

Volley源码详解

王腾

- API的实现方式
- Volley请求流程
- 源码解析
- Q&A

JSON请求

1、创建一个请求队列

```
RequestQueue mQueue = Volley.newRequestQueue(context);
```

2、创建一个请求

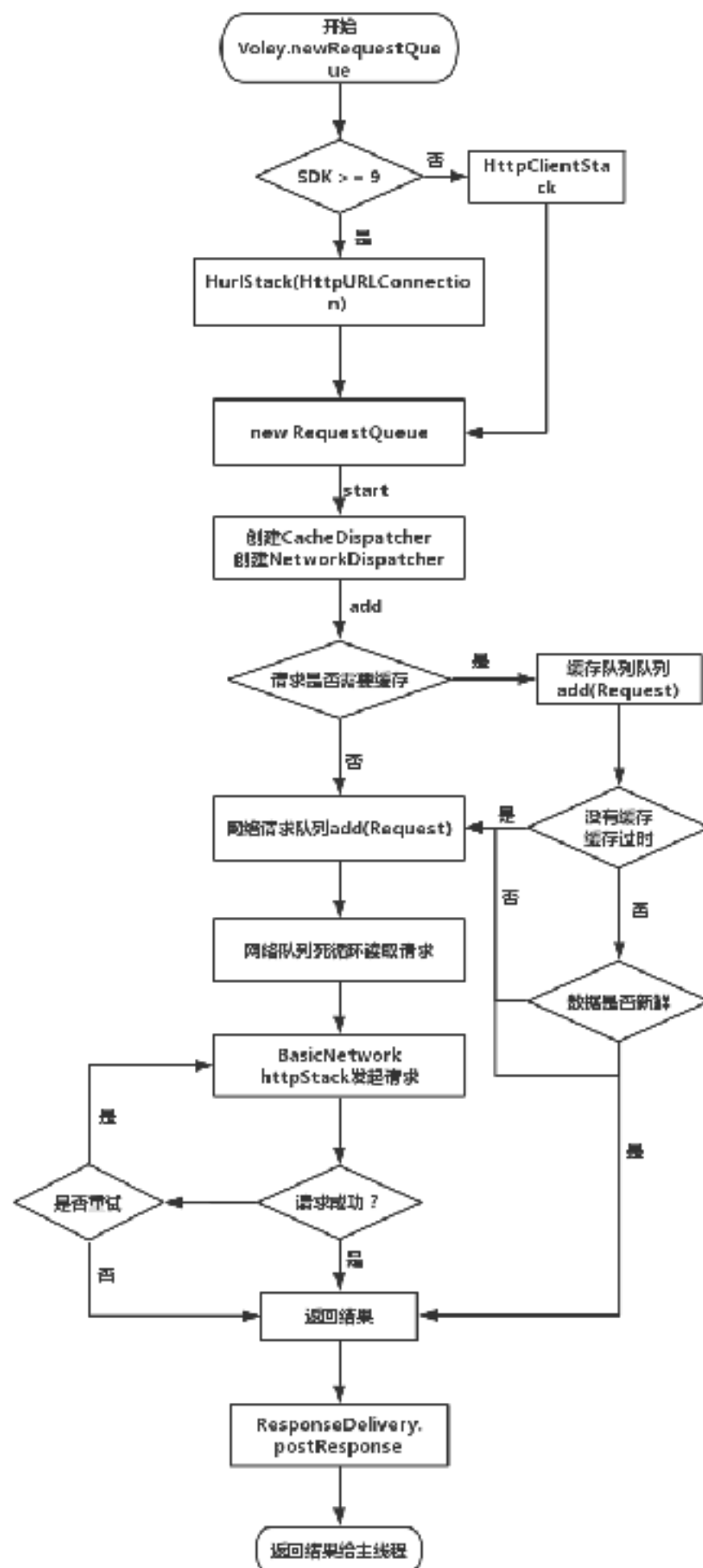
```
JsonObjectRequest request = new JsonObjectRequest();
```

