

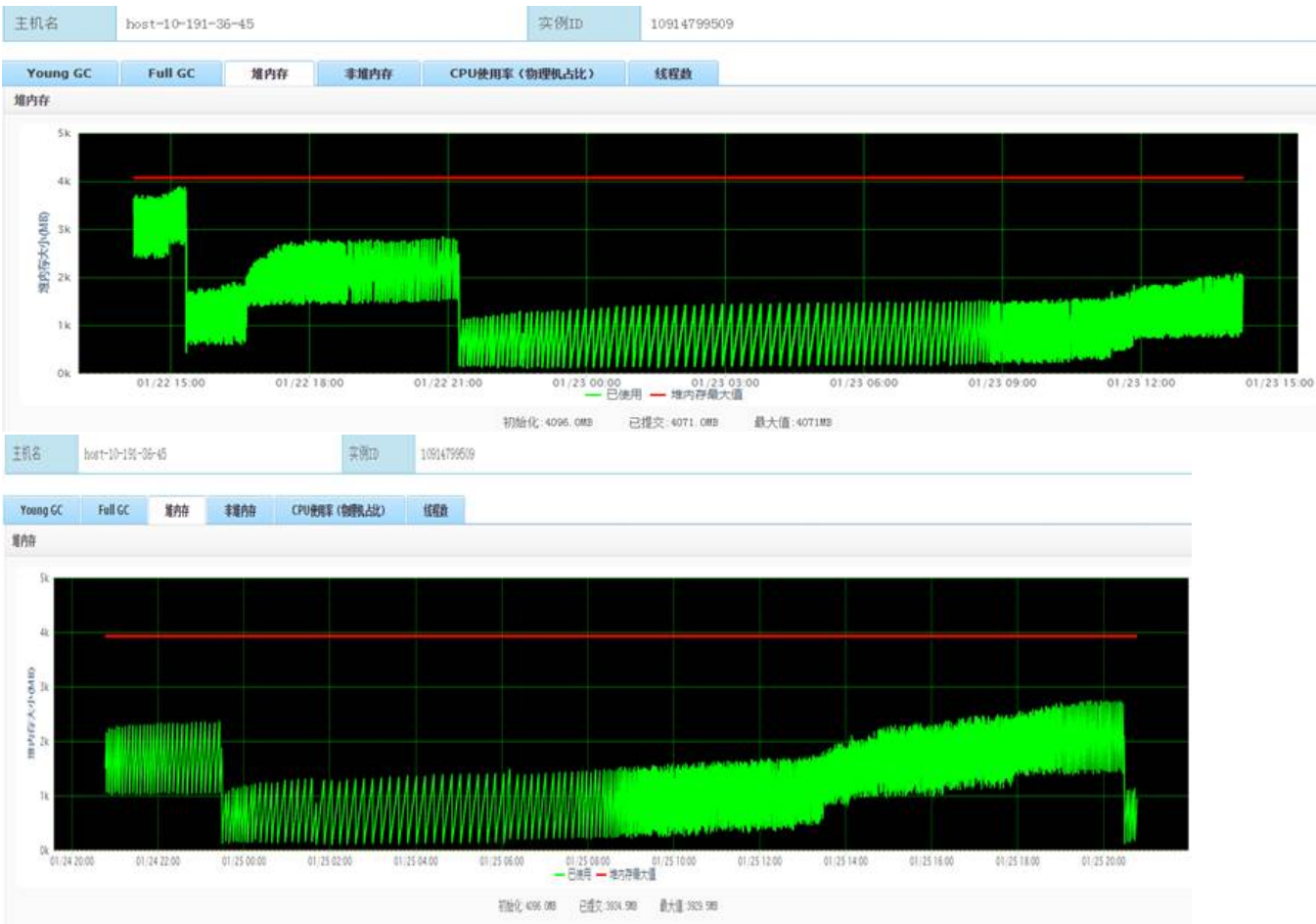
# JVM优化

## 一、测试结论：（以下图片编辑模式均可放大查看）

1、B商家批量导出功能，在gc方式为PS时，批量导出功能的对象java.util.zip.XXX在内存中持续增长，但是当达到JVM设置的-Xms4096m -Xmx4096m 时，垃圾回收正常，不存在内存泄漏问题。  
（生产环境和预发布压测环境垃圾回收都正常，不存在内存泄漏问题）

