

JasonLee的专栏

一个写着几行代码的理工科文学青年。

目录视图

摘要视图

RSS 订阅

个人资料



jasonblog



访问：1160573次

积分：12731

等级：

BLOG

7

排名：第631名

原创：192篇 转载：9篇

译文：2篇 评论：843条

文章搜索

博客专栏

- CPython学研笔记
文章：14篇
阅读：25618
- Inside The iOS Dev
文章：16篇
阅读：216593

阅读排行

- iOS中block实现的探究 (78825)
- 再谈iOS 7的手势滑动返回 (51135)
- 如何更好地限制一个UITextView (50037)
- 在iOS上实现一个简单的聊天室 (37140)

【专家问答】韦玮：Python基础编程实战专题

【知识库】Swift资源大集合

【公告】博客新皮肤上线啦

CSDN福利第二期

关于大数据量下Core Data的数据迁移

2014-01-04 14:52

9105人阅读

评论(7)

收藏

举报

分类：

职业 | 工作 (35)

iOS (85)

Core Data (14)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

Core Data版本迁移基础

通常，在使用Core Data的iOS App上，不同版本上的数据模型变更引发的数据迁移都是由Core Data来负责完成的。

这种数据迁移模式称为Lightweight Migration（可能对于开发人员来说是lightweight），开发人员只要在添加Persistent Store时设置好对应选项，其它的就交付给Core Data来做了：

```
NSDictionary *options = @{NSMigratePersistentStoresAutomaticallyOption: @(YES),
                           NSInferMappingModelAutomaticallyOption: @(YES)};
```

从命名上可以看出这两个选项分别代表：自动迁移Persistent Store，以及自动创建Mapping Model。

自动迁移Persistent Store很好理解，就是将数据从一个物理文件迁移到另一个物理文件，通常是因为物理文件结构发生了变化。

自动创建Mapping Model是为迁移Persistent Store服务的，所以当自动迁移Persistent Store选项NSMigratePersistentStoreAutomaticallyOption为@(YES)、且找不到Mapping Model时，coordinator会尝试创建一份。

其它初始化场景可以参考[Initiating the Migration Process](#)。

既然是尝试创建，便有成功和失败的不同结果。只有当数据模型的变更属于某些基本变化时，才能够成功地自动创建出一份Mapping Model，比如：新增一个字段；删除一个字段；必填字段转换成可选字段；可选字段转换成必填字段，同时提供了默认值等等。

因为可能创建Mapping Model失败，所以考虑容错性的话，可以事先判断下能否成功推断出一份Mapping Model：

```
[NSMappingModel inferredMappingModelForSourceModel:sourceModel
                                     destinationModel:destinationModel
                                     error:&error];
```

利用如上类方法，如果无法创建一份Mapping Model，则会返回nil，并带有具体原因。

来淘宝的这一年：前篇、
(34657)
iOS应用的crash日志的分
(29898)
随手记Swift基础和Optior
(22023)
使用UIImageView展现来
(20906)
Core Data浅谈系列之三
(19004)
iOS中注册功能的体验探!
(17589)

价值链接

刘未鹏 | MIND HACKS
酷壳 – CoolShell.cn
阮一峰的网络日志
弯曲评论
DBA Notes
云风的 BLOG
Erlang非业余研究
追风刀-丁奇
Joshua Zhu's Blog
Tim[后端技术]

友情博客

Judy的心情杂货铺
Nancy's Blogspot
SmoClock
KISimple 的专栏
code6 - wzc

文章分类

iOS (86)
Core Data (15)
Python (21)
C/C++ (27)
InSearchOfCS (8)
Linux (15)
Mac (2)
Qt (9)
WebDev (4)
Others About CS (15)
业界 (11)
职业 | 工作 (36)
生活 | 旅行 | 随笔 (32)

最新评论

iOS中block实现的探究
S往事随风S520: 谢谢分享, 看了一遍 云里雾里的, CPP里的内容感觉像是在看天书, 收藏了 慢慢研究
iOS中线程Call Stack的捕获和解
qweq123123123123: 有demo吗, 你的代码没有给全哈, 跑不通。
从NSTimer的失效性谈起 (二) :
jasonblog: @wwwang89123:可以参考下这里
http://stackoverflow.com/q/26...

以上都建立在Core Data能够自动找到sourceModel和destinationModel的基础上，如果无法找到对应的两份Model，则需要开发人员手工创建NSMigrationManager来进行数据迁移（可以参考[Use a Migration Manager if Models Cannot Be Found Automatically](#)）。

版本迁移过程

那么，数据迁移的过程是如何进行的？

首先，发生数据迁移需要三个基本条件：可以打开既有persistent store的sourceModel，新的数据模型destinationModel，以及这两者之间的映射关系Mapping Model。