3、将请求添加到队列中

```
mQueue.add(request);
```

4、请求结果回调主线程Callback

请求流程图



RequestQueue

JsonRequest

StringRequest

ImageRequest

自定义Request



Dispatcher Thread

CacheDispatcher

NetWorkDispatcher



Get Data

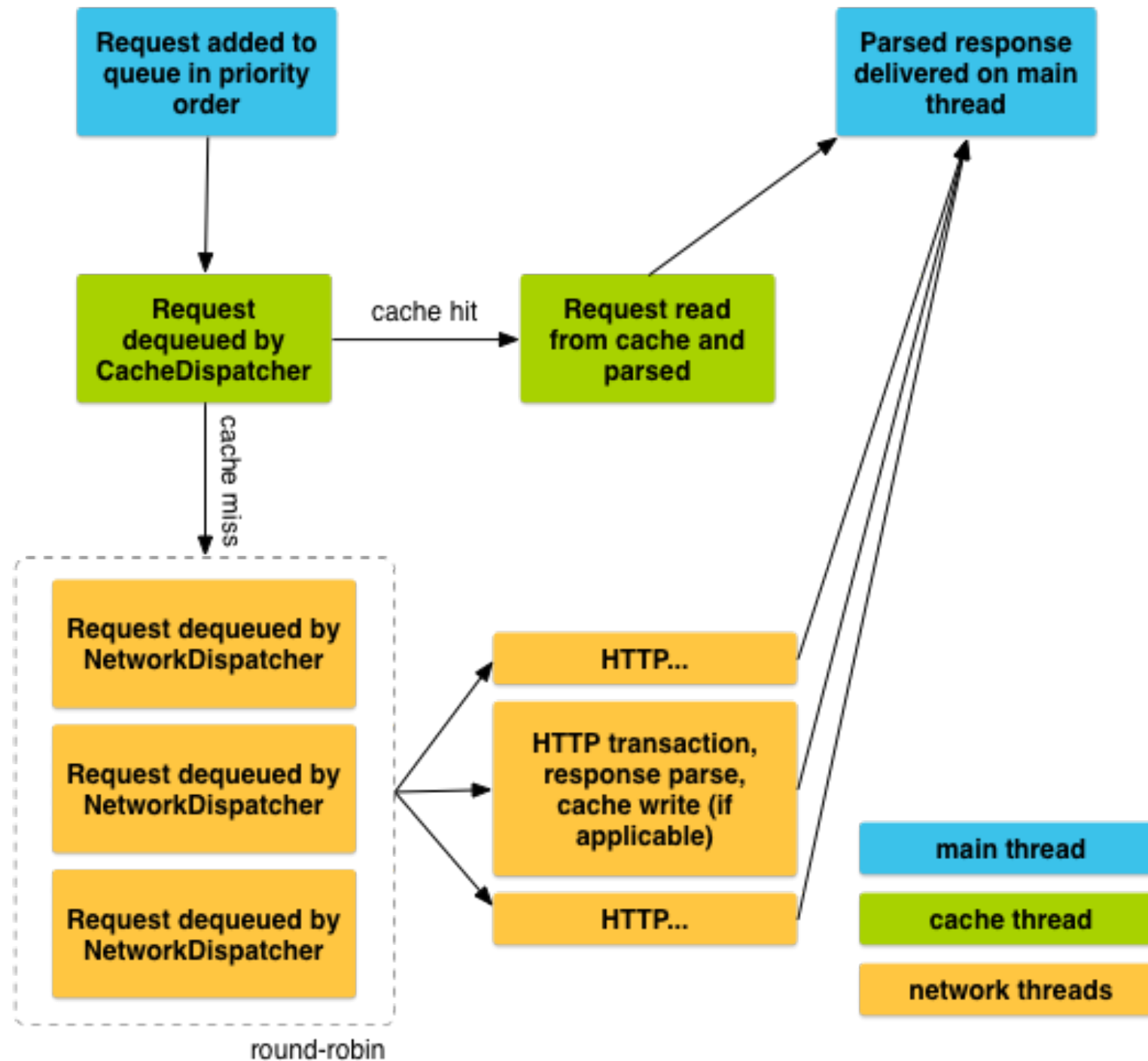
Cache

HttpClientStack/HurlStack/自定义stack

Questions

1. Volley适合做数据量小，高并发的请求？
2. Volley对请求的封装，最后结果回到主线程？
3. 并发？怎么管理线程？
4. 支持Https？怎么实现？ JDB的实现方式？
5. 和OKHttp的对比？

线程调度



Answer

1. `ByteArrayPool`? 在Volley中维护了一个`ByteArrayPool`用来缓存字节数组, 使用LRUCache的原则。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。
2. 回到主线程? 在`RequestQueue`的构造方法中创建了一个`Handler`, 拿到了`MainLooper`。通过`ExecutorDelivery`来发送给主线程
3. Volley中并没有使用线程池来管理线程, 而是启动了一个默认包含4个`NetWorkDispathter`线程的数组
4. Volley本身并不支持https协议, 需要我们写一个新的协议栈, 例如借贷宝的Volley中使用了`OkHttp`的协议栈

Volley与okHttp

1. okhttp包比较大，大约400k, Volley大约120k
2. okhttp封装比较麻烦，回调不在主线程，不能刷新UI，Volley回调是在主线程中的
3. okhttp支持Https和Http，volley只支持Http
4. okhttp中实现了差不多和URLConnection一样的API,没有版本的限制，volley有版本限制，不过可以自己扩展
5. okhttp可以做文件的上传下载，Volley由于有不适合做文件的下载

Q&A

Thanks