> 点赞再看，养成习惯，微信搜索【**三太子敖丙**】关注这个B 站用户。

> 本文 **GitHub** https://github.com/JavaFamily 已收录，有一线大厂面试完整考点、资料以及我的系列文章。

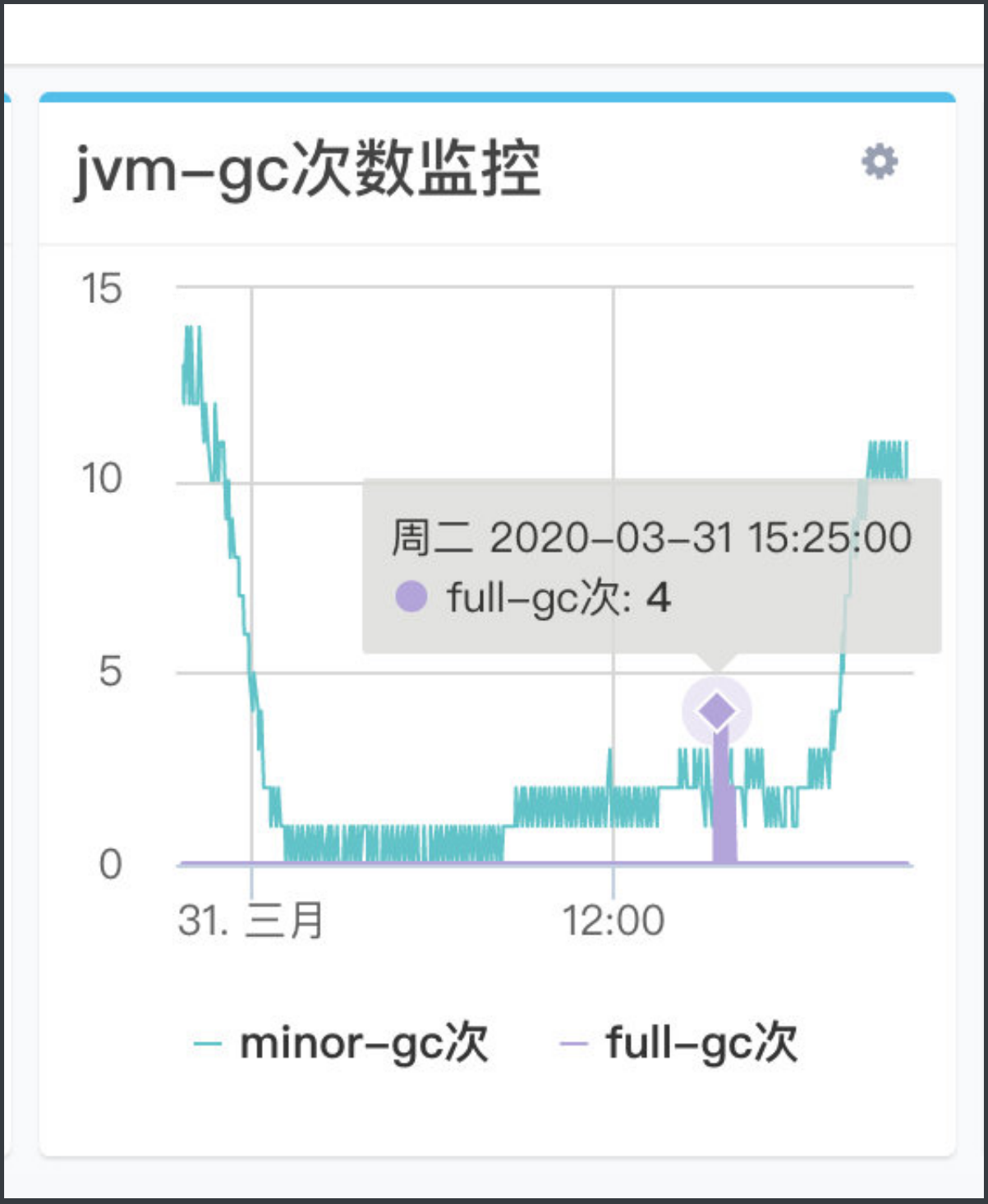
你们没发现我最近的原创原创少了很多嘛，一是最近花了很多时间做视频，本来我写文章就是利用周末的两天时间，但是现在基本上两天都要拍摄剪辑了，尽管请了小伙伴做字幕，还是得耗费大量时间在拍摄和剪辑上。

所以我只能利用工作日熬一下才能写出来了，这周因为要发布，本来是没排期写文章的，你们也看到了我发了两个视频嘛。

但是今天我一发布就吓尿了....

