服务端问题

目录

[1 Player对象问题 1](#_Toc4988)

[2 Legion对象问题 5](#_Toc4442)

[3 Global问题 6](#_Toc10844)

[4 服务器ref数据加载耗时过长 7](#_Toc18036)

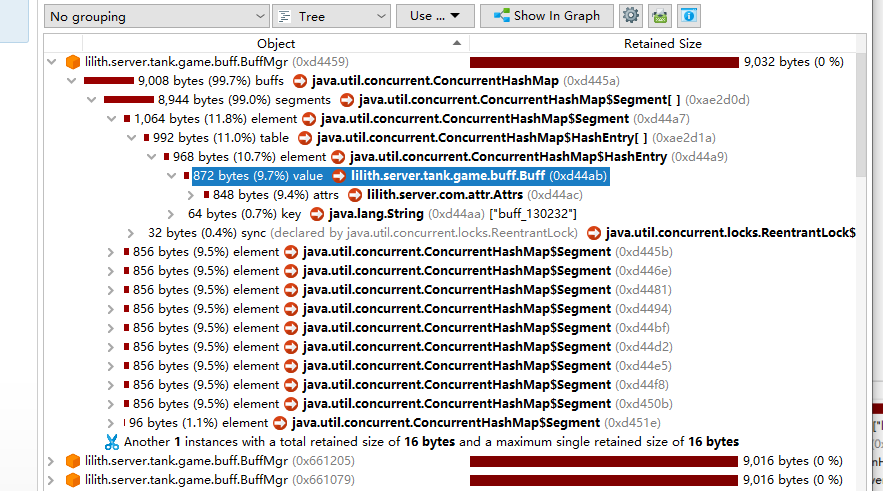
[5 编码要求 7](#_Toc8191)

[6 工具数据参考说明 7](#_Toc8280)