（1）、生产环境（10.91.36.45）：gc方式为PS，堆内存和full gc和young gc回收及资源使用情况（-Xms4096m -Xmx4096m -XX:MaxPermSize=256m ），例如在20：20左右触发一次full gc，执行完full gc后，图中堆内存释放正常， 且此时资源监控内存使用率也下降。  
每次垃圾回收，内存释放正常，垃圾回收正常，不存在内存泄漏。

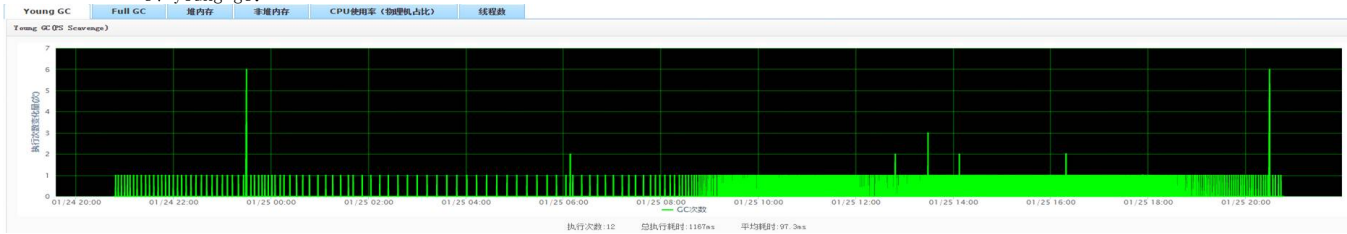
### a、堆内存：



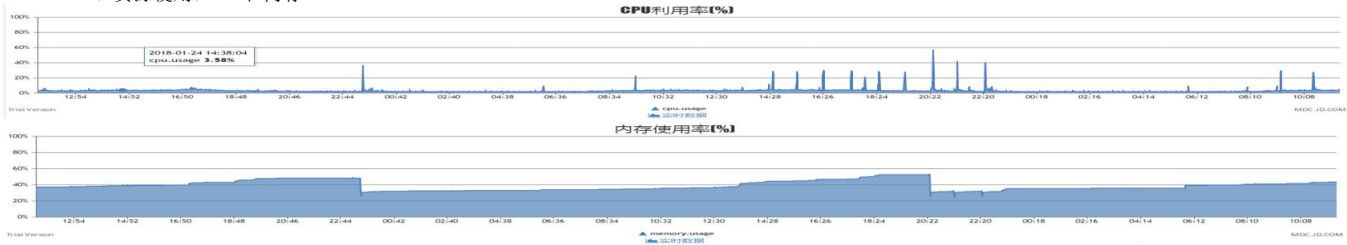
### b、full gc：



### c、young gc:

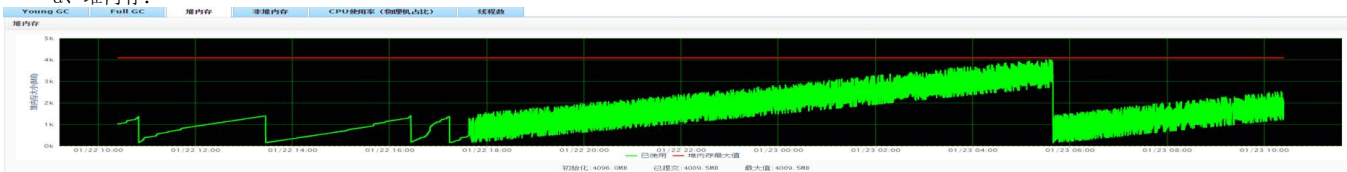


### d、资源使用：CPU和内存

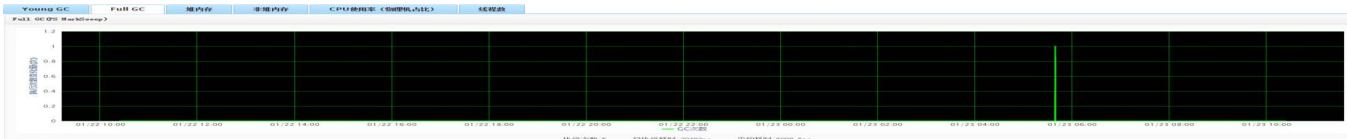


(2) 预发布压测环境：gc方式为PS，堆内存和full gc和young gc回收情况（-Xms4096m -Xmx4096m -XX:MaxPermSize=256m），垃圾回收正常，不存在内存泄漏，一次full gc耗时6098.6ms，耗时过长，导致预发布压测服务器在6098.6ms内无法提供服务

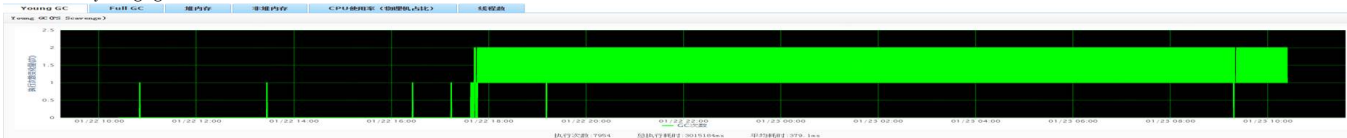
### a、堆内存:



### b、full gc:（一次full gc耗时6098.6ms）



### c、young gc:



### d、服务器错误日志：full gc耗时过长，导致预发布压测环境有6s左右无法提供服务

```
[ERROR]2018-01-23 05:39:18.304[--login] check kick user error!!!com.jd.jsf.gd.error.RpcException: [JSF-22103]Failed to call com.jd.passport.session.dubbo.UserServiceDubboRpcService.remoteCookieEffect on remote server: [jsf://10.187.127.170:22000/?safVersion=210&jsfVersion=1531&interface=com.jd.passport.session.dubbo.UserServiceDubboRpcService&alias=loginSession-mjq3.2.0]. cause by: com.jd.jsf.gd.error.ClientTimeoutException, message is: [JSF-22110]Waiting provider return response timeout. Start time: 2018-01-23 05:38:59.867, End time: 2018-01-23 05:39:18.225, Client elapsed: 0ms, Server elapsed: 18358ms, Timeout: 2000ms, MsgHeader: MessageHeader(msgId=5566311, length=657, codecType=2, msgType=1, protocolType=1, compressType=0, headerLength=19, keysMap= 1 : 2000 7 : 1601), Channel: 10.175.45.164:59486 -> 10.187.127.170:22000[com.jd.passport.remote.RemoteServiceCaller#:]http-nio-1601-exec-39]
```

## 二、遇见的问题及解决方案

### 1、存在的问题:

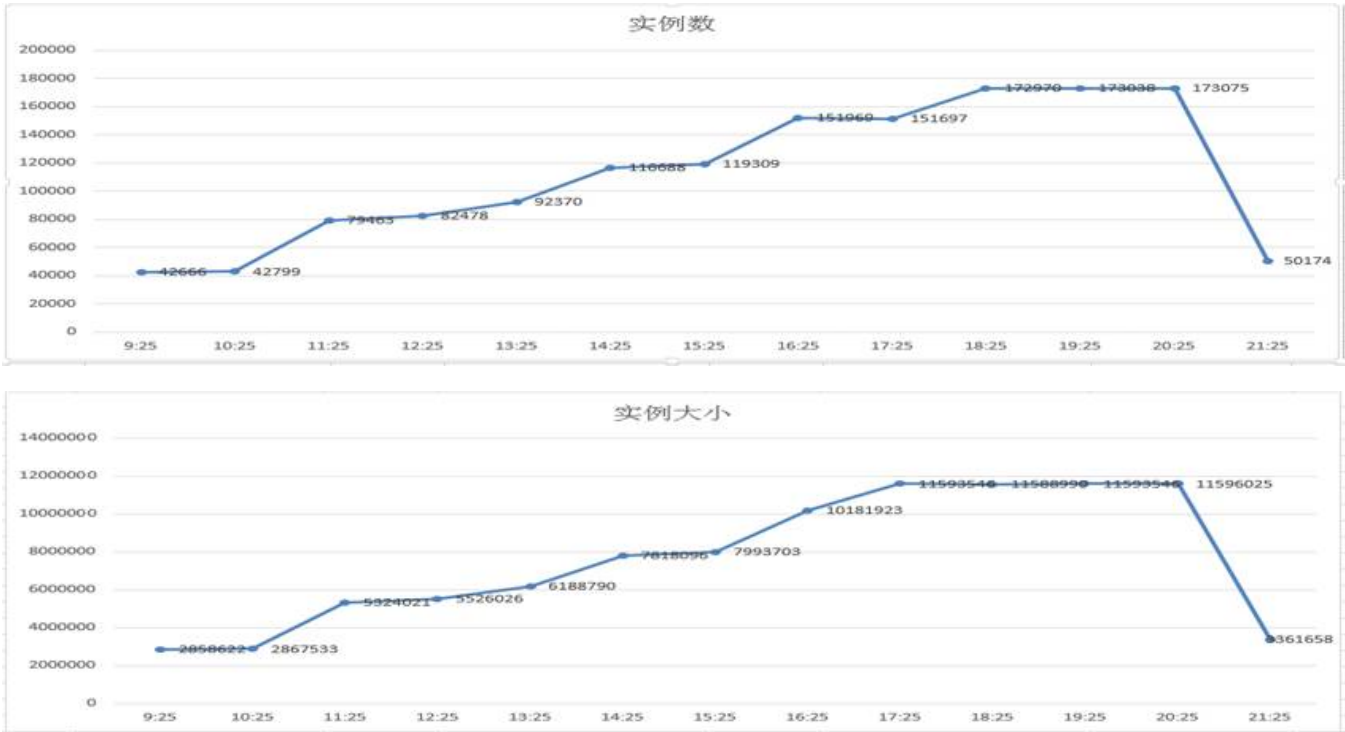
(1)、预发布压测环境，当gc方式为PS时，导出对象java.util.zip.XXX在内存中达到4096m，触发一次full gc，一次full gc耗时6098.6ms，耗时过长，导致预发布压测服务器在6098.6ms内无法提供服务。

