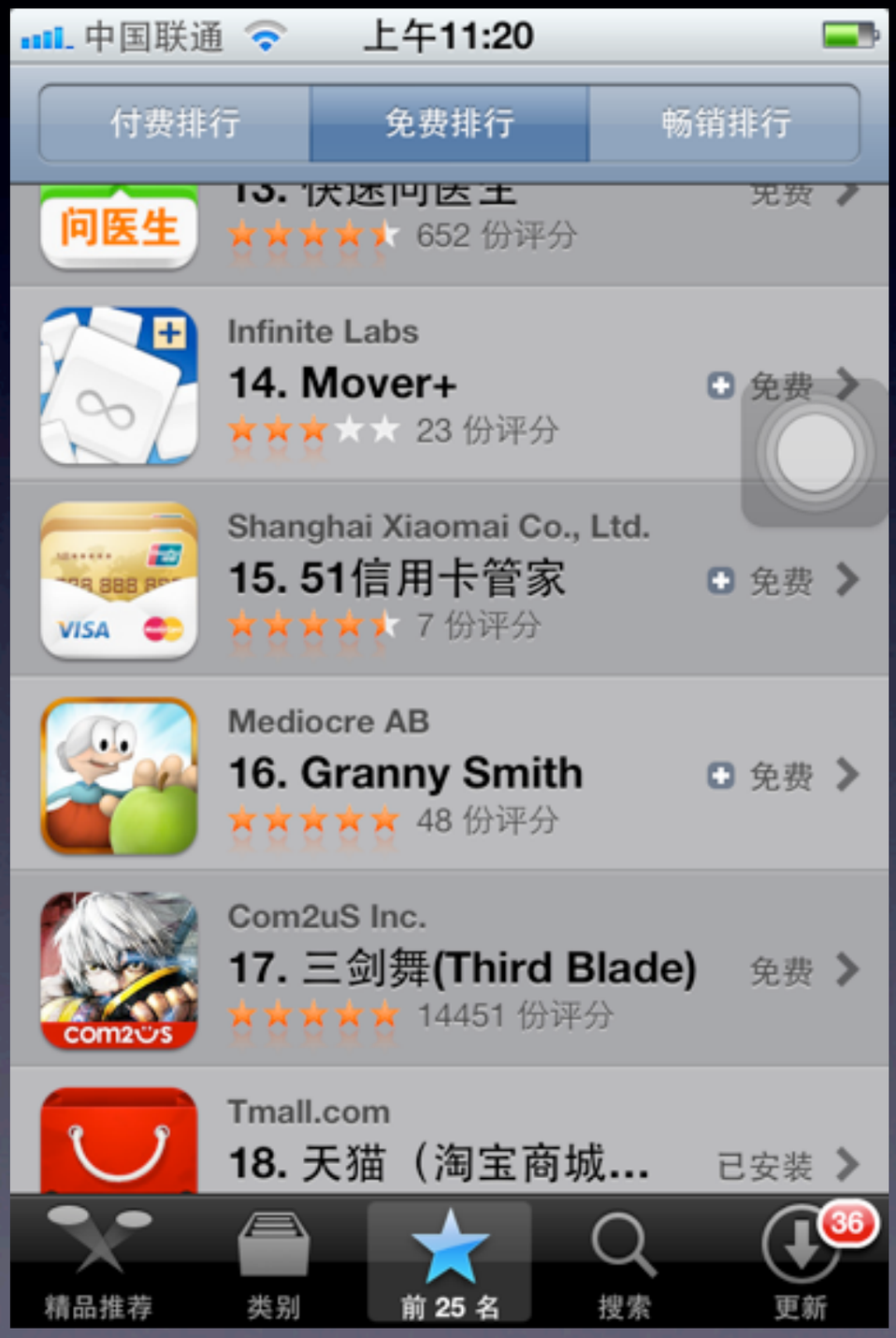
网络请求模块设计与分析

何宝晨 2012-11
腾讯无线大连研发中心

滚动列表图片的下载?


中国联通
上午11:20
付费排行
免费排行
畅销排行
652 份评分
Infinite Labs
14. Mover+
免费
23 份评分
Shanghai Xiaomai Co., Ltd.
15. 51信用卡管家
免费
7 份评分
Mediocre AB
16. Granny Smith
免费
48 份评分
Com2uS Inc.
17. 三剑舞(Third Blade)
免费
14451 份评分
Tmall.com
18. 天猫（淘宝商城...
已安装
精品推荐
类别
前 25 名
搜索
更新
36

实现方法I

- 为每一行需要显示图片发送一个HTTP请求
- 看实际运行效果

存在哪些问题?

