

# 实时通讯(长连接)--Manager



菠菜谈天说地

百家号 | 06-04 21:21

[关注](#)

## 前言

这次我们该来讲讲Manager了，本文会从四个方面介绍Manager：功能概览、消息增量更新的工程化、在线状态管理、响应低延迟优化。

## 功能概览

Manager主要有三大功能：Login、Request、Push，BreakLink。

### Login

当客户端与服务端建立链接之后，客户端会主动发起Login。Login中除了身份校验信息外，还会携带ClientID。校验Login请求后，Manager要将客户端的在线状态存储到Redis中以及将ClientID存储到Gateway的Session中。

### Request

客户端主动发起请求我们称作Request。Manager在解析Request的请求的时候，首先校验Session中是否有ClientID（即校验是否Login过），若没有Login，则通知Gateway关闭该链接，若有，则根据协议进行转发。

### Push

服务端主动下发消息我们称作Push。Manager收到Push请求后，会根据ClientID来查询是否在线，若在线则走Push流程（具体后边会讲）。

### BreakLink

Gateway检测到链接断开后（正常断开/死链清理断开），会告知Manager断开了，Manager会将Redis中的在线状态清空。

[去百度App听](#)

## 消息增量更新的工程化

之前的文章《微信如何让消息Push的可靠--消息的增量更新机制》已经介绍了消息增量更新机制，那我们这次会深入讲讲如何将消息增量更新机制工程化。

触发消息Push的有两个时间点，一个是在Login的时候，一个是在后端向Client Push消息的时候。

### Login

客户端在Login的时候，会携带MsgMaxID，这时候Manager会先将MsgMaxID存储到Redis中（BreakLink的时候会清除掉），然后走Push流程，即将大于MsgMaxID的消息推送给Client，同时会启动一个清理线程，将小于MsgMaxID的删除掉。

不过要注意一个点，一个是在推送的时候必须保障整个Manager集群中，只有一个线程在推，否则就有可能将同样的消息推多次给客户端，这里可以参考《性能优化--锁、缓存》中，分布式锁的实现。大体流程如下：

1. 加锁，若得到锁，则继续，否则退出不推
2. 基于Redis的消息存储，可以通过ZRangeByScore的方式获得大于MsgMax的方式将小于MsgMaxID的消息删除。
3. 获取消息后将消息按照顺序推送给Client，并更新MsgMaxID为最大的消息ID
4. 解锁。
5. **（注意这部分很重要）**，再检查是否有大于MsgMaxID的消息，若有，则继续重复上述步骤。这一步很关键，是为了避免在推送期间有新消息，而没有及时给客户端推出去。

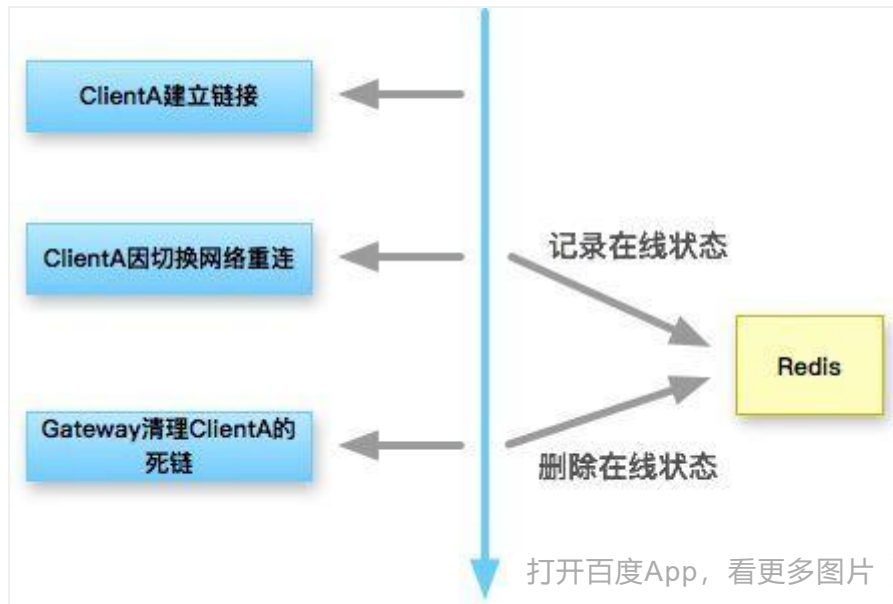
### Push

后端服务在向Client Push消息的时候，Manager会先给Msg生成一个唯一自增的ID，然后再执行上边的Push消息流程即可。

去百度App听

## 在线状态管理

通常的在线状态管理，会借助中间存储，比如Redis，以ClientID为Key，将LinkID存入进去，而这种方式在实际当中会遇到坑，就是本来实际在线的用户，而Redis中却不存在，后来发现是误删了，具体坑产生的原因如下图：



