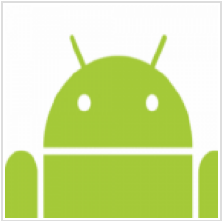




yuan514168845的专栏

目录视图 摘要视图 **RSS** 订阅

个人资料



袁阳的CSDN博客



访问：12372次

积分：536

等级：**BLOG > 3**

排名：千里之外

原创：38篇 转载：0篇

译文：0篇 评论：27条

文章搜索

博客专栏



Java基础学习

文章：17篇

阅读：8605

文章分类

今日总结 (29)

android开发经验 (6)

自定义控件起步 (3)

Android源码分析 (7)

Android属性动画 (0)

Volley源码分析 (1)

文档翻译 (0)

文章存档

2015年11月 (5)

2015年10月 (10)

2015年09月 (1)

2014年03月 (1)

2014年02月 (7)

展开

[置顶] 为什么说Volley适合数据量小，通信频繁的网络操作

标签：网络编程 Volley 源码分析

2015-11-03 15:28

69人阅读

评论(0)

收藏

举报

分类： 今日总结 (28) Android源码分析 (6)

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(?)

[+]

前言

网络编程对于客户端来说是一块及其重要的地方，使用高效的网络请求框架将为你的系统产生很大的影响。而Volley作为谷歌的一个开源项目，炙手可热。有很多中小型公司的安卓移动客户端的网络程序都是基于volley的。Volley的优点很多，光可**扩展性**这一条优点就值得我们称赞。但是我想针对的是在 Google I/O 2013 大会上发布Volley的时候的一句话：a burst or emission of many things or a large amount at once（爆炸性的事件在很短的时间内发生），意思就是：Volley特别适合**数据量小，通信量大**的客户端。同为网络请求框架，为什么Volley会有这样的特点？接下来，我就用我的理解来解释一下。

ByteArrayPool

ByteArrayPool产生背景

根据类名，知道这是一个字节数组缓存池。没错，就是用来缓存 网络请求获得的数据。当网络请求得到返回数据以后，我们需要在内存开辟出一块区域来存放我们得到的网络数据，不论是json还是图片，都会存在于内存的某一块区域，然后拿到UI显示，然而客户端请求是相当频繁的操作，想一下我们平时使用知乎等一些客户端，几乎每一个操作都要进行网络请求（虽然知乎大部分是WebView）。那么问题来了：这么频繁的数据请求，获得数据以后我们先要在堆内存开辟存储空间，然后显示，等到时机成熟，GC回收这块区域，如此往复，那么GC的负担就相当的重，然而Android客户端处理能力有限，频繁GC对客户端的性能有直接影响。我们能不能减少GC和内存的分配呢？我想这个问题就是这个类的产生背景。

实现原理（怎么实现缓存从而减少GC）

在ByteArrayPool中维护了两个List集合。

属性名	作用	类型
mBuffersByLastUse	按照最近使用对byte[]排序	LinkedList
mBuffersBySize	按照byte[]大小对byte[]排序	ArrayList

通过上述两个属性的作用可以分析出它们是ByteArrayPool类对byte[]的管理。

从缓冲区取空间

当请求数据返回以后，我们不是急于在内存开辟空间，而是从ByteArrayPool中取出一块已经分配的内存区域。此时会调用ByteArrayPool的getBuf(int)方法，来得到一块参数大小的区域，源码如下：

阅读排行

Java基础(十四)——JDBC	(725)
Java基础 (十二) ——JC	(663)
Java基础 (十三) ——JC	(634)
Java基础 (十) ——Java	(622)
java基础(十六)——GUI基	(571)
Java基础 (十一) ——Ja	(525)
Java基础(十五)——GUI基	(523)
Java基础之 (九) ——Ja	(504)
java基础 (一) ——学习ε	(498)
Java基础 (十七) ——Ot	(486)

评论排行

Java基础(十四)——JDBC	(7)
Java基础之 (九) ——Ja	(4)
安卓拍照，选取照片并裁	(3)
Java基础 (十二) ——JC	(3)
Java基础 (十一) ——Ja	(3)
Android 内存泄露的几种	(2)
Java基础(十五)——GUI基	(1)
Java基础 (十三) ——JC	(1)
java基础(十六)——GUI基	(1)
Java基础 (十) ——Java	(1)

最新评论

为什么说Volley适合数据量小，更黄枫：这么好文章，居然没人顶，那我不客气了。

Android 内存泄露的几种情形
袁阳的CSDN博客：
@u012845099:用完当然可以，但是网络请求数据是耗时操作，没等handler使用，你此时就可...

