



(21) 申请号 202010199062.5

G06F 16/28 (2019.01)

(22) 申请日 2020.03.20

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 106095863 A, 2016.11.09

申请公布号 CN 111414403 A

CN 106294499 A, 2017.01.04

(43) 申请公布日 2020.07.14

CN 107423368 A, 2017.12.01

(73) 专利权人 腾讯科技(深圳)有限公司

CN 108549725 A, 2018.09.18

地址 518000 广东省深圳市南山区高新区

US 2016140174 A1, 2016.05.19

科技中一路腾讯大厦35层

US 8572091 B1, 2013.10.29

(72) 发明人 欧霄 黄东庆 刘骏健 李建东

马镇. 多租户SaaS应用系统关键技术的研究与实现.《中国优秀硕士学位论文全文数据库 信息科技辑》. 2020, (第03期), I138-309.

(74) 专利代理机构 华进联合专利商标代理有限公司

闫湖; 周薇; 李立新; 戴娇; 韩冀中; 狄方春;. 基于分布式键值对存储技术的EMS数据库平台. 电网技术. 2012, (第09期), 167-172.

公司 44224

专利代理师 于丽君 李文渊

审查员 倪浩天

(51) Int. Cl.

G06F 16/2458 (2019.01)

权利要求书3页 说明书22页 附图10页

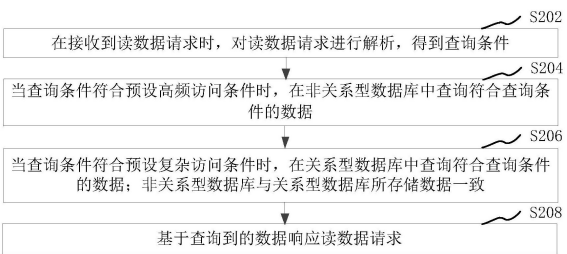
G06F 16/27 (2019.01)

(54) 发明名称

数据访问方法和装置、数据存储方法和装置

(57) 摘要

本申请涉及一种数据访问方法和装置、数据存储方法和装置。所述数据访问方法包括：在接收到读数据请求时，对所述读数据请求进行解析，得到查询条件；当所述查询条件符合预设高频访问条件时，在非关系型数据库中查询符合所述查询条件的数据；当所述查询条件符合预设复杂访问条件时，在关系型数据库中查询符合所述查询条件的数据；所述非关系型数据库与所述关系型数据库所存储数据一致；基于查询到的数据响应所述读数据请求。采用本方法既能满足高频访问需求也能满足复杂查询需求。



1. 一种数据访问方法,包括:

在接收到读数据请求时,对所述读数据请求进行解析,得到查询条件;

当所述查询条件符合预设高频访问条件时,在非关系型数据库中查询符合所述查询条件的数据;所述预设高频访问条件是预设的用于将所述读数据请求判定为来自高频简单访问场景的数据请求的指标;

当所述查询条件符合预设复杂访问条件时,在关系型数据库中查询符合所述查询条件的数据;所述非关系型数据库与所述关系型数据库所存储数据一致;所述预设复杂访问条件是预设的用于将所述读数据请求判定为来自低频复杂访问场景的数据请求的指标;

基于查询到的数据响应所述读数据请求。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述查询条件为预设的路由字段时,判定所述查询条件符合预设高频访问条件;所述路由字段表征所述非关系型数据库进行分布式存储时所依据的数据标识;

当所述查询条件包括除所述预设的路由字段之外的查询字段时,判定所述查询条件符合预设复杂访问条件。

3. 根据权利要求1所述的方法,其特征在于,所述非关系型数据库包括多个分布式子库;所述当所述查询条件符合预设高频访问条件时,在非关系型数据库中查询符合所述查询条件的数据包括:

当所述查询条件为预设的路由字段时,确定所述查询字段所属的分布式区间;

在用于存储所述分布式区间内的数据的分布式子库中,查询符合所述查询条件的数据。

4. 根据权利要求1所述的方法,其特征在于,所述非关系型数据库具有一个或多个;所述当所述查询条件符合预设高频访问条件时,在非关系型数据库中查询符合所述查询条件的数据包括:

当所述查询条件为路由字段时,在将所述路由字段作为分布式存储依据的非关系型数据库中,查询符合所述查询条件的数据。

5. 根据权利要求1所述的方法,其特征在于,所述关系型数据库包括主库和至少一个从库;所述在关系型数据库中查询符合所述查询条件的数据包括:

基于所述查询条件向所述主库发起查询请求;

当在预设时长内未接收到所述主库针对所述查询请求的查询应答时,将其中一个所述从库确定为当前的主库;

在所述当前的主库中查询符合所述查询条件的数据。

6. 根据权利要求5所述的方法,其特征在于,所述方法还包括:

获取对所述主库执行写操作时生成的日志;

将所述日志发送至一个或多个目标的从库,使所述目标的从库根据所述日志同步执行所述写操作;

当接收到所述目标的从库在执行所述写操作后返回的同步确认信息时,基于写入成功的结果来响应用于触发所述写操作的写数据请求,向其余所述从库发送所述日志。

7. 一种数据存储方法,所述方法包括:

在接收到写数据请求时,对所述写数据请求进行解析,得到待写入数据;

将所述待写入数据写入一种数据库；

将所述待写入数据从所述一种数据库同步至另一种数据库；其中，所述数据库的种类包括关系型数据库和非关系型数据库；所述非关系型数据库用于响应符合预设高频访问条件的读数据请求，所述关系型数据库用于响应符合预设复杂访问条件的读数据请求；所述预设高频访问条件是预设的用于将所述读数据请求判定为来自高频简单访问场景的数据请求的指标，所述预设复杂访问条件是预设的用于将所述读数据请求判定为来自低频复杂访问场景的数据请求的指标；

基于所述待写入数据的写入结果响应所述写数据请求。

8. 根据权利要求7所述的方法，其特征在于，所述待写入数据包括数据标识和目标数据内容；所述将所述待写入数据写入一种数据库包括：

当所述写数据请求为数据更新请求时，根据所述目标数据内容对所述非关系型数据库中存储的与所述数据标识对应的原数据内容进行更新；

所述将所述待写入数据从所述一种数据库同步至另一种数据库包括：

在所述非关系型数据库更新完毕时，通过消息队列将所述目标数据内容从所述非关系型数据库异步更新至所述关系型数据库。

9. 根据权利要求8所述的方法，其特征在于，所述目标数据内容包括待写入数据的目标版本；所述根据目标数据内容对所述非关系型数据库存储的与所述数据标识对应的原数据内容进行更新包括：

确定非关系型数据库中存储的与数据标识对应的原数据内容的存储版本；

当所述目标版本等于所述存储版本时，根据目标数据内容对所述非关系型数据库存储的与所述数据标识对应的原数据内容进行更新；

在更新成功后，对所述存储版本进行更新。

10. 根据权利要求7所述的方法，其特征在于，所述待写入数据包括数据标识和目标数据内容；所述将所述待写入数据写入一种数据库包括：

当所述写数据请求为插入数据请求时，确定所述数据标识是否已存储于所述关系型数据库中；

若所述数据标识尚未存储于所述关系型数据库中，将所述待写入数据插入所述关系型数据库及所述非关系型数据库；

若所述数据标识已存储于所述关系型数据库中，将所述待写入数据插入所述非关系型数据库，并在插入成功后，通过消息队列将所述目标数据内容从所述非关系型数据库异步更新至所述关系型数据库。

11. 根据权利要求10所述的方法，其特征在于，所述若所述数据标识已存储于所述关系型数据库中，将所述待写入数据插入所述非关系型数据库包括：

若所述数据标识已存储于所述关系型数据库中，但尚未存储于所述非关系型数据库时，将所述待写入数据插入所述非关系型数据库；

若所述数据标识已存储于所述关系型数据库及所述非关系型数据库时，将写入成果确定为写入失败。

12. 根据权利要求8至11任一项所述的方法，其特征在于，所述通过消息队列将目标数据内容从非关系型数据库异步更新至所述关系型数据库包括：

当所述待写入数据对应的入队标识为待入队时,将所述待写入数据的数据标识添加至消息队列,在所述非关系型数据库中將所述入队标识更新为排队中;

按照排队顺序对所述消息队列中每个数据标识进行遍历;

根据当前遍历顺序的数据标识在所述非关系型数据库中查询对应的目标数据内容,基于查询到的目标数据内容,对所述关系型数据库存储的与所述数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

13.根据权利要求10所述的方法,其特征在于,所述方法包括:

对所述关系型数据库中每条数据进行遍历;

在当前遍历顺序数据的数据标识未存储在所述非关系型数据库中时,将所述当前遍历顺序的数据插入所述非关系型数据库;

在当前遍历顺序数据的数据标识存储在所述非关系型数据库中,但对应的数据内容不一致时,根据所述非关系型数据库中的数据内容对所述关系型数据库中的数据内容进行更新。

14.一种数据访问装置,其特征在于,所述装置包括:

数据读取请求模块,用于在接收到读数据请求时,对所述读数据请求进行解析,得到查询条件;

数据查询模块,用于当所述查询条件符合预设高频访问条件时,在非关系型数据库中查询符合所述查询条件的数据;当所述查询条件符合预设复杂访问条件时,在关系型数据库中查询符合所述查询条件的数据;所述非关系型数据库与所述关系型数据库所存储数据一致;所述预设高频访问条件是预设的用于将所述读数据请求判定为来自高频简单访问场景的数据请求的指标,所述预设复杂访问条件是预设的用于将所述读数据请求判定为来自低频复杂访问场景的数据请求的指标;

查询结果反馈模块,用于基于查询到的数据响应所述读数据请求。

15.一种数据存储装置,其特征在于,所述装置包括:

数据写入请求模块,用于在接收到写数据请求时,对所述写数据请求进行解析,得到待写入数据;

数据准实时同步模块,用于将所述待写入数据写入一种数据库;将所述待写入数据从所述一种数据库同步至另一种数据库;其中,所述数据库的种类包括关系型数据库和非关系型数据库;所述非关系型数据库用于响应符合预设高频访问条件的读数据请求,所述关系型数据库用于响应符合预设复杂访问条件的读数据请求;所述预设高频访问条件是预设的用于将所述读数据请求判定为来自高频简单访问场景的数据请求的指标,所述预设复杂访问条件是预设的用于将所述读数据请求判定为来自低频复杂访问场景的数据请求的指标;

写入结果反馈模块,用于基于所述待写入数据的写入结果响应所述写数据请求。

16.一种计算机设备,包括存储器及处理器,所述存储器中储存有计算机程序,其特征在于,所述计算机程序被所述处理器执行时,使得所述处理器执行如权利要求1至13中任一项所述的方法的步骤。

17.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至13中任一项所述的方法的步骤。

数据访问方法和装置、数据存储方法和装置

技术领域

[0001] 本申请涉及数据库技术领域，特别是涉及一种数据访问方法和装置、数据存储方法和装置。

背景技术

[0002] 互联网中每天都会产生海量数据，为了实现对海量数据的有效管理，互联网服务提供商大多会将产生的数据存储于数据库(DataBase, DB)中。目前，主要是采用MySQL、Oracle等关系型数据库对数据进行存储。关系型数据库采用结构化查询语言进行数据查询，并支持对数据库中的数据进行增删改查操作以及跨表查询功能，使用便利，易于理解，因而得到越来越广的应用。

[0003] 然而，关系型数据库是以单张数据表的形式进行数据存储，这种单表读写的性能限制无法适应与日俱增的数据访问量，影响数据的正常访问，逐渐成为业务发展的瓶颈。

发明内容

[0004] 基于此，有必要针对上述技术问题，提供一种既能满足高并发访问需求也能满足复杂查询需求的数据访问及存储方法、装置、计算机设备和存储介质。

[0005] 一种数据访问方法，所述方法包括：

[0006] 在接收到读数据请求时，对所述读数据请求进行解析，得到查询条件；

[0007] 当所述查询条件符合预设高频访问条件时，在非关系型数据库中查询符合所述查询条件的数据；

[0008] 当所述查询条件符合预设复杂访问条件时，在关系型数据库中查询符合所述查询条件的数据；所述非关系型数据库与所述关系型数据库所存储数据一致；

[0009] 基于查询到的数据响应所述读数据请求。

[0010] 一种数据访问装置，所述装置包括：

[0011] 数据读取请求模块，用于在接收到读数据请求时，对所述读数据请求进行解析，得到查询条件；

[0012] 数据查询模块，用于当所述查询条件符合预设高频访问条件时，在非关系型数据库中查询符合所述查询条件的数据；当所述查询条件符合预设复杂访问条件时，在关系型数据库中查询符合所述查询条件的数据；所述非关系型数据库与所述关系型数据库所存储数据一致；

[0013] 查询结果反馈模块，用于基于查询到的数据响应所述读数据请求。

[0014] 在一个实施例中，所述数据访问装置还包括访问场景识别模块，用于当查询条件为预设的路由字段时，判定查询条件符合预设高频访问条件；当查询条件包括除路由字段之外的查询字段时，判定查询条件符合预设复杂访问条件。