ClientA建立链接，然后因为切换网络所以进行了重连，而老的链接因为无法及时握手关闭导致等待清理死链的时候才被发现。这样清理死链在重连之后，所以导致误删的情况出现。菠菜是借助Redis的HashMap来解决，具体操作如下：

```
// -- 添加在线状态
Del ClientID
HSet ClientID LinkID MsgMaxID

// -- 删除在线状态
HDel ClientID LinkID

// -- 获取在线状态
HGetAll ClientID
```



去百度App听  
菠菜聊代码

添加在线状态的时候，先将ClientID给删除掉，然后HSET进去，这个保证了在获取在线状态的时候，虽然使用的是HGETALL，但永远只会获取到一个。

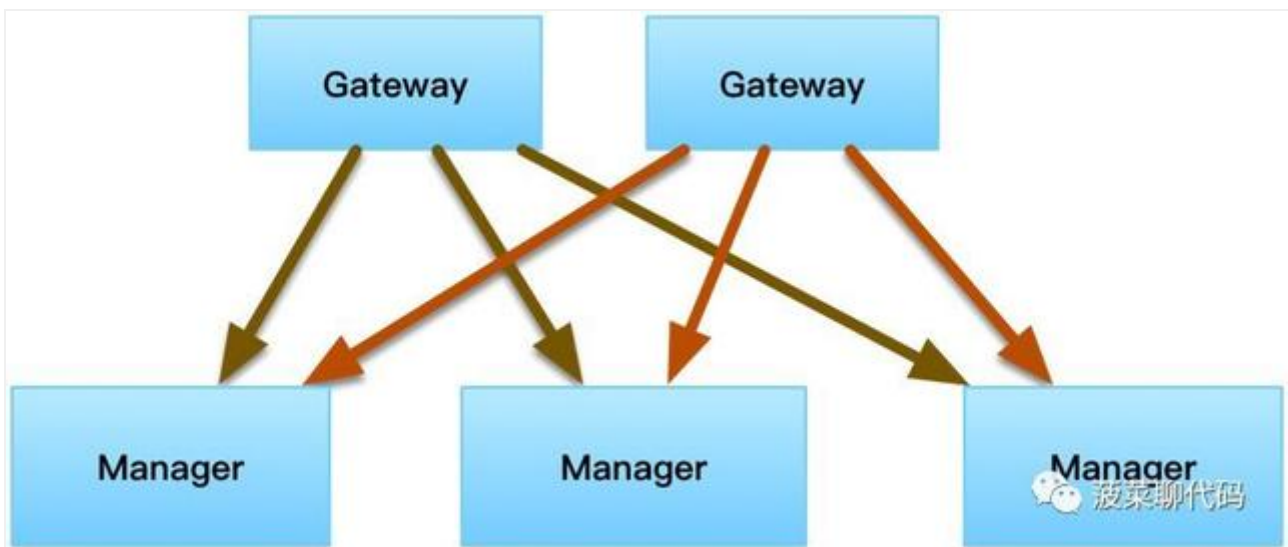
清除在线状态的时候只删除该链接的LinkID。上图的情况，HashMap中会存在NewLinkID，OldLinkID是不存在的，当清理死链的时候，去尝试删除OldLinkID而不成功，没有误删。

## 响应低延迟优化

由于接入长连接的流量越来越多，导致许多Request的耗时在几百毫秒甚至上秒，为了将请求耗时优化在100ms左右，菠菜从两个方面入手：部署方案的优化和长尾流量的优化。

