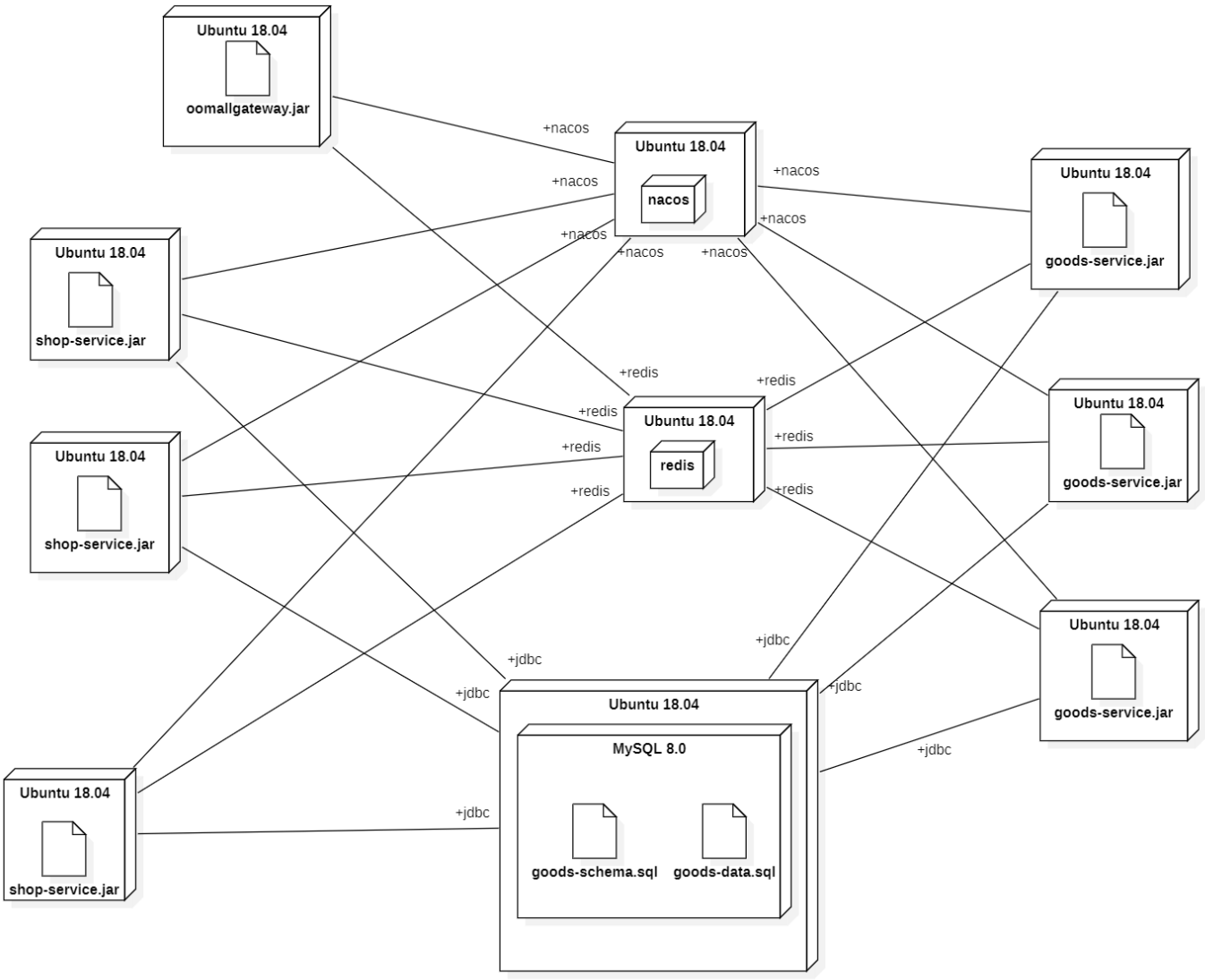