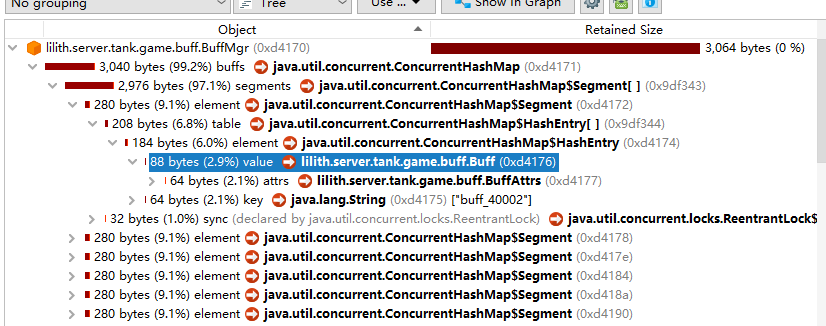
# Player对象问题

## 玩家buff问题

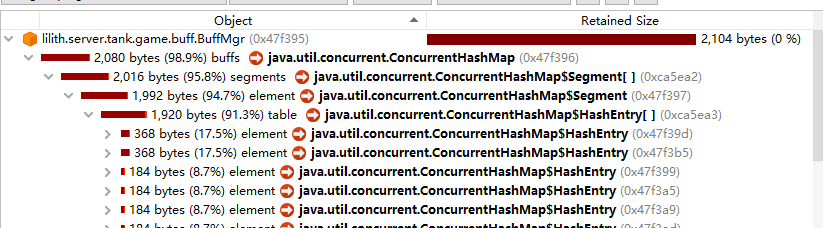
* 未优化BuffMgr内存占比图



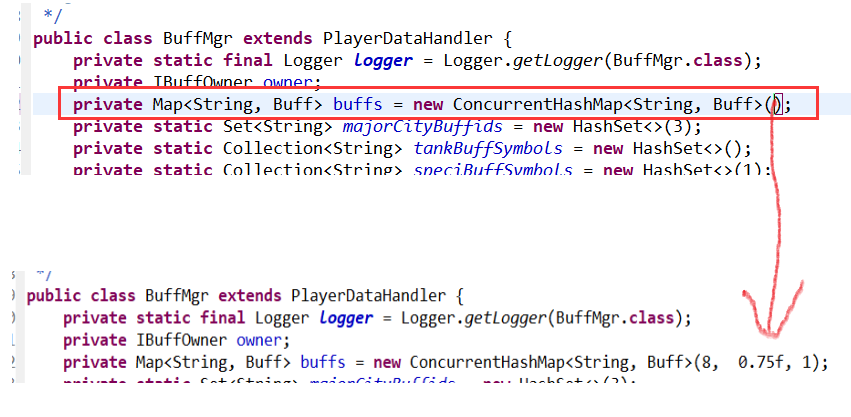
* 使用javabean替换attrs内存占比图



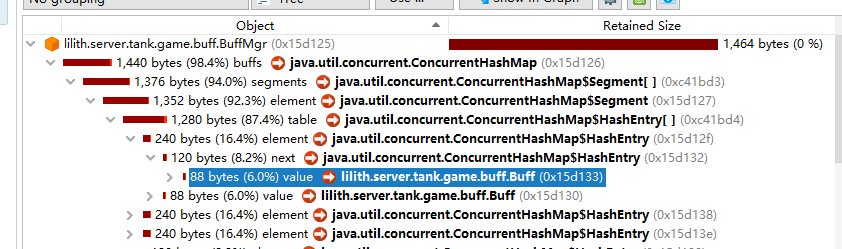
* 使用javabean替换attrs并且优化ConcurrentHashMap参数内存占比图



* buffMgr ConcurrentHashMap参数修改图

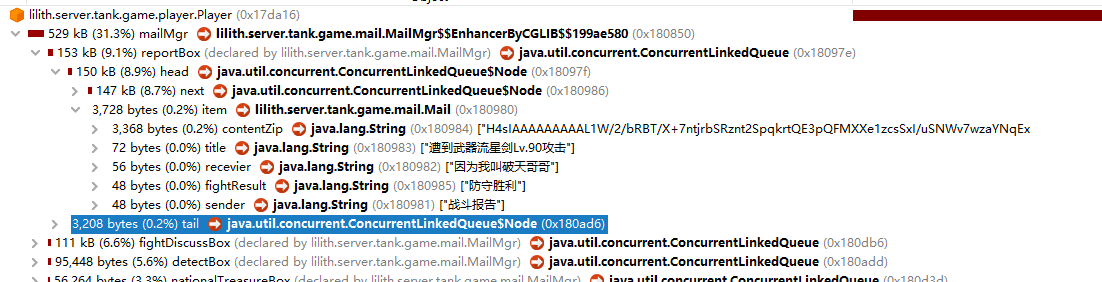


* 使用String.intern优化内存占比图

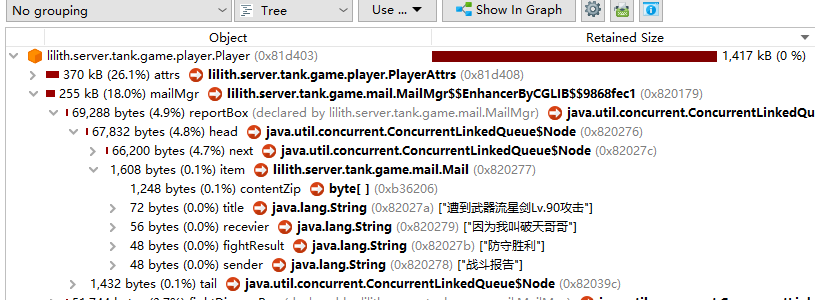


## 玩家邮件问题

* 未优化MailMgr内存占比图



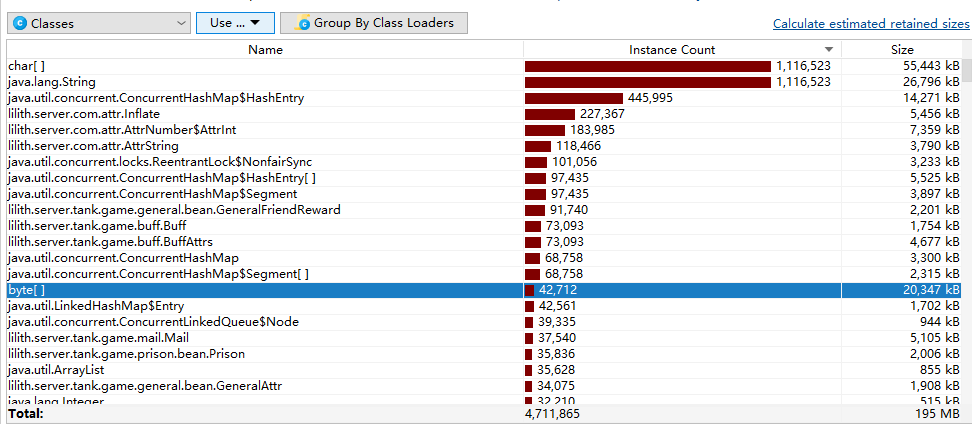
* attachmentZip/contentZip直接以压缩字节存储内存占比图