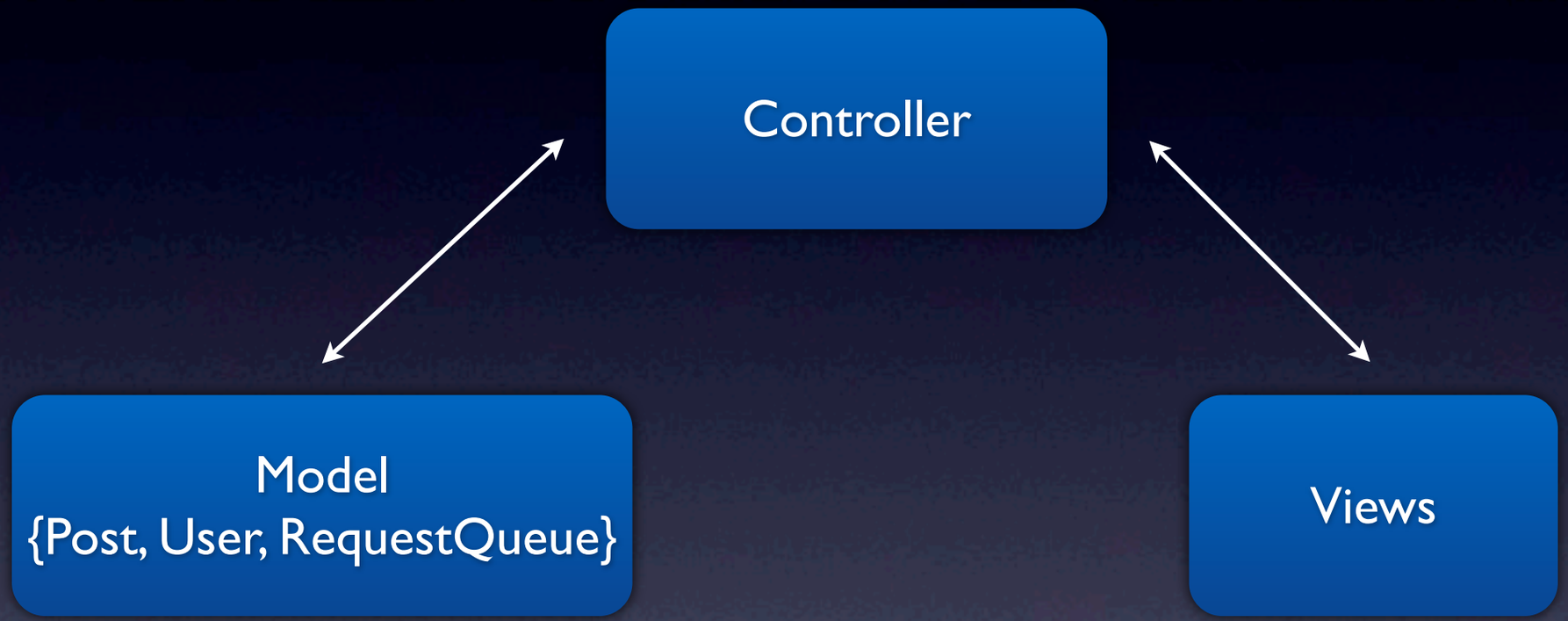
使用一个请求队列?

- 引入320Network开源库
- 可以设置并发请求的数量
- 先进先出

Controller中省略的代码有

- 拉取列表数据（还得支持分段拉取）
- 网络的各种异常没有处理
- 正常的业务处理，如果不是Demo的话
- 综上，Controller里太复杂了！

谁来收发请求合适?