事情是这样的.....

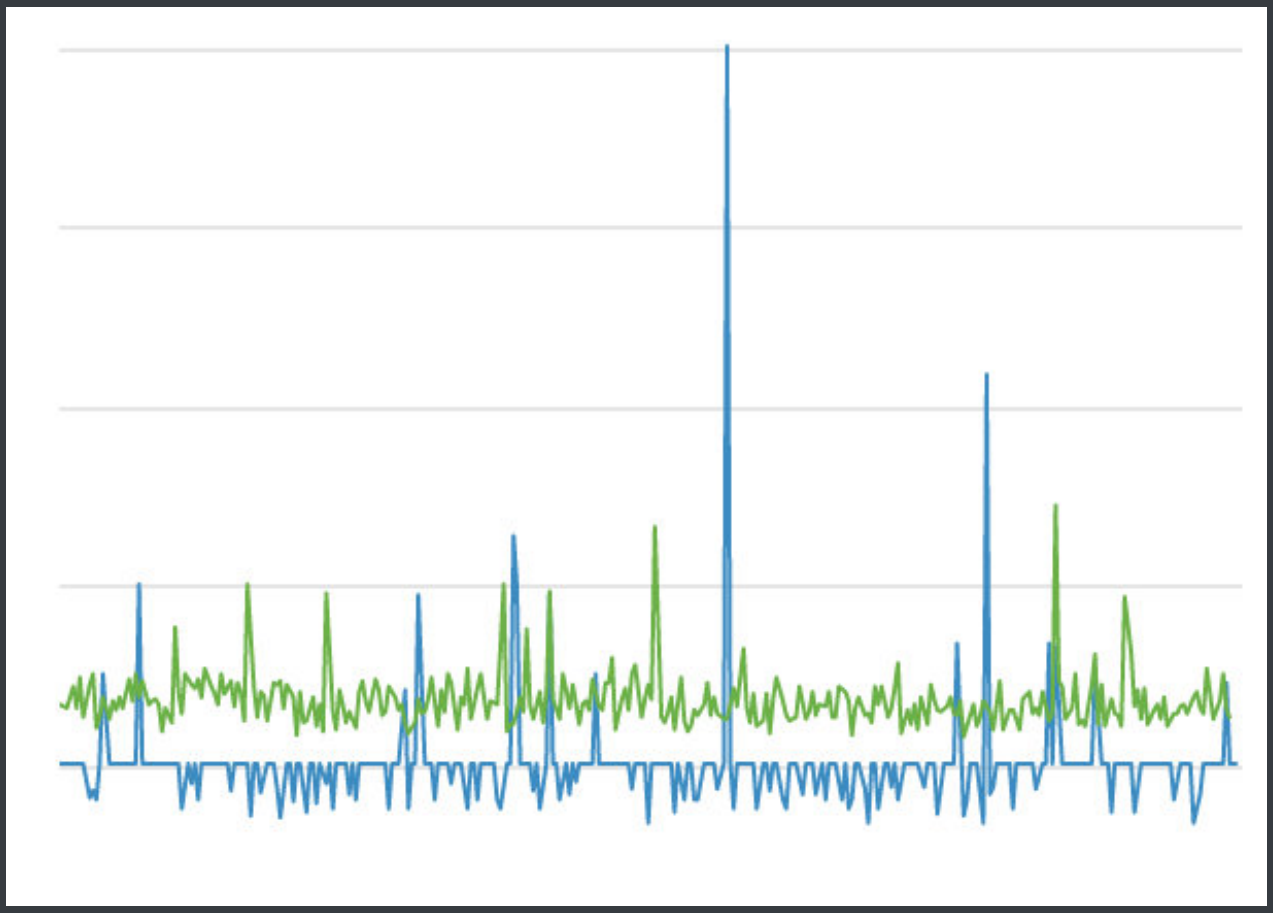
我和小组伙伴最近负责的系统今天上线了，因为涉及到的业务特别多所以选择白天发布，怕出问题找不到对应的负责人。



看到3点25那个full gc没，我发布上去一分钟不到就疯狂GC，我当时就吓尿了，马上点了回滚按钮。

	发布被终止	敖丙

而且我发现cpu几乎在我发布的瞬间，直接打到了峰值。



所幸只发布了一台，我紧张的望向了周围，发现周围的同事没注意到我，我顿时没那么紧张了，拿起桌上用剩下的纸巾搽掉了我鬓角马上留下的汗水。

我其实已经知道大概是什么问题了，无非就是死循环，或者大对象什么的。

但是我还是想找个人承担这个锅。

我马上找到三歪，我问他：你刚才是不是点过我的系统?