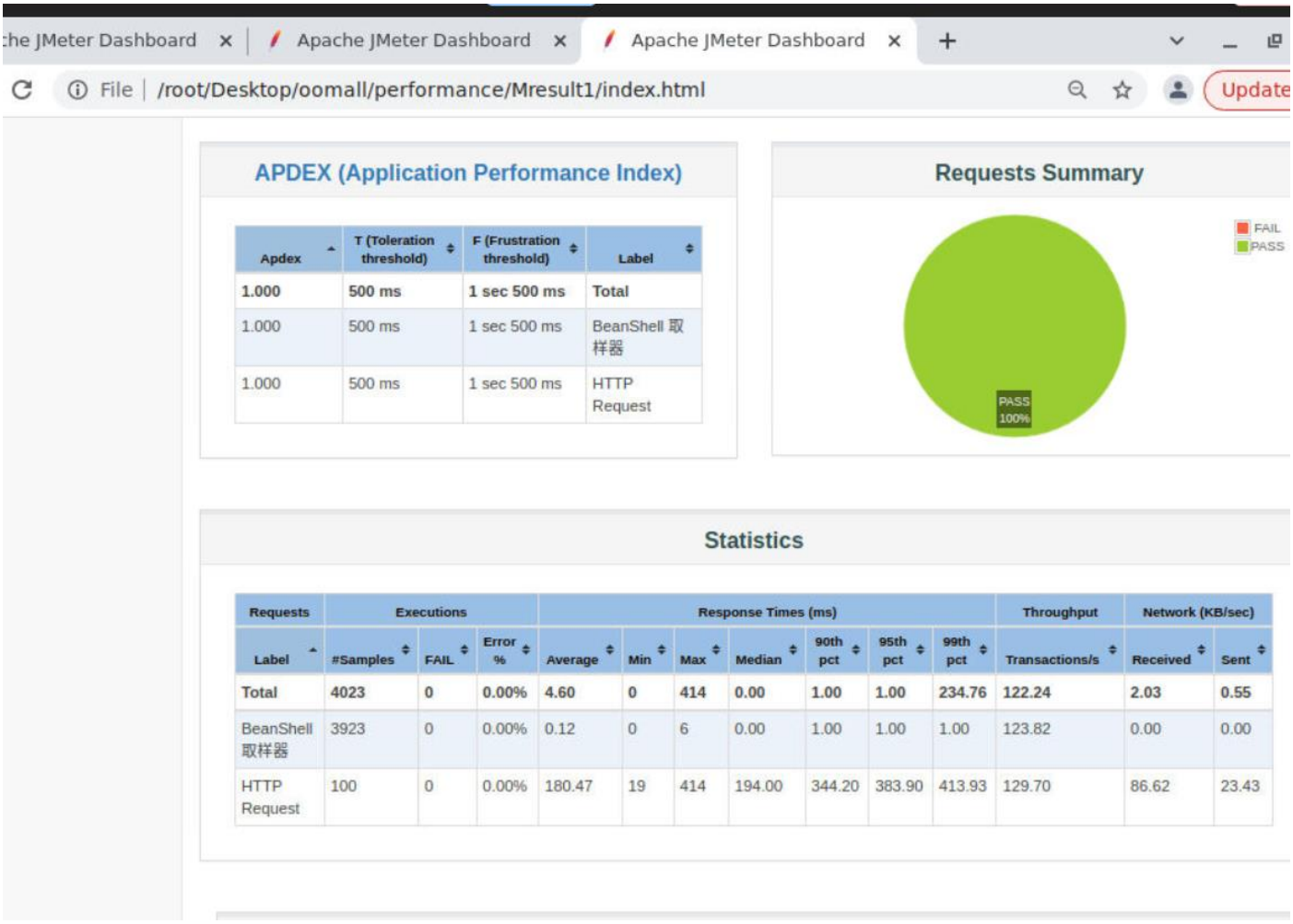
必测部分