利用这三样，当调用如下代码时（[addPersistentStore](#)）：

```
_persistentStoreCoordinator = [[NSPersistentStoreCoordinator alloc] initWithManagedObjectModel:self.managedObjectModel];
if (![ _persistentStoreCoordinator addPersistentStoreWithType:storeType
                                configuration:nil
                                URL:storeURL
                                options:options
                                error:&error])
{
    NSLog(@"Error : %@\n", [error localizedDescription]);
    NSAssert1(YES, @"Failed to create store %@ with NSSQLiteStoreType", [storeURL path]);
}
```

Core Data创建了两个stack（分别为source stack和destination stack，可以参考[Core Data stack](#)），然后遍历Mapping Model里每个entity的映射关系，做以下三件事情：

1. 基于source stack，Core Data先获取现有数据，然后在destination stack里创建当前entity的实例，只填充属性，不建立关系；
2. 重新创建entity之间的关系；
3. 验证数据的完整性和一致性，然后保存。

考虑到第二步是重新建立entity之间的关系，那么应该是在第一步就把所有entity的对象都创建好了，并且保留在内存中，为第二步服务（事实上也是如此）。

完成第二步后，所有数据还是维持在内存中（可能还有两份，因为有两个stack），在完成数据验证后才真正保存。

这样的话，会容易导致内存占用过多，因为Core Data在这个迁移过程中也没有一种机制清理响应的context。所以在数据量较多时，App可能会遇到在数据迁移过程因为内存紧张而被系统干掉。

针对这种情况，我们可以自定义迁移过程。

自定义数据迁移（解决内存问题）

自定义数据迁移的过程通畅分为三步：

第一步是判断是否需要数据进行迁移：

从NSTimer的失效性谈起（二）：
www.wang89123: 楼主，自带的NSTimer，退到后台是可以继续执行的，而GCD定义的Timer，APP退到后台就暂...

从NSTimer的失效性谈起（二）：
www.wang89123: @L_H_Y_1990: 确实，这个源码看了也忘记了，不大好理解了

从NSTimer的失效性谈起（二）：
lvable: 前辈感觉你分析的所有东东，都是通过源代码分析的。如果写一篇博客关于分析技术心得和方法就最好啦，哈哈，...

iOS中block实现的探究

lvable: 写的好强悍，绝对是我这个菜鸟的礼物

关于AssetsLibrary无法获取本地dongtinghong: 好像ios8以上的系统要用PHAsset才能检索出所有相册，而且检索出的相册组归类有点问题，楼主有没...

如何更好地限制一个UITextFieldID_超电磁炮: 没有crash啊，你们怎么操作的？

再谈multistage text input（中文qq_16839391: self.placeholder这个指的是什么

文章存档

- 2016年04月 (1)
- 2016年03月 (2)
- 2015年11月 (2)
- 2015年05月 (1)
- 2014年12月 (1)

展开

```
NSString *storeName = @"cdNBA.sqlite";
NSString *path = [[self applicationDocumentsDirectory] stringByAppendingPathComponent:storeName];
NSURL *storeURL = [NSURL fileURLWithPath:path];

NSError *error = NULL;
NSDictionary *sourceMetadata = [NSPersistentStoreCoordinator metadataForPersistentStoreOfType:NSSQLiteStoreType
                                                                    URL:storeURL
                                                                    error:&error];

NSManagedObjectModel *sourceModel = [NSManagedObjectModel mergedModelFromBundles:nil
                                                                    forStoreMetadata:sourceMetadata];

if (sourceModel == nil) {
    NSLog(@"sourceModel not found. \n");
}

NSManagedObjectModel *destinationModel = [self managedObjectModel];
BOOL pscCompatible = [destinationModel isConfiguration:nil compatibleWithStoreMetadata:sourceMetadata];
if (!pscCompatible) {
    NSLog(@"database incompatible. \n");
} else {
    NSLog(@"database compatible. \n");
}
```

http://blog.csdn.net/jasonblog

第二步是创建一个Migration Manager对象：

```
NSMigrationManager *migrationManager = [[NSMigrationManager alloc]
                                        initWithSourceModel:sourceModel
                                        destinationModel:destinationModel]
                                        autorelease];

http://blog.csdn.net/jasonblog
```

第三步是真正发生数据迁移：

```
if (![migrationManager migrateStoreFromURL:storeURL
                                         type:NSSQLiteStoreType
                                         options:nil
                                         withMappingModel:mappingModel
                                         toDestinationURL:tmpStoreURL
                                         destinationType:NSSQLiteStoreType
                                         destinationOptions:nil
                                         error:&error])
{
    NSLog(@"Error migrating %@, %@", error, [error userInfo]);
    abort();
}
```

http://blog.csdn.net/jasonblog

上面三幅图所展示的代码在内存使用量上跟lightweight migration也没什么区别，无法解决内存峰值过高的问题。

虽然Core Data专家Marcus S. Zarra比较倾向坚持使用lightweight migration，不过对于上述内存占用过多的问题，Apple官方推荐使用Multiple Passes来解决。

