压测监控及pod调优

一、压测流程

1、确认压测场景

参会的人员：架构师、后端、测试
确认事项：
1、压测场景
2、压测目标
3、中间件配置

2、压测方案评审

参会人员：架构师、后端、测试

3、发邮件通知pm及运维购买相关资源

4、开发人员搭建压测环境及数据库迁移

5、开发人员提供压测数据

6、运维人员给出监控页面

7、开始压测

二、概念介绍

1、压测的目的

压力测试是对系统不断施加压力，来获得系统最大服务能力的测试。一般而言，只有在系统基础功能测试验证完成、系统趋于稳定的情况下，才会进行压力测试。
当负载逐渐增加时，观察系统各项性能指标的变化情况是否有异常。

2、压测指标

压测的指标通常有tps、并发数、响应时间、错误率等

1）tps每秒事务处理量(TransactionPerSecond)

每秒处理的消息数。

2）qps每秒处理事务数

包括了用户请求服务器、服务器自己的内部处理、服务器返回给用户。这三个过程，每秒能够完成N个这三个过程，Tps也就是N。

3）qps与tps关系

Qps基本类似于Tps，但是不同的是，对于一个页面的一次访问，形成一个Tps；但一次页面请求，可能产生多次对服务器的请求，服务器对这些请求，就可计入“Qps”之中。

例如，访问一个 Index 页面会请求服务器 3 次，包括一次 html，一次 css，一次 js，那么访问这一个页面就会产生一个“T”，产生三个“Q”。

3）并发数

系统同时能处理的请求数量

这里有两个概念介绍（用户端并发与服务端并发）

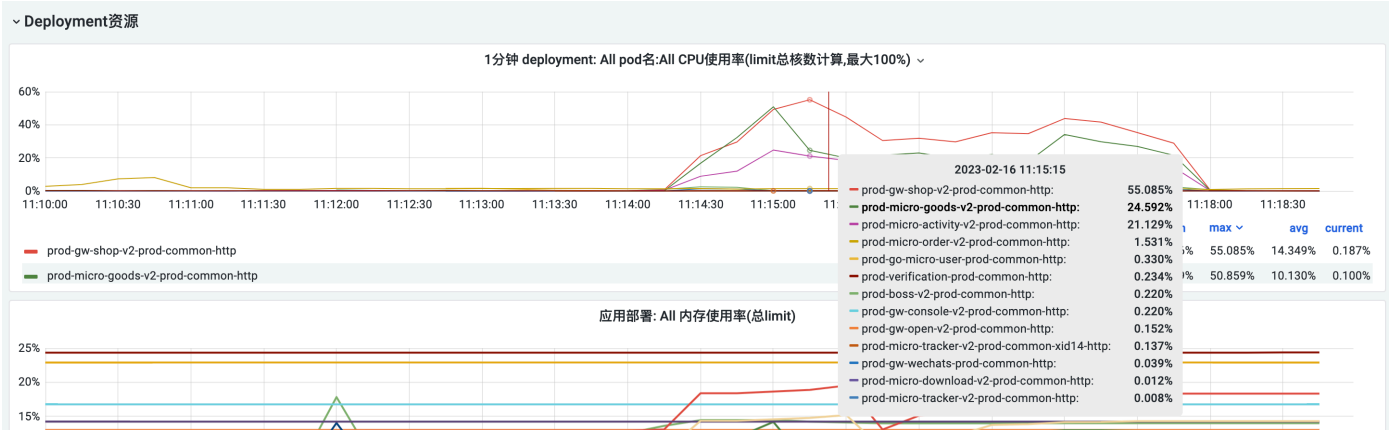
举个例子：

一款商品，晚上八点订阅用户有10W，此时用户端并发就为10W。由于不用的用户使用的硬件不同以及点击下单的时间差，导致并不是同一时间触发请求（假设这里损失1W的用户），接下来会有9W的用户同时请求了服务端，又由于每位用户的地理位置不同，假设服务器位于北京，靠近北京的用户当然比远在海南的用户先到服务端，这里再损失3W的用户。此时服务端并发就为6W了。所以服务端并发<用户端并发