服务器部署图如下：

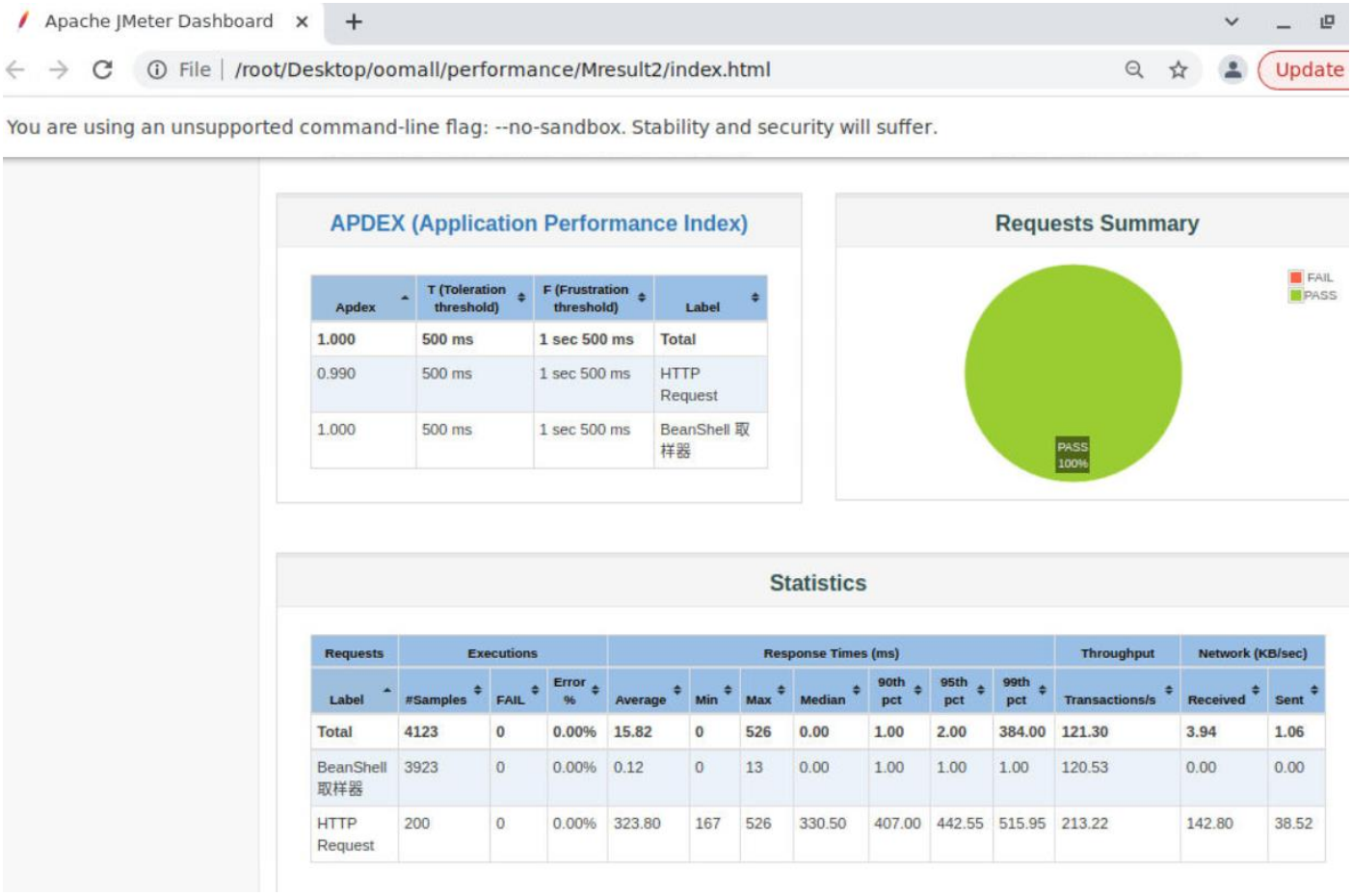