(2)、预发布压测环境当gc方式为PS时，young gc频繁，耗时相对较高，平均耗时379.1ms。

(3)、生产环境中，当gc方式为PS时，堆内存中的对象实例较多的除了导出对象java.util.zip.XXX，还有打印对象com.itextpdf.text.pdf.XXX。下图为堆内存中对象实例较多的对象：

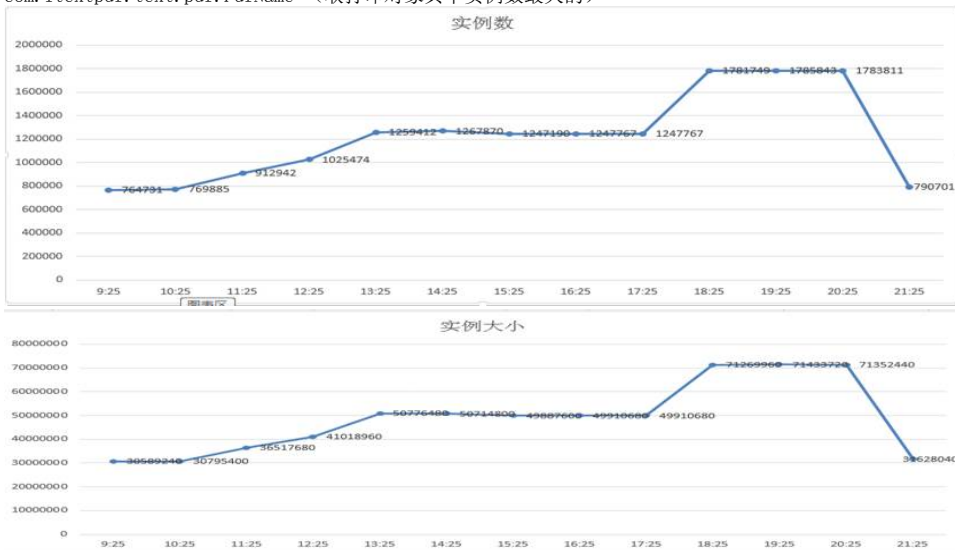
a、导出对象：（取导出对象其中实例数最大的）20:20分钟左右触发了一次full gc，导致堆内存中对象实例下降。

0 java.util.zip.Deflater



b、打印对象：

- com.itextpdf.text.pdf.PdfName （取打印对象其中实例数最大的）



## 2、解决方法：

已解决（1）、（2）中提到的full gc耗时多和young gc频繁的问题，将预发布压测环境的gc方式由PS改成CMS，且设置年轻代大小。

生产环境（10.91.36.45）gc方式暂未更改，还是PS方式。

预发布压测环境JVM优化前后对比：CMS方式配置：

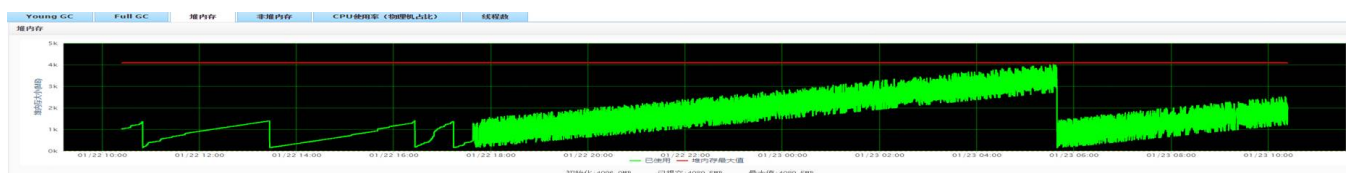
```
-Xms4096m -Xmx4096m -Xmn2048m -XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:+CMSClassUnloadingEnabled
```

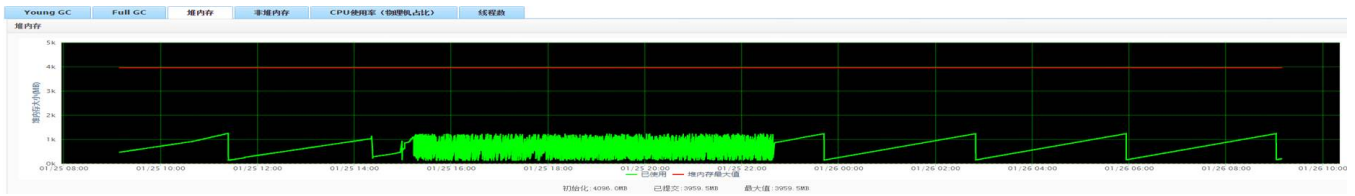
```
-XX:CMSInitiatingOccupancyFraction=80
-XX:CMSInitiatingPermOccupancyFraction=80
```

## （1）、堆内存：

优化前：PS方式

优化后：CMS方式

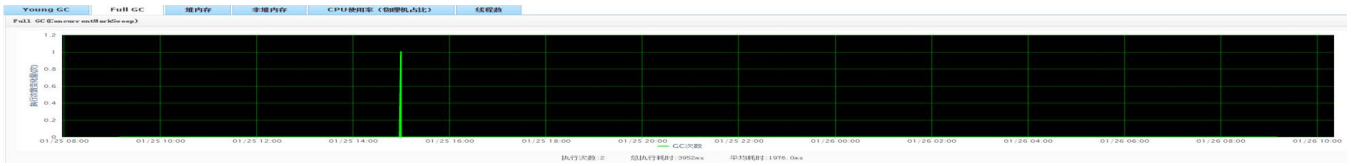
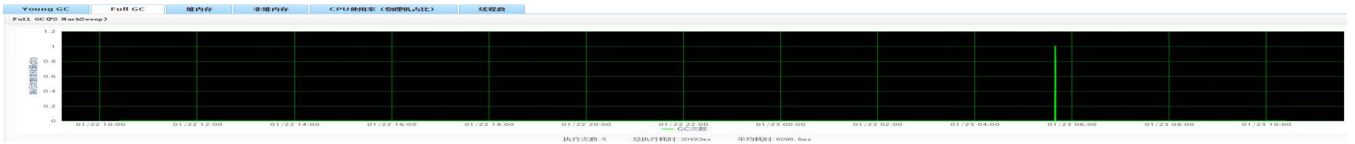




(2)、full gc: 优化前full gc耗时6098.6ms, 优化后full gc耗时1976ms

优化前: PS方式

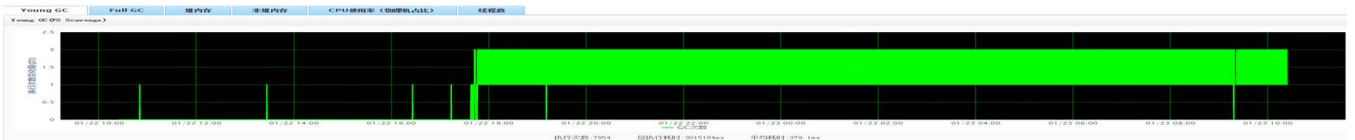
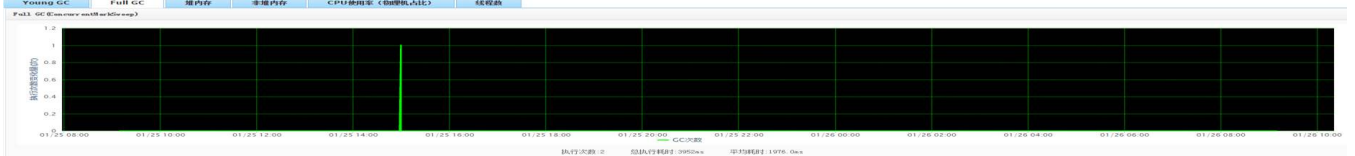
优化后: CMS方式



(3)、young gc: 优化前young gc耗时379.1ms, 优化后耗时163.7ms

优化前: PS方式

优化后: CMS方式

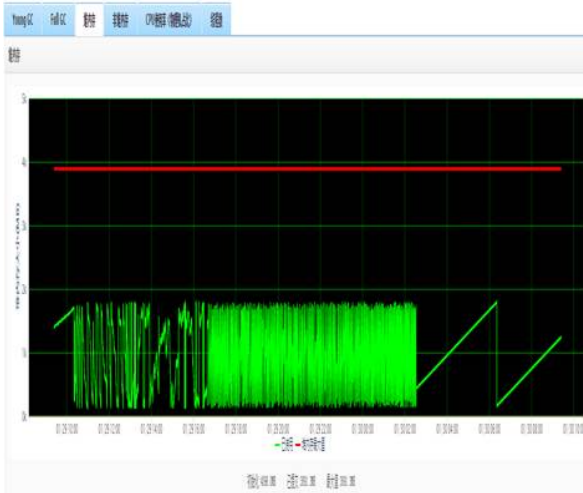
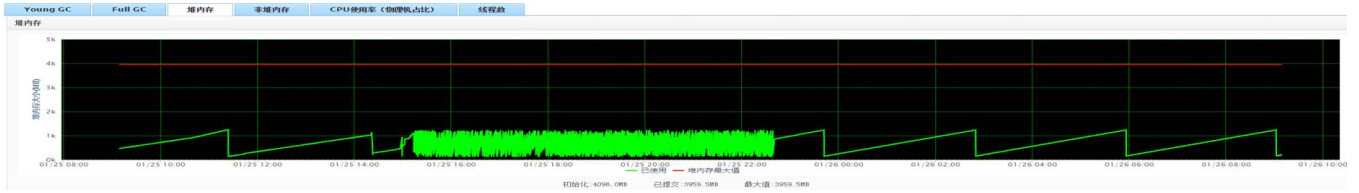


(4)

-XX:CMSInitiatingOccupancyFraction=80 -XX:CMSInitiatingPermOccupancyFraction=80 和 -XX:CMSInitiatingOccupancyFraction=70 -XX:CMSInitiatingPermOccupancyFraction=70无明显差别

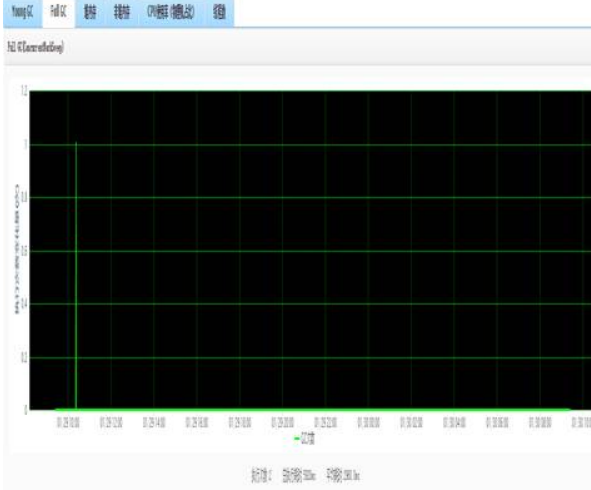
80:

70:



80:

70:



80:

70:



三、测试结果数据：

(1)、批量导出功能对象监控：

名称	9:25:00		10:25:00		11:25:00		12:25:00		13:25:00		14:25:00		15:25:00		16:25:00		17:25:00		18:25:00		19:25:00		20:25:00		21:25:00	
	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小
java.util.zip.Deflater	42666	2858622	42799	2867533	79463	5324021	82478	5526026	92370	6188790	116688	7818096	119309	7993703	151969	10181923	151697	10163699	172970	11588990	173038	11593546	173075	11596025	50174	3361658
java.util.zip.ZStreamRef	64038	1536912	64238	1541712	119497	2867928	125987	3023987	138880	3333120	175227	4205448	179163	4299912	228428	5482272	228020	5472480	259943	6238632	260056	6241344	260115	6242760	59101	1418424
java.util.zip.Inflater	21372	1239576	21439	1243462	39833	2310314	42089	2441162	46241	2581978	58480	3391840	59795	3468110	76088	4413104	75952	4405216	86602	5022916	86647	5025526	86669	5026802	24112	1398496
java.util.zip.ZipCoder	1495	73255	1494	73206	2605	127645	2845	139405	3003	147147	3734	182966	3818	186984	4799	235151	4791	234759	5412	265335	5428	265972	5434	266266	1215	59535
java.util.zip. ZipFile\$ZipFileInflaterInputStream	1297	105057	1291	104571	2429	196749	2614	211734	2818	160626	3524	285444	3614	292734	4603	372843	4595	372195	5231	423711	5244	424764	5258	425898	1307	105867
java.util.zip. ZipFile\$ZipFileInputStream	1297	76929	1291	73587	2429	138453	2601	148257	2818	160626	3524	200868	3614	205998	4603	262371	4595	261915	5231	298167	5244	298908	5258	299706	1307	74499

(2)、打印对象监控：

名称	9:25:00		10:25:00		11:25:00		12:25:00		13:25:00		14:25:00		15:25:00		16:25:00		17:25:00		18:25:00		19:25:00		20:25:00		21:25:00	
	大小		大小		大小		实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	实例数	大小	大小	

	实例数		实例数		实例数																				实例数	
com. itextpdf. text. pdf. PdfName	764 731	3058 9240	769 885	3079 5400	912 942	3651 7680	1025 474	4101 8960	1259 412	5077 6480	1267 870	5071 4800	1247 190	4988 7600	1247 767	4991 0680	1247 767	4991 0680	1781 749	7126 9960	1785 843	7143 3720	1783 811	7135 2440	790 701	3162 8040
com. itextpdf. text. pdf. PdfNumber	482 955	2125 0020	486 380	2140 0720	577 745	2542 0780	6812 65	2997 5660	8026 93	3531 8492	8077 10	3553 9240	7932 48	3490 2912	7934 02	3490 9688	7934 02	3499 9688	1130 373	4973 6412	1133 267	4986 3748	1131 821	4980 0124	420 165	1848 7260
com. itextpdf. text. pdf. PRIndirectReference	339 761	1766 7572	342 205	1779 4660	405 528	2108 7456	4942 15	2569 9180	5641 75	2933 7100	5629 30	2927 2360	5539 20	2880 3840	5542 69	2882 1988	5542 69	2882 1988	7918 80	4117 7760	7936 68	4127 0736	7927 79	4122 4508	330 111	1716 5772
com. itextpdf. text. pdf. PdfDictionary	189 430	9850 360	191 131	9938 812	229 730	1194 5960	2852 69	1483 3988	3135 52	1630 4704	3152 38	1639 2376	3061 46	1591 9592	3063 00	1592 7600	3063 00	1592 7600	4385 63	2280 5276	4403 63	2289 8876	4394 79	2285 2908	179 800	9349 600
com. itextpdf. text. pdf. PdfArray	906 90	3990 360	917 98	4039 112	108 629	4779 676	1298 10	5711 640	1507 03	6630 932	1518 14	6679 816	1489 39	6553 316	1489 77	6554 988	1489 77	6554 988	2122 49	9338 956	2128 28	9364 432	2125 41	9351 804	885 90	3897 960
com. itextpdf. text. pdf. PdfWriter\$PdfBody\$PdfCrossReference	453 75	1633 500	459 25	1653 300	574 08	2066 688	67892	2444 112	74807	2693 052	78828	2837 808	73921	2661 156	73921	2661 156	73921	2661 156	1053 37	3792 132	1063 24	3827 664	1058 34	3810 024	461 05	1659 780
com. itextpdf. text. pdf. PdfIndirectReference	449 15	1976 260	533 53	1920 708	536 03	2358 532	65891	2899 204	74704	3286 976	75083	3303 652	73888	3251 072	73700	1842 500	73700	1842 500	1053 22	4634 168	1055 65	4644 860	1054 46	4639 624	420 91	1852 004