[0015] 在一个实施例中，非关系型数据库包括多个分布式子库；所述数据查询模块包括高频访问查询模块，用于当查询条件为路由字段时，确定查询字段所属的分布式区间；在用

于存储分布式区间内的数据的分布式子库中,查询符合查询条件的数据。

[0016] 在一个实施例中,非关系型数据库具有一个或多个;所述高频访问查询模块还用于当查询条件为路由字段时,在将路由字段作为分布式存储依据的非关系型数据库中,查询符合查询条件的数据。

[0017] 在一个实施例中,关系型数据库包括主库和至少一个从库;所述数据查询模块还包括复杂访问查询模块,用于基于查询条件向主库发起查询请求;当在预设时长内未接收到主库针对查询请求的查询应答时,将其中一个从库确定为当前的主库;在当前的主库中查询符合查询条件的数据。

[0018] 在一个实施例中,所述数据访问装置还包括半同步复制模块,用于获取对主库执行写操作时生成的日志;将日志发送至一个或多个目标的从库,使目标的从库根据日志同步执行写操作;当接收到目标的从库在执行写操作后返回的同步确认信息时,基于写入成功的结果来响应用于触发写操作的写数据请求,向其余从库发送日志。

[0019] 一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现以下步骤:

[0020] 在接收到读数据请求时,对所述读数据请求进行解析,得到查询条件;

[0021] 当所述查询条件符合预设高频访问条件时,在非关系型数据库中查询符合所述查询条件的数据;

[0022] 当所述查询条件符合预设复杂访问条件时,在关系型数据库中查询符合所述查询条件的数据;所述非关系型数据库与所述关系型数据库所存储数据一致;

[0023] 基于查询到的数据响应所述读数据请求。

[0024] 一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现以下步骤:

[0025] 在接收到读数据请求时,对所述读数据请求进行解析,得到查询条件;

[0026] 当所述查询条件符合预设高频访问条件时,在非关系型数据库中查询符合所述查询条件的数据;

[0027] 当所述查询条件符合预设复杂访问条件时,在关系型数据库中查询符合所述查询条件的数据;所述非关系型数据库与所述关系型数据库所存储数据一致;

[0028] 基于查询到的数据响应所述读数据请求。

[0029] 上述数据访问方法、装置、计算机设备和存储介质,关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力;通过结合两种存储技术的优势并保证两种数据库中数据的一致性,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

[0030] 一种数据存储方法,所述方法包括:

[0031] 在接收到写数据请求时,对所述写数据请求进行解析,得到待写入数据;

[0032] 将所述待写入数据写入一种数据库;

[0033] 将所述待写入数据从所述一种数据库同步至另一种数据库;其中,所述数据库的

种类包括关系型数据库和非关系型数据库；所述非关系型数据库用于响应符合预设高频访问条件的读数据请求，所述关系型数据库用于响应符合预设复杂访问条件的读数据请求；

[0034] 基于所述待写入数据的写入结果响应所述写数据请求。

[0035] 在一个实施例中，上述数据存储方法还包括：

[0036] 当所述写数据请求为插入数据请求，且所述数据标识尚未存储于所述关系型数据库中，在将所述待写入数据插入所述关系型数据库及所述非关系型数据库后，初始化所述待写入数据的入队标识为待入队；

[0037] 当所述数据标识已存储于关系型数据库中，但尚未存储于非关系型数据库时，在将所述待写入数据插入所述非关系型数据库后，将所述待写入数据的数据标识添加至所述消息队列，并初始化所述待写入数据的入队标识为排队中。

[0038] 在一个实施例中，上述数据存储方法还包括：

[0039] 按照预设时间频率统计所述关系型数据库所存储全部数据的总数据量；

[0040] 根据所述总数据量确定进行遍历的目标时间频率；

[0041] 所述对所述关系型数据库中每条数据进行遍历包括：

[0042] 按照所述目标时间频率对所述关系型数据库中每条数据进行遍历。

[0043] 一种数据存储装置，所述装置包括：

[0044] 数据写入请求模块，用于在接收到写数据请求时，对所述写数据请求进行解析，得到待写入数据；

[0045] 数据准实时同步模块，用于将所述待写入数据写入一种数据库；将所述待写入数据从所述一种数据库同步至另一种数据库；其中，所述数据库的种类包括关系型数据库和非关系型数据库；所述非关系型数据库用于响应符合预设高频访问条件的读数据请求，所述关系型数据库用于响应符合预设复杂访问条件的读数据请求；

[0046] 写入结果反馈模块，用于基于所述待写入数据的写入结果响应所述写数据请求。

[0047] 在一个实施例中，待写入数据包括数据标识和目标数据内容；所述数据准实时同步模块包括数据更新同步模块，用于当写数据请求为数据更新请求时，根据目标数据内容对非关系型数据库中存储的与数据标识对应的原数据内容进行更新；将待写入数据从一种数据库同步至另一种数据库包括：在非关系型数据库更新完毕时，通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库。

[0048] 在一个实施例中，目标数据内容包括待写入数据的目标版本；所述数据更新同步模块还用于确定非关系型数据库中存储的与数据标识对应的原数据内容的存储版本；当目标版本等于存储版本时，根据目标数据内容对非关系型数据库存储的与数据标识对应的原数据内容进行更新；在更新成功后，将存储版本更新为目标版本。当待写入数据版本低于存储版本时，确定写入结果为写入失败。

[0049] 在一个实施例中，待写入数据包括数据标识和目标数据内容；所述数据准实时同步模块还包括数据插入同步模块，用于当写数据请求为插入数据请求时，确定数据标识是否已存储于关系型数据库中；若数据标识尚未存储于关系型数据库中，将待写入数据插入关系型数据库及非关系型数据库；若数据标识已存储于关系型数据库中，将待写入数据插入非关系型数据库，并在插入成功后，通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库。

[0050] 在一个实施例中,所述数据插入同步模块还用于若数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,将待写入数据插入非关系型数据库;若数据标识已存储于关系型数据库及非关系型数据库时,将写入成果确定为写入失败。

[0051] 在一个实施例中,所述数据更新同步模块还用于当待写入数据对应的入队标识为待入队时,将待写入数据的数据标识添加至消息队列,在非关系型数据库中将入队标识更新为排队中;按照排队顺序对消息队列中每个数据标识进行遍历;根据当前遍历顺序的数据标识在非关系型数据库中查询对应的目标数据内容,基于查询到的目标数据内容,对关系型数据库存储的与数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

[0052] 在一个实施例中,所述数据存储装置还包括数据异步更新模块,用于当写数据请求为插入数据请求,且数据标识尚未存储于关系型数据库中,在将待写入数据插入非关系型数据库后,初始化待写入数据的入队标识为待入队;当数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,在将待写入数据插入非关系型数据库后,将待写入数据的数据标识添加至消息队列,并初始化待写入数据的入队标识为排队中。

[0053] 在一个实施例中,所述数据存储装置还包括数据兜底对账模块,用于对关系型数据库中每条数据进行遍历;在当前遍历顺序数据的数据标识未存储在非关系型数据库中时,将当前遍历顺序的数据插入非关系型数据库;在当前遍历顺序数据的数据标识存储在非关系型数据库中,但对应的数据内容不一致时,根据非关系型数据库中的数据内容对关系型数据库中的数据内容进行更新。

[0054] 一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现以下步骤:

[0055] 在接收到写数据请求时,对所述写数据请求进行解析,得到待写入数据;

[0056] 将所述待写入数据写入一种数据库;

[0057] 将所述待写入数据从所述一种数据库同步至另一种数据库;其中,所述数据库的种类包括关系型数据库和非关系型数据库;所述非关系型数据库用于响应符合预设高频访问条件的读数据请求,所述关系型数据库用于响应符合预设复杂访问条件的读数据请求;

[0058] 基于所述待写入数据的写入结果响应所述写数据请求。

[0059] 一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现以下步骤:

[0060] 在接收到写数据请求时,对所述写数据请求进行解析,得到待写入数据;

[0061] 将所述待写入数据写入一种数据库;

[0062] 将所述待写入数据从所述一种数据库同步至另一种数据库;其中,所述数据库的种类包括关系型数据库和非关系型数据库;所述非关系型数据库用于响应符合预设高频访问条件的读数据请求,所述关系型数据库用于响应符合预设复杂访问条件的读数据请求;

[0063] 基于所述待写入数据的写入结果响应所述写数据请求。

[0064] 上述数据存储方法、装置、计算机设备和存储介质,当发生写数据操作时,先将数据写入一种数据库,再将数据从该数据库同步至另一种数据库,由于是数据库中的数据是同步得到的,可以保证两种数据库中数据的一致性,由于是基于写数据操作事件驱动数据的写入和同步,可以保证数据同步的实时性。关系型数据库支持低并发的数据访问操作,能

够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力。在保证两种数据库中数据一致性的前提下,通过结合两种存储技术的优势,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

附图说明

- [0065] 图1为一个实施例中数据访问及存储方法的应用环境图;
- [0066] 图2为一个实施例中数据访问方法的流程示意图;
- [0067] 图3为一个实施例中数据访问方法的原理示意图;
- [0068] 图4为一个实施例中基于主从保护机制的关系型数据库的示意图;
- [0069] 图5为一个具体实施例中数据访问方法的流程示意图;
- [0070] 图6为一个实施例中数据存储方法的流程示意图;
- [0071] 图7为一个实施例中数据存储方法的原理示意图;
- [0072] 图8为一个实施例中响应数据更新请求在关系型数据库和非关系型数据库之间进行准实时同步的流程示意图;
- [0073] 图9为一个实施例中响应数据插入请求时在关系型数据库和非关系型数据库之间进行准实时同步的流程示意图;
- [0074] 图10为一个实施例中在关系型数据库和非关系型数据库中进行兜底对账的流程示意图;
- [0075] 图11为一个具体实施例中数据存储方法的流程示意图;
- [0076] 图12为一个实施例中数据访问装置的结构示意图;
- [0077] 图13为另一个实施例中数据访问装置的结构示意图;
- [0078] 图14为一个实施例中数据存储装置的结构示意图;
- [0079] 图15为另一个实施例中数据存储装置的结构示意图;
- [0080] 图16为一个实施例中数据服务设备实现为服务器时的内部结构图。

具体实施方式

[0081] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0082] 本申请提供的数据访问及数据存储方法,可以应用于如图1所示的应用环境中。其中,数据访问设备110与数据服务设备120通过有线或无线通信方式进行直接或间接地连接。数据服务设备120部署有对应的数据库130。数据库130包括关系型数据库130a和非关系型数据库130b。非关系型数据库130b可以部署一个,也可以部署有多个。数据访问方可以通过数据访问设备110向数据服务设备120发起读数据请求或写数据请求。数据服务设备120为数据访问方提供数据调度服务,根据读数据请求从关系型数据库130a或非关系型数据库130b中查询数据,或根据写数据请求向从关系型数据库130a和非关系型数据库130b写入数

据,并保证关系型数据库130a与非关系型数据库130b之间的数据一致性。

[0083] 数据访问设备110具体可以是终端,也可以是服务器,还可以是终端与服务器的组合。比如,数据访问方通过终端向服务器发起读数据请求,服务器将读数据请求转发至数据服务设备120。数据服务设备120可以是服务器。其中,终端具体可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表等,但并不局限于此。服务器具体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,还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN(Content Delivery Network,内容分发网络)、以及大数据和人工智能平台等基础云计算服务的云服务器。

[0084] 在一个实施例中,如图2所示,提供了一种数据访问方法,以该方法应用于图1中的数据服务设备120为例进行说明,包括以下步骤:

[0085] 步骤202,在接收到读数据请求时,对读数据请求进行解析,得到查询条件。

[0086] 其中,数据访问设备中运行有可执行本申请提供的数据访问方法及数据存储方法的目标应用,如社交应用、支付应用、游戏应用等。目标应用具体可以是母应用还可以是子应用(Mini Program,又称小程序)。母应用可以是原生APP(Application,应用程序),也可以是基于页面浏览引用访问的网络页面。读数据请求可以是数据访问方在数据访问设备通过目标应用发起的用于指示数据服务设备对数据库中的数据进行读操作的数据请求。查询条件是数据访问方发起读数据请求时在目标应用输入的查询字段。

[0087] 具体地,参考图3,图3示出了一个实施例中数据访问方法的原理示意图。如图3所示,当需要查询数据时,数据访问方可以基于数据访问设备上的目标应用设定查询条件。查询条件可以包含一个查询字段,也可以包括多个查询字段。目标应用根据查询字段生成读数据请求,将读数据请求发送至数据服务设备。数据服务设备部署有对应的数据库。数据服务设备在接收到读数据请求后,对读数据请求进行解析,得到查询条件,根据查询条件为数据访问方提供数据调度服务。

[0088] 步骤204,当查询条件符合预设高频访问条件时,在非关系型数据库中查询符合查询条件的数据。

[0089] 其中,数据库(DB,Database),又称为数据管理系统,简而言之可视为电子化的文件柜——存储电子文件的处所,用户可以对文件中的数据运行新增、截取、更新、删除等操作。所谓“数据库”系以一定方式储存在一起、能予多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。通常意义上的数据库,指信息化中的数据存储载体。根据数据存储结构的不同,可以将数据库区分为关系型数据库和非关系型数据库。