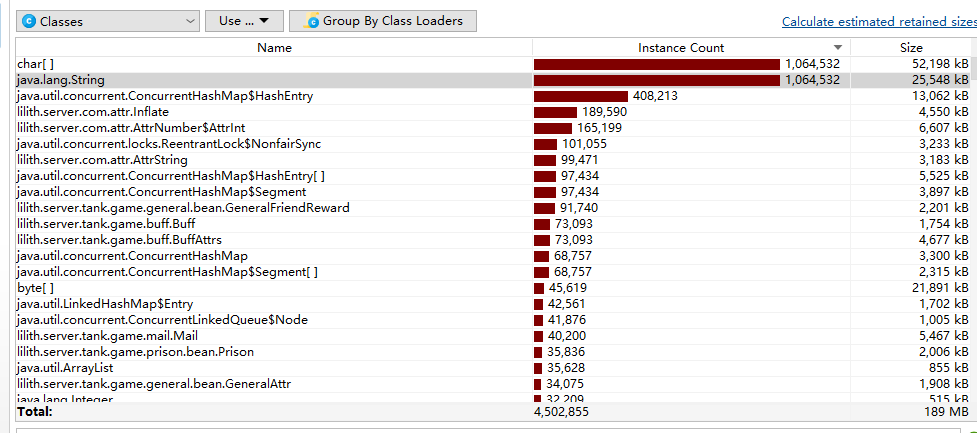
可以看到mailMgr从529kb降低到255kb，减少了50%

## 玩家attr组合key问题

* 对玩家属性key进行采样分析
* 总共3067个key
* 其中TodayMoveCityCount\_xxx相关的key占了1992个属性key
* 213玩家内存占比图