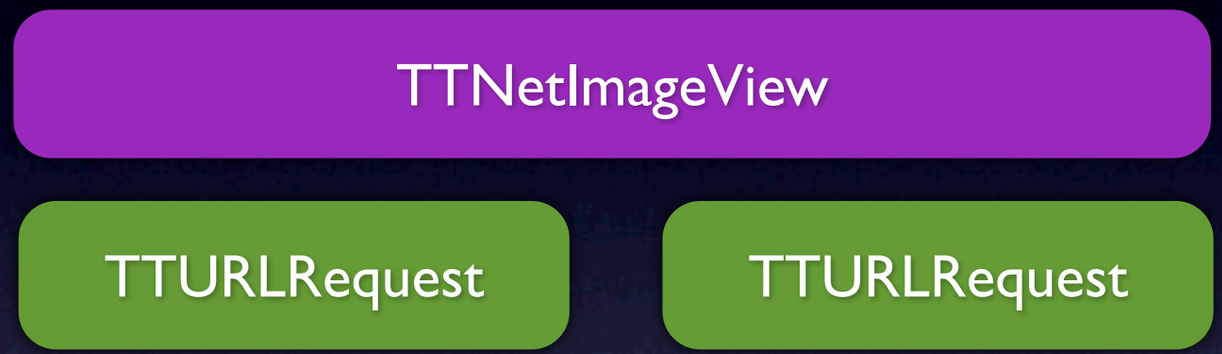


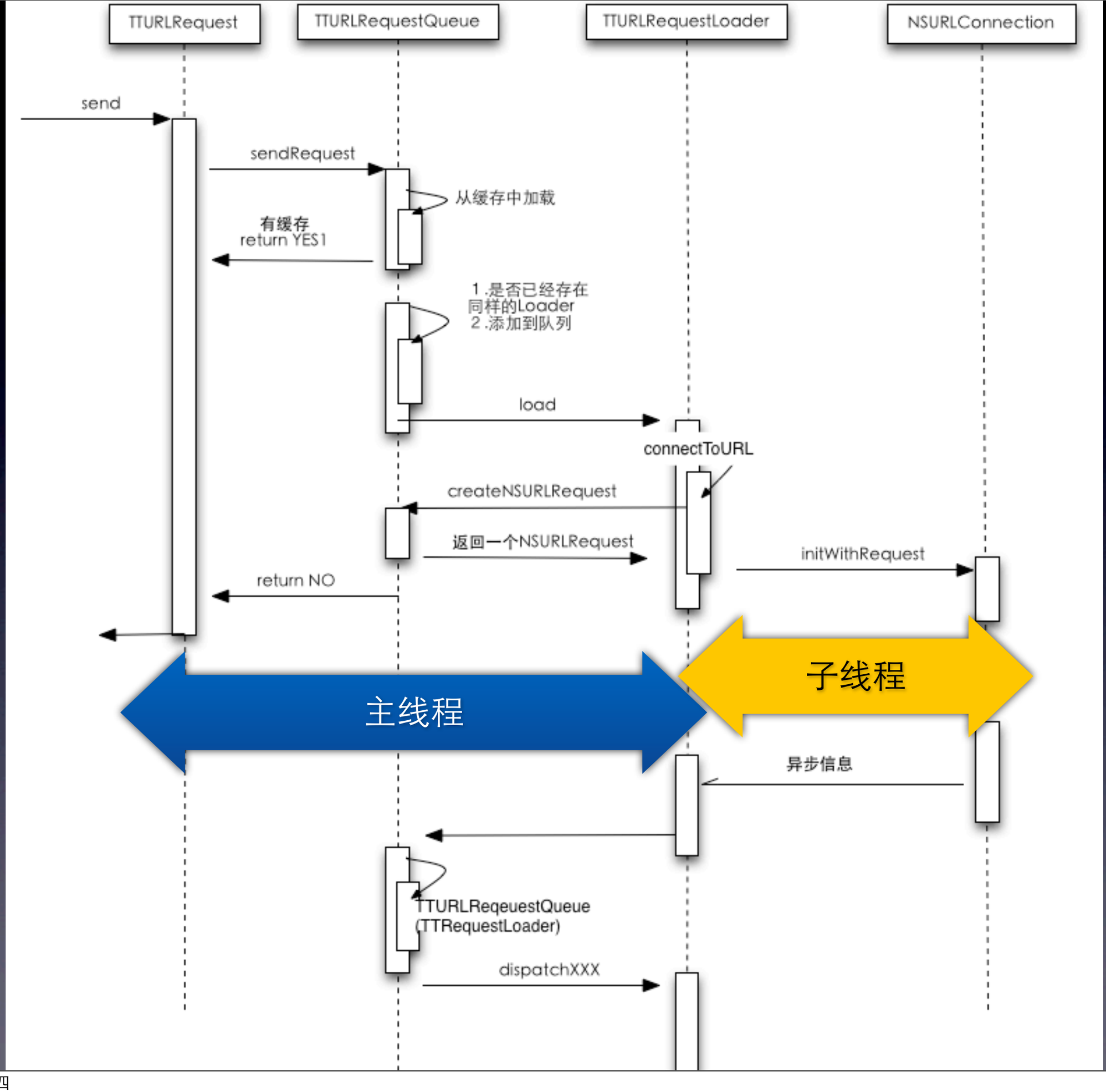
```
imageView.image = [UIImage imageNamed:@"xxx.png"];
```

如果能像上面这样就最好了！

```
imageView.url = “http://wo-pi-zun-ni-shi-zhu.jpg”
```

320Network


TTNetImageView
TTURLRequest
TTURLRequest
TTRequestQueue
TTRequ
TTRequ
TTRequ
TTRequestLoader
TTURLCache


TTURLRequest
TTURLRequestQueue
TTURLRequestLoader
NSURLConnection
send
sendRequest
从缓存中加载
有缓存
return YES1
1.是否已经存在
同样的Loader
2.添加到队列
load
connectToURL
createNSURLRequest
返回一个NSURLRequest
initWithRequest
return NO
主线程
子线程
异步信息
TTURLReqeuestQueue
(TTRequestLoader)
dispatchXXX

TTNetImageView

- 增了urlPath，不仅支持网络连接，还支持本地路径
- 自己管理请求的收发

TTURLRequest

- 代表一个请求实体（代理）
- 支持多delegate的回调
- 最终会被转换成系统API提供的NSURLRequest对象，TTRequestLoader来执行真正的网络请求

TTURLRequest

- cacheKey
  - 根据请求的URL和Header生成
  - 用来判断两个请求是否可以合并
  - 取消请求用来查找Loader
- cacheExpirationAge
  - 默认值是一周

TTURLRequest

- cachePolicy（缓存策略）
- 内存, 硬盘, 服务端, etag, 本地, 默认
- 默认类型=内存|硬盘|服务端

```
TTURLRequestCachePolicyDefault  = (TTURLRequestCachePolicyMemory
                                 | TTURLRequestCachePolicyDisk
                                 | TTURLRequestCachePolicyNetwork)
```