他说：对啊，我刚才点了，我本来就经常使用你的系统，咋了？（刚好正中下怀嘻嘻）

我说过多少次啊，叫你不要乱点，这已经不是第一次了，现在把我的系统点坏了，怎么办？都怪你。

他：委屈巴巴一副要哭的样子。

我：诶，算了算了，下次注意哈，我去排查下什么情况，看看你点坏了哪里。

他：丙哥真好，丙哥真棒，丙哥我爱你。

总算是找到背锅侠了，我就开始排查问题吧。

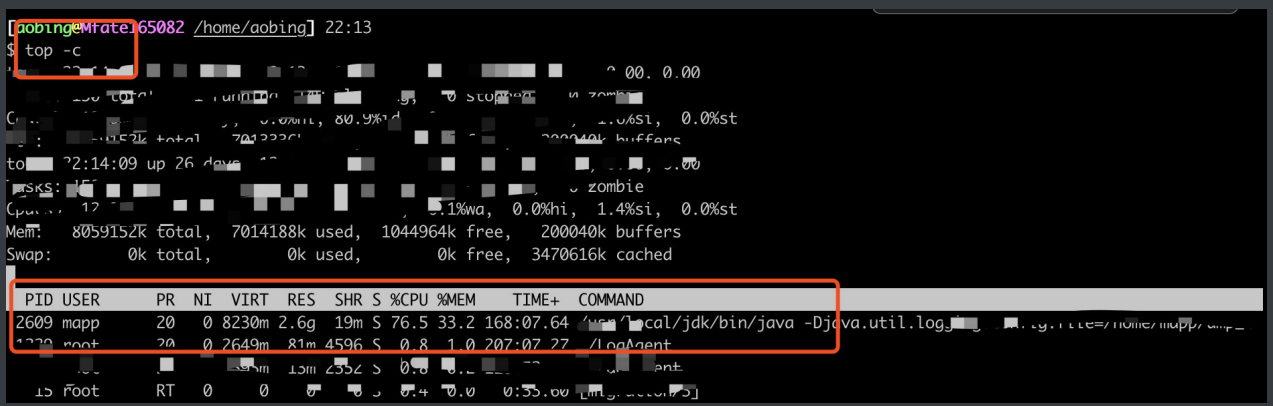
注：以下代码都是伪代码，为了还原排查过程

一般CPU100%疯狂GC，都是死循环的锅，那怎么排查呢?