关于Multiple Passes，官方文档的说明很简明扼要，如有需要，可以参考Stackoverflow上的这么一篇帖子。

用我的话往简单里说就是对数据模型进行划分，把一份Mapping Model拆分成多份，然后分成多次迁移，从而降低内存峰值。这需要对数据库进行全盘的考虑（甚至可能需要变更部分设计），然后通过合理的划分把相关联的Entity放在一份Mapping Model里面（因为要建立关联）。

新的问题

采用上述方案来解决数据迁移过程中内存峰值的问题，我们仍然需要关注迁移所耗费的时间、内存，从而能够在数据上验证方案的有效性，并且在用户交互方面进行一些必要的更改（总不能让用户傻傻地在那边等数据迁移吧）。

虽然可以解决内存峰值的问题，但也引进了其它问题。

- 1. 需要对数据模型进行划分（以及变更）， 存在一定的工作量和风险；
- 2. 需要手工建立多份Mapping Model；
- 3. 需要手工编写Multiple Passes迁移代码；
- 4. 需要在每个版本变迁中都再次创建新的Mapping Model， 且在跨版本迁移过程存在着其它问题；
- 5. 数据模型版本多起来，就面临着跨版本迁移的问题，是要为每个历史版本创建到最新模型的Mapping Model， 还是只维护最近两个版本的Mapping Model（更早的版本通过相邻版本的Mapping Model依次迁移过来，比较耗

时)？

6. 对数据模型重新划分后，无关的Entity简单变更也会引起整个store和model的不兼容，需要迁移，那么是否考虑分库？

7. 这么大的动作服务的用户数是很少的（只有少数用户会遇到，或者是很少），但却是比较资深的（因为消息记录多），疼。。。

8. 这无法解决单个Entity数据量过大的问题，针对这种场景，只能自己手工编码进行小批量的数据迁移；

Jason

2014.01.02 @ Hangzhou

Evernote

顶

4

踩

0


上一篇 使用代码跟踪iOS App的内存使用情况

下一篇 2013-2014小结留白

我的同类文章

职业 工作 (35)	iOS (85)	Core Data (14)
<div>• 记一次帮同事排查由命名引... 2016-04-20 阅读 1867</div>	<div>• 从NSTimer的失效性谈起 (... 2016-03-15 阅读 926</div>	<div>• 从NSTimer的失效性谈起 (... 2016-03-15 阅读 1085</div>
<div>• 从NSTimer的失效性谈起 (... 2016-03-15 阅读 926</div>	<div>• Auto Layout和UILabel 2015-05-25 阅读 2993</div>	
<div>• Inside ARC — to see the co... 2014-05-23 阅读 2613</div>	<div>• 使用Objective-C处理UBB标签 2014-05-23 阅读 1637</div>	
<div>• 又是一年WWDC，暨回顾去... 2014-05-13 阅读 1406</div>	<div>• iOS应用的crash日志的分析... 2014-02-10 阅读 29884</div>	
<div>• 2013-2014小结留白 2014-01-19 阅读 1564</div>	<div>• 关于互相尊重和直言有讳 2013-08-31 阅读 2286</div>	
<div>更多文章</div>		

参考知识库



MySQL知识库

10164 关注 | 1351 收录



Swift知识库

2194 关注 | 385 收录

猜你在找

- iOS8开发技术（Swift版）：iOS基础知识

大数据编程语言：Java基础

从零练就iOS高手实战班

大数据之编程语言：Scala

数据结构和算法
- Core Data 模型版本升级和数据迁移三映射概述

Core Data 深入理解5 一版本数据迁移


缺失version model的情况下能否完成Core Data的数据

Core Data 模型版本升级和数据迁移二轻量级数据迁移


Oracle 大数据量下的数据迁移

查看评论


4楼 [u2takey](#) 2014-12-09 11:48发表

 如果是内置数据（数据不是用户产生）一定要使用多store+bundle内置sqlite文件的方式，否则 一点点数据（1~2M）迁移都会在iphone4这样的手机上运行1~2分钟。血的教训。。


Re: [jasonblog](#) 2014-12-18 10:57发表

 回复u2takey：多谢分享：)


3楼 [u2takey](#) 2014-12-09 11:40发表

 Lightweight Migration基本满足所有需求，使用多个model似乎麻烦比好处多，多configuration+多soter可以解决部分问题，实在很慢开机像游戏一样加loading状态就好了。


2楼 [xdream86](#) 2014-02-25 23:02发表

 验证过了，表名发生了变化，的确是替换了物理文件。


Re: [jasonblog](#) 2014-03-03 18:01发表

 回复xdream86：赞一个！

1楼 [xdream86](#) 2014-02-24 17:49发表

 “自动迁移Persistent Store很好理解，就是将数据从一个物理文件迁移到另一个物理文件，通常是因为物理文件结构发生了变化。”你确定数据是被迁移到另一个物理文件中，而不是保持之前的物理文件不变，只是使用最新版本的数据模型模式调整了映射规则。

Re: [jasonblog](#) 2014-03-03 18:01发表

 回复xdream86：CSDN的通知不是很靠谱啊，这两条评论我都没看到

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack
- VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery
- BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity
- Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC
- coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo
- Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr
- Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap