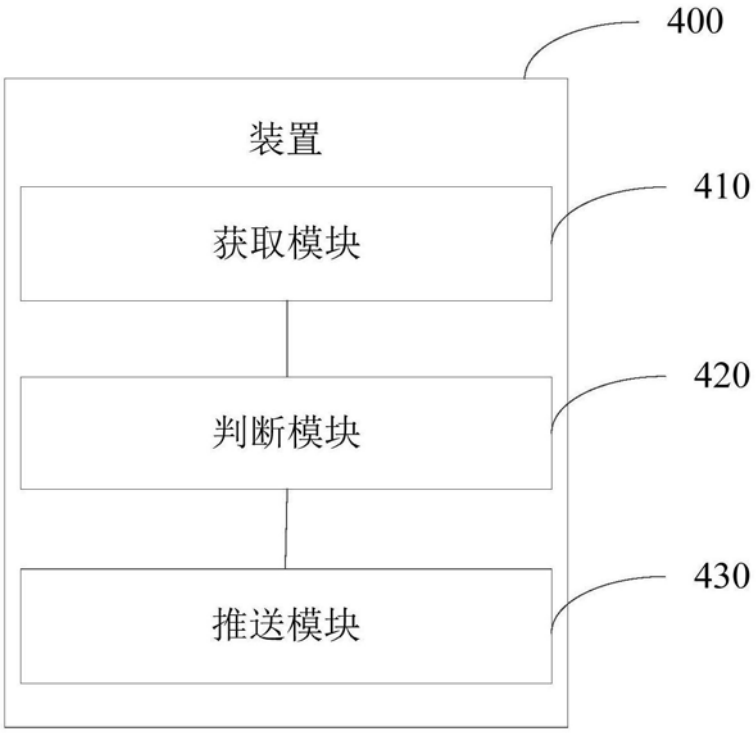


图2

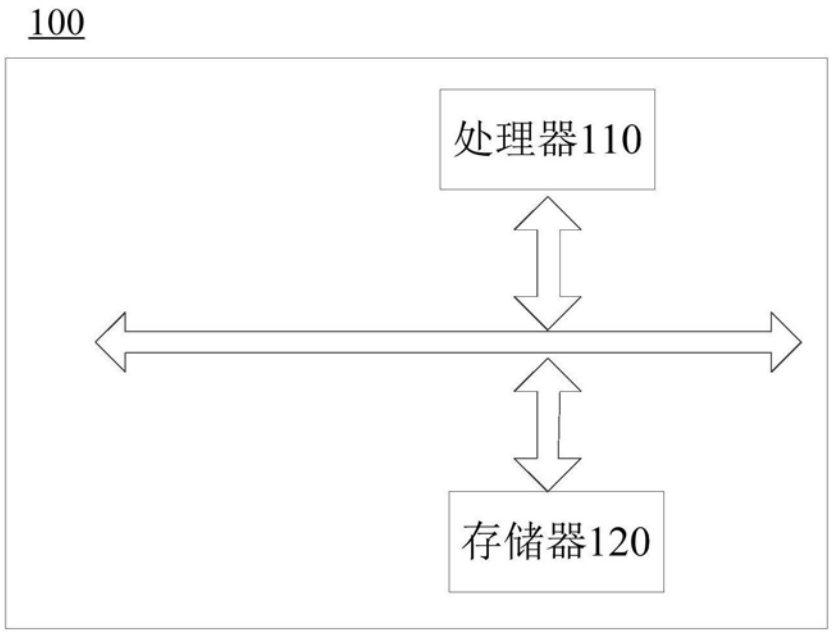


图3