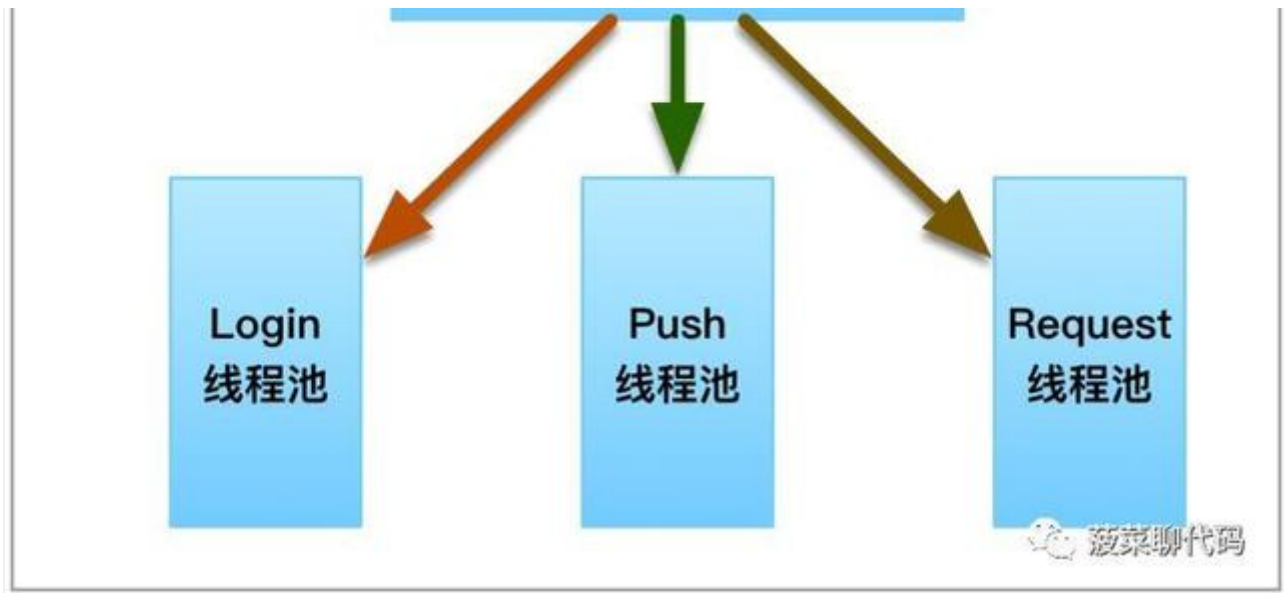
### 部署方案的优化

第一版的部署方案是混部，如下图：

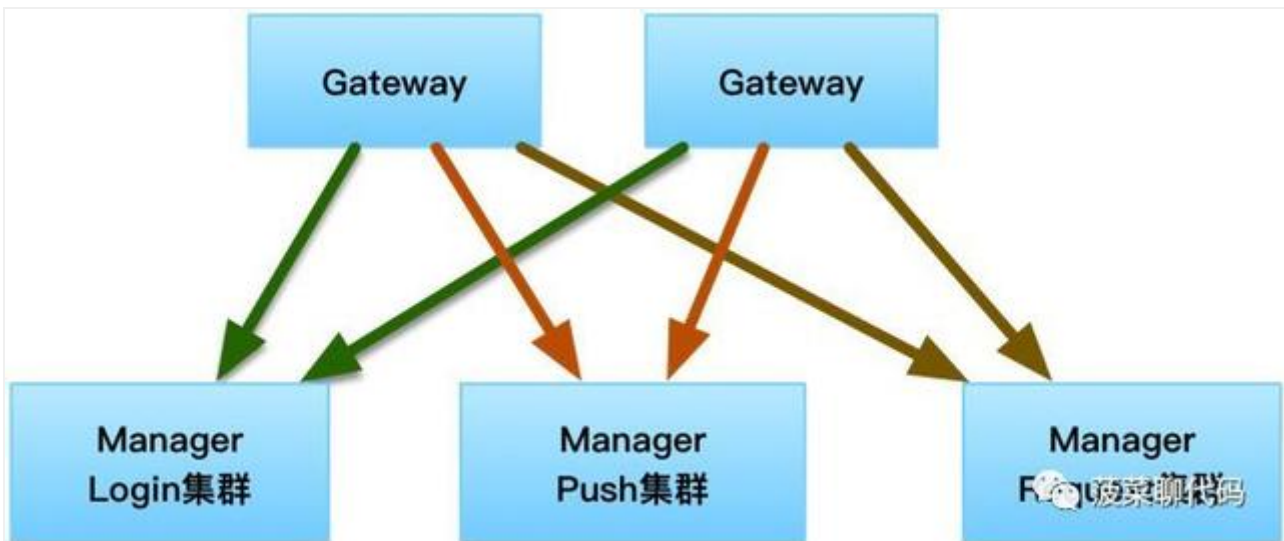


所有的Manager都处理Login、Request、Push，发现彼此会争夺资源，导致响应延迟高（占流量的5%），针对这个现象，菠菜优化了一版，如下图：





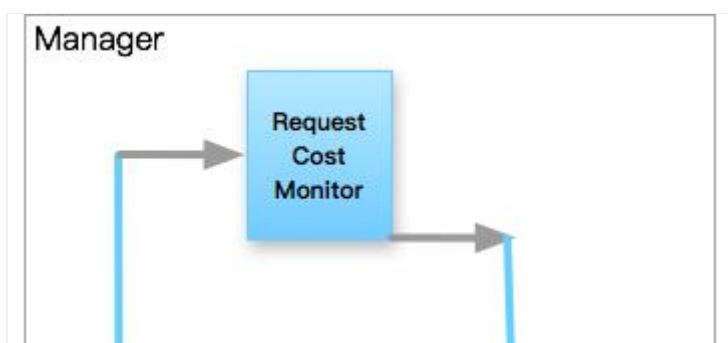
整体部署方案不变，在Manager内部设置不同的线程池做不同的事儿，后来发现这种软隔离也不是很有效，仅从5%降低到4.5%，然后继续改进如下：



不同功能分集群后，改善了很多，然后针对不同的集群深度优化响应延迟，这样最终控制在1%左右的流量会有高延迟。

### 长尾1%流量的优化

为了优化这1%，菠菜去追寻原因发现，就是因为偶发的CPU调度或IO延迟导致的，菠菜用了一个简单粗暴的方式来解决，如下图：



去百度App听





当发起请求的时候，RequestCostMonitor 就启动监控，如果 Monitor 发现 100ms（这个时间需要根据具体业务来定制）内请求还未结束，就再发起一个请求。不过要控制重试的流量，否则会带来服务雪崩的风险，那我们是为了优化1%的流量，我们就将重试的流量控制在1.5%左右。这样从1%的高延迟成功的优化到了0.01%。

最后

呼，长连接系列文章到此结束，其实还有很多细节没有在文章中体现，不过菠菜已经尽力将关键点罗列出来，如果小伙伴们有不理解的，欢迎前来讨论。

本文由百家号作者上传并发布，百家号仅提供信息发布平台。文章仅代表作者个人观点，不代表百度立场。未经作者许可，不得转载。

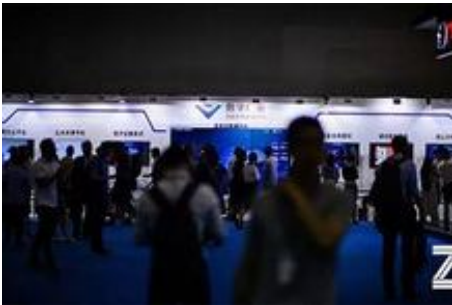
举报

相关阅读