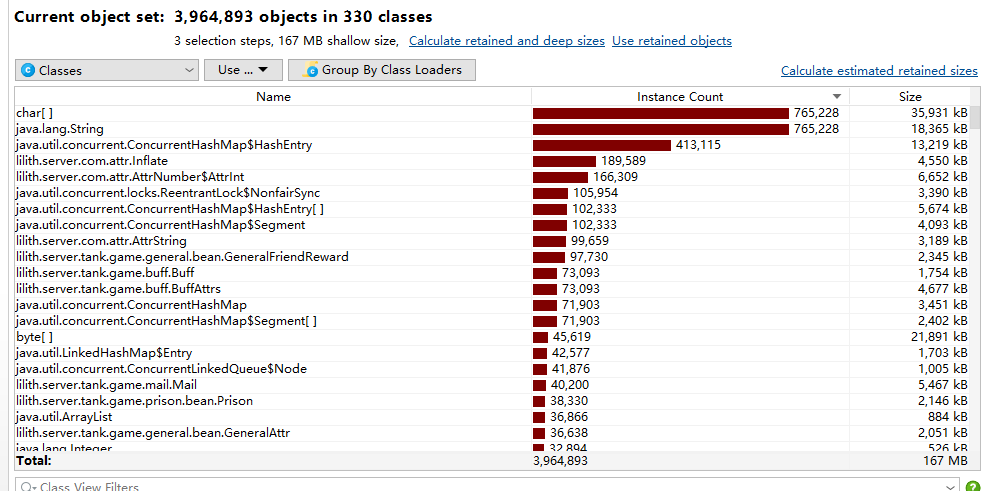


* 删除TodayMoveCityCount相关key内存占比图



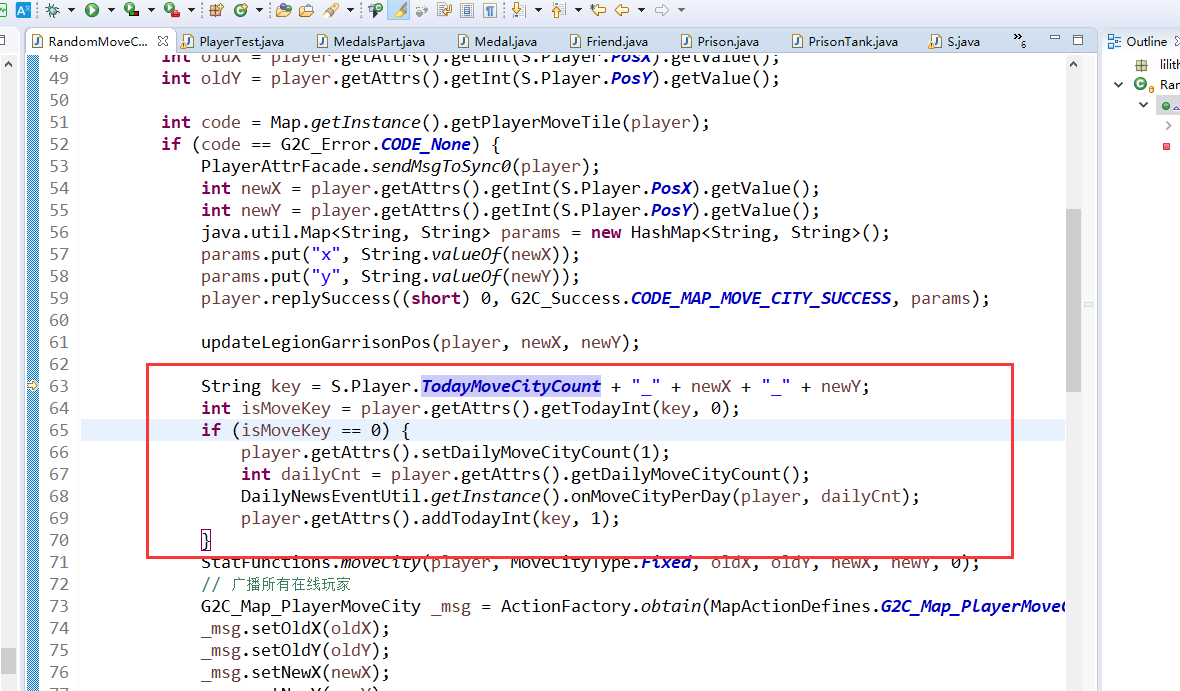
213个玩家节约了5M内存

* Attr里面的symbol.intern内存占比图



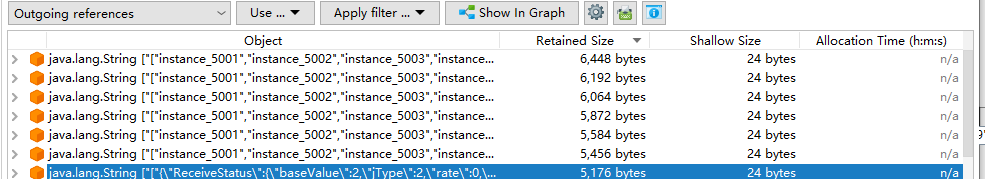
213个玩家节约大概20M内存

* TodayMoveCityCount相关代码

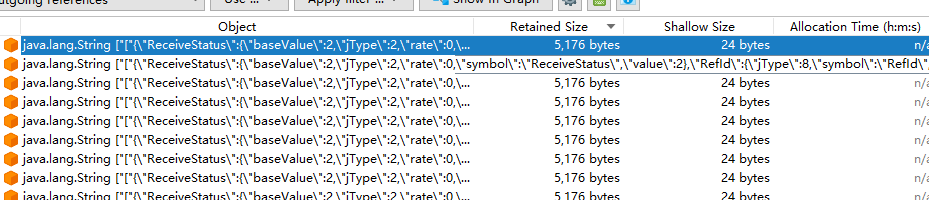