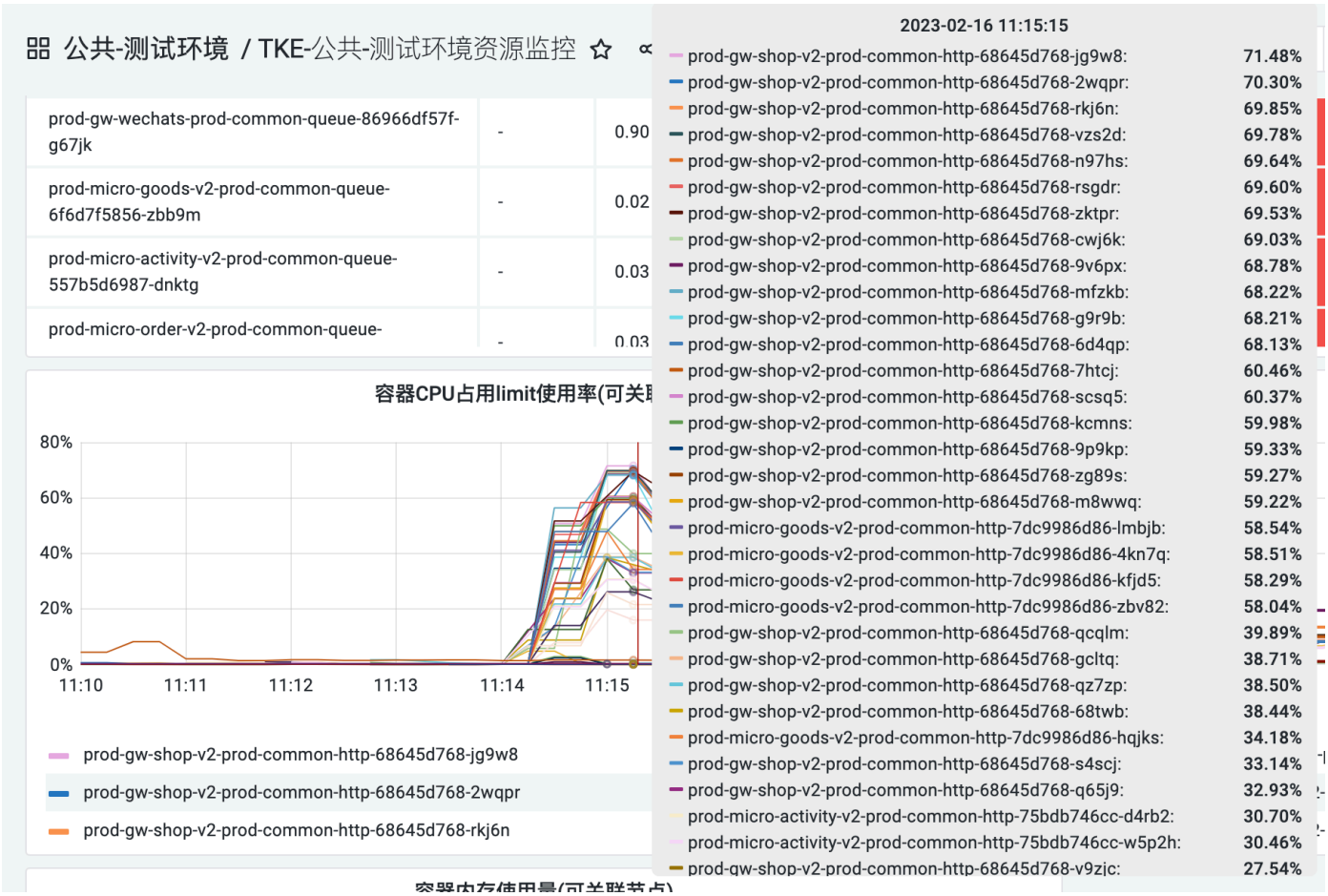
三、压测关注点

1、cpu监控

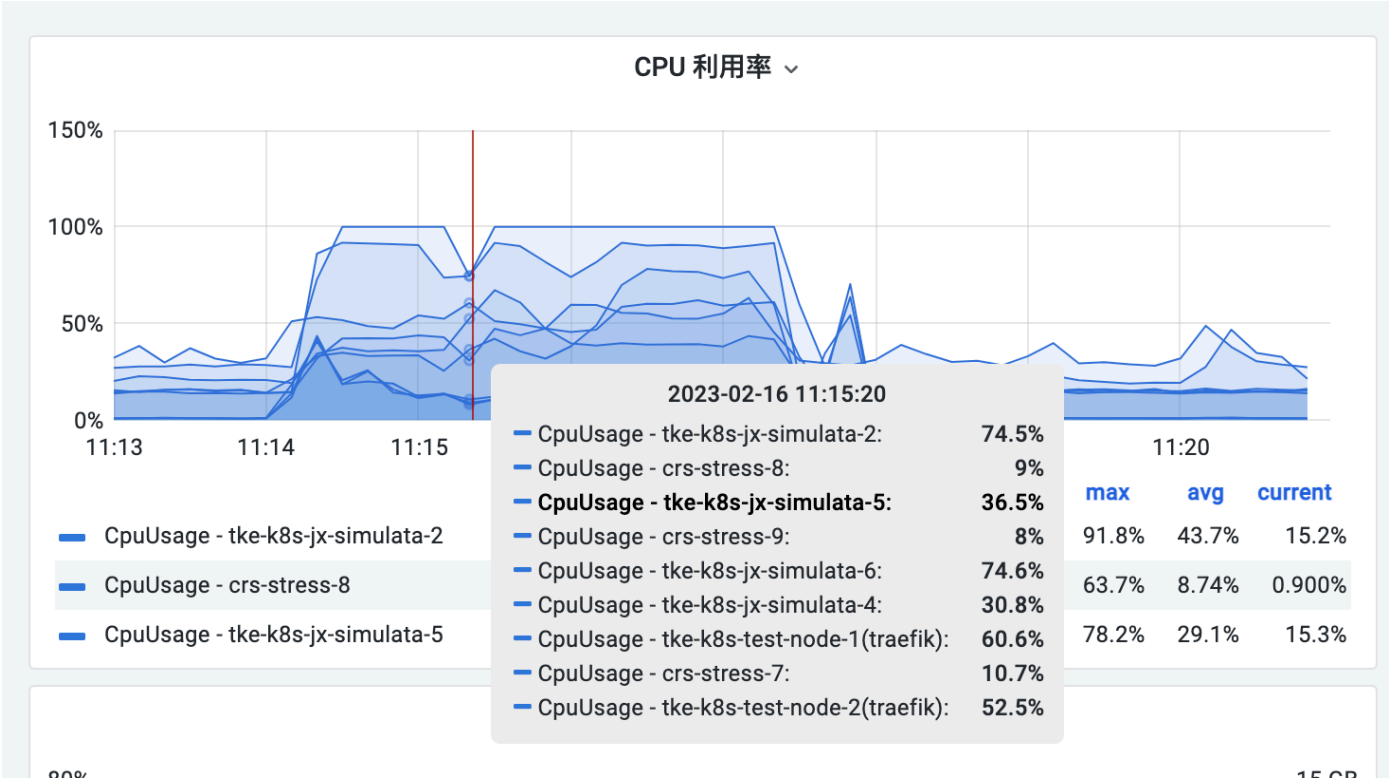
微服务cpu监控



服务pod cpu监控


公共-测试环境 / TKE-公共-测试环境资源监控
prod-gw-wechats-prod-common-queue-86966df57f-g67jk - 0.90
prod-micro-goods-v2-prod-common-queue-6f6d7f5856-zbb9m - 0.02
prod-micro-activity-v2-prod-common-queue-557b5d6987-dnktg - 0.03
prod-micro-order-v2-prod-common-queue- - 0.03
容器CPU占用limit使用率(可关
80%
60%
40%
20%
0%
11:10
11:11
11:12
11:13
11:14
11:15
prod-gw-shop-v2-prod-common-http-68645d768-jg9w8
prod-gw-shop-v2-prod-common-http-68645d768-2wqpr
prod-gw-shop-v2-prod-common-http-68645d768-rkj6n
2023-02-16 11:15:15
prod-gw-shop-v2-prod-common-http-68645d768-jg9w8: 71.48%
prod-gw-shop-v2-prod-common-http-68645d768-2wqpr: 70.30%
prod-gw-shop-v2-prod-common-http-68645d768-rkj6n: 69.85%
prod-gw-shop-v2-prod-common-http-68645d768-vzs2d: 69.78%
prod-gw-shop-v2-prod-common-http-68645d768-n97hs: 69.64%
prod-gw-shop-v2-prod-common-http-68645d768-rsgdr: 69.60%
prod-gw-shop-v2-prod-common-http-68645d768-zktpr: 69.53%
prod-gw-shop-v2-prod-common-http-68645d768-cwj6k: 69.03%
prod-gw-shop-v2-prod-common-http-68645d768-9v6px: 68.78%
prod-gw-shop-v2-prod-common-http-68645d768-mfzkb: 68.22%
prod-gw-shop-v2-prod-common-http-68645d768-g9r9b: 68.21%
