(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 110427251 A
(43)申请公布日 2019.11.08

(21)申请号 201910678639.8

(22)申请日 2019.07.25

(71)申请人 吉旗(成都)科技有限公司
地址 610000 四川省成都市天府新区天府大道南段846号

(72)发明人 冯磊

(74)专利代理机构 北京棘龙知识产权代理有限公司 11740
代理人 谢静

(51)Int.Cl.
G06F 9/46(2006.01)
G06F 16/215(2019.01)
G06F 16/27(2019.01)

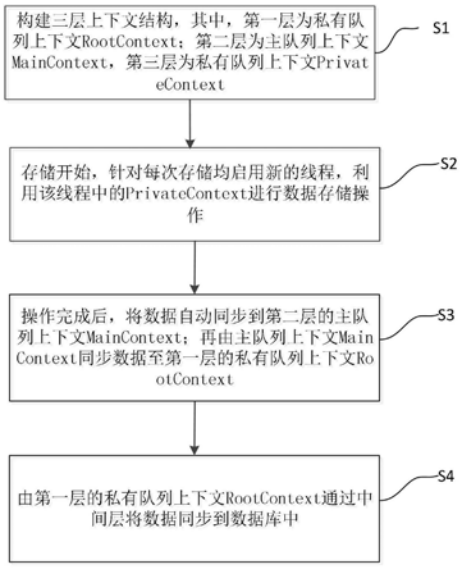
权利要求书1页 说明书4页 附图4页

(54)发明名称

一种针对iOS CoreData数据库存储的优化方法

(57)摘要

本发明提出了一种针对iOS CoreData数据库存储的优化方法,包括:构建三层上下文结构,其中,第一层为私有队列上下文RootContext,该层私有队列上下文RootContext与中间层连接,中间层进一步与数据库连接;第二层为主队列上下文MainContext,第三层为私有队列上下文PrivateContext;存储开始,针对每次存储均启用新的线程,利用该线程中的PrivateContext进行数据存储操作;操作完成后,将数据自动同步到第二层的主队列上下文MainContext;再由主队列上下文MainContext同步数据至第一层的私有队列上下文RootContext;由第一层的私有队列上下文RootContext通过中间层将数据同步到数据库中。本发明通过构建三层上下文结构,使得数据存储都是异步执行,不会对主线程造成卡顿,并且可以同时批量操作数据,提高数据的存储效率,保证线程安全。



CN 110427251 A

1.一种针对iOS CoreData数据库存储的优化方法,其特征在于,包括如下步骤:

步骤S1,构建三层上下文结构,其中,第一层为私有队列上下文RootContext,该层私有队列上下文RootContext与中间层连接,所述中间层进一步与数据库连接;第二层为主队列上下文MainContext,第三层为私有队列上下文PrivateContext,该层私有队列上下文PrivateContext包括:多个线程的私有队列上下文,并且保证每个线程独享一个上下文;

步骤S2,存储开始,针对每次存储均启用新的线程,利用该线程中的PrivateContext进行数据存储操作;

步骤S3,操作完成后,将数据自动同步到第二层的主队列上下文MainContext;再由主队列上下文MainContext同步数据至第一层的私有队列上下文RootContext;

步骤S4,由第一层的私有队列上下文RootContext通过中间层将数据同步到数据库中。

2.如权利要求1所述的针对iOS CoreData数据库存储的优化方法,其特征在于,在所述步骤S1中,利用所述中间层对所述第一层为私有队列上下文RootContext和数据库文件中间进行数据同步。

3.如权利要求1所述的针对iOS CoreData数据库存储的优化方法,其特征在于,在所述步骤S1中,每个所述线程仅持有一个私有队列上下文PrivateContext。

4.如权利要求1所述的针对iOS CoreData数据库存储的优化方法,其特征在于,在所述步骤S2中,针对每次存储均启用新的线程,首先检测该线程中是否已经存在私有队列上下文PrivateContext,如果有则利用该私有队列上下文PrivateContext进行数据操作;如果线程中没有私有队列上下文PrivateContext,再生成新的私有队列上下文PrivateContext,并将该新的私有队列上下文PrivateContext关联到新的线程上,利用该私有队列上下文PrivateContext进行数据操作。