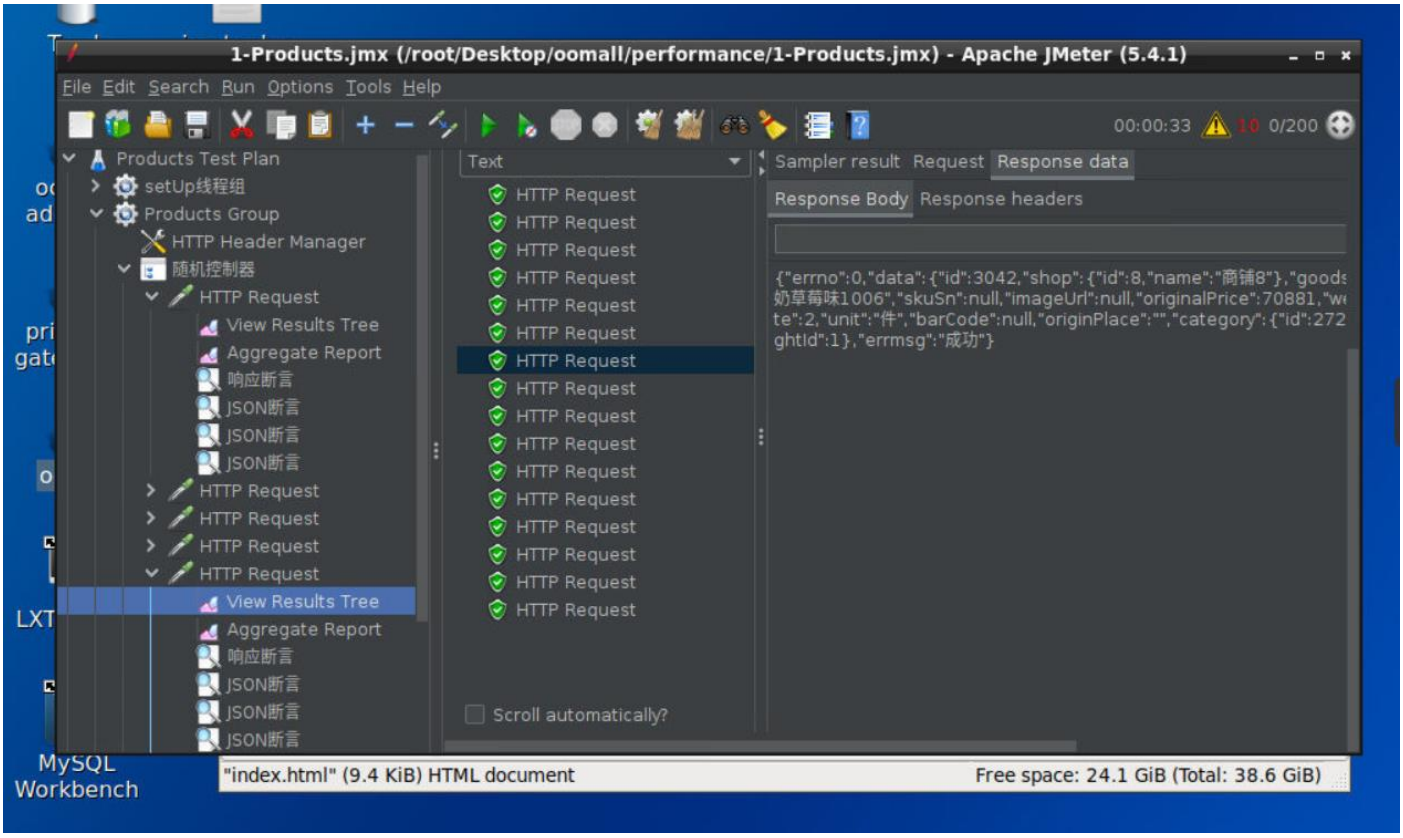
必测部分总共测试两次，第一次测试线程数为 100，第二次测试线程数为 600
第一次测试结果如下：