TTURLRequestQueue

- 通过控制同时执行的TTRequestLoader数量，来维护网络连接的并发数
- 管理请求队列，和正在执行的请求队列，按照先进先出的方式执行请求，没有优先级
- 通过两个维度来取消请求，一是单个请求的取消，二是以delegate为参数来取消请求，可以取消多个请求
- 检查TTURLCache中是否存在有效的缓存数据，如果有，侧直接返回，而不发送实际的请求

TTURLRequestQueue

- suspended的作用
- 尽量的避免阻塞主线程
- YES的时候，不执行新的请求，不读缓存，全部放入等待队列

TTRequestLoader

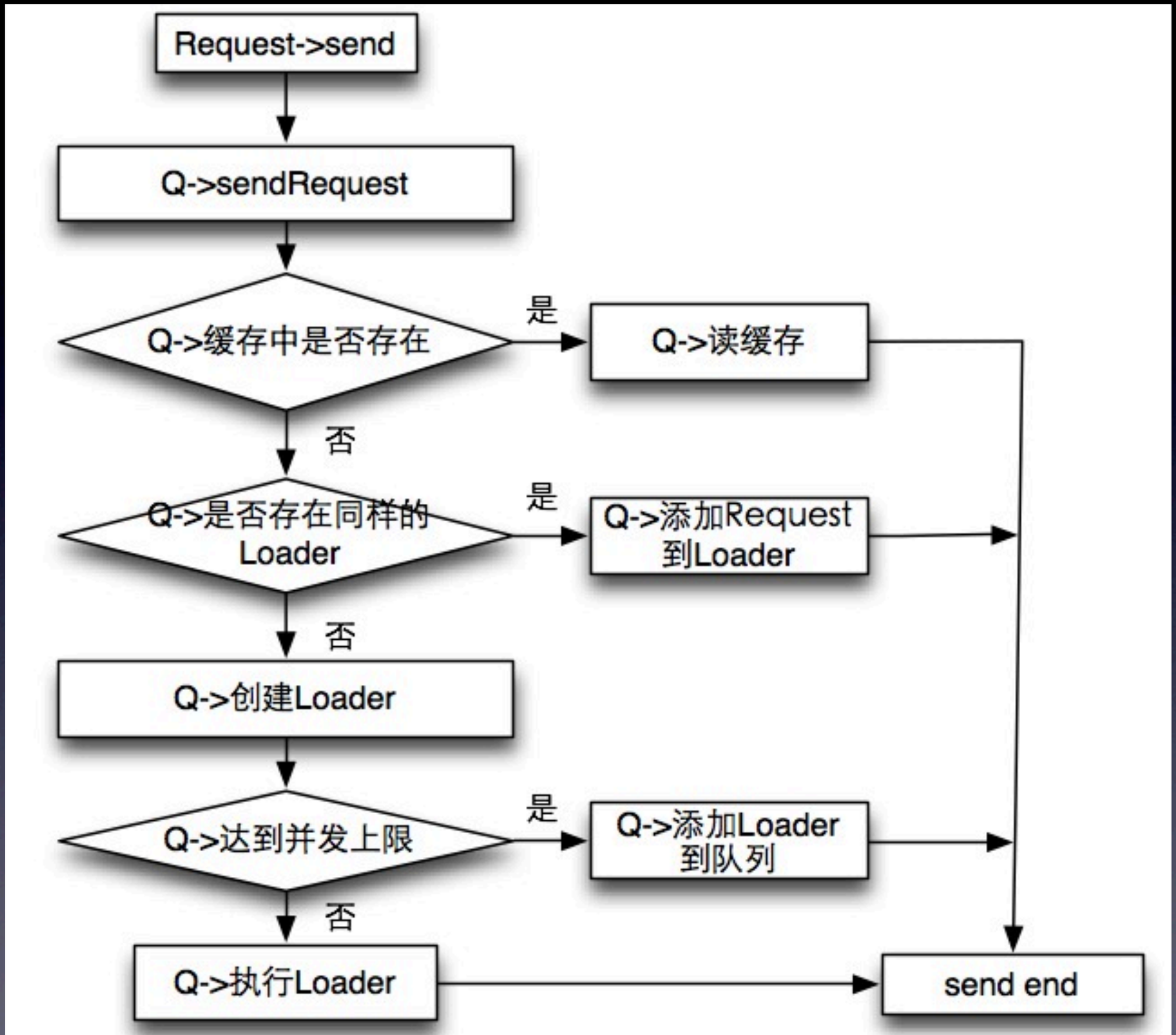
- 代替Controller来收发请求
- 是最终调用系统API来发请求的对象
- 有出错重试机制，移动网络下非常有用
- 这个对象的存在是个比较巧的一个设计