[0090] 非关系型数据库(Not Only SQL,NoSQL)去掉了关系型数据库的关系型特征,数据之间无关系,正是得益于这种无关系性,使非关系型数据库结构简单,具有非常高的读写性能,能够满足高频的访问需求。非关系型数据库中的使用聚合模型来处理。聚合模型主要有键值对(Key-Value)、BSON、列族、文档、图形等方式。在本申请的实施例中所采用的非关系型数据库主要是键值存储数据库。键值存储数据库将数据存储为键值对(Key-Value)对集合,其中键Key作为唯一标识符指向值Value。值Value无结构化,通常只被当作字符串或二进制数据,如何解释由使用者自定义。

[0091] 键值存储数据库具体可以是KV(Ktable,一种分布式存储系统)、Redis(Remote

Dictionary Server,远程字典服务)、DynamoDB、Apache Cassandra等。其中,KV是微信自研的一种分布式存储系统,包括KV server端和KV client端,同时提供了一套结构化的数据语义库,比如类似SQL (Structured Query Language,结构化查询语言)的数据操纵接口Select、Update、Insert、Delete等。Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型的键值存储数据库,提供了多种语言的API。Redis支持存储的value类型很多,包括string(字符串)、list(链表)、set(集合)、zset(sorted set,有序集合)和hash(哈希类型)等。

[0092] 实际应用中,根据查询条件的复杂度可以将数据访问场景区分为简单访问场景和复杂访问场景。简单访问场景需要根据数据记录的主要标识来进行高频的数据读写操作;复杂访问场景需要根据数据记录的多个属性内容为条件来筛选符合此多种条件的数据集,从而进行读写操作。这两种场景在大量目标应用的业务场景中都是经常出现的,而这两种场景的出现频率以及对数据层的要求也是不一致的。简单访问场景往往是高频的,且只需要根据数据的唯一标识符来进行数据记录的筛选。复杂访问场景则往往是非高频的,但需要数据层能提供根据多个字段条件来进行范围筛选的能力。

[0093] 预设高频访问条件是预设的用于将当前的读数据请求判定为来自高频简单访问场景的数据请求的指标,具体可以是查询条件中仅包含一个查询字段且该查询字段为一条数据记录的唯一标识(即查询主键),或者数据访问方发起读数据请求时所采用的数据通道为第一通道等。预设复杂访问条件是预设的用于将当前的读数据请求判定为来自低频复杂访问场景的数据请求的指标,具体可以是查询条件中包含查询字段的数量多于一个,或者数据访问方发起读数据请求时所采用的数据通道为第二通道等。

[0094] 具体地,目标应用在数据访问页面提供多种访问通道,如第一通道和第二通道等。当数据访问方在输入查询字段后基于第一通道的按钮发起读数据请求时,数据服务设备根据读数据请求携带的第一通道标识,将该读数据请求判定为符合预设高频访问条件的请求。如图3所示,当查询请求符合预设高频访问条件时,数据服务设备根据预设的非关系型数据库的访问语法规则,如XML (Extensible Markup Language,可扩展标记语言),生成查询条件对应的查询语句,基于该查询语句在非关系型数据库中查询符合查询条件的数据。

[0095] 步骤206,当查询条件符合预设复杂访问条件时,在关系型数据库中查询符合查询条件的数据;非关系型数据库与关系型数据库所存储数据一致。

[0096] 其中,关系型数据库是指采用了关系模型来组织数据的数据库,其以行和列的形式存储数据,以便于用户理解,关系型数据库这一系列的行和列被称为表,一组表组成了数据库。表中每一行的数据构成一条记录。每条记录中的各数据之间是有关系的。用户可以通过限定一个或多个查询字段来检索数据库中的数据,比如,通过限定“得分”和“性别”字段,可以在学生信息表中查询得分在80-90之间的女生人数。关系模型可以简单理解为二维表格模型,而一个关系型数据库就是由二维表及其之间的关系组成的一个数据组织。主流的关系型数据库有Oracle、DB2、MySQL、Microsoft SQL Server、Microsoft Access等。

[0097] 综合而言,关系型数据库支持低并发的数据访问操作,能够提供根据包含多个查询字段的复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据用于分布式存储的唯一查询主键Key进行数据筛选的能力。

[0098] 具体地,当数据访问方在输入查询字段后基于第二通道的按钮发起读数据请求

时,数据服务设备根据读数据请求携带的第二通道标识,将该读数据请求判定为符合预设复杂访问条件的请求。如图3所示,当查询请求符合预设复杂访问条件时,数据服务设备根据预设的关系型数据库的访问语法规则,如SQL (Structured Query Language,结构化查询语言),生成查询条件对应的查询语句,基于该查询语句在关系型数据库中查询符合查询条件的数据。

[0099] 在一个实施例中,上述数据访问方法还包括:当查询条件为预设的路由字段时,判定查询条件符合预设高频访问条件;当查询条件包括除路由字段之外的查询字段时,判定查询条件符合预设复杂访问条件。

[0100] 在非关系型数据库中,键Key和值Value都可以是从简单对象到复杂复合对象的任何内容,但键Key是唯一能够查询到该条数据的主键。例如,用户登录社交应用进行会话期间,社交应用可以将所有与会话相关的数据存储在非关系型数据库中。会话数据可能包括用户资料信息、消息、个性化数据和主题、建议、有针对性的促销和折扣等。每个用户会话具有唯一的标识符,该标识符即为能够在非关系型数据库中查询到该会话数据的唯一主键。路由字段是非关系型数据库进行分布式存储时所依据的数据标识,即为能够查询到非关系型数据库中数据的唯一查询主键。

[0101] 数据服务设备确定查询条件所包含的查询字段是否属于路由字段。若是,数据服务设备判定该查询条件符合预设高频访问条件。当查询条件包含除了路由字段之外的其他查询字段时,即当查询条件包含多个查询字段,或者查询条件仅包含一个查询字段,但该查询字段并不属于路由字段时,数据服务设备判定该查询条件符合预设复杂访问条件。

[0102] 如下表一所示,本实施例中,在高频简单访问场景,基于唯一数据标识在非关系型数据库中查询数据;在复杂访问场景中,基于多个查询字段在关系型数据库中查询数据。

[0103] 表一

[0104]	访问场景	操作频率	筛选范围	查询数据库
	简单访问	高频	路由字段	非关系型数据库
	复杂访问	低频	包含除路由字段之外的一个或多个查询字段	关系型数据库

[0105] 值得说明的是,本申请实施例中,关系型数据库与非关系型数据库中存储的数据内容一致,从而保证从其中任一种数据库查询到的数据一致。在一个实施例中,可以定期或不定期对关系型数据库与非关系型数据库中的数据进行一致性校验,并在不一致时进行数据同步处理,以保证关系型数据库与非关系型数据库中数据的一致性。

[0106] 步骤208,基于查询到的数据响应读数据请求。

[0107] 具体地,数据服务设备将从关系型数据库或者非关系型数据库查询到的数据发送至数据访问设备,以响应读数据请求。对于数据访问方而言,直接通过数据服务设备提供的数据调度服务进行数据访问操作,无需关系其实现细节。

[0108] 上述数据访问方法中,关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力;通过结合两种存储技术的优势并保证两种数据库中

数据的一致性,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

[0109] 在一个实施例中,非关系型数据库包括多个分布式子库;当查询条件符合预设高频访问条件时,在非关系型数据库中查询符合查询条件的数据包括:当查询条件为路由字段时,确定查询字段所属的分布式区间;在用于存储分布式区间内的数据的分布式子库中,查询符合查询条件的数据。

[0110] 其中,非关系型数据库是高度可分区的,并且允许以其他类型的数据库无法实现的规模进行水平扩展,如果现有分区填满了容量,并且需要更多的存储空间,非关系型数据库就会将额外的分区分配给表,从而实现分布式存储。综合而言,非关系型数据库能够支持海量数据的分布式存储。

[0111] 非关系型数据库包括多个分布式子库。分布式区间是非关系型数据库进行分布式存储时每个分布式子库所存储数据的路由字段的取值范围。比如,假设每条数据的数据标识(即路由字段)为用户ID,则可以将每100位用户的数据存储在一个分布式子库中,如将用户[ID0, ID100]的数据存储至分布式子库A1,将用户[ID101, ID200]的数据存储至分布式子库A2,如此类推。

[0112] 当判定查询条件符合预设高频访问条件时,数据服务设备在非关系型数据库中查询数据。具体地,数据服务设备确定查询条件中路由字段所属的分布式区间,进一步确定用于存储分布式区间内的数据的分布式子库,在所确定的分布式子库中查询符合查询条件的数据。

[0113] 本实施例中,相比集中存储,分布式存储可以提高数据安全性,此外,基于分布式子库进行数据查询,可以减少数据查询时数据处理量,进而提高数据查询效率。

[0114] 在一个实施例中,非关系型数据库具有一个或多个;当查询条件符合预设高频访问条件时,在非关系型数据库中查询符合查询条件的数据包括:当查询条件为路由字段时,在将路由字段作为分布式存储依据的非关系型数据库中,查询符合查询条件的数据。

[0115] 其中,如上文所述,每个非关系型数据库具有对应的路由字段,基于该路由字段能够向相应的非关系型数据库发起高并发访问请求。如果期望针对多种字段实现高并发访问,则可以分别部署每个字段对应的非关系型数据库。比如,可以非关系型数据库A和B分别将字段A和B作为路由字段。可以理解,每个非关系型数据库中所存储的数据内容一致,只是将不同的字段作为路由字段。每个关系型数据库可以包括多个分布式子库。

[0116] 具体地,在确定读数据请求来自高频访问场景后,若非关系型数据库具有多个,数据服务设备根据查询字段属于哪一非关系型数据库对应的路由字段,进一步确定在哪一非关系型数据库中查询数据。比如,在上述举例中,后续在接收到仅包含查询字段A的读数据请求时,可以在非关系型数据库A中查询数据;而在接收到仅包含查询字段B的读数据请求时,可以在非关系型数据库B中查询数据。

[0117] 本实施例中,针对细分的不同高频访问场景分布部署对应的非关系型数据库,可以同时满足多种高频访问场景的数据访问请求。

[0118] 在一个实施例中,关系型数据库包括主库和至少一个从库;在关系型数据库中查

询符合查询条件的数据包括:基于查询条件向主库发起查询请求;当在预设时长内未接收到主库针对查询请求的查询应答时,将其中一个从库确定为当前的主库;在当前的主库中查询符合查询条件的数据。

[0119] 其中,除了数据访问性能,数据可用性也至关重要。高可用的数据库是由一系列的数据库构成的总体系统,要求在任何时刻,至少有一个数据库节点能够响应用户的数据访问请求并提供数据服务。关系型数据库将数据集中存储在单张数据表,对数据安全性要求高,因而针对关系型数据库采用主从备用的高可用机制。高可用的关系型数据库系统包括主库和至少一个从库。主库用于响应数据访问方发起的数据访问请求,并将数据访问方对数据的写操作信息同步至从库。从库用于对主库中的数据进行备用。可以理解,主库与从库均为关系型数据库。

[0120] 具体地,在确定读数据请求来自复杂访问场景后,数据服务设备从关系型数据库系统的主库中查询数据,即根据查询条件向主库发送查询请求。当在预设时长内仍未接收到主库针对查询请求的查询应答时,数据服务设备判定主库发生故障。参考图4,图4示出了一个实施例中基于主从保护机制的关系型数据库的示意图。如图4所示,当主库发生故障时,数据服务设备随机将一个从库确定为新的主库,或者将当前性能最优的一个从库确定为新的主库等,而后在当前的主库中查询符合查询条件的数据。

[0121] 在一个实施例中,当在预设时长内仍未接收到主库针对查询请求的查询应答时,数据服务设备再次向主库发起查询请求,如此重试预设次数,若预设次数查询请求均为得到应答,则判定主库发生故障。

[0122] 在一个实施例中,非关系型数据库也可以按照上述主从备用的机制提高数据安全性。由于非关系型数据库是分布式存储的,可以针对每个分布式子库分别部署对应的备用数据库。可以理解,备用数据库与分布式子库均为非关系型数据库。

[0123] 本实施例中,有一个主库处理主要的数据库访问请求,还有若干备用的从库用于容灾切换,当主库不能提供服务时,备用的从库将自动称为主库并继续提供服务,能够保证整个关系型数据库系统的可用和稳定。

[0124] 在一个实施例中,上述数据库访问方法还包括:获取对主库执行写操作时生成的日志;将日志发送至一个或多个目标的从库,使目标的从库根据日志同步执行写操作;当接收到目标的从库在执行写操作后返回的同步确认信息时,基于写入成功的结果来响应用于触发写操作的写数据请求,向其余从库发送日志。