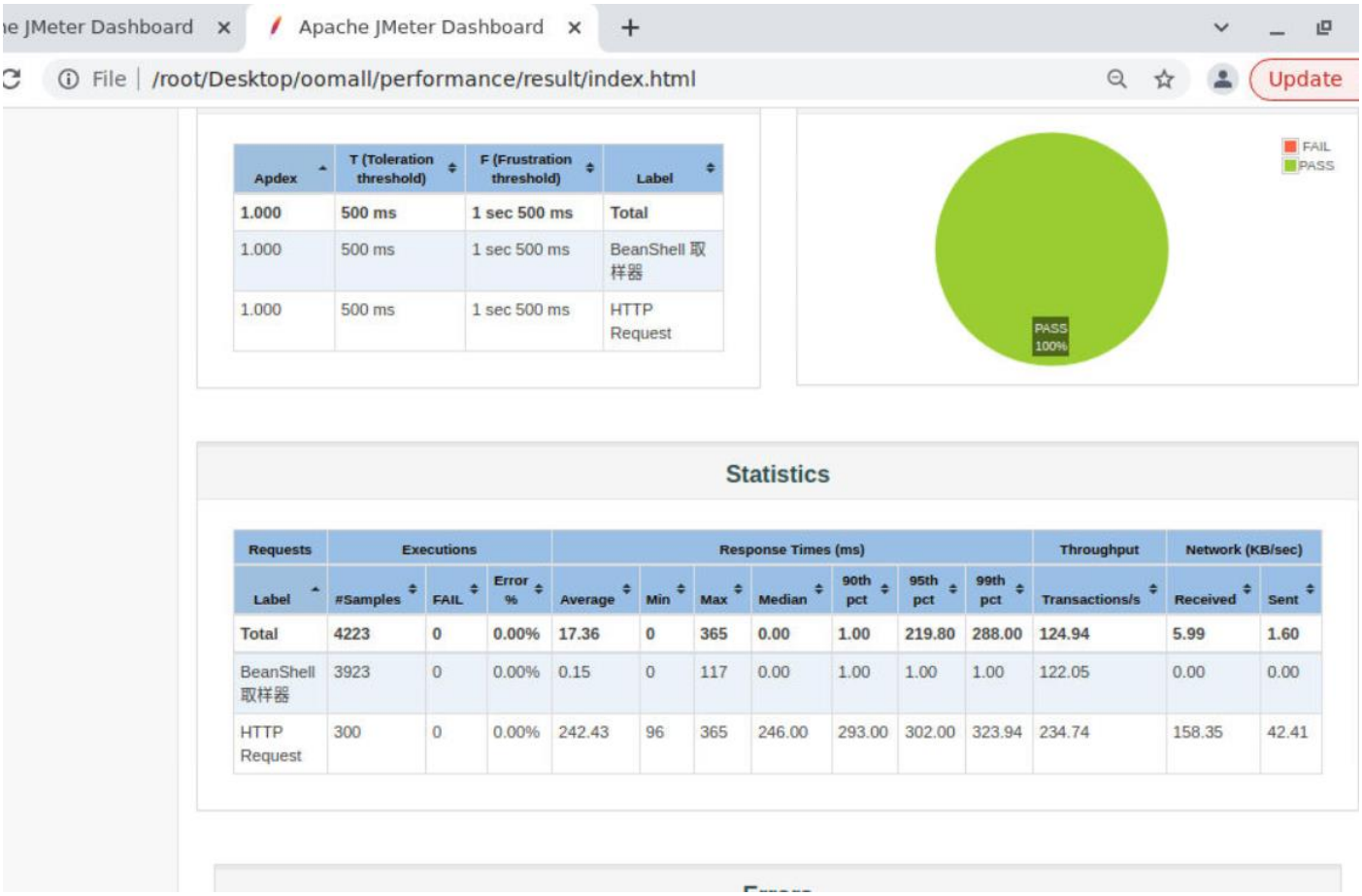
可以看到，1s 内可完成查看 100 次商品

第二次测试结果如下：





之后，继续增加线程至 300，测试结果如下：



Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
1.000	500 ms	1 sec 500 ms	Total
1.000	500 ms	1 sec 500 ms	BeanShell 取样器
1.000	500 ms	1 sec 500 ms	HTTP Request

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	4223	0	0.00%	17.36	0	365	0.00	1.00	219.80	288.00	124.94	5.99	1.60
BeanShell 取样器	3923	0	0.00%	0.15	0	117	0.00	1.00	1.00	1.00	122.05	0.00	0.00
HTTP Request	300	0	0.00%	242.43	96	365	246.00	293.00	302.00	323.94	234.74	158.35	42.41