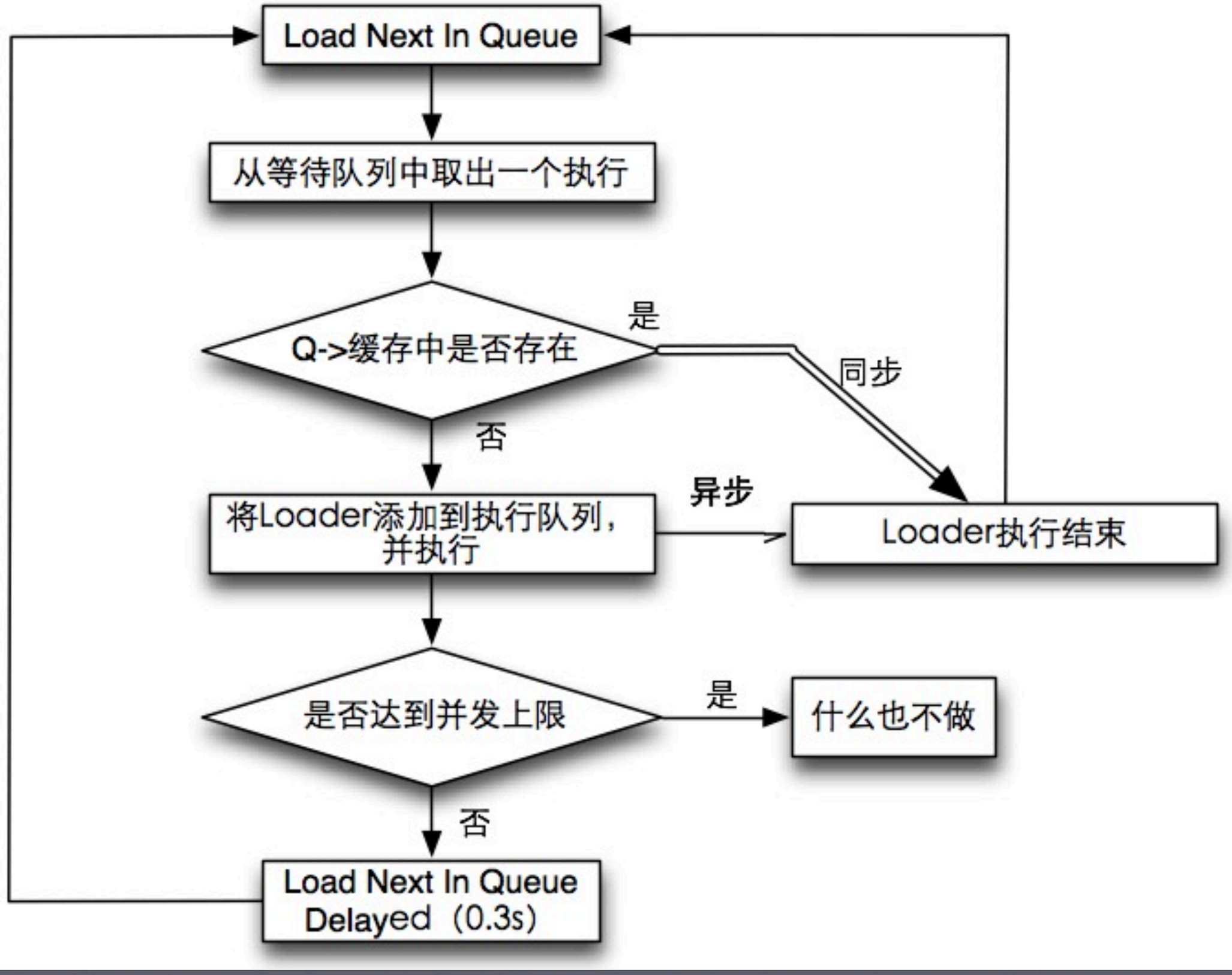
TTURLCache

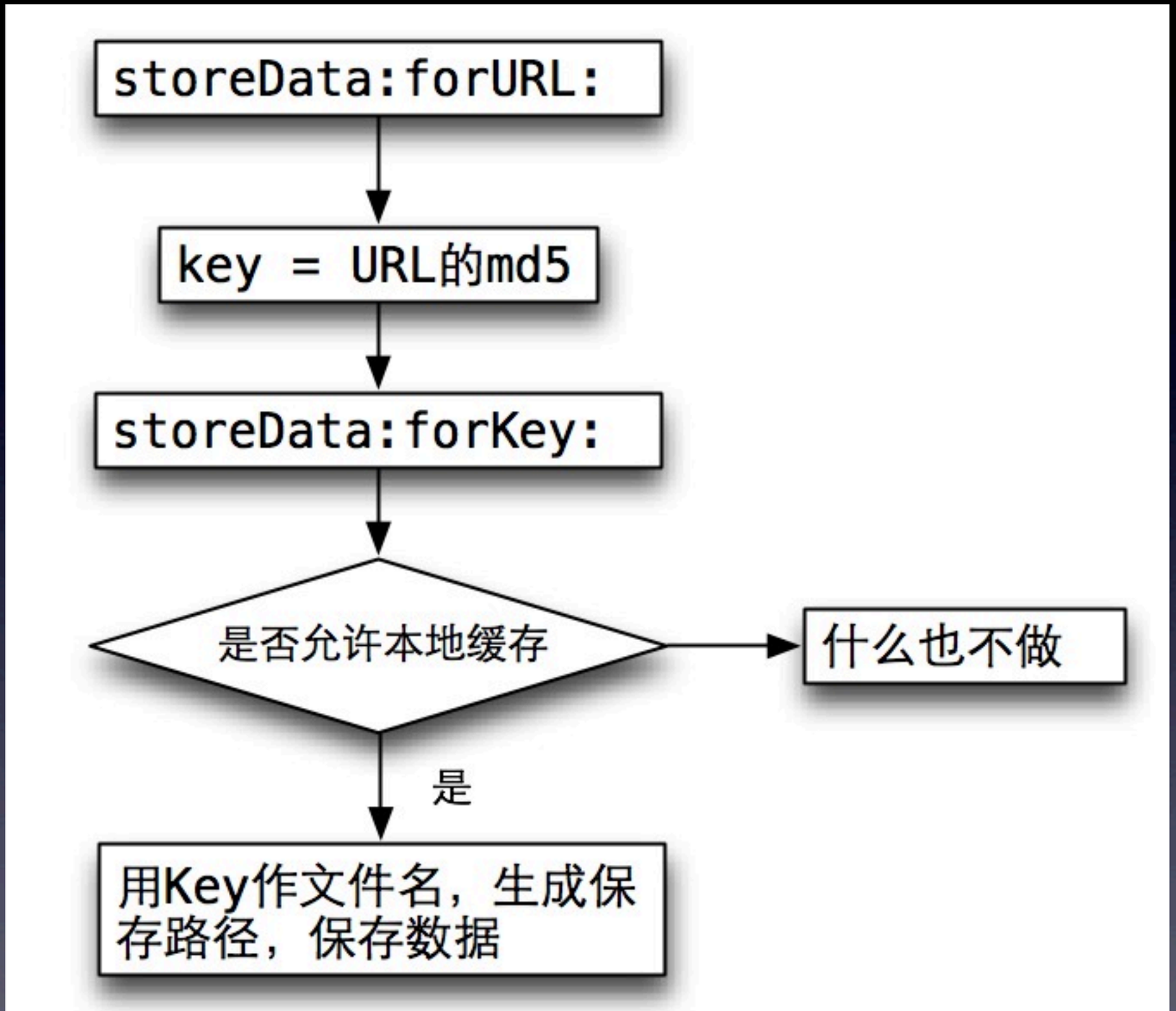
- 负责缓存数据的管理，内存和本地
- 对缓存的数据块大小有限制，可设置
- 支持缓存的有效期
- 内存告警时，内存的数据可清掉