[0125] 除了持续稳定的数据服务,还应当保证主库和从库之间数据的一致性。本申请实施例基于复制日志的数据同步模式来保证主库和从库之间的数据一致性。复制日志的数据同步模式是指对主库的数据操作以日志的形式发送给各个从库,从库接收到日志后即进行同样的数据操作,完成数据备用。数据操作包括读操作和写操作。写操作包括更新操作、插入操作等。复制日志的数据同步模式中,主库与至少一个从库连接,能够方便的实现读写分离,同时因为每个从库都在运行中,从库里面的数据均为热数据,可以快速实现容灾切换。

[0126] 复制日志的数据同步模式具体可以采用异步复制(Asynchronous replication)、半同步复制(Semisynchronous replication)和全同步复制(Fully synchronous replication)其中任一种。其中,异步复制是指主库将新生成的日志发送给各个从库后,无需等待从库的确认回复信息即认为已完成同步,向数据访问方反馈数据操作已成功的信

息。MySQL等数据库默认的复制是异步的,异步复制的方式虽然可以提高数据操作的速度,减少耗时,但数据可靠性降低。在极端情况下,若主库刚提交日志,其他从库尚未接收到相关日志时,主库便发生故障,由于主库此时已经向数据访问方返回数据操作已成功的信息,导致该日志的数据操作内容全部丢失。

[0127] 全同步复制方式是指主库将新生成的日志发送给各个从库后,需要等待所有从库的确认回复信息才认为完成同步。全同步的复制方式虽然能够保证数据可靠性,但数据操作速率收到严重影响。

[0128] 半同步复制是介于异步复制和全同步复制之间的一种数据同步方式,是指数据服务设备在将主库新生成的日志发送给各个主库之前,需要发送日志到预设数量目标的从库,等待从库返回确认信息后,数据服务设备再提交日志给其余从库并直接视为同步完成。目标的从库可以是随机的从库,也可以是预先指定的从库。预设数量可以根据需求自由设定,如1等。可以理解,预设数量越多,数据同步所需耗时越长,数据可靠性越高。当预设数量为1时,可以保证至少一个从库已真正完成同步时,才向数据访问方返回数据操作已成功的信息,保证数据可靠性。其余从库按照异步复制的方式,将日志发出即认为完成其余从库的同步,减少耗时。

[0129] 由于半同步复制的方式在未接收到目标从库返回的确认回复信息时并不向数据访问方返回数据操作已完成的信息,当出现主库刚提交日志,其他从库尚未接收到相关日志时,主库便发生故障的极端情况时,数据访问方需要重复数据操作,进而再次触发一次日志的复制,从而面对这种极端情况,半同步复制的方式也可以保证数据可靠性,保证容灾切换时不丢失数据。

[0130] 本实施例中,基于日志的主从半复制模式在主库和从库之间进行数据同步,还可以保障主库和不同从库之间数据的一致性,且半同步的复制模式相比异步复制模式能够提高数据可靠性,相比全同步的复制模式能够有效减少耗时,提高数据库操作效率。

[0131] 在一个具体地实施例中,如图5所示,上述数据访问方法包括:

[0132] S502,在接收到读数据请求时,对读数据请求进行解析,得到查询条件。

[0133] S504,当查询条件为路由字段时,在将路由字段作为分布式存储依据的非关系型数据库中,查询符合查询条件的数据。

[0134] S506,当查询条件包括除路由字段之外的查询字段时,基于查询条件向主关系型数据库发起查询请求。

[0135] S508,当在预设时长内未接收到主关系型数据库针对查询请求的查询应答时,将其中一个从关系型数据库确定为当前的主关系型数据库。

[0136] S510,在当前的主关系型数据库中查询符合查询条件的数据。

[0137] S512,基于查询到的数据响应读数据请求。

[0138] 上述数据访问方法,关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力;通过结合两种存储技术的优势并保证两种数据库中数据的一致性,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提

高数据访问性能。

[0139] 应该理解的是,虽然图2和5的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,图2和5中的至少一部分步骤可以包括多个步骤或者多个阶段,这些步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤中的步骤或者阶段的至少一部分轮流或者交替地执行。

[0140] 在一个实施例中,如图6所示,提供了一种数据存储方法,以该方法应用于图1中的数据服务设备120为例进行说明,包括以下步骤:

[0141] 步骤602,在接收到写数据请求时,对写数据请求进行解析,得到待写入数据。

[0142] 其中,写数据请求是数据访问方在数据访问设备通过目标应用发起的用于指示数据服务设备对数据库中的数据进行写操作的数据请求。写操作包括更新操作、插入操作等。更新操作是指对数据进行修改或删除的操作。删除操作通常不会真的将数据从数据库删除,而是采用更新的方式将数据的有效性字段配置为无效。

[0143] 具体地,当需要向数据库中写入数据时,数据访问方可以基于数据访问设备上的目标应用输入待写入数据。目标应用根据待写入数据生成写数据请求,将写数据请求发送至数据服务设备。数据服务设备对写数据请求进行解析,得到待写入数据,根据待写入数据为数据访问方提供数据调度服务。

[0144] 步骤604,将待写入数据写入一种数据库。

[0145] 步骤606,将待写入数据从一种数据库同步至另一种数据库;其中,数据库的种类包括关系型数据库和非关系型数据库;非关系型数据库用于响应符合预设高频访问条件的读数据请求,关系型数据库用于响应符合预设复杂访问条件的读数据请求。

[0146] 对数据库中的数据的操作包括读操作和写操作。读数据不会引起数据的变更,因而无需在关系型数据库和非关系型数据库之间进行数据同步。而写操作则会造成数据变更,因而需要在关系型数据库和非关系型数据库之间进行数据同步操作,以保证数据一致性。具体的同步策略可以是将待写入数据写入一种数据库后,再将待写入数据从该数据库同步至另一种数据库。比如,在进行更新操作时,可以将待写入数据先更新至非关系型数据库,再根据非关系型数据库来更新关系型数据库。再比如,在进行插入操作时,可以将待写入数据先插入关系型数据库,再根据关系型数据库来更新非关系型数据库。

[0147] 参考图7,图7示出了一个实施例中数据存储方法的原理示意图。如图7所示,可以基于准实时同步策略在关系型数据库和非关系型数据库进行数据同步。关系型数据库和非关系型数据库之间的数据同步操作是基于写操作事件驱动的,换言之,一旦发生写数据请求,立即将待写入数据写入一种数据库,并随即同步至另一种数据库。这种基于事件驱动的数据同步策略可以使关系型数据与非关系型数据库的同步只有微小的时间间隔,虽然不能实现完全实时,可以实现准实时同步。这种基于事件驱动的同步策略即为准实时同步策略。

[0148] 步骤608,基于待写入数据的写入结果响应写数据请求。

[0149] 其中,写入结果是指待写入数据是否已成功写入数据库的结果信息,具体包括写入成功和写入失败。对于不同的写操作,写入结果进一步可以区分为更新结果、插入结果等。写入结果的判定规则可以是在待写入数据成功写入一种数据库时即将写入结果确定为

写入成功,也可以只有待写入数据在两种数据库中均写入成功时才确定写入结果为写入成功。

[0150] 具体地,数据服务设备根据写入结果的判定规则,在确定待写入数据的写入结果后将写入结果返回给数据访问方。比如,在进行更新操作时,可以待写入数据成功更新至非关系型数据库时即向数据访问方返回数据更新成功的信息。再比如,在进行插入操作时,可以将待写入数据先插入关系型数据库,再根据关系型数据库来更新非关系型数据库,在待写入数据成功插入关系型数据库以及非关系型数据库时,向数据访问方返回数据插入成功的信息。

[0151] 上述数据存储方法中,当发生写数据操作时,先将数据写入一种数据库,再将数据从该数据库同步至另一种数据库,由于是数据库中的数据是同步得到的,可以保证两种数据库中数据的一致性,由于是基于写数据操作事件驱动数据的写入和同步,可以保证数据同步的实时性。关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力。在保证两种数据库中数据一致性的前提下,通过结合两种存储技术的优势,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

[0152] 在一个实施例中,待写入数据包括数据标识和目标数据内容;将待写入数据写入一种数据库包括:当写数据请求为数据更新请求时,根据目标数据内容对非关系型数据库中存储的与数据标识对应的原数据内容进行更新;将待写入数据从一种数据库同步至另一种数据库包括:在非关系型数据库更新完毕时,通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库。

[0153] 其中,数据更新请求是用于指示数据服务设备对数据库中的数据执行更新操作的数据请求。消息队列(messagequeuing,MQ)是一种应用程序对应用程序的通信方法。应用程序通过写和检索出入队列的针对应用程序的数据(消息)来通信,而无需专用连接来链接它们。消息传递指的是程序之间通过在消息中发送数据进行通信,而不是通过直接调用彼此来通信,直接调用通常是用于诸如远程过程调用的技术。排队指的是应用程序通过队列来通信。队列的使用除去了接收和发送应用程序同时执行的要求。

[0154] 具体地,参考图8,图8示出了一个实施例中响应数据更新请求在关系型数据库和非关系型数据库之间进行准实时同步的流程示意图。如图8所示,响应数据更新请求的步骤包括:

[0155] S802,当接收到数据更新请求时,数据服务设备首先根据数据更新请求携带的待写入数据的数据标识在非关系型数据库中确定需要更新的数据(以下称目标数据),以待写入数据中的目标数据内容替换该条数据的原数据内容。由于宕机等情况,可能导致目标数据在非关系型数据库中更新失败,此时向数据访问方反馈数据更新失败的信息。

[0156] S804,在完成对非关系型数据库中目标数据的更新后,数据服务设备根据待写入数据或其数据标识生成更新任务,将更新任务添加至MQ。

[0157] S806,数据服务设备执行更新任务,对关系型数据库的目标数据进行更新。比如,

将待写入数据加入MQ,数据服务设备按需从MQ中提取待写入数据进行消费,即根据提取到的待写入数据对关系型数据库中的目标数据进行更新。

[0158] 本实施例中,先更新非关系型数据库,再借助消息队列来异步更新关系型数据库,这种基于更新时间驱动的数据同步策略可以使关系型数据与非关系型数据库的同步只有微小的时间间隔,可以实现准实时同步。

[0159] 在一个实施例中,通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库包括:当待写入数据对应的入队标识为待入队时,将待写入数据的数据标识添加至消息队列,在非关系型数据库中将入队标识更新为排队中;按照排队顺序对消息队列中每个数据标识进行遍历;根据当前遍历顺序的数据标识在非关系型数据库中查询对应的目标数据内容,基于查询到的目标数据内容,对关系型数据库存储的与数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

[0160] 其中,入队标识是数据服务设备为执行本申请提供的数据库访问及存储方法在数据库中新增加的字段。每条数据具有对应的入队标识。为了避免高频更新产生大量入队操作导致超出MQ长度范围,从而丢失更新任务,本申请实施例引入入队标识字段,在非关系型数据库侧做标记。入队标识字段只需在非关系型数据库的数据表中添加。入队标识用于表征一条数据的更新任务是否位在消息队列中的状态信息,具体包括“待入队”和“排队中”。

[0161] 具体地,如图8所示,上述向消息队列中添加更新任务的步骤S804包括:数据服务设备确定非关系型数据库中待写入数据的数据标识对应的入队标识。在一个实施例中,入队标识也可以采用其他字符表征,如“待入队”可以用0表征,“排队中”可以用1表征。更新非关系型数据库后,数据服务设备检查待写入数据的入队标识字段是否为1。若入队标识为1则表示待写入数据对应的更新任务已在MQ中,不需要再次入队。若入队标识为0,则表示MQ中没有该条数据的更新任务,则需要加入MQ。数据服务设备只需将待写入数据的数据标识加入消息队列,并确定是否加入成功。当数据标识成功加入消息队列时,数据服务设备将更新结果确定为更新成功,向数据库访问方反馈数据更新成功的信息。

[0162] 如图8所示,上述消费消息队列中的更新任务,对关系型数据库中的数据进行更新的步骤S806包括:在将待写入数据对应的更新任务添加至消息队列后,数据服务设备在非关系型数据库中将入队标识更新为排队中。数据服务设备按照排队顺序对消息队列中每个更新任务进行遍历执行。当遍历至该待写入数据对应的更新任务时,数据服务设备首先根据待写入数据的数据标识,在关系型数据库中查询是否该数据标识对应的目标数据。若不存在则重试,若重试预设次数后仍无法查询到该数据标识对应的目标数据,则跳过该更新任务,并发出告警,继续执行下一顺序更新任务。当存在时,数据服务设备根据更新任务中的数据标识在非关系型数据库中查询对应的目标数据内容,在查询到的目标数据内容的数据版本高于关系型数据库中原数据内容的存储版本时,基于查询到的目标数据内容,对关系型数据库存储的与数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

[0163] 本实施例中,借助消息队列来异步更新关系型数据库,由于消息队列是不断处理队列的更新任务的,因而关系型数据库实际上可以做到准实时的数据更新同步,且由于消息队列中的更新任务是逐个处理的,不会因为请求量大导致关系型数据库压力过大。

[0164] 在一个实施例中,目标数据内容包括待写入数据的目标版本;根据目标数据内容

对非关系型数据库存储的与数据标识对应的原数据内容进行更新包括：确定非关系型数据库中存储的与数据标识对应的原数据内容的存储版本；当目标版本等于存储版本时，根据目标数据内容对非关系型数据库存储的与数据标识对应的原数据内容进行更新；在更新成功后，对存储版本进行更新。

[0165] 其中，数据版本是数据服务设备为执行本申请提供的数据库访问及存储方法在数据库中新增加的字段。每条数据具有对应的存储版本。为保证更新顺序的准确性，避免MQ更新关系型数据库存在顺序问题，对非关系型数据库和关系型数据库中为每条数据均加上数据版本字段，来记录每条数据的数据版本信息。数据版本字段在非关系型数据库和关系型数据库的数据表中均需添加。数据版本字段具体可以是版本号数值，如0,1等。读取出数据时，将数据版本号一同读出，之后更新时，对此数据版本号加一。

[0166] 数据服务设备根据通过数据版本字段构建乐观锁机制，基于乐观锁机制完成在非关系型数据库中的写操作。具体地，当根据数据更新请求对非关系型数据库中的目标数据进行更新时，数据服务设备对比非关系型数据库中当前存储的与待写入数据中数据标识对应的原数据内容的数据版本（以下称存储版本），只有待写入数据中目标数据内容的数据版本（以下称待写入数据版本）较新时才执行更新，并在更新或插入完成后对存储版本加一。当待写入数据版本低于存储版本时，确定写入结果为写入失败，向数据访问方返回数据写入失败的信息。

[0167] 本实施例中，提交的数据版本号等于数据库表当前版本号，则予以更新，否则认为是过期数据，这种乐观锁机制来保证写数据操作的准确性。

[0168] 在一个实施例中，待写入数据包括数据标识和目标数据内容；将待写入数据写入一种数据库包括：当写数据请求为插入数据请求时，确定数据标识是否已存储于关系型数据库中；若数据标识尚未存储于关系型数据库中，将待写入数据插入关系型数据库及非关系型数据库；若数据标识已存储于关系型数据库中，将待写入数据插入非关系型数据库，并在插入成功后，通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库。

[0169] 其中，数据插入请求是用于指示数据服务设备对数据库中的数据执行插入操作的数据请求。

[0170] 具体地，参考图9，图9示出了一个实施例中响应数据插入请求时在关系型数据库和非关系型数据库之间进行准实时同步的流程示意图。如图9所示，响应数据插入请求的步骤包括：

[0171] S902，当接收到数据插入请求时，数据服务设备首先根据数据插入请求携带的待写入数据的数据标识在关系型数据库中查询是否已经存在该数据标识。

[0172] S904，若关系型数据库中尚不存在该数据标识，表明该数据标识为首次插入，数据服务设备将待写入数据插入关系型数据库中，并初始化该待写入数据在关系型数据库的数据版本号为0。

[0173] 若待写入数据在关系型数据库中插入成功，在将待写入数据成功插入关系型数据库之后，数据服务设备将待写入数据插入非关系型数据库，并初始化该待写入数据在非关系型数据库的数据版本号为0。

[0174] 若待写入数据在关系型数据库中插入失败，或者在非关系型数据库中插入失败，

则数据服务设备向数据访问方返回插入失败的信息。

[0175] S906,若关系型数据库中已存在该数据标识,表明是重复插入关系型数据库,数据服务设备根据数据标识在非关系型数据库中查询是否已经存在该数据标识。

[0176] 若非关系型数据库中也已存在该数据标识,表明也是重复插入非关系型数据库,数据服务设备向数据访问方返回插入错误的提示信息。在插入操作中,由于需要插入两种数据库,任何一种数据库的插入失败均有可能导致数据访问方重复发起插入,而本实施例在关系型数据库中已存在该条数据时不再重复插入,当非关系型数据库也已存在该条数据时返回错误信息,如此可以保障数据不被重复插入以及数据的一致性。

[0177] 在一个实施例中,若数据标识已存储于关系型数据库中,将待写入数据插入非关系型数据库包括:若数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,将待写入数据插入非关系型数据库;若数据标识已存储于关系型数据库及非关系型数据库时,将写入成果确定为写入失败。

[0178] 若非关系型数据库中尚不存在该数据标识,表明是首次插入非关系型数据库,数据服务设备将待写入数据插入非关系型数据库中,并初始化该待写入数据在非关系型数据库的数据版本号为0。考虑到数据可能发生变化,数据服务设备在将待写入数据插入非关系型数据库后,往消息队列中添加待写入数据的更新任务,以对关系型数据库中已存储的对应目标数据进行更新。具体更新逻辑可参考上述实施例,在此不再赘述。

[0179] 在一个实施例中,如图9所示,上述数据存储方法还包括:当写数据请求为插入数据请求,且数据标识尚未存储于关系型数据库中,在将待写入数据插入非关系型数据库后,初始化待写入数据的入队标识为待入队。即当一条待写入数据为首次插入时,则按照正常更新逻辑,该待写入数据对应的更新任务等待入队。

[0180] 在一个实施例中,如图9所示,上述数据存储方法还包括:当数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,在将待写入数据插入非关系型数据库后,将待写入数据的数据标识添加至消息队列,并初始化待写入数据的入队标识为排队中。即当一条待写入数据为重复插入关系型数据库时,则将该待写入数据对应的更新任务立即添加至消息队列,提高数据同步实时性。

[0181] 本实施例中,由于非关系型数据库能够承受高频快速的插入操作,采用先插入关系型数据库,再同步插入非关系型数据库的插入操作同步策略,不再借助消息队列来将待写入数据异步插入非关系型数据库,如此非关系型数据无需承担异步操作存在的同步失败的风险。

[0182] 在一个实施例中,上述数据存储方法还包括:对关系型数据库中每条数据进行遍历;在当前遍历顺序数据的数据标识未存储在非关系型数据库中时,将当前遍历顺序的数据插入非关系型数据库;在当前遍历顺序数据的数据标识存储在非关系型数据库中,但对应的数据内容不一致时,根据非关系型数据库中的数据内容对关系型数据库中的数据内容进行更新。

[0183] 基于上述写操作准实时同步策略,基本可以保证两种数据库中数据的一致性。如图7所示,为了避免出现一些极端错误(如数据插入非关系型数据库持续失败,或消息队列更新重试多次依然失败等)带来数据不一致的问题,数据服务设备基于预设的兜底对账机制来保障两种数据库中数据的最终一致性。

[0184] 具体地,参考图10,图10示出了一个实施例中在关系型数据库和非关系型数据库中进行兜底对账的流程示意图。如图10所示,对关系型数据库和非关系型数据库中的数据
进行兜底对账的步骤包括:

[0185] S1002,数据服务设备按照目标时间频率对关系型数据库中的每条数据进行遍历,对遍历到的每条数据的数据标识是否同时存储在非关系型数据库中进行验查询。

[0186] 在一个实施例中,上述数据存储方法还包括:按照预设时间频率统计关系型数据库所存储全部数据的总数据量;根据总数据量确定进行遍历的目标时间频率;对关系型数据库中每条数据进行遍历包括:按照目标时间频率对关系型数据库中每条数据进行遍历。

[0187] 该兜底对账机制通过脚本每隔一定时间(如30分钟)进行一次。兜底对账脚本执行频率(即目标时间频率)可以是根据关系型数据库中所存储数据的总数据量动态确定的。可以理解,目标时间频率与总数据量正相关。

[0188] S1004,若当前遍历顺序的一条数据的数据标识未存储在非关系型数据库中,则数据服务设备在非关系型数据库中新增该条数据,即将当前遍历顺序的数据插入非关系型数据库中,并初始化新增数据的数据版本为0。

[0189] S1006,若当前遍历顺序的一条数据的数据标识存储在非关系型数据库中,则数据服务设备比对该数据标识在关系型数据库中的数据内容与非关系型数据库中的数据内容是否一致。若一致,则继续遍历下一顺序数据。若不一致,则数据服务设备比对该数据标识在非关系型数据库中数据内容的数据版本是否高于关系型数据库中数据内容的数据版本。若是,数据服务设备将该数据标识在非关系型数据库中的数据内容替换关系型数据库中的数据内容,并将该数据标识对应数据在非关系型数据库中的入队标识配置为待入队。兜底对账过程中进行数据更新完全以非关系型数据库中的数据为准,不参考数据版本。

[0190] 值得说明的是,正是这种兜底对账机制,可以保证两种数据库中数据的最终一致性。因而,如图8所示,在步骤S804中,即便在数据标识未成功加入消息队列时,依然可以将更新结果确定为更新成功。

[0191] 本实施例中,在两种数据库之间进行准实时同步后,进一步进行兜底对账,可以避免出现一些极端错误带来数据不一致的问题,达到数据的最终一致性。

[0192] 在一个具体地实施例中,如图11所示,本申请提供的数据存储方法包括:

[0193] S1102,在接收到写数据请求时,对写数据请求进行解析,得到待写入数据;待写入数据包括数据标识和目标数据内容;目标数据内容包括待写入数据的目标版本。

[0194] S1104,当写数据请求为插入数据请求时,确定数据标识是否已存储于用于响应符合预设复杂访问条件的读数据请求的关系型数据库中。

[0195] S1106,若数据标识尚未存储于关系型数据库中,将待写入数据插入关系型数据库及非关系型数据库,初始化待写入数据的入队标识为待入队,初始化待写入数据的存储版本为0。非关系型数据库用于响应符合预设高频访问条件的读数据请求。

[0196] S1108,若数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,将待写入数据插入非关系型数据库;并在插入成功后,将待写入数据的数据标识立即添加至消息队列,并初始化待写入数据的入队标识为排队中,初始化待写入数据的存储版本为0。按照排队顺序对消息队列中每个数据标识进行遍历;根据当前遍历顺序的数据标识在非关系型数据库中查询对应的目标数据内容,基于查询到的目标数据内容,对关系型数据库

存储的与数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

[0197] S1110,若数据标识已存储于关系型数据库及非关系型数据库时,将写入成果确定为写入失败。

[0198] S1112,当写数据请求为数据更新请求时,确定非关系型数据库中存储的与数据标识对应的原数据内容的存储版本。

[0199] S1114,当目标版本等于存储版本时,根据目标数据内容对非关系型数据库存储的与数据标识对应的原数据内容进行更新;在更新成功后,将存储版本更新为目标版本。

[0200] S1116,当待写入数据版本低于存储版本时,确定写入结果为写入失败。

[0201] S1118,在非关系型数据库更新完毕,且待写入数据对应的入队标识为待入队时,将待写入数据的数据标识添加至消息队列,在非关系型数据库中将入队标识更新为排队中;按照排队顺序对消息队列中每个数据标识进行遍历;根据当前遍历顺序的数据标识在非关系型数据库中查询对应的目标数据内容,基于查询到的目标数据内容,对关系型数据库存储的与数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

[0202] S1120,基于待写入数据的插入结果响应数据插入请求。

[0203] S1122,获取对主库执行写操作时生成的日志。

[0204] S1124,将日志发送至一个或多个目标的从库,使目标的从库根据日志同步执行写操作。

[0205] S1126,当接收到目标的从库在执行写操作后返回的同步确认信息时,基于写入成功的结果来响应用于触发写操作的写数据请求,向其余从库发送日志,使其余从库根据日志同步执行写操作。

[0206] S1128,对关系型数据库中每条数据进行遍历。

[0207] S1130,在当前遍历顺序数据的数据标识未存储在非关系型数据库中时,将当前遍历顺序的数据插入非关系型数据库。

[0208] S1132,在当前遍历顺序数据的数据标识存储在非关系型数据库中,但对应的数据内容不一致时,根据非关系型数据库中的数据内容对关系型数据库中的数据内容进行更新。

[0209] 上述数据存储方法中,综合使用两种存储方式,结合两者优点,同时提供数据高可用保障,并设计了一种准实时且有最终一致性保障的数据存储访问方案。该方案能满足业务高并发场景下的数据访问需求,解决传统关系型数据库的性能问题;对于复杂查询等场景又能提供低频的复杂查询方案,解决非关系型数据库在复杂查询场景下的劣势;对于数据的准确性与一致性,有可靠的数据同步方案来满足,并有兜底对账脚本来保障数据的最终一致性。极大地提高了数据层对业务的支撑能力,解决数据访问上的瓶颈,为业务的稳步发展提供坚强有力的基础。

[0210] 当发生写数据操作时,先将数据写入一种数据库,再将数据从该数据库同步至另一种数据库,由于是数据库中的数据是同步得到的,可以保证两种数据库中数据的一致性,由于是基于写数据操作事件驱动数据的写入和同步,可以保证数据同步的实时性。关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非

关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力。在保证两种数据库中数据一致性的前提下,通过结合两种存储技术的优势,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

[0211] 应该理解的是,虽然图6以及图8至图11的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,图6以及图8至图11中的至少一部分步骤可以包括多个步骤或者多个阶段,这些步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤中的步骤或者阶段的至少一部分轮流或者交替地执行。

[0212] 在一个实施例中,如图12所示,提供了一种数据访问装置1200,该装置可以采用软件模块或硬件模块,或者是二者的结合成为计算机设备的一部分,该装置具体包括:数据读取请求模块1202、数据查询模块1204和查询结果反馈模块1206,其中:

[0213] 数据读取请求模块1202,用于在接收到读数据请求时,对读数据请求进行解析,得到查询条件。

[0214] 数据查询模块1204,用于当查询条件符合预设高频访问条件时,在非关系型数据库中查询符合查询条件的数据;当查询条件符合预设复杂访问条件时,在关系型数据库中查询符合查询条件的数据;非关系型数据库与关系型数据库所存储数据一致。

[0215] 查询结果反馈模块1206,用于基于查询到的数据响应读数据请求。

[0216] 在一个实施例中,如图13所示,上述数据访问装置1200还包括访问场景识别模块1208,用于当查询条件为预设的路由字段时,判定查询条件符合预设高频访问条件;当查询条件包括除路由字段之外的查询字段时,判定查询条件符合预设复杂访问条件。

[0217] 在一个实施例中,非关系型数据库包括多个分布式子库;如图13所示,数据查询模块1204包括高频访问查询模块12042,用于当查询条件为路由字段时,确定查询字段所属的分布式区间;在用于存储分布式区间内的数据的分布式子库中,查询符合查询条件的数据。

[0218] 在一个实施例中,非关系型数据库具有一个或多个;高频访问查询模块12042还用于当查询条件为路由字段时,在将路由字段作为分布式存储依据的非关系型数据库中,查询符合查询条件的数据。

[0219] 在一个实施例中,关系型数据库包括主库和至少一个从库;如图13所示,数据查询模块1204还包括复杂访问查询模块12044,用于基于查询条件向主库发起查询请求;当在预设时长内未接收到主库针对查询请求的查询应答时,将其中一个从库确定为当前的主库;在当前的主库中查询符合查询条件的数据。

[0220] 在一个实施例中,如图13所示,上述数据访问装置1200还包括半同步复制模块1210,用于获取对主库执行写操作时生成的日志;将日志发送至一个或多个目标的从库,使目标的从库根据日志同步执行写操作;当接收到目标的从库在执行写操作后返回的同步确认信息时,基于写入成功的结果来响应用于触发写操作的写数据请求,向其余从库发送日志。

[0221] 上述数据访问装置,关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力;通过结合两种存储技术的优势并保证两种数据库中数据的一致性,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

[0222] 关于数据访问装置的具体限定可以参见上文中对于数据访问方法的限定,在此不再赘述。上述数据访问装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中,也可以以软件形式存储于计算机设备中的存储器中,以便于处理器调用执行以上各个模块对应的操作。

[0223] 在一个实施例中,如图14所示,提供了一种数据存储装置1400,该装置可以采用软件模块或硬件模块,或者是二者的结合成为计算机设备的一部分,该装置具体包括:数据写入请求模块1402、数据准实时同步模块1404和写入结果反馈模块1406,其中:

[0224] 数据写入请求模块1402,用于在接收到写数据请求时,对写数据请求进行解析,得到待写入数据。

[0225] 数据准实时同步模块1404,用于将待写入数据写入一种数据库;将待写入数据从一种数据库同步至另一种数据库;其中,数据库的种类包括关系型数据库和非关系型数据库;非关系型数据库用于响应符合预设高频访问条件的读数据请求,关系型数据库用于响应符合预设复杂访问条件的读数据请求。

[0226] 写入结果反馈模块1406,用于基于待写入数据的写入结果响应写数据请求。

[0227] 在一个实施例中,待写入数据包括数据标识和目标数据内容;如图15所示,数据准实时同步模块1404包括数据更新同步模块14042,用于当写数据请求为数据更新请求时,根据目标数据内容对非关系型数据库中存储的与数据标识对应的原数据内容进行更新;将待写入数据从一种数据库同步至另一种数据库包括:在非关系型数据库更新完毕时,通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库。

[0228] 在一个实施例中,目标数据内容包括待写入数据的目标版本;数据更新同步模块14042还用于确定非关系型数据库中存储的与数据标识对应的原数据内容的存储版本;当目标版本等于存储版本时,根据目标数据内容对非关系型数据库存储的与数据标识对应的原数据内容进行更新;在更新成功后,将存储版本更新为目标版本。当待写入数据版本低于存储版本时,确定写入结果为写入失败。

[0229] 在一个实施例中,待写入数据包括数据标识和目标数据内容;如图15所示,数据准实时同步模块1404还包括数据插入同步模块14044,用于当写数据请求为插入数据请求时,确定数据标识是否已存储于关系型数据库中;若数据标识尚未存储于关系型数据库中,将待写入数据插入关系型数据库及非关系型数据库;若数据标识已存储于关系型数据库中,将待写入数据插入非关系型数据库,并在插入成功后,通过消息队列将目标数据内容从非关系型数据库异步更新至关系型数据库。

[0230] 在一个实施例中,数据插入同步模块14044还用于若数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,将待写入数据插入非关系型数据库;若数据标识

已存储于关系型数据库及非关系型数据库时,将写入成果确定为写入失败。

[0231] 在一个实施例中,数据更新同步模块14042还用于当待写入数据对应的入队标识为待入队时,将待写入数据的数据标识添加至消息队列,在非关系型数据库中将入队标识更新为排队中;按照排队顺序对消息队列中每个数据标识进行遍历;根据当前遍历顺序的数据标识在非关系型数据库中查询对应的目标数据内容,基于查询到的目标数据内容,对关系型数据库存储的与数据标识对应的原数据内容进行更新,将当前遍历顺序的入队标识更新为待入队。

[0232] 在一个实施例中,上述数据存储装置1400还包括数据异步更新模块1408,用于当写数据请求为插入数据请求,且数据标识尚未存储于关系型数据库中,在将待写入数据插入非关系型数据库后,初始化待写入数据的入队标识为待入队;当数据标识已存储于关系型数据库中,但尚未存储于非关系型数据库时,在将待写入数据插入非关系型数据库后,将待写入数据的数据标识添加至消息队列,并初始化待写入数据的入队标识为排队中。

[0233] 在一个实施例中,如图15所示,上述数据存储装置1400还包括数据兜底对账模块1410,用于对关系型数据库中每条数据进行遍历;在当前遍历顺序数据的数据标识未存储在非关系型数据库中时,将当前遍历顺序的数据插入非关系型数据库;在当前遍历顺序数据的数据标识存储在非关系型数据库中,但对应的数据内容不一致时,根据非关系型数据库中的数据内容对关系型数据库中的数据内容进行更新。

[0234] 上述数据存储装置,当发生写数据操作时,先将数据写入一种数据库,再将数据从该数据库同步至另一种数据库,由于是数据库中的数据是同步得到的,可以保证两种数据库中数据的一致性,由于是基于写数据操作事件驱动数据的写入和同步,可以保证数据同步的实时性。关系型数据库支持低并发的数据访问操作,能够提供根据复杂查询条件进行数据筛选的能力;非关系型数据库支持高并发的数据访问操作,能够根据简单查询条件进行数据筛选的能力。在保证两种数据库中数据一致性的前提下,通过结合两种存储技术的优势,根据数据访问场景区分查询数据的来源,即高并发的数据访问场景从非关系型数据库中查询数据,而并发度不高但查询条件复杂的数据访问场景则可以从关系型数据库中查询数据,从而既能满足高并发的数据访问请求,也能够满足复杂的数据访问请求,提高数据访问性能。

[0235] 关于数据存储装置的具体限定可以参见上文中对于数据存储方法的限定,在此不再赘述。上述数据存储装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中,也可以以软件形式存储于计算机设备中的存储器中,以便于处理器调用执行以上各个模块对应的操作。

[0236] 在一个实施例中,提供了一种计算机设备,该计算机设备可以是服务器,其内部结构图可以如图16所示。该计算机设备包括通过系统总线连接的处理器、存储器和网络接口。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库包括关系型数据库和非关系型数据库,所存储的数据一致。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种数据访问及存储方法。

[0237] 本领域技术人员可以理解,图16中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0238] 在一个实施例中,还提供了一种计算机设备,包括存储器和处理器,存储器中存储有计算机程序,该处理器执行计算机程序时实现上述各方法实施例中的步骤。

[0239] 在一个实施例中,提供了一种计算机可读存储介质,存储有计算机程序,该计算机程序被处理器执行时实现上述各方法实施例中的步骤。

[0240] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,的计算机程序可存储于一非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和易失性存储器中的至少一种。非易失性存储器可包括只读存储器(Read-Only Memory,ROM)、磁带、软盘、闪存或光存储器等。易失性存储器可包括随机存取存储器(Random Access Memory,RAM)或外部高速缓冲存储器。作为说明而非局限,RAM可以是多种形式,比如静态随机存取存储器(Static Random Access Memory,SRAM)或动态随机存取存储器(Dynamic Random Access Memory,DRAM)等。

[0241] 以上实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0242] 以上实施例仅表达了本申请的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本申请构思的前提下,还可以做出若干变形和改进,这些都属于本申请的保护范围。因此,本申请专利的保护范围应以所附权利要求为准。

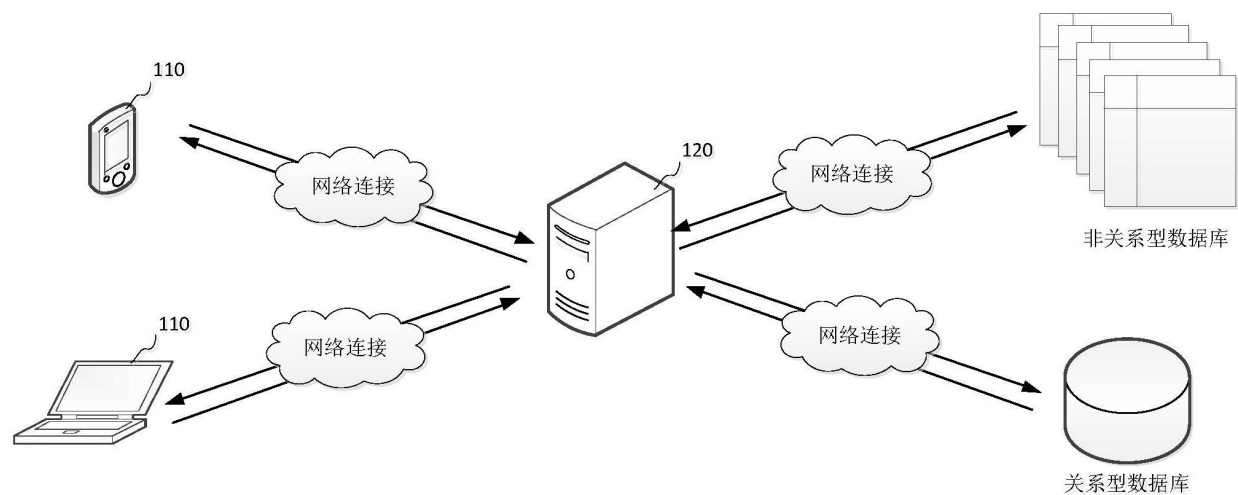


图1

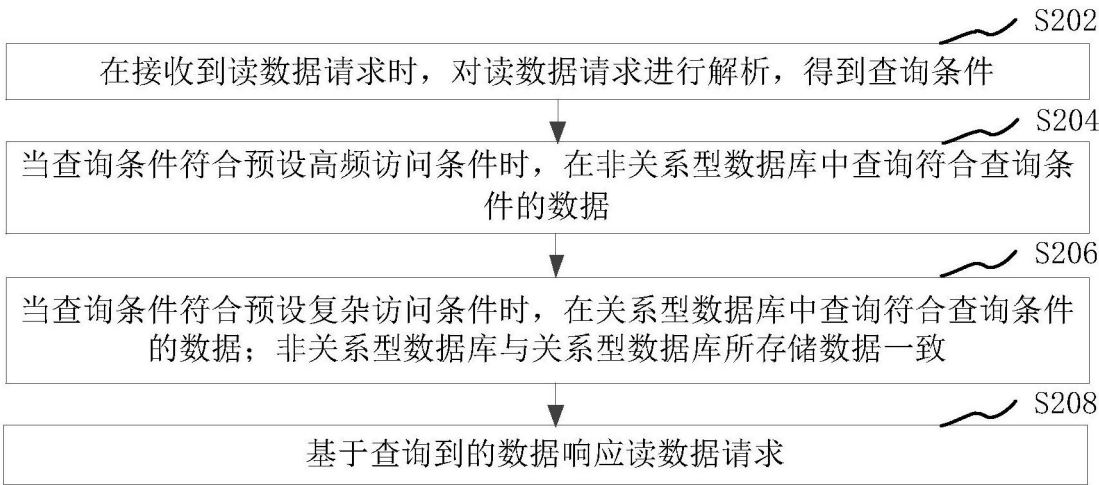


图2

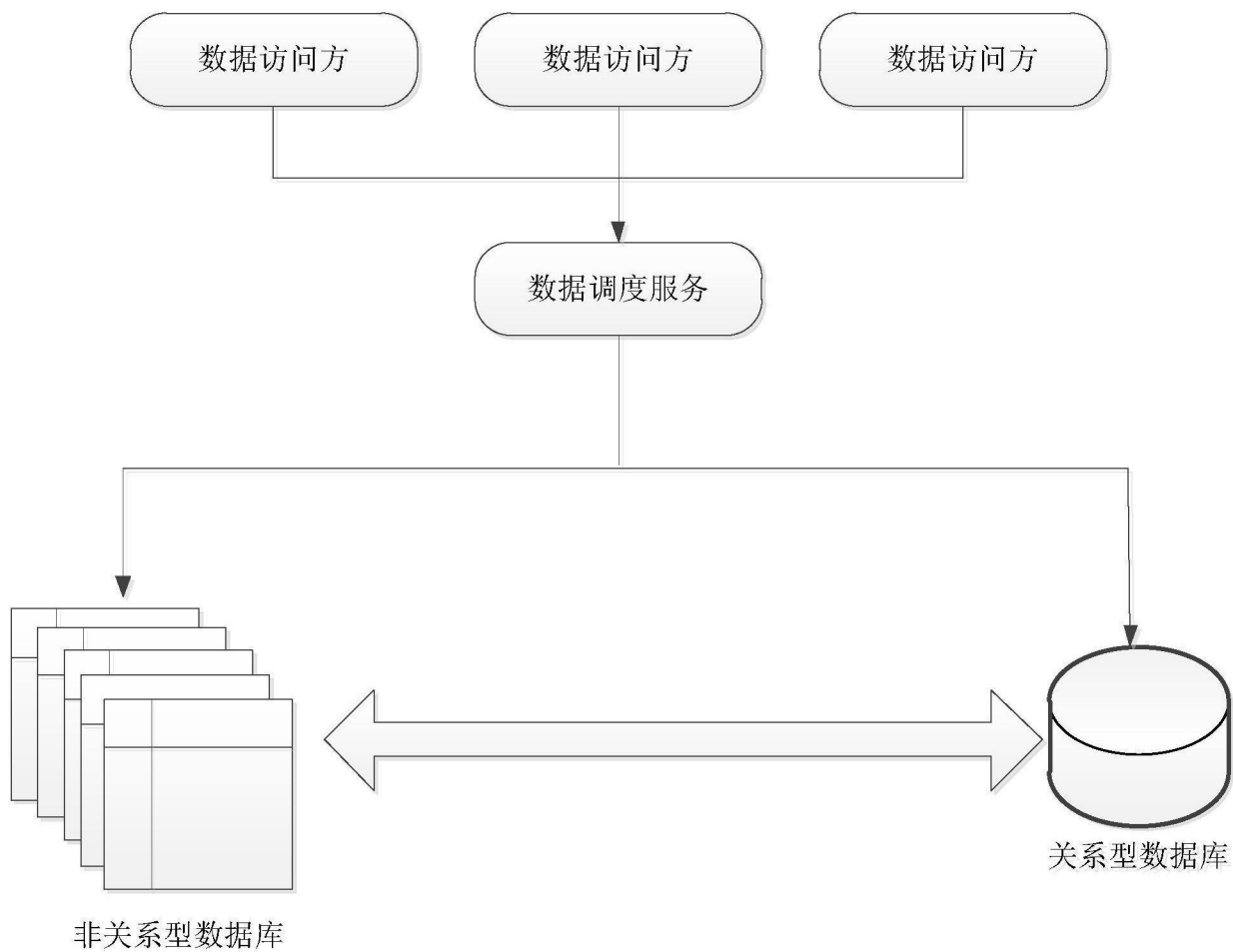


图3

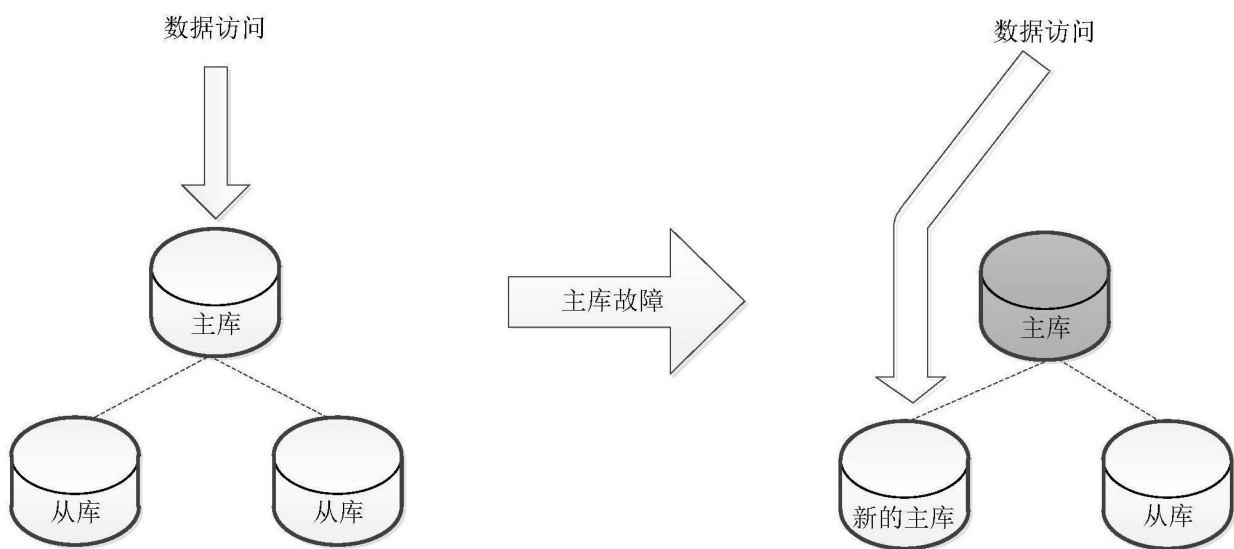


图4

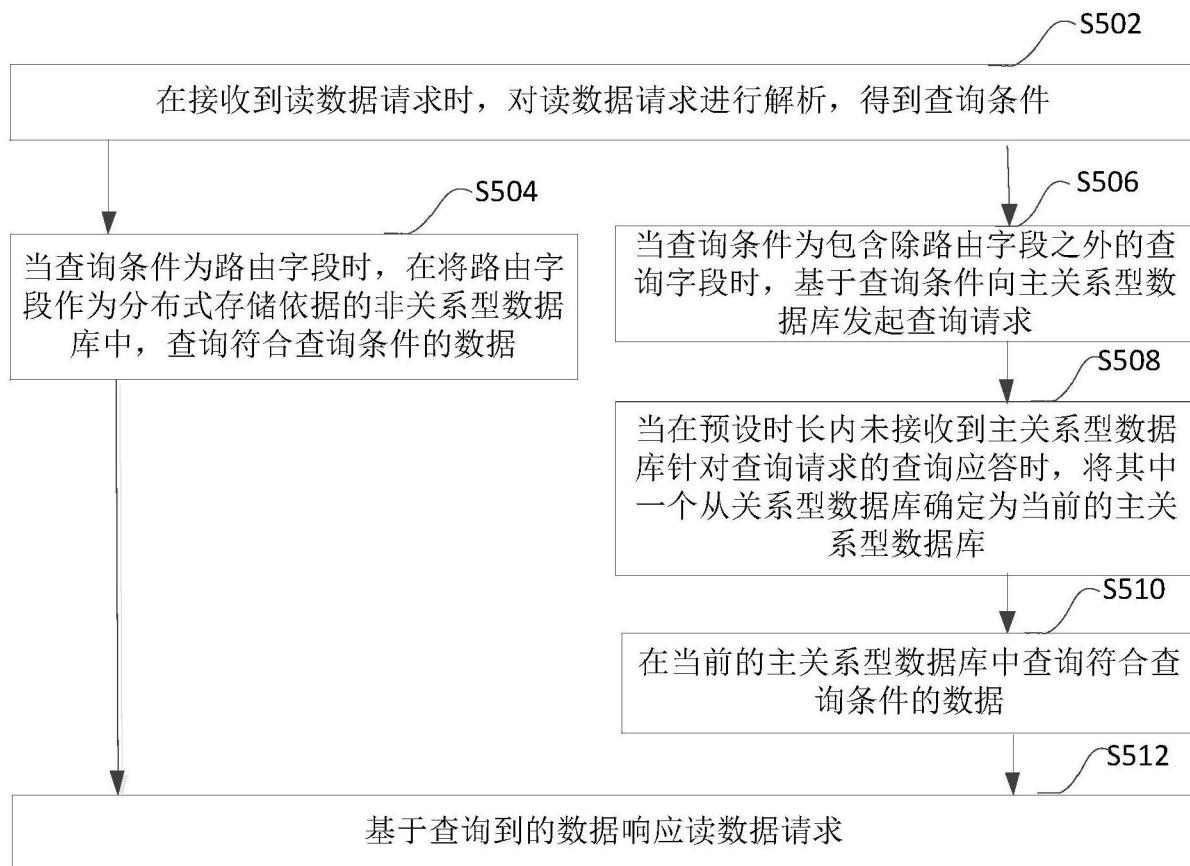


图5

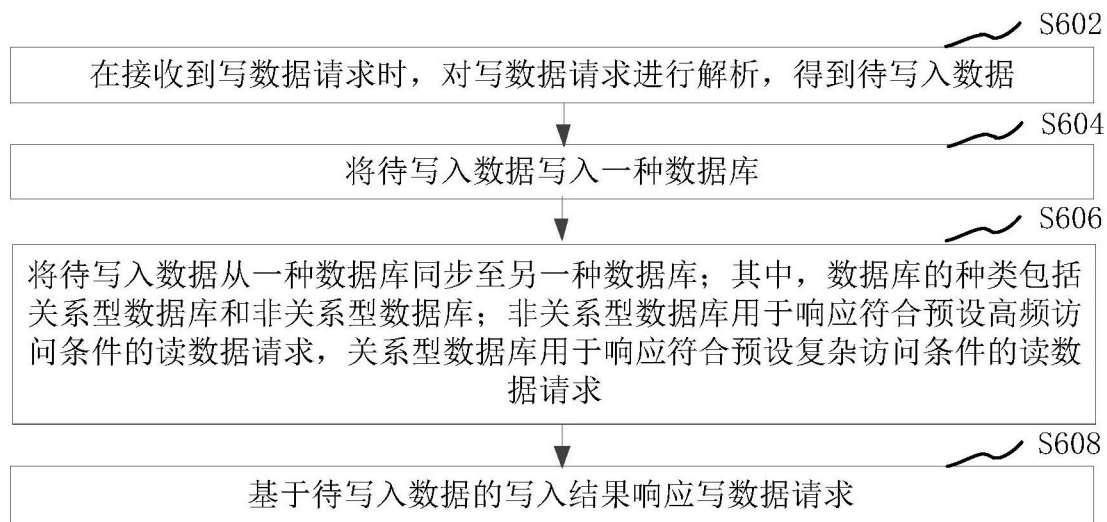


图6



图7

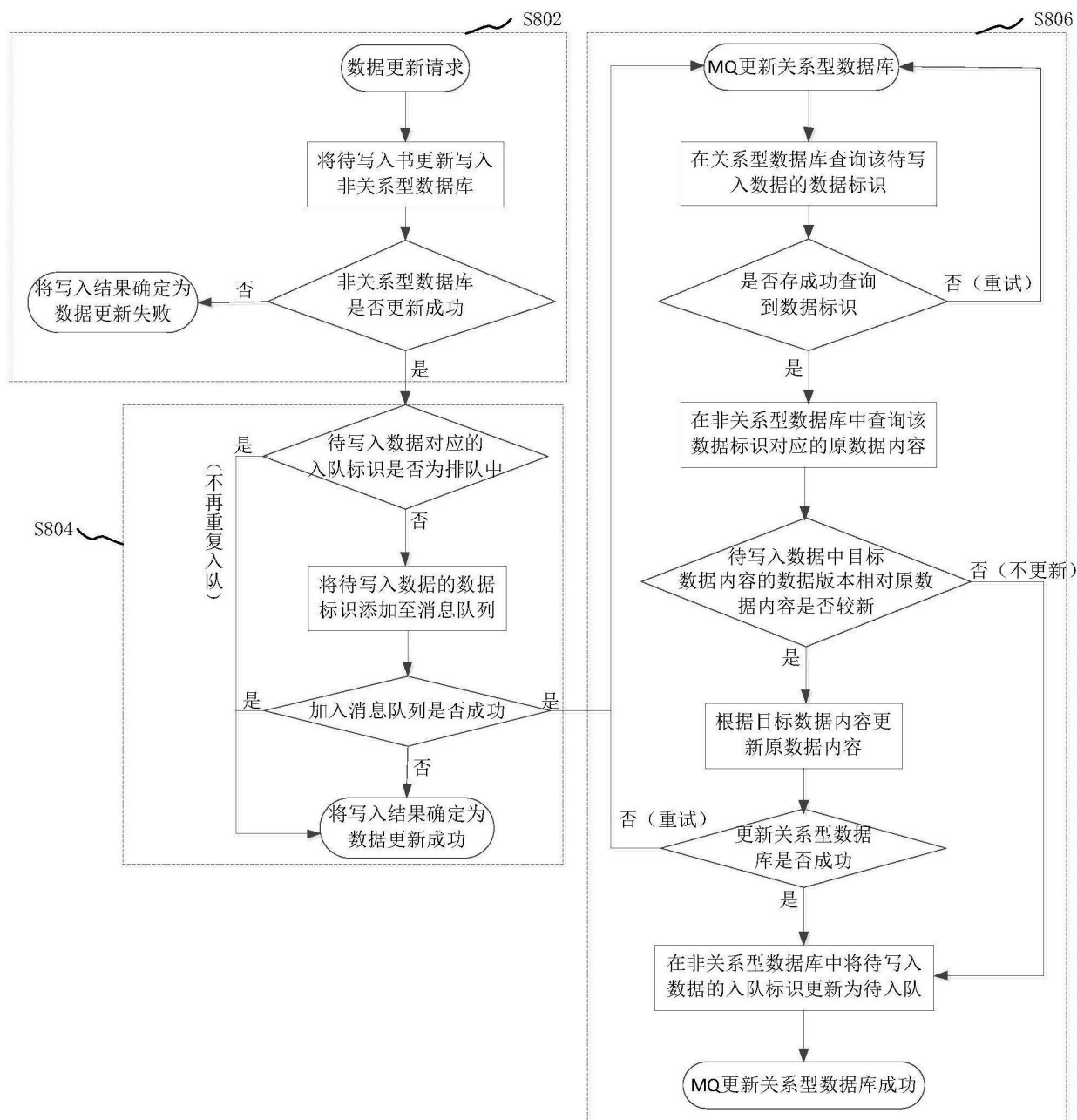