prod-gw-shop-v2-prod-common-http-68645d768-6d4qp: 68.13%
prod-gw-shop-v2-prod-common-http-68645d768-7htcj: 60.46%
prod-gw-shop-v2-prod-common-http-68645d768-scsq5: 60.37%
prod-gw-shop-v2-prod-common-http-68645d768-kcmns: 59.98%
prod-gw-shop-v2-prod-common-http-68645d768-9p9kp: 59.33%
prod-gw-shop-v2-prod-common-http-68645d768-zg89s: 59.27%
prod-gw-shop-v2-prod-common-http-68645d768-m8wwq: 59.22%
prod-micro-goods-v2-prod-common-http-7dc9986d86-lmbjb: 58.54%
prod-micro-goods-v2-prod-common-http-7dc9986d86-4kn7q: 58.51%
prod-micro-goods-v2-prod-common-http-7dc9986d86-kfjd5: 58.29%
prod-micro-goods-v2-prod-common-http-7dc9986d86-zbv82: 58.04%
prod-gw-shop-v2-prod-common-http-68645d768-qcqlm: 39.89%
prod-gw-shop-v2-prod-common-http-68645d768-gcltq: 38.71%
prod-gw-shop-v2-prod-common-http-68645d768-qz7zp: 38.50%
prod-gw-shop-v2-prod-common-http-68645d768-68twb: 38.44%
prod-micro-goods-v2-prod-common-http-7dc9986d86-hqjks: 34.18%
prod-gw-shop-v2-prod-common-http-68645d768-s4scj: 33.14%
prod-gw-shop-v2-prod-common-http-68645d768-q65j9: 32.93%
prod-micro-activity-v2-prod-common-http-75bdb746cc-d4rb2: 30.70%
prod-micro-activity-v2-prod-common-http-75bdb746cc-w5p2h: 30.46%
prod-gw-shop-v2-prod-common-http-68645d768-v9zjc: 27.54%

cvm单机 cpu监控

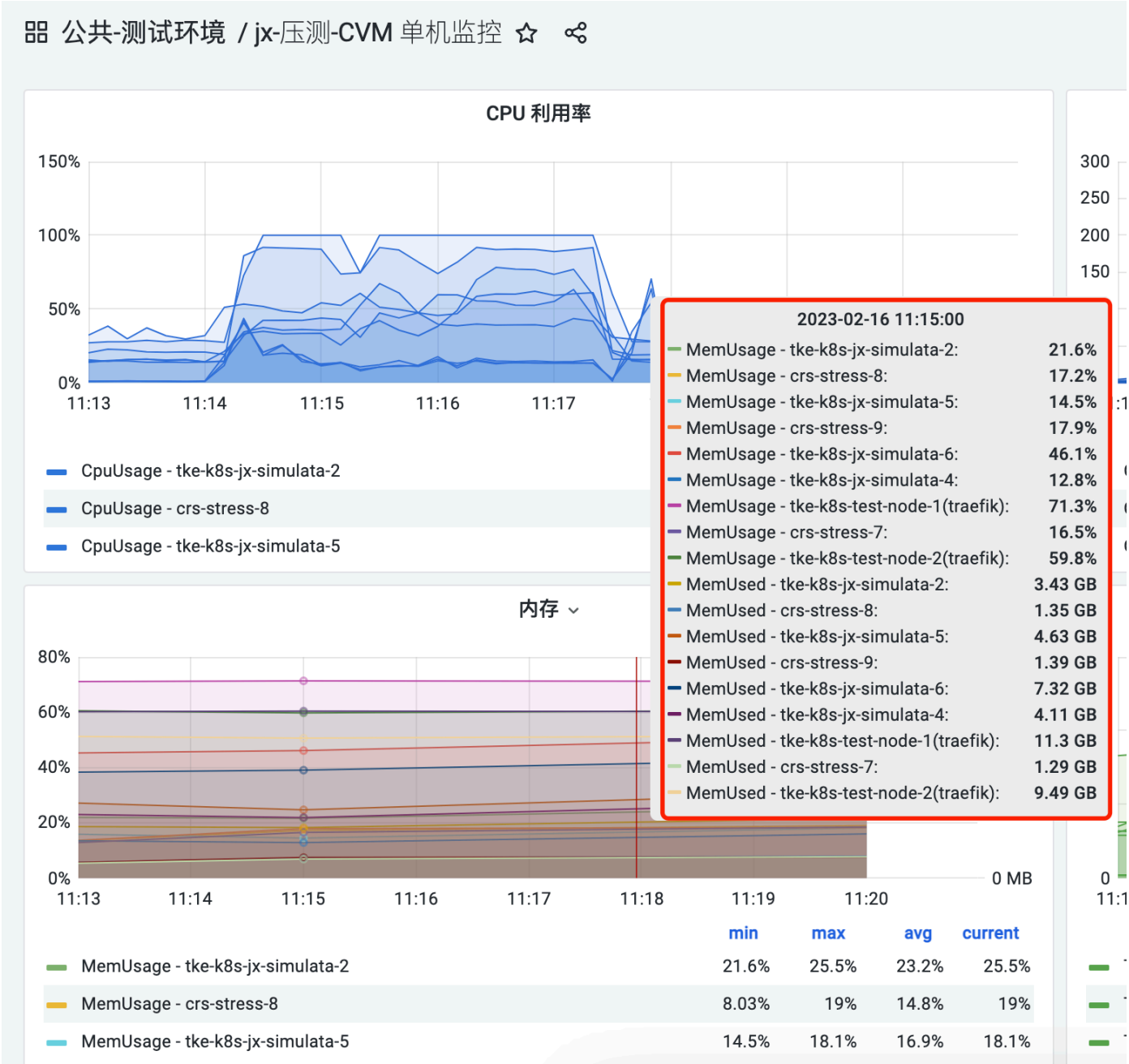

CPU 利用率
150%
100%
50%
0%
11:13
11:14
11:15
11:20
max
avg
current
CpuUsage - tke-k8s-jx-simulata-2 91.8% 43.7% 15.2%
CpuUsage - crs-stress-8 63.7% 8.74% 0.900%
CpuUsage - tke-k8s-jx-simulata-5 78.2% 29.1% 15.3%
2023-02-16 11:15:20
CpuUsage - tke-k8s-jx-simulata-2: 74.5%
CpuUsage - crs-stress-8: 9%
CpuUsage - tke-k8s-jx-simulata-5: 36.5%
CpuUsage - crs-stress-9: 8%
CpuUsage - tke-k8s-jx-simulata-6: 74.6%
CpuUsage - tke-k8s-jx-simulata-4: 30.8%
CpuUsage - tke-k8s-test-node-1(traefik): 60.6%
CpuUsage - crs-stress-7: 10.7%
CpuUsage - tke-k8s-test-node-2(traefik): 52.5%

2、内存监控

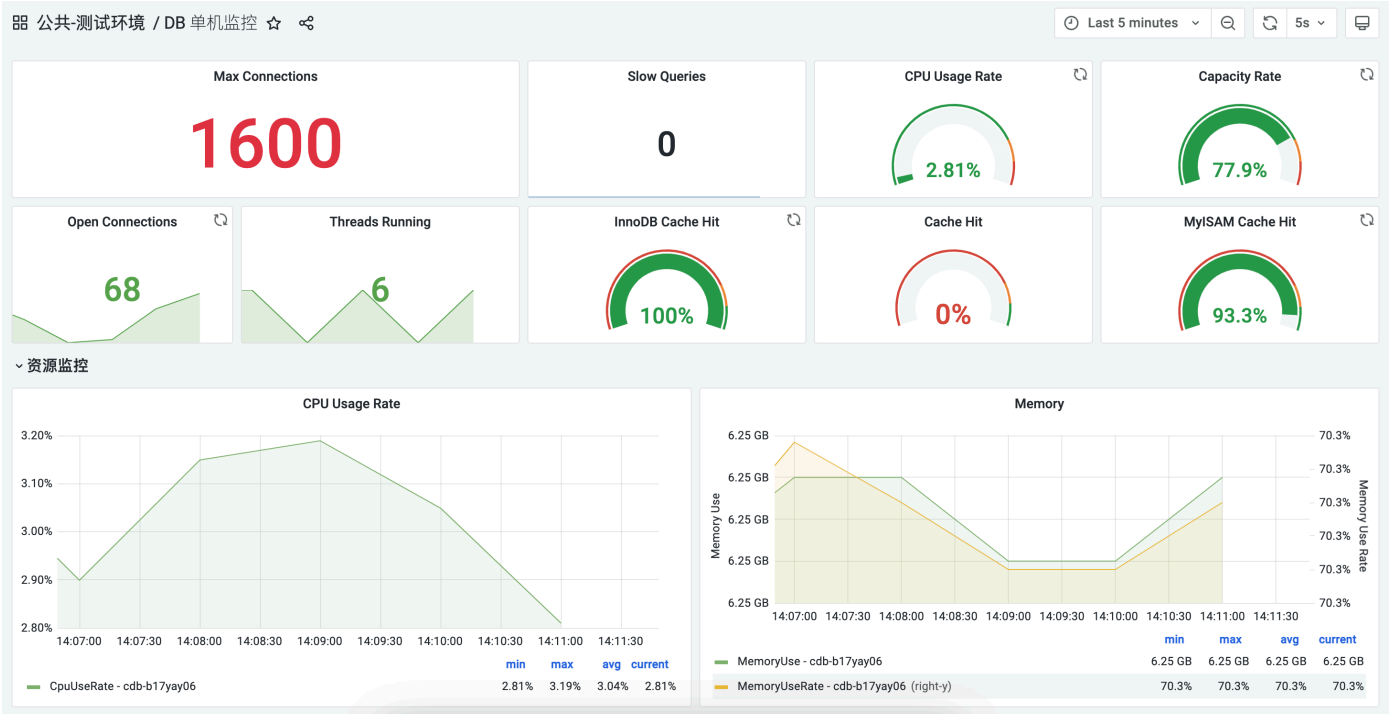
微服务内存监控

服务pod 内存监控

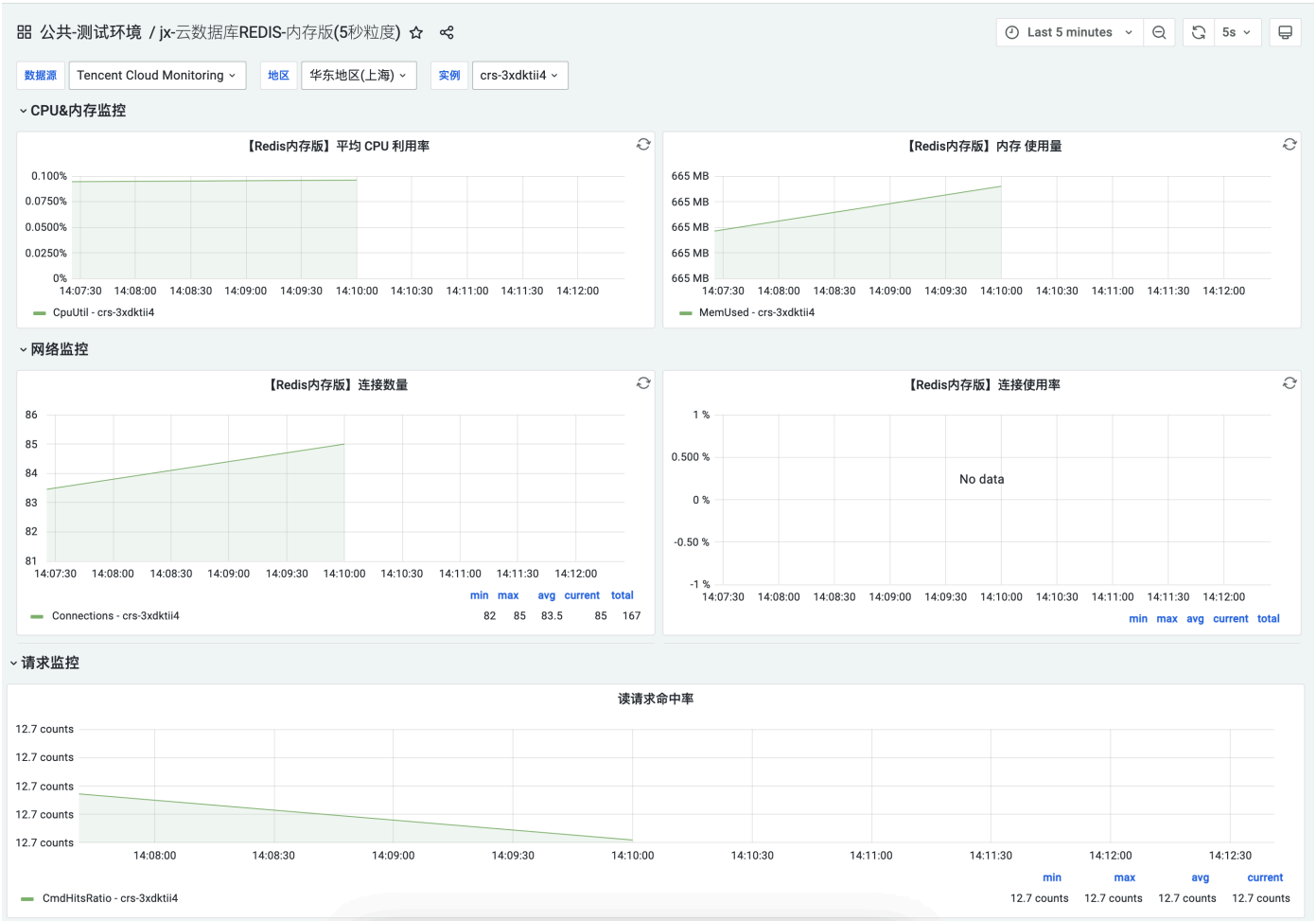
cvm单机内存监控


公共-测试环境 / jx-压测-CVM 单机监控
CPU 利用率
150%
100%
50%
0%
11:13
11:14
11:15
11:16
11:17
CpuUsage - tke-k8s-jx-simulata-2
CpuUsage - crs-stress-8
CpuUsage - tke-k8s-jx-simulata-5
2023-02-16 11:15:00
MemUsage - tke-k8s-jx-simulata-2: 21.6%
MemUsage - crs-stress-8: 17.2%
MemUsage - tke-k8s-jx-simulata-5: 14.5%
MemUsage - crs-stress-9: 17.9%
MemUsage - tke-k8s-jx-simulata-6: 46.1%
MemUsage - tke-k8s-jx-simulata-4: 12.8%
MemUsage - tke-k8s-test-node-1(traefik): 71.3%
MemUsage - crs-stress-7: 16.5%
MemUsage - tke-k8s-test-node-2(traefik): 59.8%
MemUsed - tke-k8s-jx-simulata-2: 3.43 GB
MemUsed - crs-stress-8: 1.35 GB
MemUsed - tke-k8s-jx-simulata-5: 4.63 GB
MemUsed - crs-stress-9: 1.39 GB
MemUsed - tke-k8s-jx-simulata-6: 7.32 GB
MemUsed - tke-k8s-jx-simulata-4: 4.11 GB
MemUsed - tke-k8s-test-node-1(traefik): 11.3 GB
MemUsed - crs-stress-7: 1.29 GB
MemUsed - tke-k8s-test-node-2(traefik): 9.49 GB
300
250
200
150
内存
80%
60%
40%
20%
0%
0 MB
11:13
11:14
11:15
11:16
11:17
11:18
11:19
11:20
min
max
avg
current
MemUsage - tke-k8s-jx-simulata-2 21.6% 25.5% 23.2% 25.5%
MemUsage - crs-stress-8 8.03% 19% 14.8% 19%
MemUsage - tke-k8s-jx-simulata-5 14.5% 18.1% 16.9% 18.1%

3、观察DB使用情况



4、观察Redis使用情况



四、监控流程

1、制作表格

服务	pod		cpu		memory		cpu（总数）			memory（总数）	
	request	limit	request	limit	request	limit	request(pod)*request(cpu)	limit(pod)*request(cpu)	request(pod)*limit(cpu)	request	limit
gw-console-v2	1	1	0.5	0.8	500	800	0.5	0.5	0.8	500	500
gw-shop-v2	15	20	0.5	0.8	500	800	7.5	10	12	7500	10000
boss-v2	2	2	0.5	0.8	500	800	1	1	1.6	1000	1000
gw-open-v2	1	1	0.5	0.8	500	800	0.5	0.5	0.8	500	500
micro-goods-v2	5	10	0.5	0.8	500	800	2.5	5	4	2500	5000
micro-order-v2	1	1	0.5	0.8	500	800	0.5	0.5	0.8	500	500
micro-activity-v2	5	10	0.5	0.8	500	800	2.5	5	4	2500	5000
go-micro-user	1	1	0.3	0.5	500	800	0.3	0.3	0.5	500	500
gw-wechats	2	2	0.3	0.5	500	800	0.6	0.6	1	1000	1000
micro-download-v2	1	1	0.3	0.5	500	800	0.3	0.3	0.5	500	500
micro-tracker	1	1	0.3	0.5	500	800	0.3	0.3	0.5	500	500
总数			32		64000		16.5	24	26.5	17500	25000

表格方便计算每次调整pod数、cpu数、内存数。其中H列与J列需小于总CPU核数（32），相对应的内存计算应小于总内存数（64G）。
如下图，每个cvm主机不光运行了业务pod外，还运行了基础服务，这些也是会占用cpu及内存，所以需要预留cpu与内存给基础服务。

监测pod所属cvm

```
Namespace                  Name                                                    CPU Requests  CPU Limits  Memory Requests  Memory Limits  AGE
---------                  ----                                                    ------------  ----------  ---------------  -------------  ---
argocd                     argocd-redis-ha-haproxy-5bc5dbc689-6jj24                0 (0%)        0 (0%)      0 (0%)           0 (0%)         35d
argocd                     argocd-repo-server-765b6445cd-zrkdk                     0 (0%)        0 (0%)      0 (0%)           0 (0%)         35d
argocd                     argocd-server-6f8d69c7dc-4w6s6                          0 (0%)        0 (0%)      0 (0%)           0 (0%)         35d
hivesec                    hiveagent-qzgxl                                         500m (6%)     1 (12%)     500Mi (1%)       1000Mi (3%)    2d20h
infra                      rocketmq-dashboard-774c79ddb9-d5b7t                     0 (0%)        0 (0%)      0 (0%)           0 (0%)         19d
kube-system                csi-coslauncher-xrsh6                                   250m (3%)     8 (102%)    250Mi (0%)       8Gi (28%)      36d
kube-system                csi-cosplugin-7kb2k                                     0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                csi-nodeplugin-cfsplugin-7hkhp                          0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                ip-masq-agent-nq2l5                                     0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                kube-proxy-nbjkl                                        0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                log-pilot-lwv9r                                         0 (0%)        0 (0%)      0 (0%)           0 (0%)         8d
kube-system                node-problem-detector-fdrfd                             100m (1%)     500m (6%)   200Mi (0%)       200Mi (0%)     36d
kube-system                serf-agent-9f4v9                                        0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                tke-bridge-agent-n9tg4                                  0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                tke-cni-agent-8ttgs                                     0 (0%)        0 (0%)      0 (0%)           0 (0%)         36d
kube-system                tke-eni-agent-mxjsb                                     10m (0%)      0 (0%)      0 (0%)           0 (0%)         36d
kube-system                tke-monitor-agent-fxdph                                 0 (0%)        100m (1%)   0 (0%)           100Mi (0%)     36d
kubesphere-monitoring-system  node-exporter-lf5kp                                  112m (1%)     2 (25%)     200Mi (0%)       600Mi (2%)     36d
shopping-mall-baseline-prod  prod-boss-v2-prod-common-http-849fbd4d6b-gn2bd        300m (3%)     500m (6%)   500Mi (1%)       800Mi (2%)     10m
shopping-mall-baseline-prod  prod-go-micro-user-prod-common-http-5755744b55-s4z8x  100m (1%)     100m (1%)   500Mi (1%)       800Mi (2%)     9m32s