## Attrs.getJsonList/setJsonList相关问题

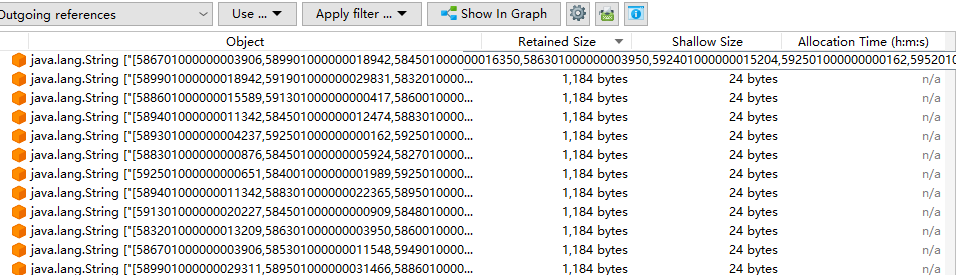
* instUltimateExplorePasses属性



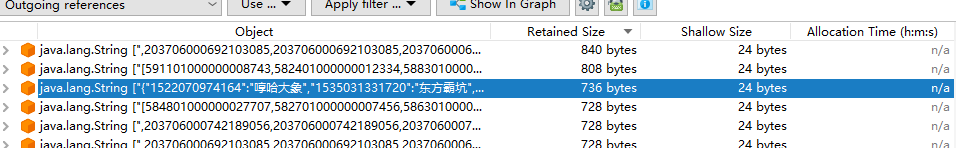
* PointRewards属性



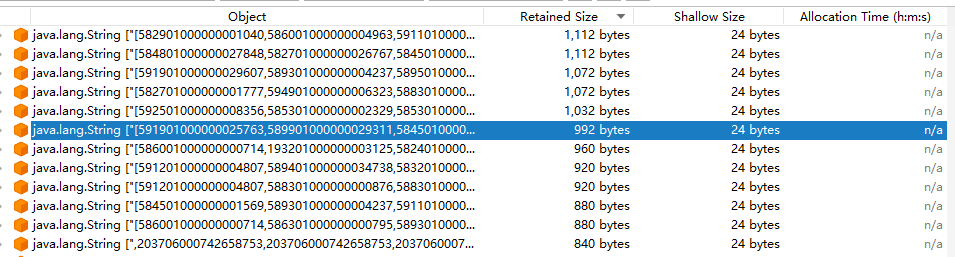
* LatelyRelation 属性



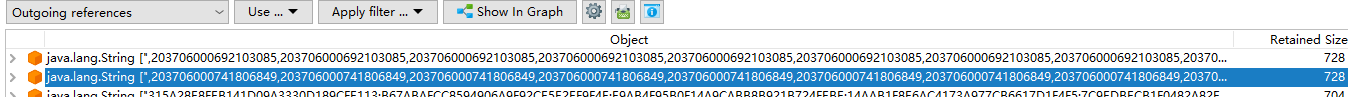
* usedNames属性(无限)



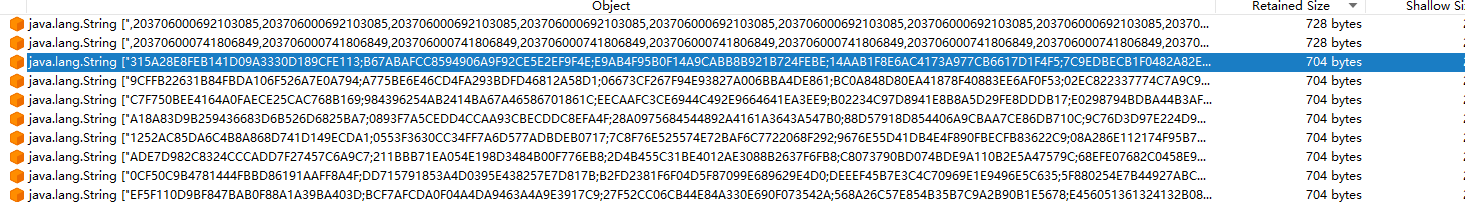
* MineList属性



* ballotReportIds属性



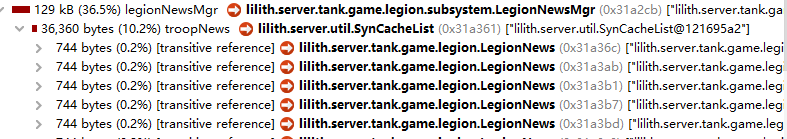
* LastOrderIds属性



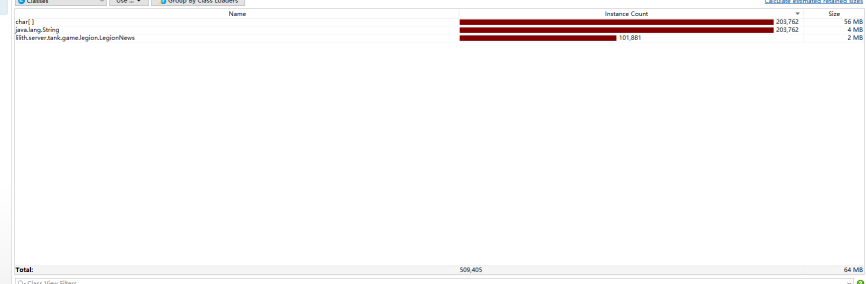
# Legion对象问题

## LegionNews问题

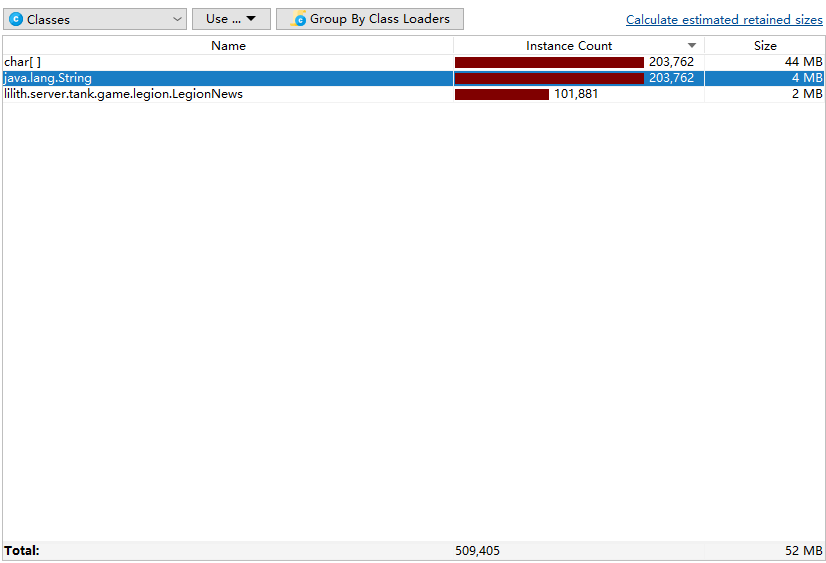
* LegionNews数量也非常多，单个对象占比接近了1kb



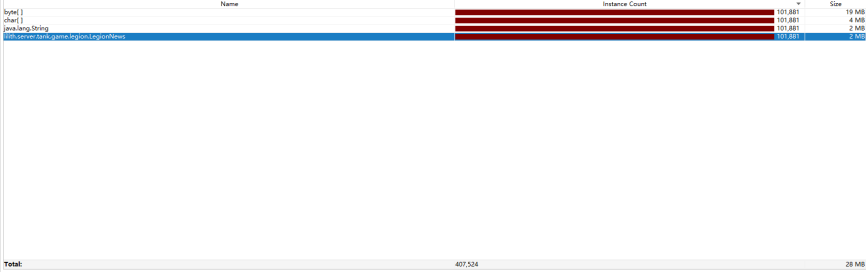
* 所有LegionNews，legionNewsDesc压缩之后进行base64编码图



* LegionNews.legionNewsDesc不压缩直接存储图



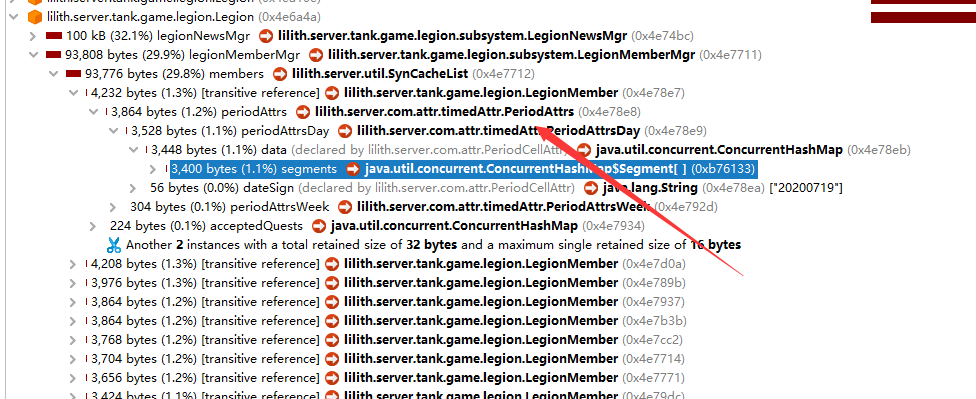
* LegionNews.legionNewsDesc直接以压缩格式存储内存占比图



直接改成压缩LegionNews.legionNewsDesc，不以字符串形式存储，LegionNews在现有的基础上可以降低50%

## PeriodAttrs问题

* PeriodAttrs内存占比图

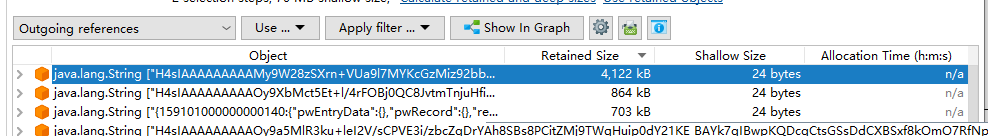


PeriodAttrs 底层存储了Attr相关的对象设计，沿袭了Attr耗费内存的特性

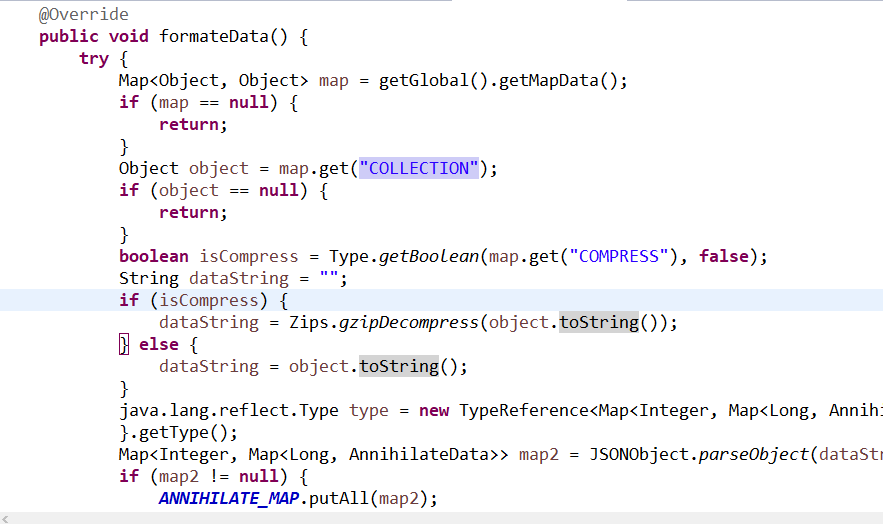
# Global问题

## COLLECTION问题

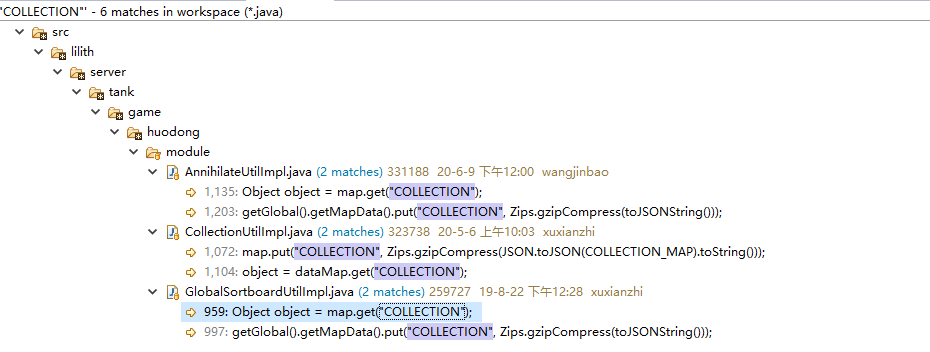
* COLLECTION单个字符串问题,占比4M



* 将数据反序列化之后原来的数据不清除



* 压缩之后使用base64编码，导致数据量变大
* Save 之后不会进行相应的清除
* 这种问题不止存在一个模块



* 不删除是因为Global会自动对数据进行保存，删除之后就会清空数据

# 服务器ref数据加载耗时过长

现在服务器加载ref数据需要1分钟左右，加载所有ref数据耗时：56187ms

# 编码要求

* 废弃attrs相关模块，新的功能模块能别用就不用
* 新的功能模块不能使用attrs.getJsonList，相关的的函数废弃
* Zips.gzipDecompress/Zips.gzipCompress如果不是用在http传输中，尽量别用，会导致压缩率降低，新的模块存储都不要用
* 使用Global属性模块问题，把Global当成用来存储的中间对象，Global的自动存储功能都尽量别用，使用之后会导致内存中有双份数据
* Attrs里面尽量别用组合key

# 工具数据参考说明

* JProfiler
* <https://www.jb51.net/softs/608640.html>
* 线上正式服数据
* 内网ip: 192.168.16.33
* 目录: /home/ccj/mysql.sql