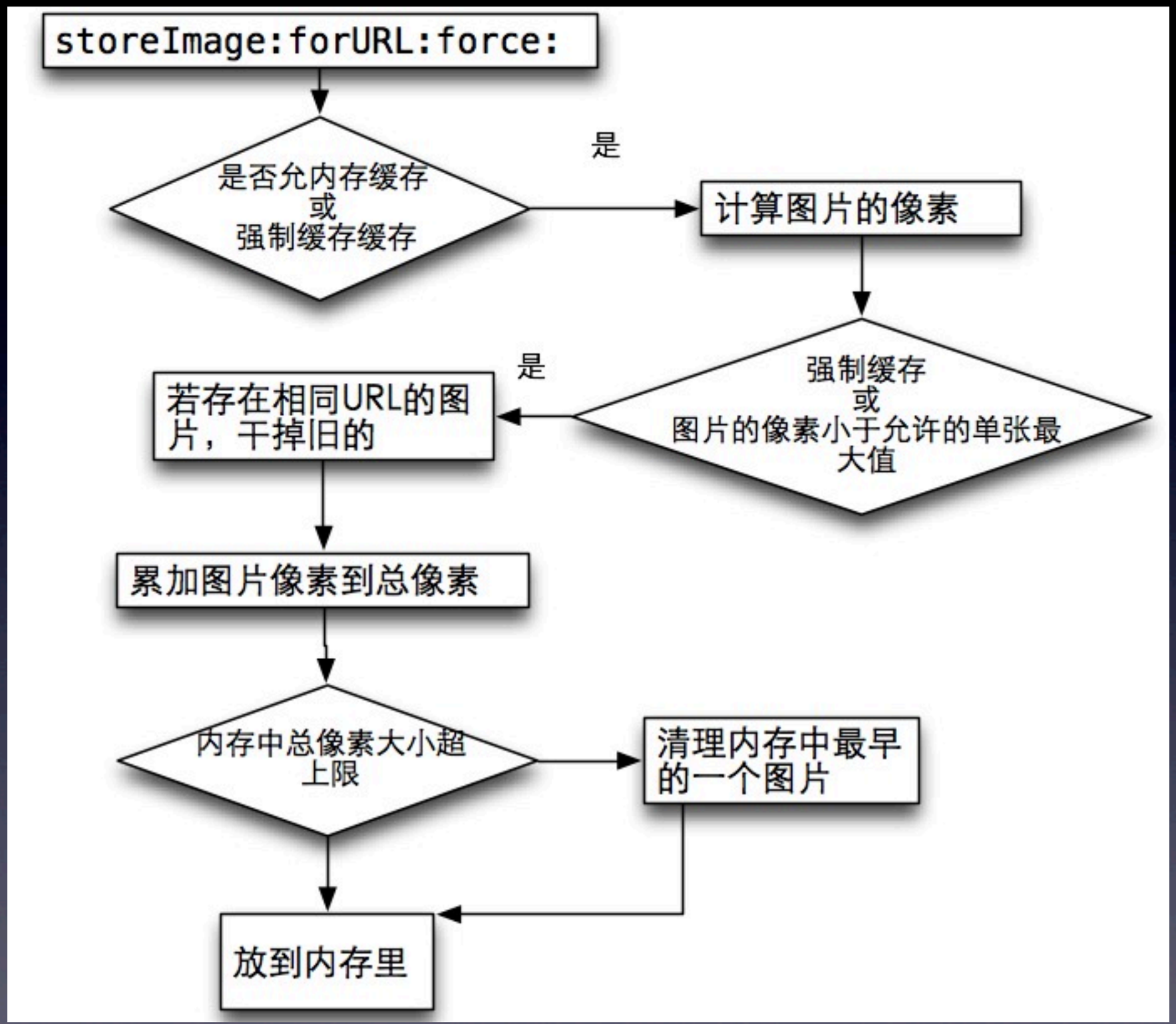
TTURLCache

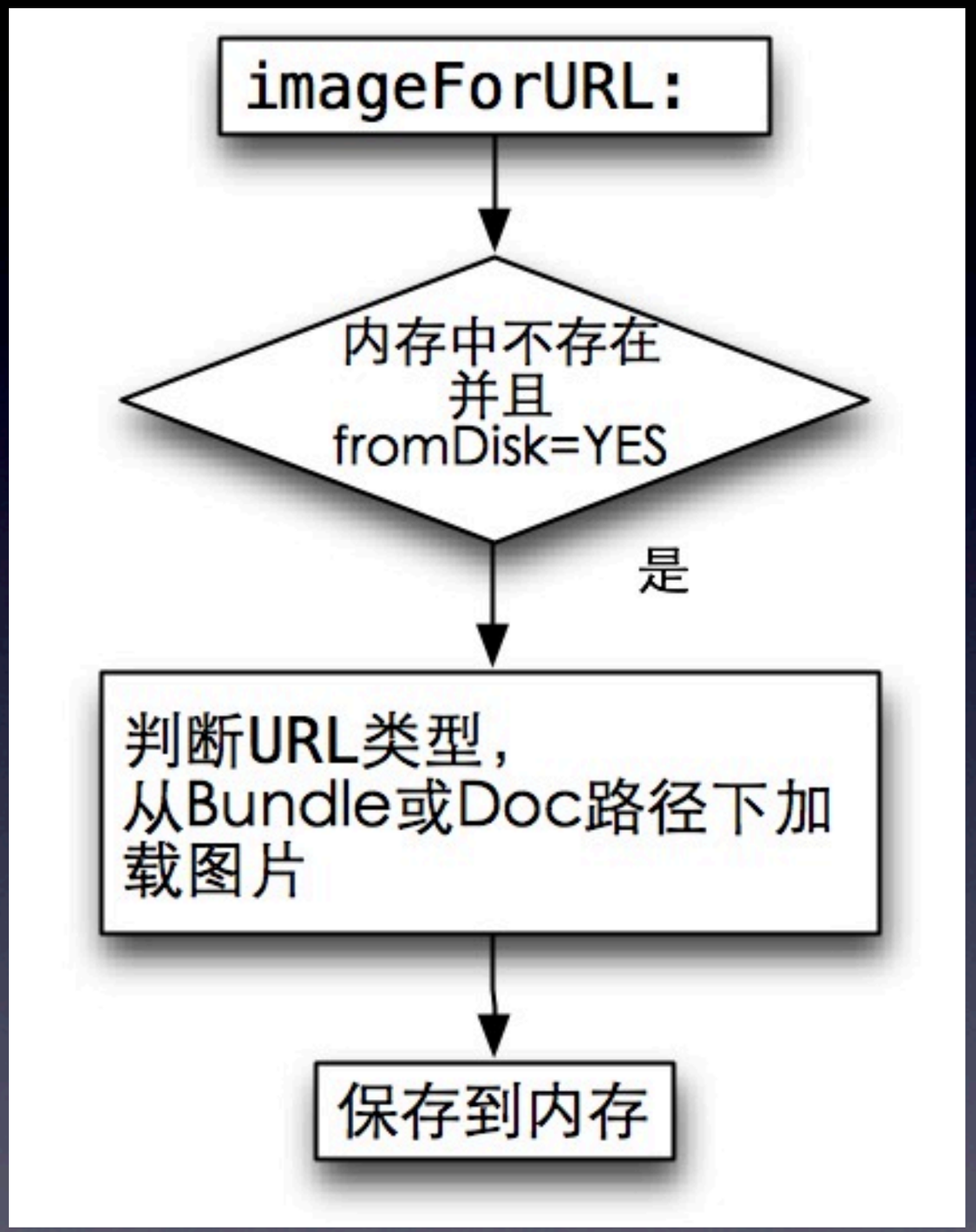
- 可以通过静态方法CacheWithName创建多个实例，用于不同的业务目的
- 每个实例可单独设置占用内存的大小
- 仍然存在很多可以改进和完善的空间