Android 内存泄露的几种情形
小图包: 用完Handler可以在Activity的时候释放Handler的数据

安卓拍照，选取照片并裁减完全/袁阳的CSDN博客: @lk832400: 邮箱发来

安卓拍照，选取照片并裁减完全/lk832400: 请求源码啊，我现在被照相机裁剪虐死了快

自定义控件之仿IOS搜索框的实现
袁阳的CSDN博客: 大家留个言，点个赞，让我知道你来过

安卓拍照，选取照片并裁减完全/袁阳的CSDN博客: 欢迎大家批评指正

Java基础(十四)——JDBC(三)JD
袁阳的CSDN博客：
@u013506018:你好 已经开了J2se学习的专栏。可有来看看。

Java基础(十四)——JDBC(三)JD
明月涟漪: 开个专栏吧，我天天来听课

Java基础之 (九) ——Java中的I
袁阳的CSDN博客：
@wx3957156:建议把IO包的结构先弄清晰，然后在深入，这样事半功倍，希望对你有帮助

```
1 public synchronized byte[] getBuf(int len) {
2     for (int i = 0; i < mBuffersBySize.size(); i++) {
3         byte[] buf = mBuffersBySize.get(i);
4         if (buf.length >= len) {
5             mCurrentSize -= buf.length;
6             mBuffersBySize.remove(i);
7             mBuffersByLastUse.remove(buf);
8             return buf;
9         }
10    }
11    return new byte[len];
12 }
```

方法的第2行代码，遍历mBuffersBySize，找到最适合len大小的byte[]。第6~8行更新缓存池中数据的大小，并从两个数组中去除分配出去的byte[]。如果在缓存池中没有要求的byte[]，此时会从内存分配一跨区域返回。此方法主要的功能：**不必每次存数据都要进行内存分配，而是先查找缓冲池中有无适合的内存区域，如果有，直接拿来用，从而减少内存分配的次数。**

其实这个方法有改进空间：由于在类中有一个mSizeLimit属性，表示此缓冲区的最大值。我们可以在方法体的第一行判断len与mSizeLimit的大小，如否len>mSizeLimit,直接进入到最后一句运行，否则，循环。修改后的方法如下：

```
1 public synchronized byte[] getBuf(int len) {
2     //有的条件不需循环
3     if(len<=mSizeLimit)
4     {
5         for (int i = 0; i < mBuffersBySize.size(); i++) {
6             byte[] buf = mBuffersBySize.get(i);
7             if (buf.length >= len) {
8                 mCurrentSize -= buf.length;
9                 mBuffersBySize.remove(i);
10                mBuffersByLastUse.remove(buf);
11                return buf;
12            }
13        }
14    }
15    return new byte[len];
16 }
```

将空间返回给缓存池

如果只是拿数据，缓存区的只会越来越小，我们还需要向缓冲区中加入存储空间。这个时候涉及到一个方法：
returnBuf(byte[])。

```
1 public synchronized void returnBuf(byte[] buf) {
2     if (buf == null || buf.length > mSizeLimit) {
3         return;
4     }
5     mBuffersByLastUse.add(buf);
6     int pos = Collections.binarySearch(mBuffersBySize, buf, BUF_COMPARATOR);
7     if (pos < 0) {
8         pos = -pos - 1;
9     }
10    mBuffersBySize.add(pos, buf);
11    mCurrentSize += buf.length;
12    trim();
13 }
```

方法首先检查 要插入的数据大小有没有超出边界，如果没有，利用二分法找到插入位置，将数据插入到上述的两个集合，完成排序。然后更新缓冲池的大小，以方便从缓冲区中取存储空间。

结语

ByteArrayPool利用getBuf和returnBuf以及mBuffersByLastUse和mBuffersBySize完成字节数组的缓存。当需要使内存区域的时候，先从已经分配的区域中获得以减少内存分配次数。当空间用完以后，在将数据返回给此缓冲区。这样，就会减少内存区域堆内存的波动和减少GC的回收，让CPU把更多的性能留给页面的渲染，提高性能。通过

这个类发现，谷歌对技术的细节十分考究。

上一篇 [WebView使用指南（下）](#)

下一篇 [Volley的cache之硬盘缓存--DiskBasedCache](#)

顶

2

踩

0

我的同类文章

今日总结（28）

[Android源码分析（6）](#)

主题推荐

[网络编程](#)

[android](#)

[ui](#)

[通信](#)

[网络](#)

[数据](#)

[开源项目](#)

猜你在找

查看评论

暂无评论

发表评论

用户名：

scboyhj__

评论内容：



提交

* 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](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