一种针对iOS CoreData数据库存储的优化方法

技术领域

[0001] 本发明涉及iOS CoreData数据库技术领域,特别涉及一种针对iOS CoreData数据库存储的优化方法。

背景技术

[0002] Core Data是iOS编程中使用持久化数据存储的一种方式。在一些情况下,可以使用Core Data来持久化数据。需要知道的是,Core Data并不是数据库本身,是Apple提供的对象持久化技术(Object Persistent Technology)。Core Data框架为数据变更管理、对象存储、对象读取恢复的功能提供了支持。它可以使用SQLite作为持久化存储的类型,但本身并不是一个数据库。

[0003] 但是,现有的iOS CoreData数据库存储效率较低,在主线程使用主队列上下文存储会有阻塞现场,造成UI卡顿。如果使用私有队列异步存储,又不是线程安全的。传统CoreData存储层级设计如图4所示。全局使用一个主队列存储上下文。如果在主线程遇到大量数据存储时,会造成严重的UI卡顿。

发明内容

[0004] 本发明的目的旨在至少解决所述技术缺陷之一。

[0005] 为此,本发明的目的在于提出一种针对iOS CoreData数据库存储的优化方法。

[0006] 为了实现上述目的,本发明的实施例提供一种针对iOS CoreData数据库存储的优化方法,包括如下步骤:

[0007] 步骤S1,构建三层上下文结构,其中,第一层为私有队列上下文RootContext,该层私有队列上下文RootContext与中间层连接,所述中间层进一步与数据库连接;第二层为主队列上下文MainContext,第三层为私有队列上下文PrivateContext,该层私有队列上下文PrivateContext包括多个线程的私有队列上下文,并且保证每个线程独享一个上下文;

[0008] 步骤S2,存储开始,针对每次存储均启用新的线程,利用该线程中的PrivateContext进行数据存储操作;

[0009] 步骤S3,操作完成后,将数据自动同步到第二层的主队列上下文MainContext;再由主队列上下文MainContext同步数据至第一层的私有队列上下文RootContext;

[0010] 步骤S4,由第一层的私有队列上下文RootContext通过中间层将数据同步到数据库中。

[0011] 进一步,在所述步骤S1中,利用所述中间层对所述第一层为私有队列上下文RootContext和数据库文件中间进行数据同步。

[0012] 进一步,在所述步骤S1中,每个所述线程仅持有一个私有队列上下文PrivateContext。

[0013] 进一步,在所述步骤S2中,针对每次存储均启用新的线程,首先检测该线程中是否已经存在私有队列上下文PrivateContext,如果有则利用该私有队列上下文

PrivateContext进行数据操作;如果线程中没有私有队列上下文PrivateContext,再生成新的私有队列上下文PrivateContext,并将该新的私有队列上下文PrivateContext关联到新的线程上,利用该私有队列上下文PrivateContext进行数据操作。

[0014] 根据本发明实施例的针对iOS CoreData数据库存储的优化方法,通过将私有队列上下文和主队列上下文两种存储模式混合使用,构建三层上下文结构,提供一种线程安全的高效存储方法,可以优化iOS CoreData数据库存储效率,支持多线程并发存储,保证线程安全。本发明通过构建三层上下文结构,使得数据存储都是异步执行,不会对主线程(UI)造成卡顿,并且可以同时批量操作数据,提高数据的存储效率,保证线程安全。

[0015] 本发明附加的方面和优点将在下面的描述中部分给出,部分将从下面的描述中变得明显,或通过本发明的实践了解到。

附图说明

[0016] 本发明的上述和/或附加的方面和优点从结合下面附图对实施例的描述中将变得明显和容易理解,其中:

[0017] 图1为根据本发明实施例的针对iOS CoreData数据库存储的优化方法的流程图;

[0018] 图2为根据本发明实施例的上下文层次架构图;

[0019] 图3为根据本发明实施例的数据存储的流程图;

[0020] 图4为传统CoreData存储层级设计示意图。

具体实施方式

[0021] 下面详细描述本发明的实施例,所述实施例的示例在附图中示出,其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的,旨在用于解释本发明,而不能理解为对本发明的限制。

[0022] 本发明提供一种针对iOS CoreData数据库存储的优化方法,该方法对iOS CoreData数据存储提供了两种存储模式:主队列存储、私有队列存储。由于主队列存储会存在卡顿UI的问题;私有队列存储不存在卡顿UI,但不是线程安全的。本发明通过将两种存储模式混合使用,搭建出一种线程安全的高效存储方案。

[0023] 下面对本发明中的上下文、主队列上下文和私有队列上下文进行说明:

[0024] (1)上下文:对应NSManagedObjectContext类的实例,用于查询,修改,删除,增加数据库中的数据。

[0025] (2)主队列上下文:对数据库的操作都在主队列中完成。如果耗时长,会造成明显的UI卡顿。

[0026] (3)私有队列上下文:对数据库的操作在私有队列中完成,不会卡顿UI,但不是线程安全的。

[0027] 如图1所示,本发明实施例的针对iOS CoreData数据库存储的优化方法,包括如下步骤:

[0028] 步骤S1,构建三层上下文结构。

[0029] 参考图2,第一层为私有队列上下文RootContext,该层私有队列上下文RootContext与中间层Persistent Store Coordinator直接连接,中间层Persistent

Store Coordinator进一步与数据库DB连接。利用中间层Persistent Store Coordinator对第一层为私有队列上下文RootContext和数据库文件中间进行数据同步。由于是私有队列上下文,在和Persistent Store Coordinator通信时,可以实现并发存储,提高效率。

[0030] 第二层为主队列上下文MainContext,其父上下文为RootContext。由于是主队列上下文,所有的操作都会依次执行。

[0031] 第三层为私有队列上下文PrivateContext,该层私有队列上下文PrivateContext包括多个线程的私有队列上下文,并且保证每个线程独享一个上下文。其中,私有队列上下文PrivateContext的父上下文为MainContext。由于私有队列上下文不是线程安全的,本发明需要保证每个线程只能持有一个私有队列上下文PrivateContext。同一个线程针对数据库的所有操作都通过这个上下文来实现。

[0032] 步骤S2,存储开始,针对每次存储均启用新的线程,利用该线程中的PrivateContext进行数据存储操作。

[0033] 参考图3,每一次存储都新开子线程来使用PrivateContext进行操作。由于是子线程,针对数据的操作不会阻塞UI。

[0034] 具体的,针对每次存储均启用新的线程,首先检测该线程中是否已经存在私有队列上下文PrivateContext,如果有则利用该私有队列上下文PrivateContext进行数据操作;如果线程中没有私有队列上下文PrivateContext,再生成新的私有队列上下文PrivateContext,并将该新的私有队列上下文PrivateContext关联到新的线程上,利用该私有队列上下文PrivateContext进行数据操作。

[0035] 步骤S3,操作完成后,将数据自动同步到第二层的主队列上下文MainContext;由于MainContext是主队列上线文,可以保证多个PrivateContext同时同步数据到MainContext时不会产生线程问题。再由主队列上下文MainContext同步数据至第一层的私有队列上下文RootContext;

[0036] 步骤S4,由第一层的私有队列上下文RootContext通过中间层将数据同步到数据库中。由于RootContext是私有队列上下文,在真正将数据同步到数据库时,也不会对UI造成阻塞。

[0037] 根据本发明实施例的针对iOS CoreData数据库存储的优化方法,通过将私有队列上下文和主队列上下文两种存储模式混合使用,构建三层上下文结构,提供一种线程安全的高效存储方法,可以优化iOS CoreData数据库存储效率,支持多线程并发存储,保证线程安全。本发明通过构建三层上下文结构,使得数据存储都是异步执行,不会对主线程(UI)造成卡顿,并且可以同时批量操作数据,提高数据的存储效率,保证线程安全。

[0038] 在本说明书的描述中,参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中,对上述术语的示意性表述不一定指的是相同的实施例或示例。而且,描述的具体特征、结构、材料或者特点可以在任何的一个或多个实施例或示例中以合适的方式结合。

[0039] 尽管上面已经示出和描述了本发明的实施例,可以理解的是,上述实施例是示例性的,不能理解为对本发明的限制,本领域的普通技术人员在不脱离本发明的原理和宗旨的情况下在本发明的范围内可以对上述实施例进行变化、修改、替换和变型。本发明的范围

由所附权利要求及其等同限定。

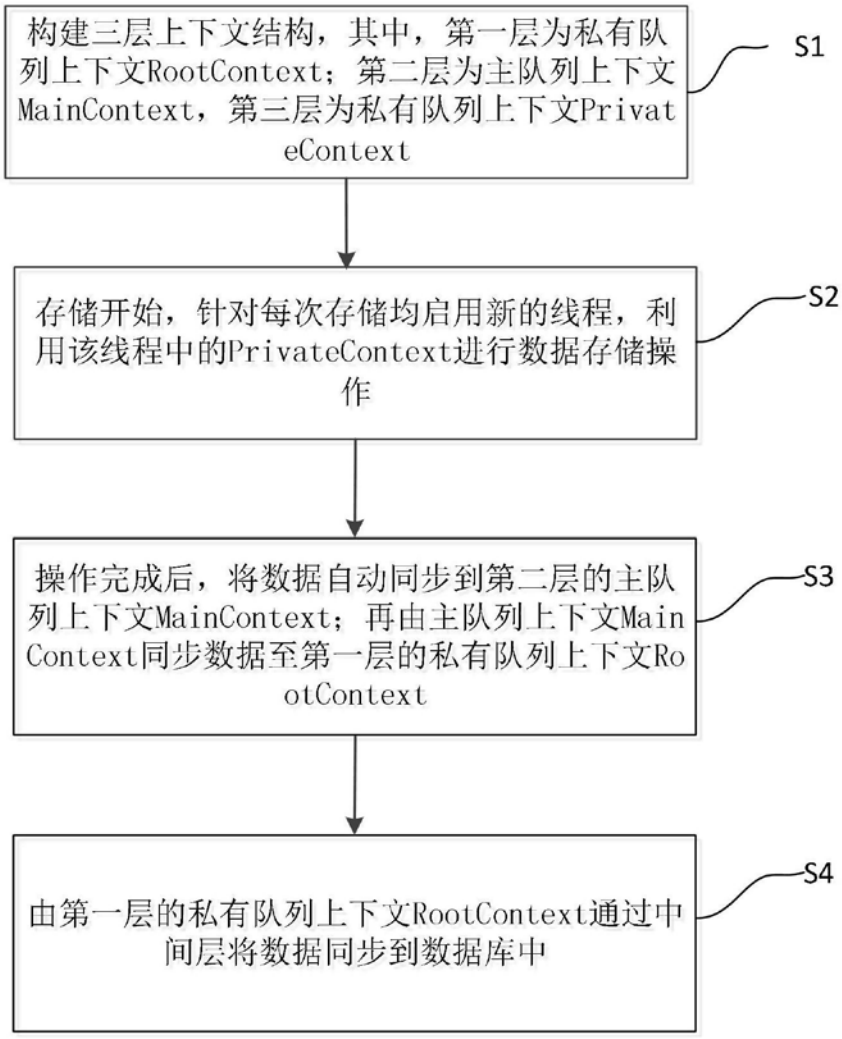

构建三层上下文结构，其中，第一层为私有队列上下文RootContext；第二层为主队列上下文MainContext，第三层为私有队列上下文PrivateContext
S1
存储开始，针对每次存储均启用新的线程，利用该线程中的PrivateContext进行数据存储操作
S2
操作完成后，将数据自动同步到第二层的主队列上下文MainContext；再由主队列上下文MainContext同步数据至第一层的私有队列上下文RootContext
S3
由第一层的私有队列上下文RootContext通过中间层将数据同步到数据库中
S4

图1

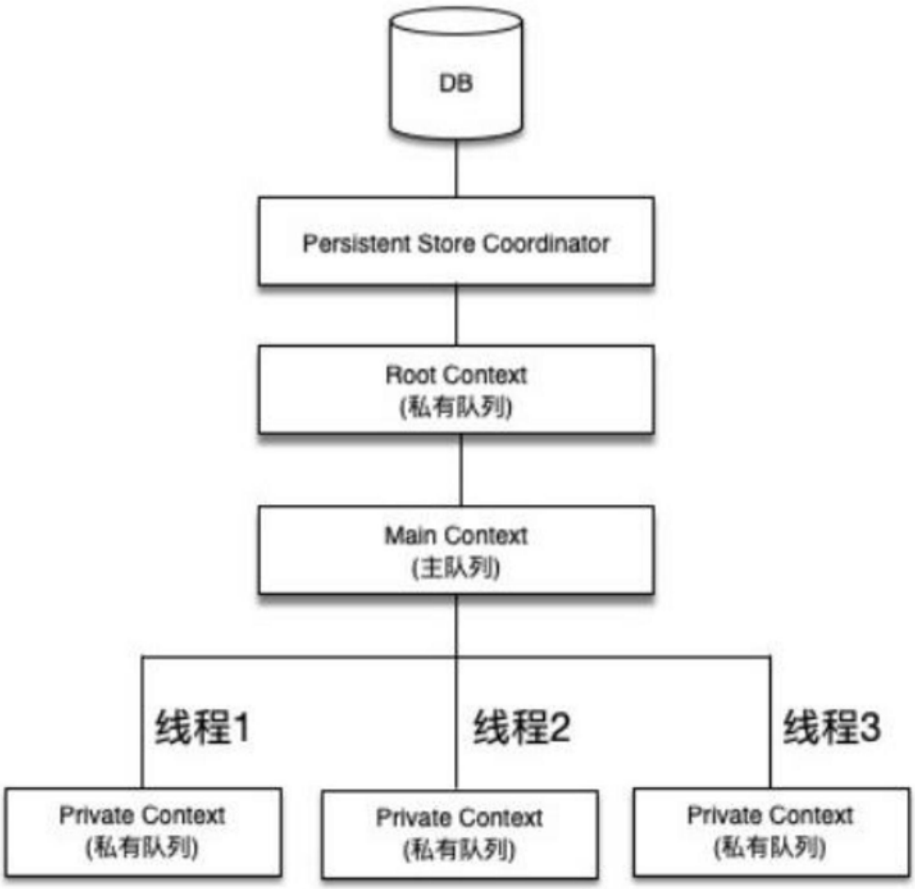

DB
Persistent Store Coordinator
Root Context
(私有队列)
Main Context
(主队列)
线程1
线程2
线程3
Private Context
(私有队列)
Private Context
(私有队列)
Private Context
(私有队列)

图2

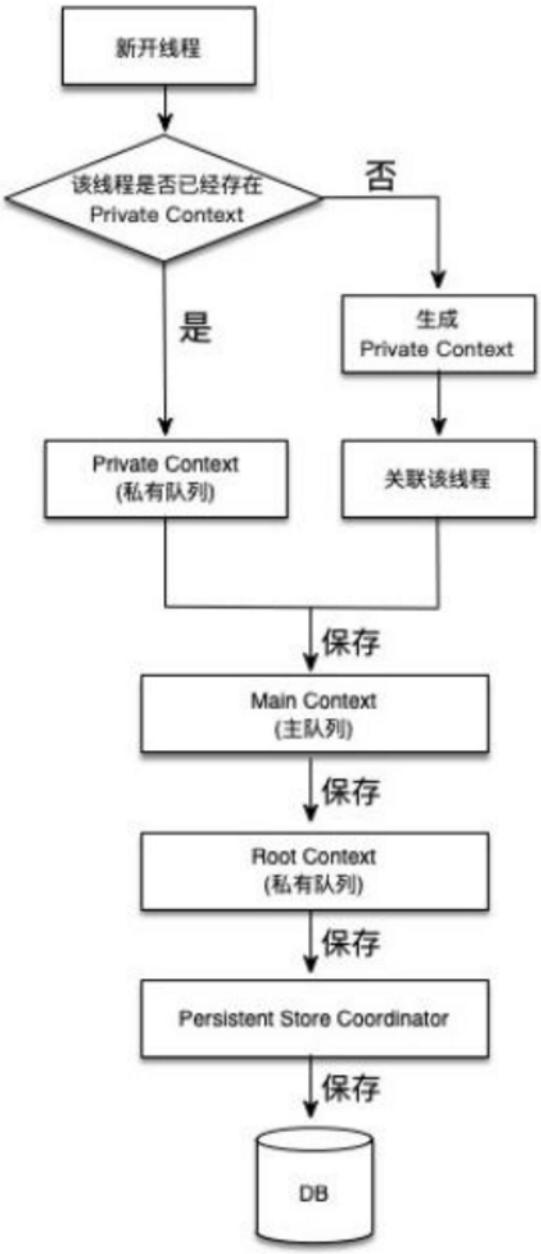

新开线程
该线程是否已经存在
Private Context
否
是
生成
Private Context
Private Context
(私有队列)
关联该线程
保存
Main Context
(主队列)
保存
Root Context
(私有队列)
保存
Persistent Store Coordinator
保存
DB

图3

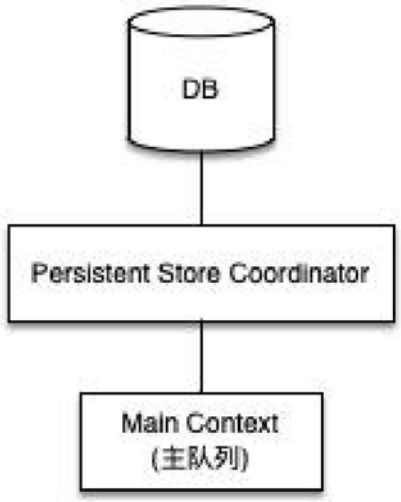


图4