图8

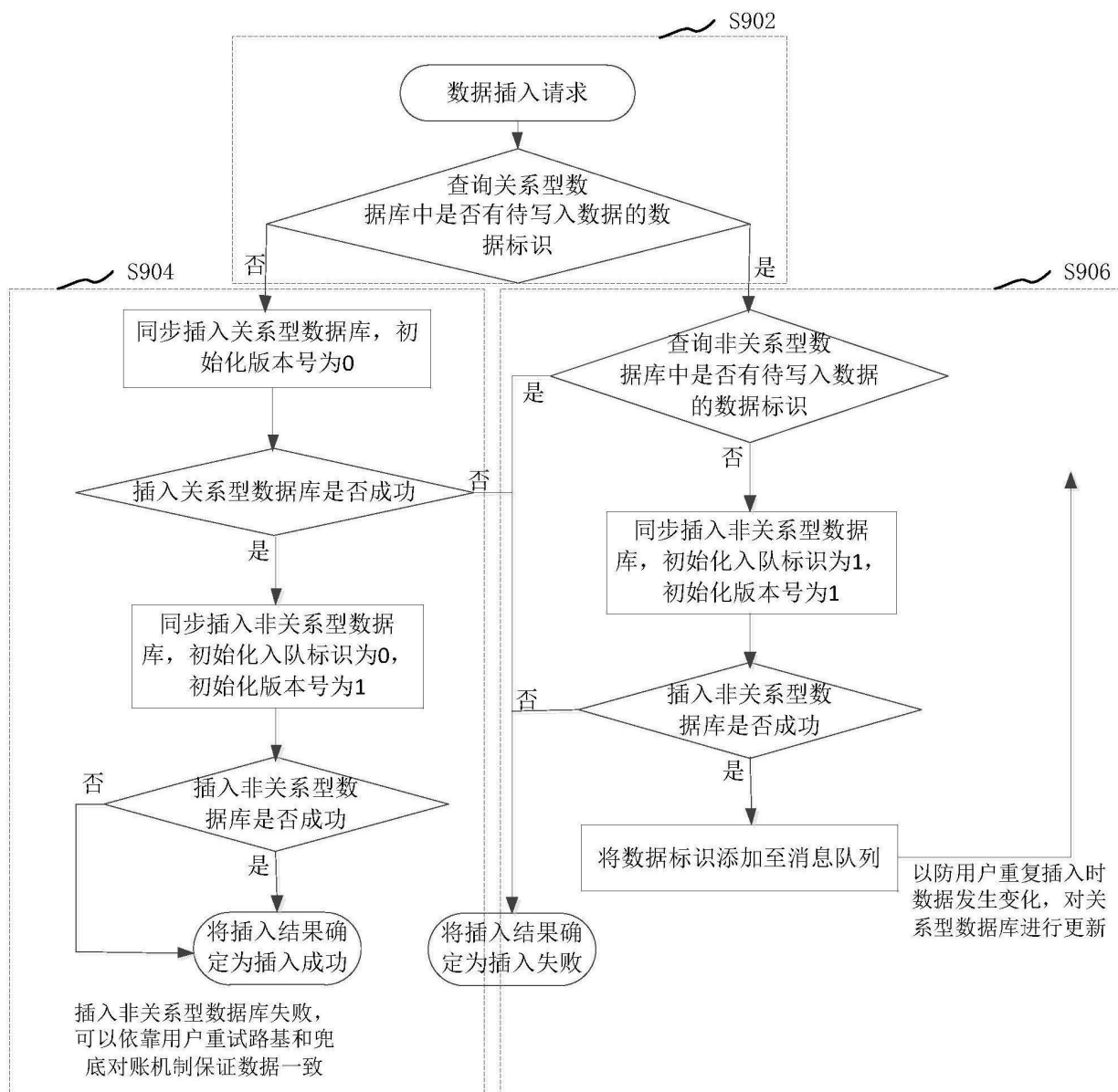


图9

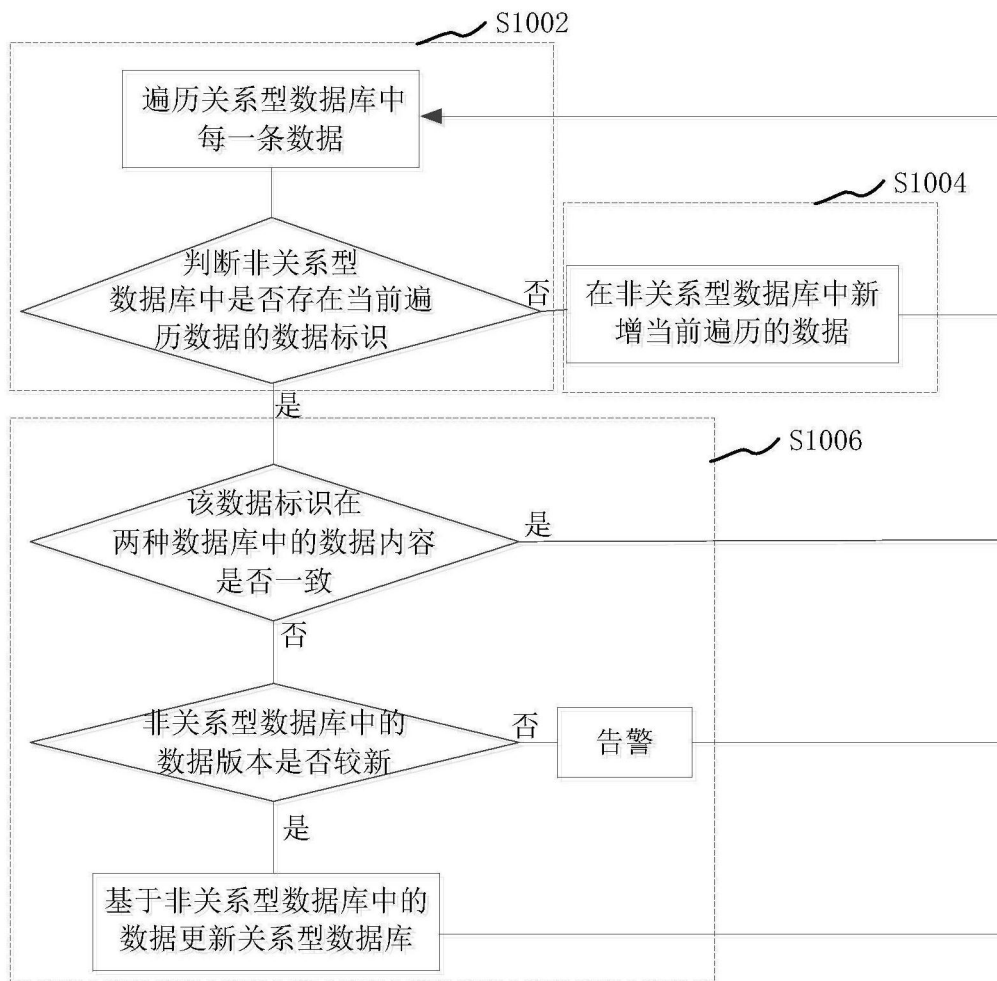


图10



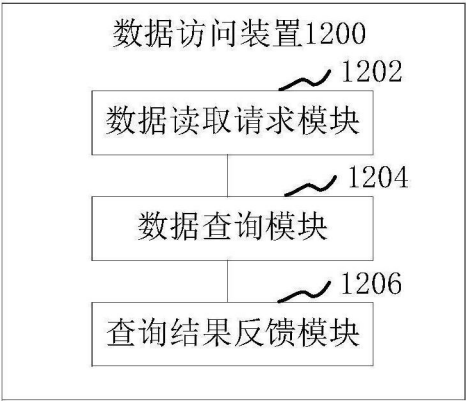


图12

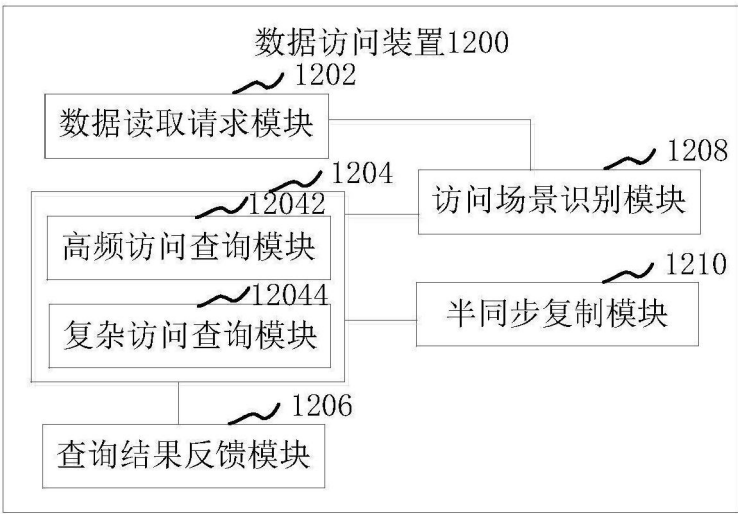


图13



图14

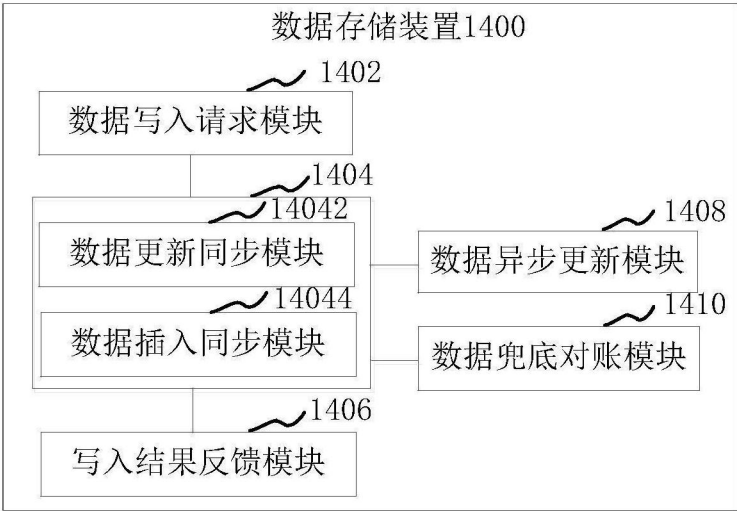


图15

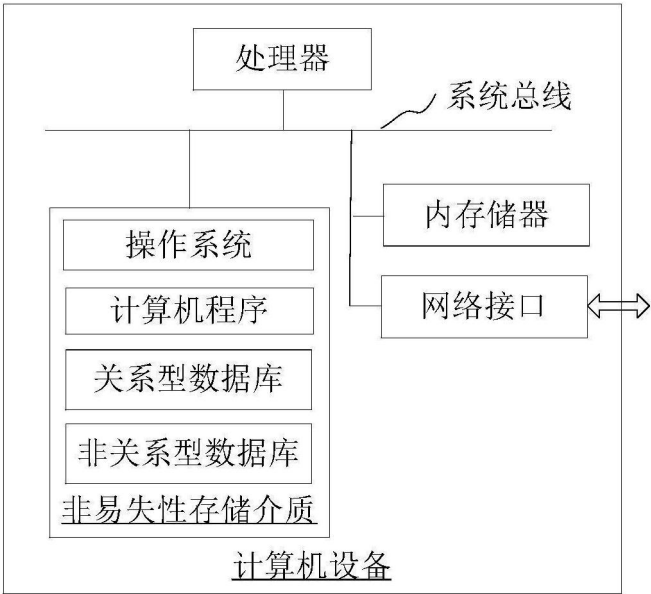


图16