shopping-mall-baseline-prod  prod-gw-open-v2-prod-common-http-648b68b6b7-vccgd     500m (6%)     800m (10%)  500Mi (1%)       800Mi (2%)     20h
shopping-mall-baseline-prod  prod-gw-open-v2-prod-common-queue-5b46c699bf-h6lhm    0 (0%)        0 (0%)      0 (0%)           0 (0%)         20h
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-47svc     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-4b6rc     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-7rcwf     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-fnz92     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-hhw2f     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-q48n2     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-qcqlm     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     15m
shopping-mall-baseline-prod  prod-gw-shop-v2-prod-common-http-68645d768-v9zjc     300m (3%)     800m (10%)  500Mi (1%)       800Mi (2%)     6m15s
shopping-mall-baseline-prod  prod-gw-wechats-prod-common-http-75b7c4bf4f-psjrc    300m (3%)     300m (3%)   500Mi (1%)       800Mi (2%)     9m50s
shopping-mall-baseline-prod  prod-micro-activity-v2-prod-common-http-75bdb746cc-4cpxq  300m (3%)  800m (10%)  500Mi (1%)  800Mi (2%)  10m
shopping-mall-baseline-prod  prod-micro-activity-v2-prod-common-http-75bdb746cc-l62pm  300m (3%)  800m (10%)  500Mi (1%)  800Mi (2%)  4m56s
shopping-mall-baseline-prod  prod-micro-activity-v2-prod-common-http-75bdb746cc-nw4mm  300m (3%)  800m (10%)  500Mi (1%)  800Mi (2%)  11m
shopping-mall-baseline-prod  prod-micro-goods-v2-prod-common-http-7dc9986d86-65kxm  300m (3%)  800m (10%)  500Mi (1%)  800Mi (2%)  6m44s
shopping-mall-baseline-prod  prod-micro-goods-v2-prod-common-http-7dc9986d86-hqjks  300m (3%)  800m (10%)  500Mi (1%)  800Mi (2%)  12m
shopping-mall-baseline-prod  prod-micro-tracker-v2-prod-common-xid14-http-684b546cbc-vnrk2  100m (1%)  100m (1%)  500Mi (1%)  800Mi (2%)  9m7s
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource                    Requests       Limits
  --------                    --------       ------
  cpu                         6172m (79%)    23800m (305%)
  memory                      10150Mi (35%)  24492Mi (86%)
  ephemeral-storage           0 (0%)         0 (0%)
  hugepages-1Gi               0 (0%)         0 (0%)
  hugepages-2Mi               0 (0%)         0 (0%)
  tke.cloud.tencent.com/eip   0              0
  tke.cloud.tencent.com/eni-ip  0            0
Events:                       <none>
```

2、估算所压场景所涉及的服务

例如：
　压测确认订单场景，梳理代码可得出涵盖的服务范围有（shop、activity、goods、order、boss、user）。
　其中shop为网关入口，order主负责逻辑运算、activity、goods主负责查询业务、user与boss则为辅助服务。此时可以得出order的cpu相对于shop、activity、goods高一点，shop的pod数相对于order、activity、goods高点，user与boss因有缓存，给出对应的两个则行。
　最终第一次压测所得出的结论：shop给出8-10个、order与goods与activity给出5个pod、user与boss给出1个。

3、根据第一次压测结果调整修改pod

观察cpu、内存、DB以及Redis的数据（因此次压测中出现的瓶颈问题都为cpu，故在这里只分析根据cpu的调整）

4、常见问题分析

1）并发小、错误率大。首先排查脚本自身原因与业务层面的原因。
2）响应时常长、TPS小。检查压力机资源消耗，首先确保不是压力机的性能原因导致相应时常长。
3）并发增加、cpu压不上去。排查数据库、php-fpm、nginx等原因。
4）TPS低、响应时常长。确认业务流向，例如：jmeter->nginx->php-fpm->php->缓存->mysql->返回。