先进服务器，用**top -c** 命令找出当前进程的运行列表

按一下 **P** 可以按照CPU使用率进行排序

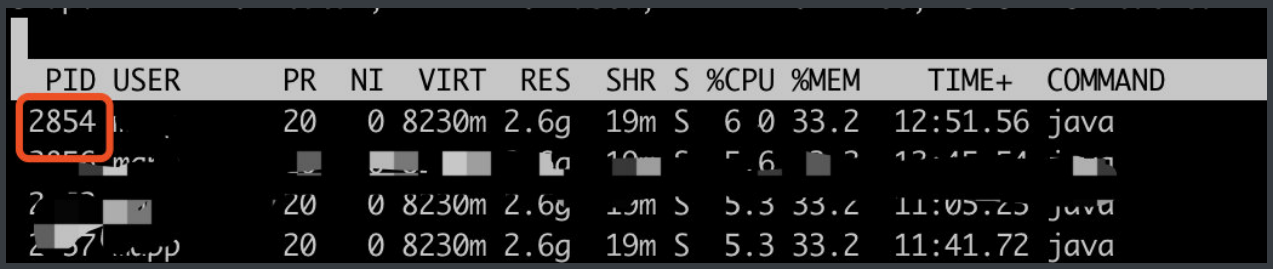
显示Java进程 PID 为 2609 的java进程消耗最高



然后我们需要根据PID 查出CPU里面消耗最高的进程

使用命令 **top -Hp 2609** 找出这个进程下面的线程，继续按P排序

可以看到 2854 CPU消耗最高



2854是十进制的，我们需要转换为十六进制，转换结果：b26

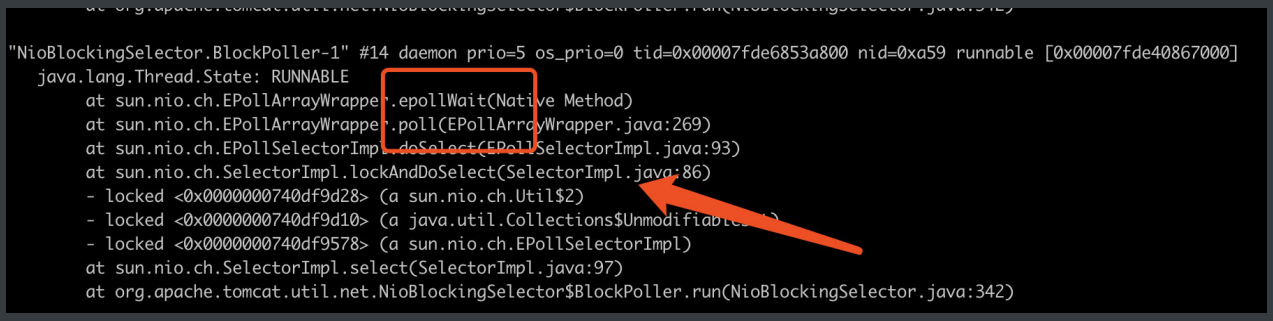
接下来就需要导出我们的进程快照了，看看这个线程做了啥

```
jstack -l 2609 > ./2609.stack
```

再用grep查看一下线程在文件里做了啥

```
cat 2609.stack |grep 'b26' -C 8
```

我这里就随便定位一个，基本上这样查都可以定位到你死循环的那个类，那一行，这里你还可以在jstack出来的文件中看到很多熟悉的名词，至于是啥，你们留言告诉我好了，就当是个课后作业了。



我写了个伪代码，看看当时我为啥会写出这个死循环，对了当时我上线的是预发，也是后台系统非线上的，虽然都是自己在玩，但是大家还是要引以为戒。

```
public void checkDate() {
    int index = 0;
    for (; ; ) {
        Map<String, Object> map = new HashMap();
        map.put("pageIndex", index);
        map.put("pageSize", 10);
        [illegible];
        dmpTags.forEach(e -> {
            String liveEndTime = e.getLiveEndTime();
            SimpleDateFormat sdf = new SimpleDateFormat( pattern: "yyyy-MM-dd"
            try {
                Date parse = sdf.parse(liveEndTime);
                if (parse.before(new Date())) {
                    return;
                }

                Calendar cal = Calendar.getInstance();
                cal.add(Calendar.DATE,  amount: 30);

                Date date = cal.getTime();
                e.setLiveEndTime(sdf.format(date));

            } catch (ParseException ex) {
                ex.printStackTrace();
            }

            [illegible]e);
        });

        if (dmpTags.size() != 10) {
            break;
        }
        index++;
    }
```

我当时写了个代码准备去查出数据库的数据，订正下日期，仔细看没问题，但是我忘了数据库的偏移值自己去计算了，因为以前都是框架，自己临时写的就没管。

导致每次都能查出10个数据，在最后判断的时候就一直true不退出了，其实最后会退出，但是得循环很多次。

我这里退出的逻辑比较取巧，就是想着最后一次查询肯定跟我的页数不一样，那就是最后一页了，那我就处理完退出。

结果没想到也是个坑了。

大家写的时候也要注意很多小坑，还有代码一定要本地测了再发，我这次其实是本地发现了，然后想着干脆就写个排查经过的吧，机制吧。

这个demo顺便带大家温习一下线上100%cpu故障的排查，下次我可能搞点内存泄露，集群宕机什么的故障，这样就有素材了呀。

如果真出问题，第一时间找个三歪这样的背锅侠，排查过程自己去电脑上操作一下，最近有点忙，准备写个分布式锁的文章，如何?

我是敖丙，一个在互联网苟且偷生的工具人。

最好的关系是互相成就，各位的**「三连」**就是丙丙创作的最大动力，我们下期见！

> 文章持续更新，可以微信搜索「**三太子敖丙**」第一时间阅读，回复【**资料**】【**面试**】【**简历**】有我准备的一线大厂面试资料和简历模板，本文 **GitHub** https://github.com/JavaFamily 已经收录，有大厂面试完整考点，欢迎Star。