



后台性能测试

主讲：沙明

2016.01.09

金阳光测试，个人培训永远免费！

目录

1 什么是性能测试

2 性能测试方案设计

3 性能测试数据分析

4 案例分享



一、什么是性能测试？

压力测试

强调极端暴力

稳定性测试

在一定压力下，长时间运行的情况

基准测试

在特定条件下的性能测试

负载测试

不同负载下的表现

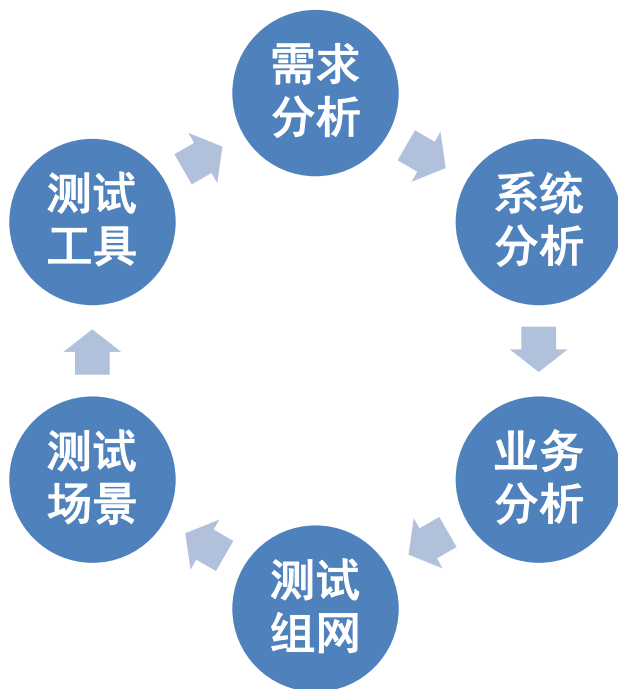
容量测试

最优容量



二、性能测试方案设计

方案设计6部曲



二、性能测试方案设计

需求分析

开发GG或者老板们的要求一般都比较抽象，比如：

- ❑ 帮我测试一下性能
- ❑ 你们测试专业，性能你们看着整
- ❑ 拍一个，目标性能3万

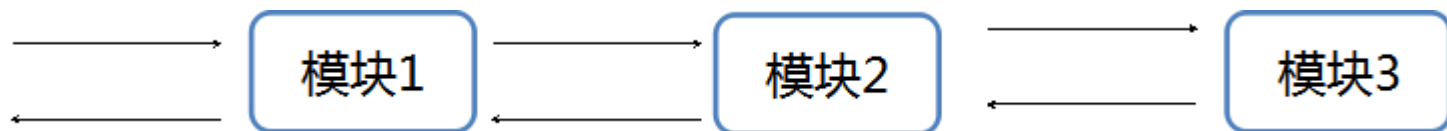
我们需要挖掘，把量化，达成共识：

- ❑ 测试目的
- ❑ 测试场景
- ❑ 性能目标



二、性能测试方案设计

系统分析



- ❑ 了解系统整体架构，列出可能的瓶颈点
- ❑ 关注是否多进程/多线程架构，网络模块间搭配，锁，缓存队列等
- ❑ 资源消耗类型：CPU型，内存型，IO型
- ❑ 集群规模对性能影响



二、性能测试方案设计

业务分析

选取 9 种典型的请求类型，每种请求所占总请求数的比例如下：

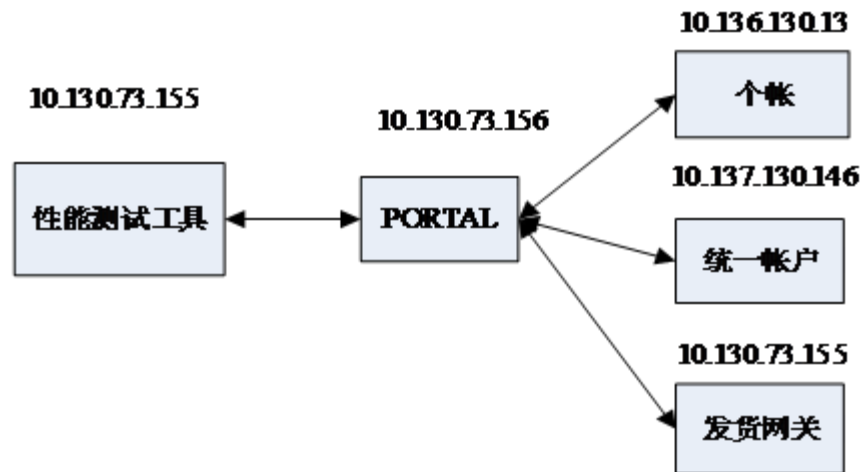
请求类型	所占比例
手机开通检查	9%
预付费开通检查	41%
手机开通	8%
预付费开通	18%
手机关闭	7%
预付费关闭	6%

- 了解生产主要的业务类型，每种业务的占比及重要度等
- 关注生产的业务数据模型，明确基础数据的构成和数据量



二、性能测试方案设计

测试组网



- 了解系统的部署，关注各个模块间的网络耗时
- 了解测试部署和生产实际部署的比例(1:1)
- 了解机器的硬件类型：CPU,内存，磁盘，网卡等



二、性能测试方案设计

测试场景

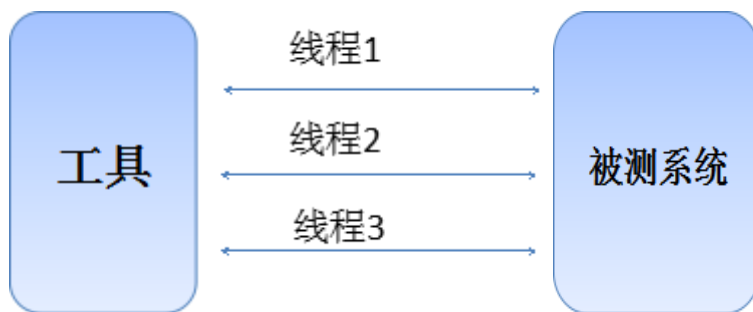
- 明确每个场景的测试目标，比如性能指标等。

场景编号	场景名称	场景说明	压力	关注性能指标
T01	代付深同城邮路	智能路由深同城	应用服务器、数据库	TPS, 时耗, 成功率, CPU, IO, 网络
T02	代付二代大额邮路	智能路由大额	应用服务器、数据库	TPS, 时耗, 成功率, CPU, IO, 网络
T03	代付二代小额邮路	智能路由小额	应用服务器、数据库	TPS, 时耗, 成功率, CPU, IO, 网络
T04	代付超级网银邮路	智能路由超级网银	应用服务器、数据库	TPS, 时耗, 成功率, CPU, IO, 网络
T05	代收走企业代收付	代收	应用服务器、数据库	TPS, 时耗, 成功率, CPU, IO, 网络
T06	查询接口	查询接口	应用服务器、数据库	TPS, 时耗, 成功率, CPU, IO, 网络



二、性能测试方案设计

测试工具

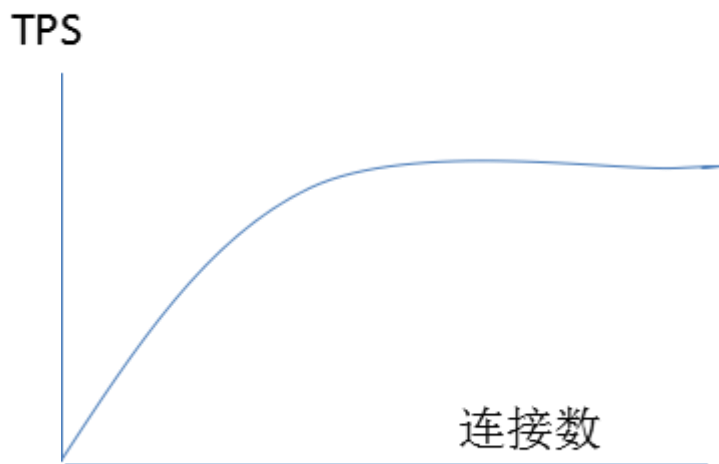


- ❑ 工具开源/自己调用API开发
- ❑ 确定是同步，异步
- ❑ 关注长连接，短连接
- ❑ 关注并发数



二、性能测试方案设计

测试工具

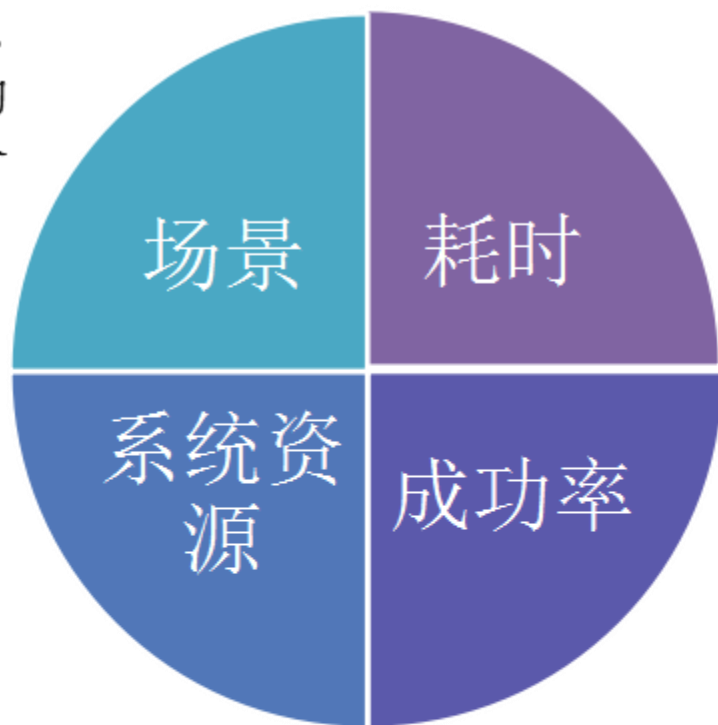


随着连接数增加，性能会逐渐平稳。



三、性能测试数据分析

TPS，每秒处理事务数。
一般用于衡量服务器的
处理能力，或者说服务
器的吞吐率。



场景:	测试场景
系统资源:	主要是CPU、内存、IO
成功率:	超时、业务失败
耗时:	业务处理时间



三、性能测试数据分析

性能指标-vmstat

参数	描述	参考值
vmstat的id	CPU空闲率	小于75%，大于20%
vmstat的us	用户进程使用CPU时间百分比	不低于60%
vmstat的sy	系统进程占用CPU时间百分比	不高于15%
vmstat的wa	IO等待占用CPU时间百分比	不高于30%
top 的Load average	CPU的负载，活跃的进程数	每个CPU不大于3



三、性能测试数据分析

性能指标-vmstat

```
procs -----memory-----swap-- ----io---- --system-- ----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
0  0   25348 4113852 720948 39037568  0    0    28    309    0    0    2    1  97  0  0
1  0   25348 4107196 720948 39037608  0    0     4    172 2573 2420  0    3  96  0  0
0  0   25348 4113596 720948 39037784  0    0    56   2020 2565 3381  1    1  98  0  0
0  0   25348 4113636 720948 39038060  0    0     0   1808 2163 3129  1    1  99  0  0
2  0   25348 4096964 720948 39038096  0    0     0    276 3255 2689  1    4  95  0  0
0  0   25348 4114856 720948 39038104  0    0     0    748 2005 2447  0    0  99  0  0
0  0   25348 4112896 720948 39038116  0    0     0    168 1940 2525  0    1  99  0  0
0  0   25348 4113908 720952 39038404  0    0    68   1936 3885 3923  2    4  94  0  0
0  0   25348 4112640 720952 39038560  0    0     0   1172 1469 2109  0    0  99  0  0
1  0   25348 4109192 720952 39038564  0    0     0   1696 6211 19477  0    1  99  0  0
0  0   25348 4113592 720952 39038604  0    0     0    168 3510 2757  1    4  95  0  0
1  0   25348 4113020 720952 39038632  0    0    68   2088 1754 3158  0    0 100  0  0
1  0   25348 4104880 720952 39039072  0    0     0   1788 2422 3051  1    1  98  0  0
0  0   25348 4112232 720952 39039084  0    0     4    276 3063 2741  1    3  96  0  0
```



三、性能测试数据分析

性能指标-CPU

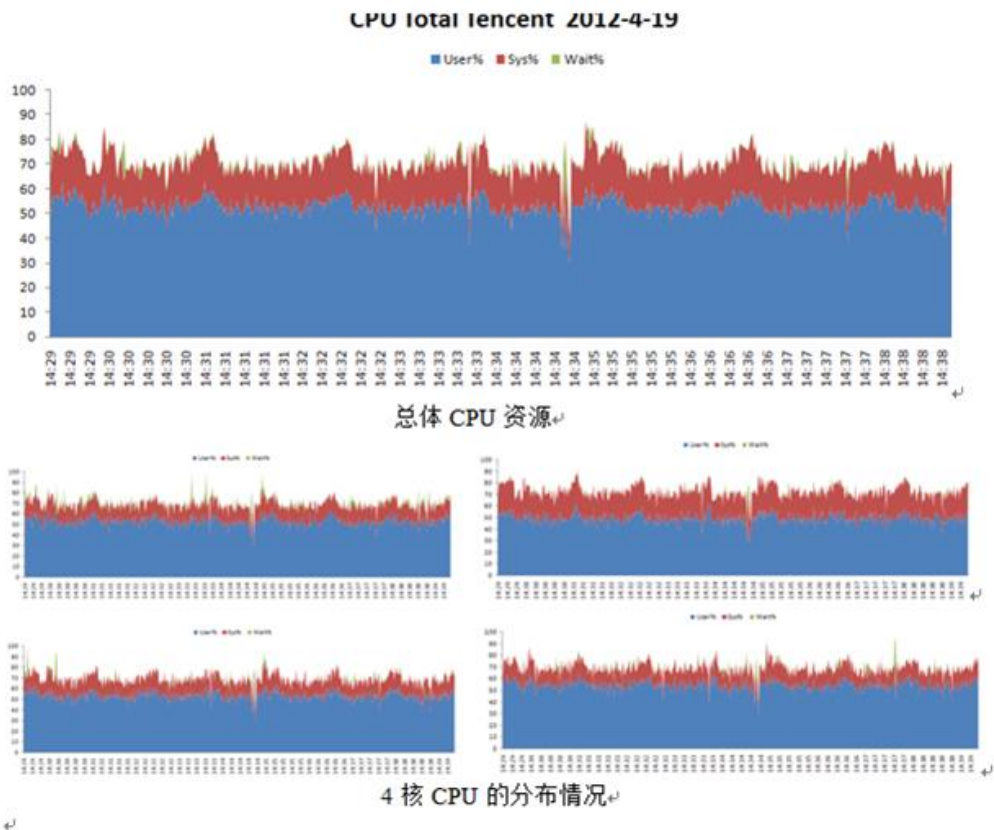
```
top - 16:02:27 up 396 days, 2:38, 2 users, load average: 1.45, 1.79, 1.82
Tasks: 557 total, 1 running, 556 sleeping, 0 stopped, 0 zombie
Cpu0  : 1.0%us, 1.9%sy, 0.0%ni, 97.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1  : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 1.0%us, 21.6%sy, 0.0%ni, 77.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  : 0.0%us, 18.2%sy, 0.0%ni, 81.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu4  : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu5  : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu6  : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu7  : 1.0%us, 0.0%sy, 0.0%ni, 99.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu8  : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu9  : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu10 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu11 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu12 : 1.0%us, 1.0%sy, 0.0%ni, 98.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu13 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu14 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu15 : 0.0%us, 1.0%sy, 0.0%ni, 99.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu16 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu17 : 1.0%us, 0.0%sy, 0.0%ni, 99.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu18 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu19 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu20 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu21 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu22 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu23 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 65795100k total, 65272156k used, 522944k free, 721444k buffers
Swap: 2097144k total, 25348k used, 2071796k free, 39414500k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  997 root        20   0 4037m  2.7g 1696  S  23.6   4.3   1712:42  mysqlreport
 5223 root         0   0  9348 1352 1092  S   6.9   0.0     0:00.07  sh
 5135 root         0   0 13504 1636  956  R   2.0   0.0     0:00.07  top
 2964 root         0   0 5622m  3.4g 1348  S   1.0   5.4  249:42.50  mysqlreport
21658 root         0   0  634m 115m 2348  S   1.0   0.2     4:57.05  binlogtofile
23613 root         0   0  702m 1468  896  S   1.0   0.0    58:01.26  mysqltransfer
27973 root         0   0  601m  14m 1268  S   1.0   0.0   106:11.09  binlogtofile
     1 root         0   0 19204  332  136  S   0.0   0.0     2:31.81  init
```



三、性能测试数据分析

性能指标-CPU



从整体 CPU 消耗来看，整体资源在 70%左右，资源利用率达到了目标。其中红色的颜色代表 sys 系统进程占用快达到 20%，比较高，主要是因为个帐渠道进程间的切换造成的。

从每个 CPU 分布情况来看，每个 CPU 的利用均衡，正常。

Wait 的值很小，证明 IO 耗时不是问题。



三、性能测试数据分析

性能指标-磁盘IO

参数	描述	参考值
%util	一秒中有百分之多少的时间用于 I/O 操作	IO繁忙时,大于 95%
b wa	b 参数(等待资源的进程数)	wa小于 30%
	wa参数(IO等待所占用的CPU时间的百分比)	
await	await: 平均每次设备I/O操作的等待时间(毫秒)。	两者不要差别太大
svctm	svctm: 平均每次设备I/O操作的服务时间(毫秒)。	

```
^C[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ iostat -t 1 -x
Linux 2.6.32.43-tlinux-1.0.8-default (TENCENT64.site) 05/20/15 _x86_64_ (24 CPU)

05/20/15 15:40:59
avg-cpu:  %user   %nice   %system %iowait  %steal   %idle
           1.29    0.00    1.34    0.01    0.00   97.35

Device:            rrqm/s   wrqm/s     r/s     w/s    rsec/s    wsec/s  avgrq-sz  avgqu-sz   await  svctm   %util
sda                 0.20    128.71    0.32    21.42   19.42   1201.34   56.15     0.09     4.01   0.22   0.47
```



三、性能测试数据分析

性能指标-网络IO

- 对比网卡的带宽，可以查看ifconfig，dstat 网卡流量

```
[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ dstat
----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int   csw
  1   1  97   0   0   0|9942B  601k|   0    0 | 418B 1369B|3646   19k
  0   0 100   0   0   0|   0   308k|  85k  88k|   0    0 |9985   19k
  4   2  94   0   0   0|   0   340k| 558k  61k|   0    0 | 13k   22k
  2   0  97   0   0   0|   0  2296k|  87k  88k|   0    0 | 11k   20k
  0   0 100   0   0   0|   0   296k|  36k  31k|   0    0 | 12k   20k
  1   1  98   0   0   0|   0 1392k| 731k 118k|   0    0 | 11k   20k
  0   0 100   0   0   0|   0   320k|  94k  91k|   0    0 | 10k   19k
  3   6  91   0   0   0|   0   428k|  33k  30k|   0    0 | 17k   40k
```



三、性能测试数据分析

性能指标-网络IO

```
[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ ethtool eth1
Settings for eth1:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Advertised pause frame use: Symmetric
    Advertised auto-negotiation: Yes
    Speed: 1000Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 1
    Transceiver: internal
    Auto-negotiation: on
    MDI-X: on
    Supports Wake-on: pumbg
    Wake-on: g
    Current message level: 0x00000007 (7)
    Link detected: yes
[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$
```



三、性能测试数据分析

性能指标-网络IO

```
[root@TENCENT64 ~/data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ ifconfig
eth0      Link encap:Ethernet  HWaddr AC:85:3D:96:F3:1E
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth1      Link encap:Ethernet  HWaddr AC:85:3D:96:F3:1F
          inet addr:10.224.148.45  Bcast:10.224.148.63  Mask:255.255.255.192
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14779397540 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13390584097 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4832499367120 (4.3 TiB)  TX bytes:3378530611506 (3.0 TiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1955259850 errors:108 dropped:108 overruns:0 frame:0
          TX packets:1955259850 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1668980575464 (1.5 TiB)  TX bytes:1668980575464 (1.5 TiB)
```



三、性能测试数据分析

性能指标-内存(free)

```
[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ free -m
[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ free -m
              total        used         free       shared    buffers     cached
Mem:           128905      115209       13696            0         721      89616
-/+ buffers/cache:        24870       104034
Swap:           2047          37         2010
```

- ❑ Memory=free memory+buffers+cached
- ❑ Total 内存总数
- ❑ Used 已经使用的内存数
- ❑ Free 空闲的内存数
- ❑ Shared 多个进程共享的内存总额
- ❑ Buffers buffer Cache和cached page cache 磁盘缓存的大小



三、性能测试数据分析

性能指标-内存(free)

```
[root@TENCENT64 /data/home/tdsql/scheduler2.0/scheduler_ShardMode/bin]$ top -d 1
top - 16:44:03 up 278 days, 6:57, 12 users, load average: 1.19, 0.81, 0.51
Tasks: 785 total, 2 running, 766 sleeping, 17 stopped, 0 zombie
Cpu(s): 4.6%us, 0.4%sy, 0.0%ni, 95.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 131999268k total, 119177144k used, 12822124k free, 739580k buffers
Swap: 2097144k total, 38368k used, 2058776k free, 91719820k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
18050	root	20	0	5828m	3.1g	9912	S	0.0	2.4	22:33.34	java
6971	root	20	0	9664m	2.7g	9476	S	0.0	2.2	21:00.96	java
10560	mmp	20	0	2823m	1.8g	59m	S	0.0	1.4	64:36.21	uacctInterSvr
10602	mmp	20	0	2823m	1.8g	58m	S	0.0	1.4	63:41.68	uacctInterSvr

- ❑ VIRI是虚拟内存
- ❑ RES是物理内存



二、性能测试方案设计

耗时分析

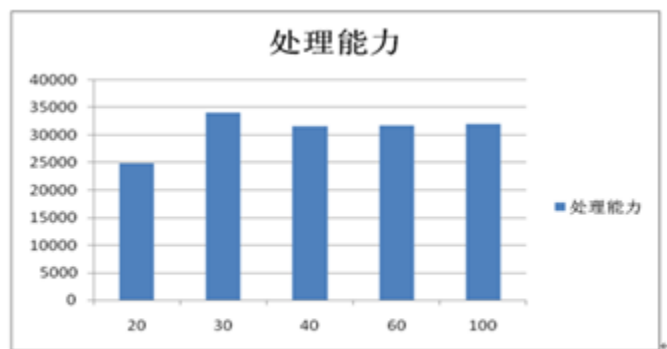
接口耗时分析结果: (ms)													
模块	接口名称	平均耗时	记录总数	成功记录	失败记录	最小耗时	耗时25%	耗时50%	耗时75%	耗时90%	最大耗时	失败原因	异常记录
dc_miacct	UniacctPic	259.808	5143	5143	0	6.796	7.894	9.015	506.543	1003.373	1016.736	查看	查看
pcpdc	Podrequest	265.699	5157	5157	0	9.467	14.093	16.077	513.529	1009.914	1023.769	查看	查看
pcpdc	PodUniResPrice	266.160	5143	5143	0	9.361	13.974	15.955	513.606	1009.812	1023.663	查看	查看
pcpdc	Podrequest	151.761	538	492	46	-1043.615	26.971	43.003	103.159	329.370	4365.748	查看	查看
pcpdc	PodUniResProvide	152.975	527	481	46	-1043.758	25.408	42.853	158.110	329.326	4365.640	查看	查看
qqacct	DoFrising	274.883	5131	5131	0	10.535	15.503	17.647	515.446	1011.360	6649.340	查看	查看
qqacct	AcctCheckPay	3.153	5131	510	4621	1.370	2.955	3.064	3.351	3.930	10.144	查看	查看
qqacct	QQPointPay	16.176	510	510	0	10.060	11.361	12.035	13.707	20.138	192.923	查看	查看
qqacct	DoProvide	101.325	510	477	33	5.285	28.370	44.186	56.563	327.004	341.487	查看	查看
qqacct	QQPointCm	4.070	477	477	0	2.584	2.622	2.971	3.366	5.685	101.629	查看	查看

- 对于单模块的系统，关注业务逻辑处理耗时，DB耗时
- 对于多模块的系统，关注接口耗时



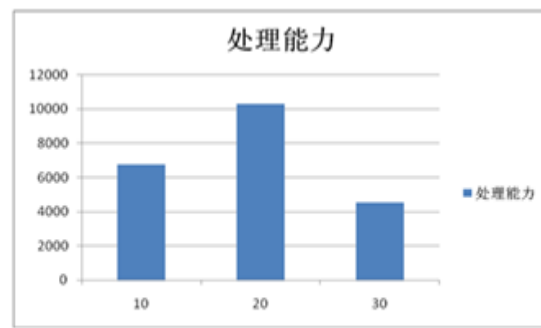
二、性能测试方案设计

配置调优



C4B 线程数调整

从上面的图可见,当 c4b 的线程数是 30 的时候上最优的。



从上面的图可见,当 boss server 的进程数是 20 的时候上最优的,调整到 30 个进程的时候,会出现性能的明显下降。

- 根据配置重要性,列出影响系统性能的列表
- 先测试影响最大的参数,参数之间一般不会相互影响
- 调整参数值,得出处理能力变化曲线



四、案例分享

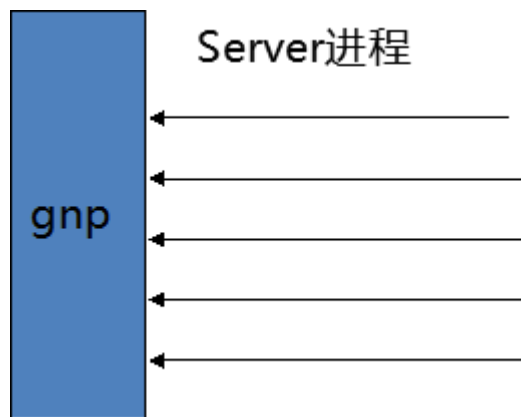
案例1

- 事件:2012.09.02晚上10点, 无线业务反馈喜乐券的请求耗时增加, 接近翻了一倍。我们通过监控发现总请求量并没有增加, 业务那边的变更是增加了并发进程数。
- 方案: 增加服务器的进程个数, 时耗恢复



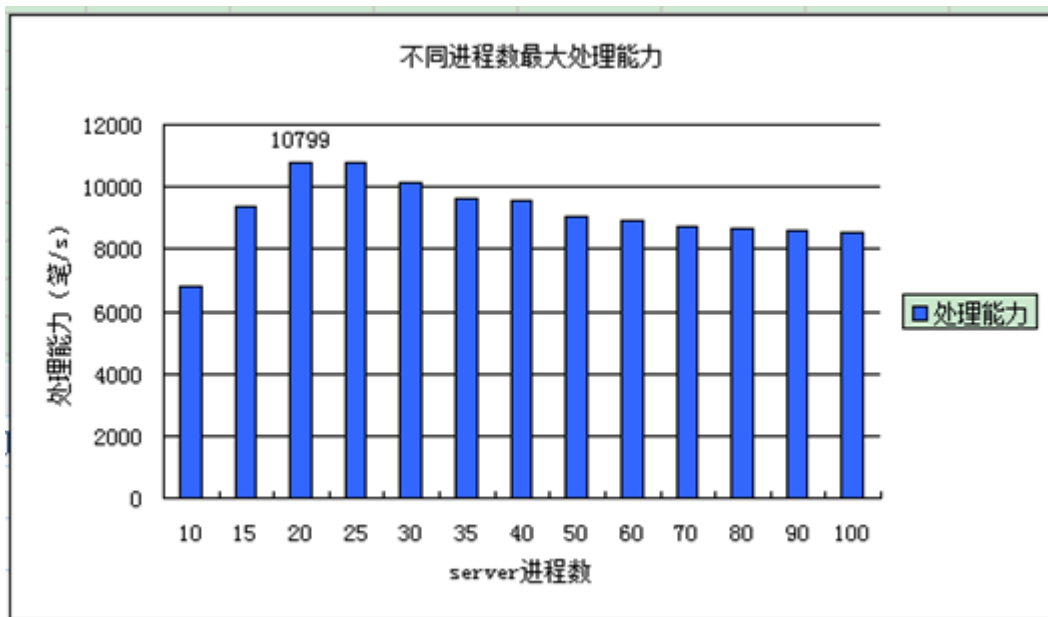
四、案例分享

- 分析原因：Server的并发处理能力太低，是不是进程个数越多越好？



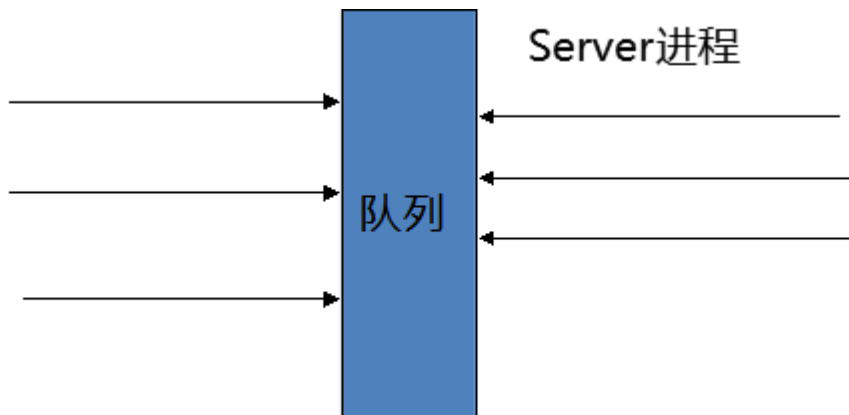
四、案例分析

- 现网可以根据这个数据来进行配置调优

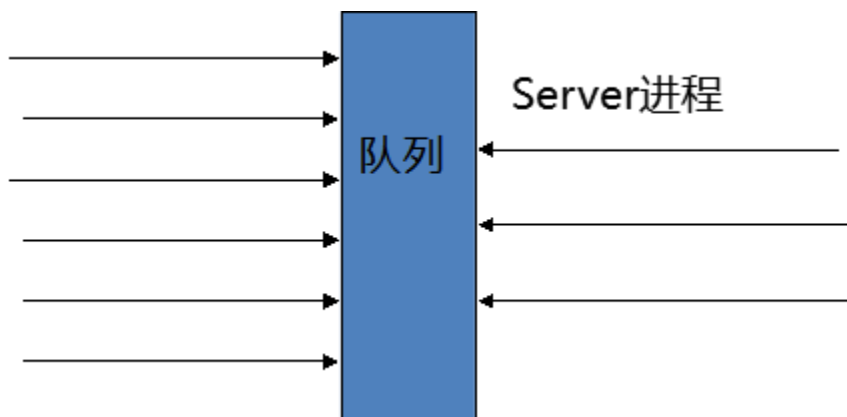


四、案例分析

- 客户端个数和时耗分析，当客户端并发数为30

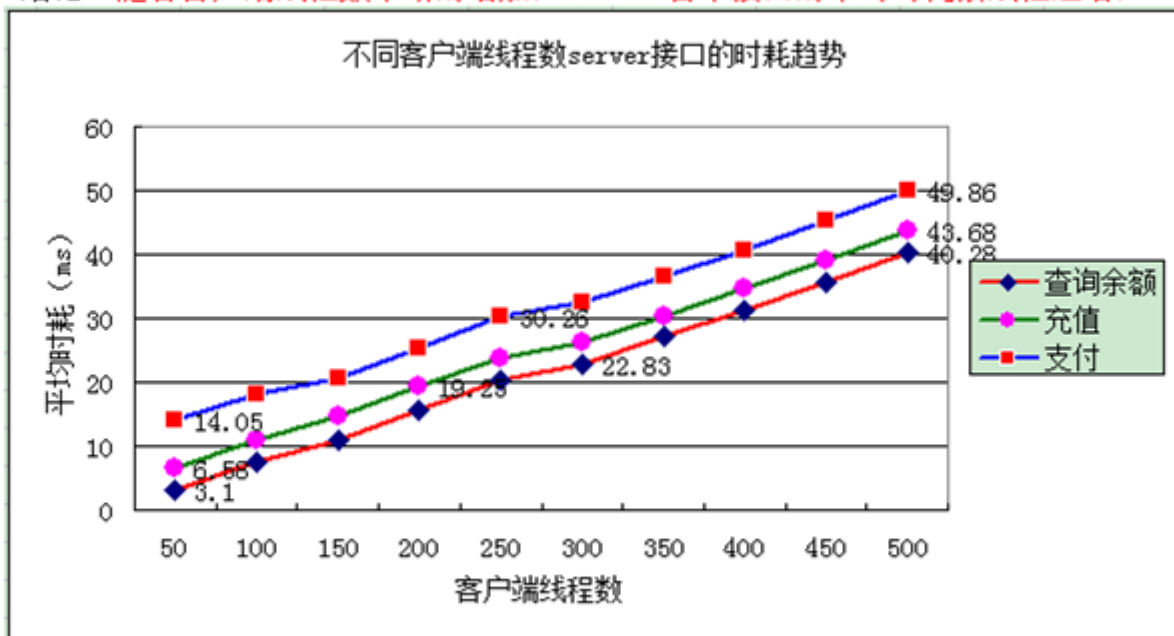


- 客户端个数和时耗分析，当客户端并发数为60



四、数据分析

结论：随着客户端线程数不断的增加，server 各个接口的平均时耗按线性递增，



有了这个结论，后续业务端的客户端如果增加到一定的数目，又对时耗有很强制的要求，我们就可以根据这个数据来进行扩容。



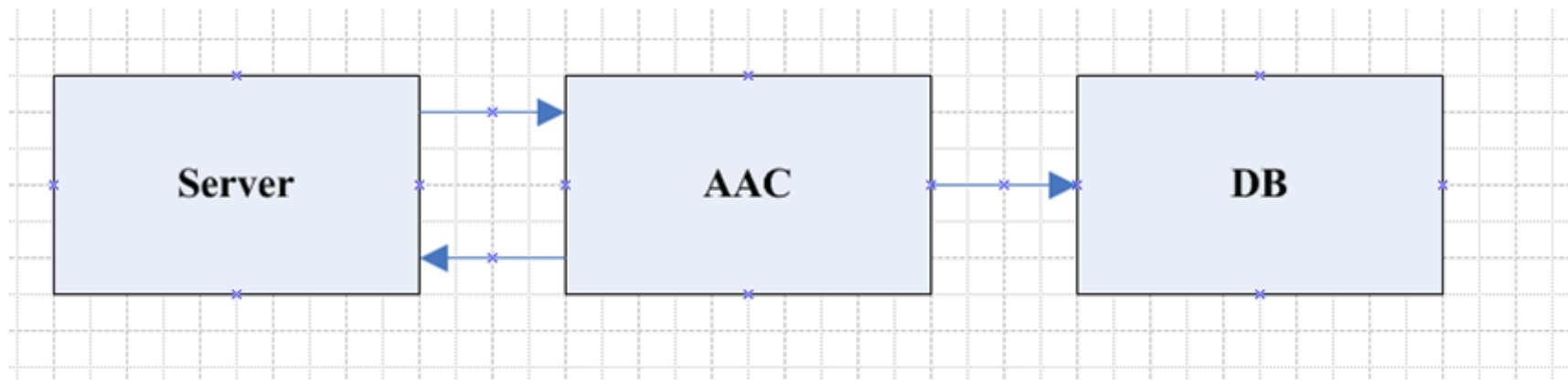
一、性能测试方案设计

案例2

- 事件：国庆期间，云账户系统某个set因为单个业务请求量突增导致整体时耗增加，也影响了其他第三方业务。
- 原因：该类业务大部分是新开户业务，在aac的命中率很低(10%)
- 方案：紧急扩容



四、案例分析

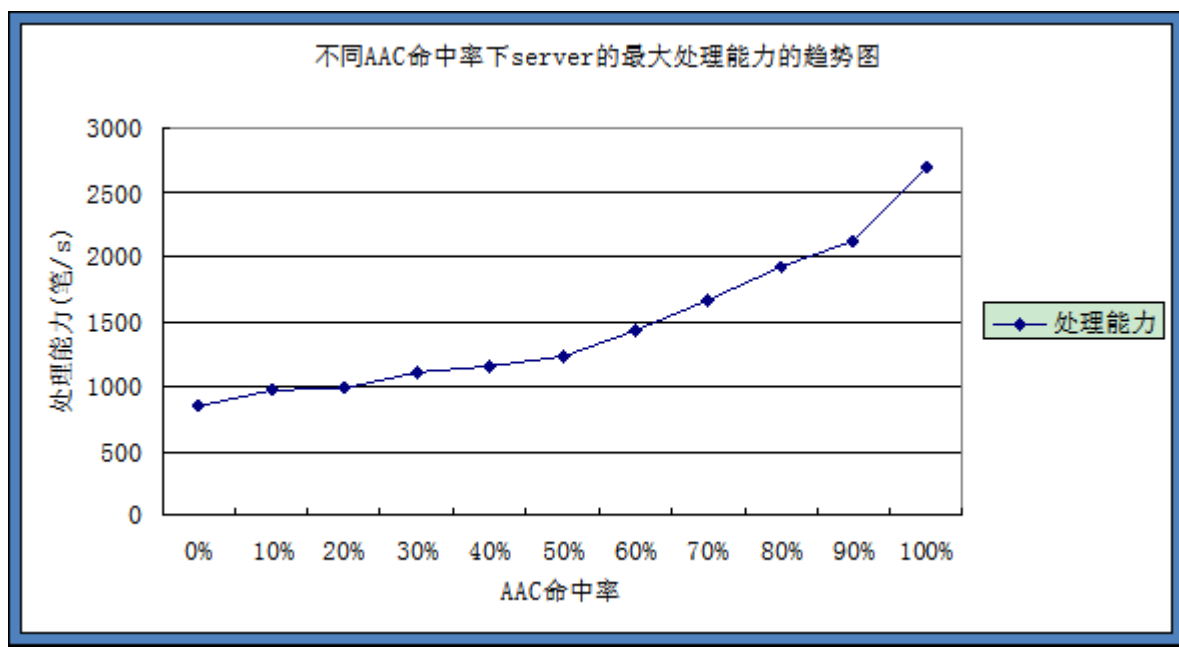


- ❑ AAC是一个高速缓存，如果有数据直接返回结果，没有数据就会去DB拉取数据并更新缓存
- ❑ AAC的命中率公式= $\text{AAC包含的QQ数目} / \text{总请求量的QQ数目}$



四、案例分析

□ 测试了0%到100%的命中率的性能



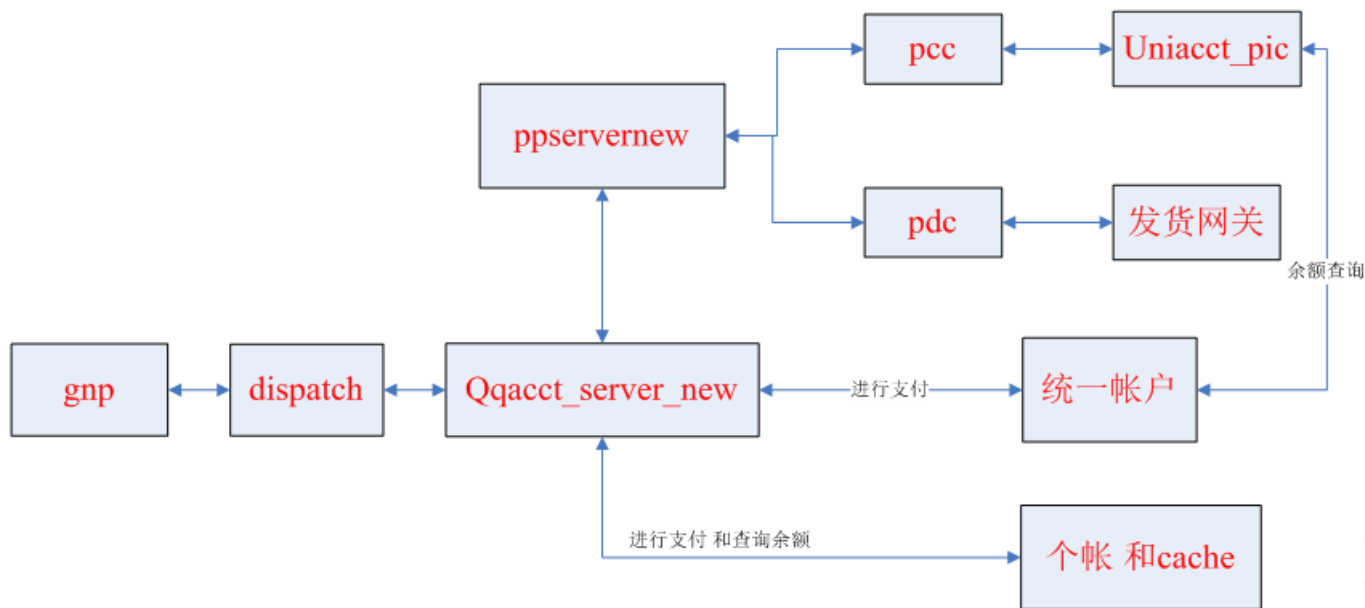
性能测试一定要结合当前系统的业务特性，有针对性的测试。这样测出来的数据才能对现网的扩容有参考意义。



四、案例分析

案例3

- 特点：多模块，模块都是独立的进程，而且外部接口多



四、案例分析

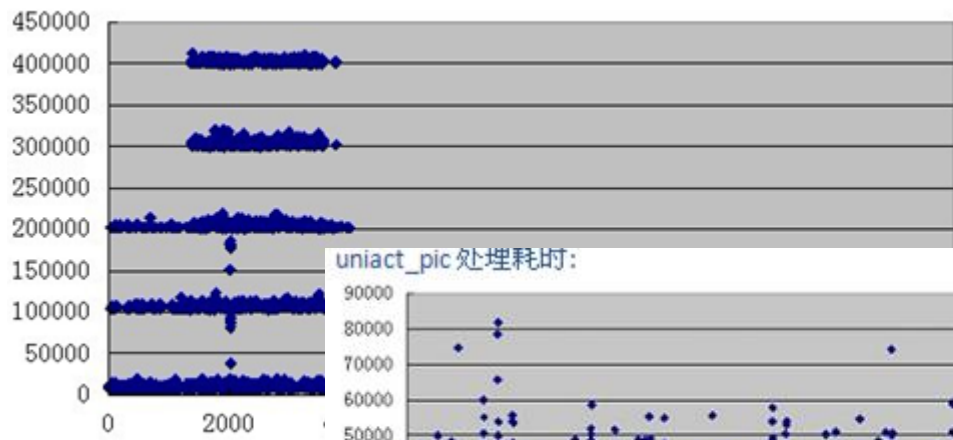
- 问题:接口和模块多,不知道耗时在哪里?找不到优化的点?
- 分析过程:先分析每个接口的耗时,逐步找出问题模块

接口耗时分析结果: (ms)

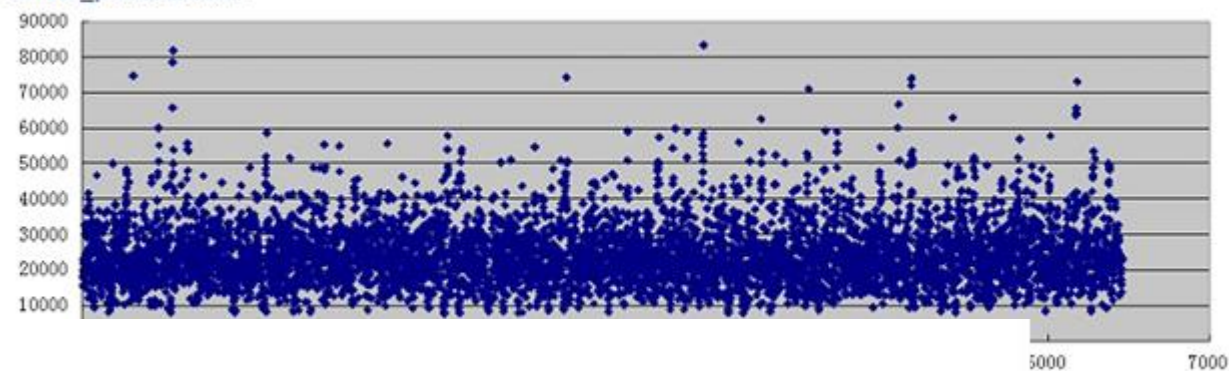
模块	接口名称	平均耗时	记录总数	成功记录	失败记录	最小耗时	耗时25%<	耗时50%<	耗时75%<	耗时90%<	最大耗时	失败明细	异常记录
dc_uniacct	UniacctPic	259.808	5143	5143	0	6.796	7.894	9.015	506.543	1003.373	1016.736	查看	查看
pcpdc	PccRequest	265.699	5157	5157	0	9.467	14.093	16.077	513.529	1009.914	1023.769	查看	查看
pcpdc	PccIhiResFrice	266.160	5143	5143	0	9.361	13.974	15.955	513.606	1009.812	1023.663	查看	查看
pcpdc	PdcRequest	151.761	538	492	46	-1043.615	26.971	43.003	103.159	329.370	4365.748	查看	查看
pcpdc	PdcIhiResProvide	152.975	527	481	46	-1043.758	25.428	42.853	158.110	329.326	4365.640	查看	查看
qqqacct	DoPricing	274.683	5131	5131	0	10.535	15.503	17.647	515.446	1011.360	6649.340	查看	查看
qqqacct	AcctCheckPay	3.153	5131	510	4621	1.370	2.955	3.064	3.351	3.930	10.144	查看	查看
qqqacct	QQPointPay	16.176	510	510	0	10.060	11.361	12.035	13.707	20.138	192.923	查看	查看
qqqacct	DoProvide	101.325	510	477	33	5.285	28.370	44.186	56.583	327.004	341.487	查看	查看
qqqacct	QQPointCfm	4.070	477	477	0	2.584	2.822	2.971	3.366	5.685	101.629	查看	查看



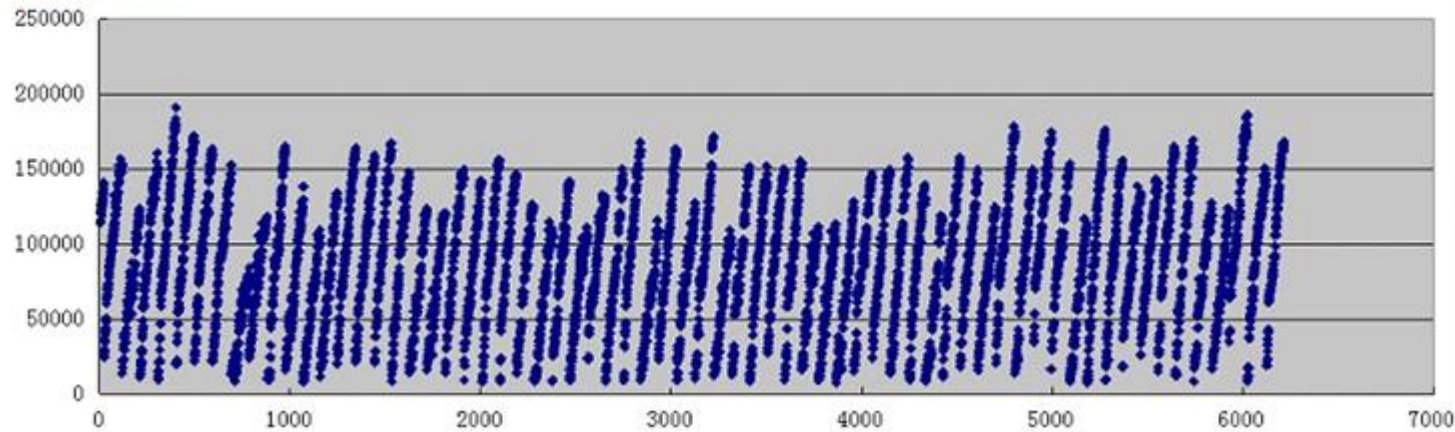
四、案例分析



uniact_pic 处理耗时:



Gnp 处理耗时:



四、案例分析

- ❑ 分析到具体模块，可以在函数级别加上耗时日志打印
- ❑ 把细化的函数时间做统计，最后判断出最耗时的函数

```
gettimeofday(&tv, NULL);
uiBegin = tv.tv_sec*1000000 + tv.tv_usec;

uacct_public_op stUacctOp;
stUacctOp.initServer(n_iAgentKey, n_strUniAcctDnsMasterIP.c_str(), n_iUniAcctDnsMasterPort,
                    n_strUniAcctDnsSlaveIP.c_str(), n_iUniAcctDnsSlavePort,
                    n_iUniAcctTimeOut, n_strUniAcctIP.c_str(), n_iUniAcctPort,
                    n_strUniAcctName.c_str());

gettimeofday(&tv, NULL);
uiEnd = tv.tv_sec*1000000 + tv.tv_usec;
g_stSysLog.error("initServer Use: %u", uiEnd - uiBegin);
```



四、案例分析

```
char szValue[OC_MAX_BUF_LEN];
//memset(szValue, 0x0, sizeof(szValue));
size_t iValueLen = sizeof(szValue);

uint32_t uiUpdateTimeSec = 0;
uint32_t uiUpdateTimeUsec = 0;
int iResultCode = 0;

bool bRet = m_stCloudApi.get(m_iPayDBType
));

if( !bRet )
{
    return iResultCode;
}
szValue[iValueLen] = '\0';
```

解决方法:

#define OC_MAX_BUF_LEN 8192 //这有8k,内存很大

1)把memset(szValue, 0x0, sizeof(szValue));代码删除.

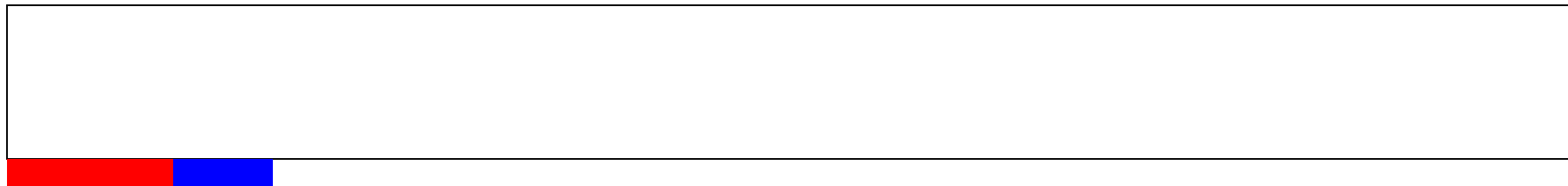
2)加上szValue[iValueLen]='\0';



【金阳光测试】

- 1、官方网站：<http://gstest.ke.qq.com>
- 2、百度搜索：金阳光测试
- 3、新浪微博：金阳光woody
- 4、微信订阅：金阳光自动化测试
- 5、QQ群组：211574787（金阳光测试②）
338924804（金阳光深圳区）





Thanks!