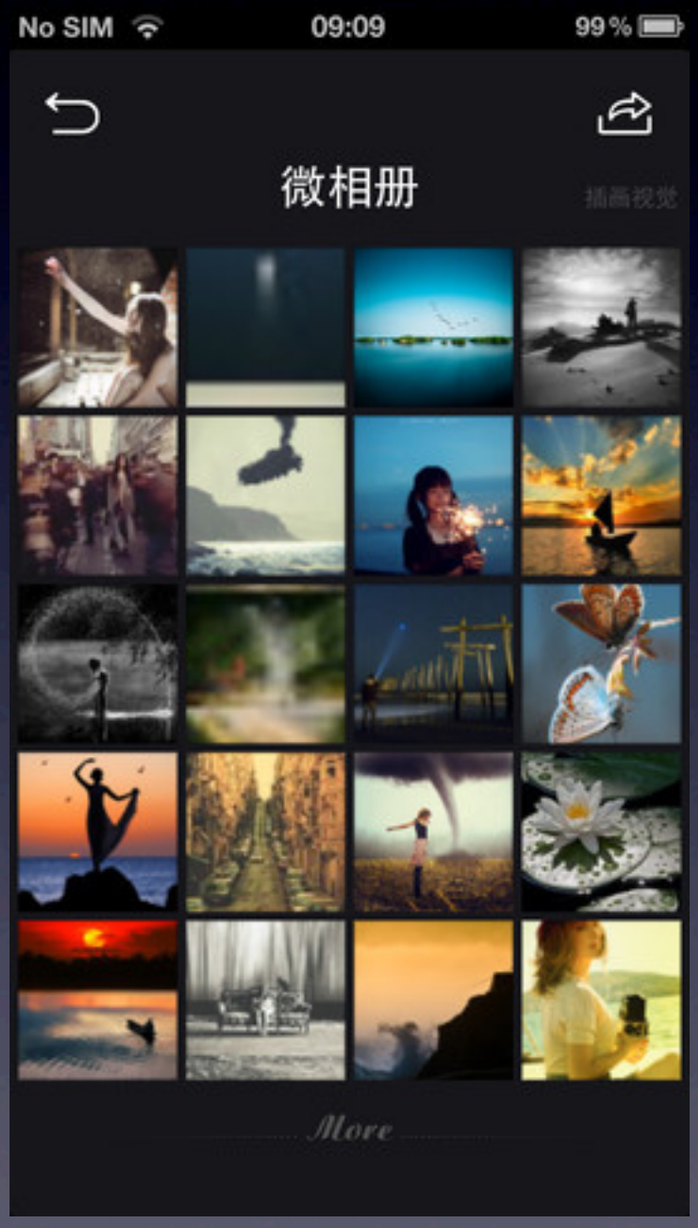

Request->send
Q->sendRequest
Q->缓存中是否存在
是
Q->读缓存
否
Q->是否存在同样的Loader
是
Q->添加Request到Loader
否
Q->创建Loader
Q->达到并发上限
是
Q->添加Loader到队列
否
Q->执行Loader
send end


Load Next In Queue
从等待队列中取出一个执行
Q->缓存中是否存在
是
同步
否
异步
将Loader添加到执行队列，
并执行
Loader执行结束
是否达到并发上限
是
什么也不做
否
Load Next In Queue
Delayed（0.3s）


storeData:forURL:
key = URL的md5
storeData:forKey:
是否允许本地缓存
什么也不做
是
用Key作文件名，生成保存路径，保存数据


storeImage:forURL:force:
是否允内存缓存
或
强制缓存缓存
是
计算图片的像素
强制缓存
或
图片的像素小于允许的单张最大值
是
若存在相同URL的图片，干掉旧的
累加图片像素到总像素
内存中总像素大小超上限
清理内存中最早的一个图片
放到内存里


imageForURL:
内存中不存在
并且
fromDisk=YES
是
判断URL类型，
从Bundle或Doc路径下加
载图片
保存到内存


No SIM
09:09
99%
微相册
插画视觉
More


No SIM
09:09
99%
返回
微穿越
广播
德国新天鹅堡
德国 Marienbrücke
9100公里
君霍茨
Jungholz
匹恩斯王格
穿越到这里，想说点什么呢~
2372

可优化的点?

Q & A

- More Framework for iOS Network Module
- ASIHTTPRequest
- AFNetwork