20年前它叫QQ，20年后它是连接一切的腾讯



[打开百度App](#) [和讯网](#)



讯达云教你零门槛搭建实时音视频通信平台

[去百度App听](#)

[打开百度App](#) [SpeedyCloud讯达云](#)



# 数据看李霄鹏继承和改变 manager模式致鲁能飙升

[打开百度App](#) 新浪体育

打开百度App看更多资讯

## 精彩视频

隐患最低的智能宠物门，连接手机实时监测，钥匙丢了也不用担心

01:06

[打开百度App](#) 新小白研究所

在海里跑的火车见过么？11公里长连接德国小岛和大陆，票价50块

02:01

[打开百度App](#) 溜溜达达看世界

南二环沿线8个电子抓拍系统将启用，对违法违规行为实时抓拍

02:35

[打开百度App](#) 要闻望远镜

冰清实时聊天：成都市区喜提今冬第一片雪

00:10

[打开百度App](#) 封面新闻

## 热门推荐

最便宜的跨界SUV轿跑，主打操控，百公里仅6个油，仅卖8万很超值

[去百度App听](#)



打开百度App 老白说车

夏天穿着几款鞋子美爆了，怪不得朋友给我推荐，舒适又时尚！



打开百度App 小苹果的时尚衣柜

还在买途观L？这辆车动力比它强、配置比它高，还便宜整整9万



打开百度App 侃车e族

腰间盘突出严重影响生活！骨科专家力推的一个方法，一劳永逸！



打开百度App 东方头条

搞笑趣图

没有什么比这更直白纯粹了

去百度App听





打开百度App 蠢萌爱宠



爆笑趣图：一千万一次，一千万两次，成交。



打开百度App 奇葩小伙伴

中国现代四大壁画：办证，开锁，助考，通下水。



打开百度App 搞笑奇趣汇

爆笑动态：女神就是一种折磨人的动物



打开百度App 熊孩子基地

热门评论

去百度App听

打开百度App,说说我的想法