可以看到此时甚至比之前的 250 线程更快

继续增加线程至 400，测试结果如下：

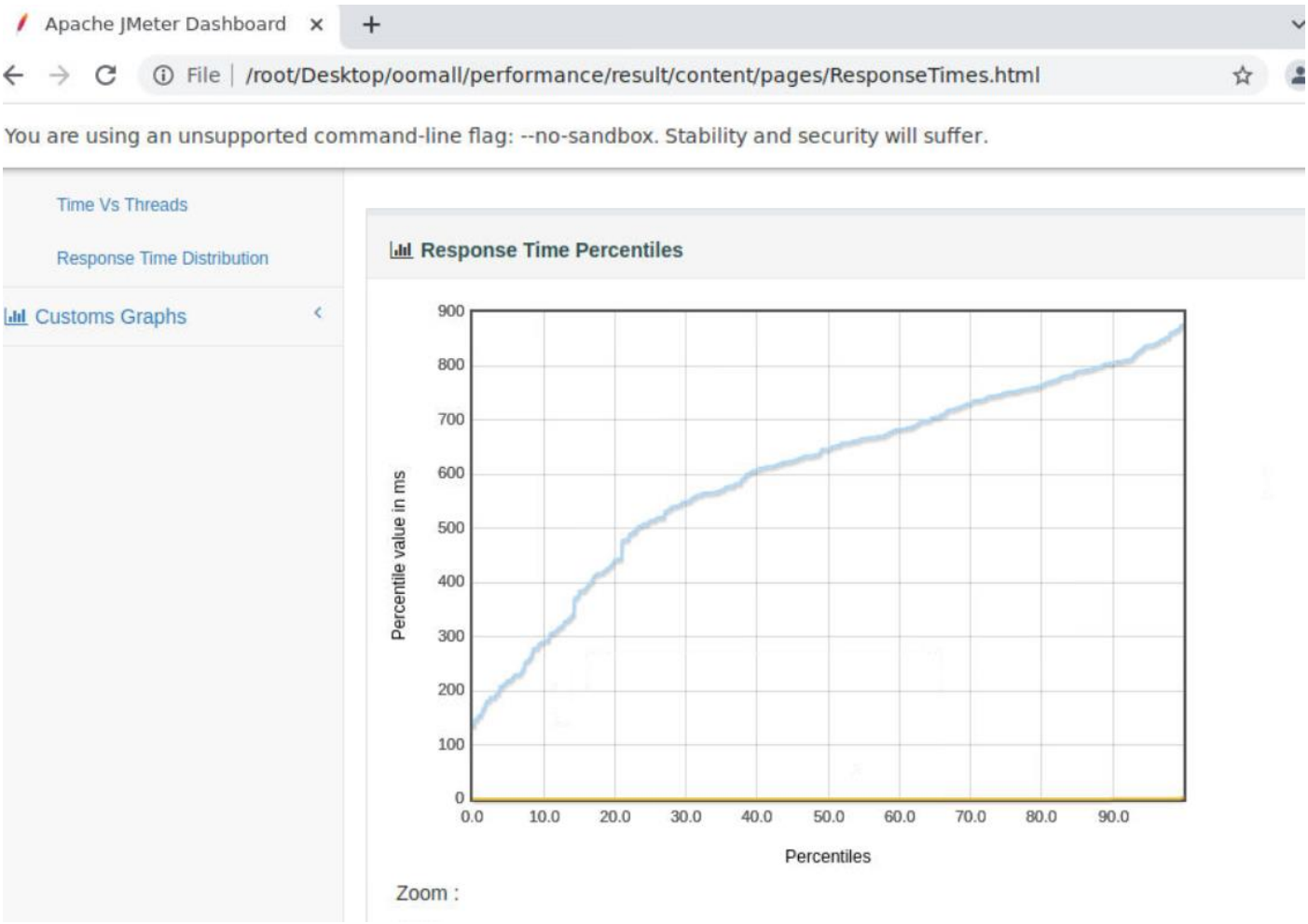
rted command-line flag: --no-sandbox. Stability and security will suffer.

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.964	**500 ms**	**1 sec 500 ms**	**Total**
0.615	500 ms	1 sec 500 ms	HTTP Request
1.000	500 ms	1 sec 500 ms	BeanShell 取样器

Statistics

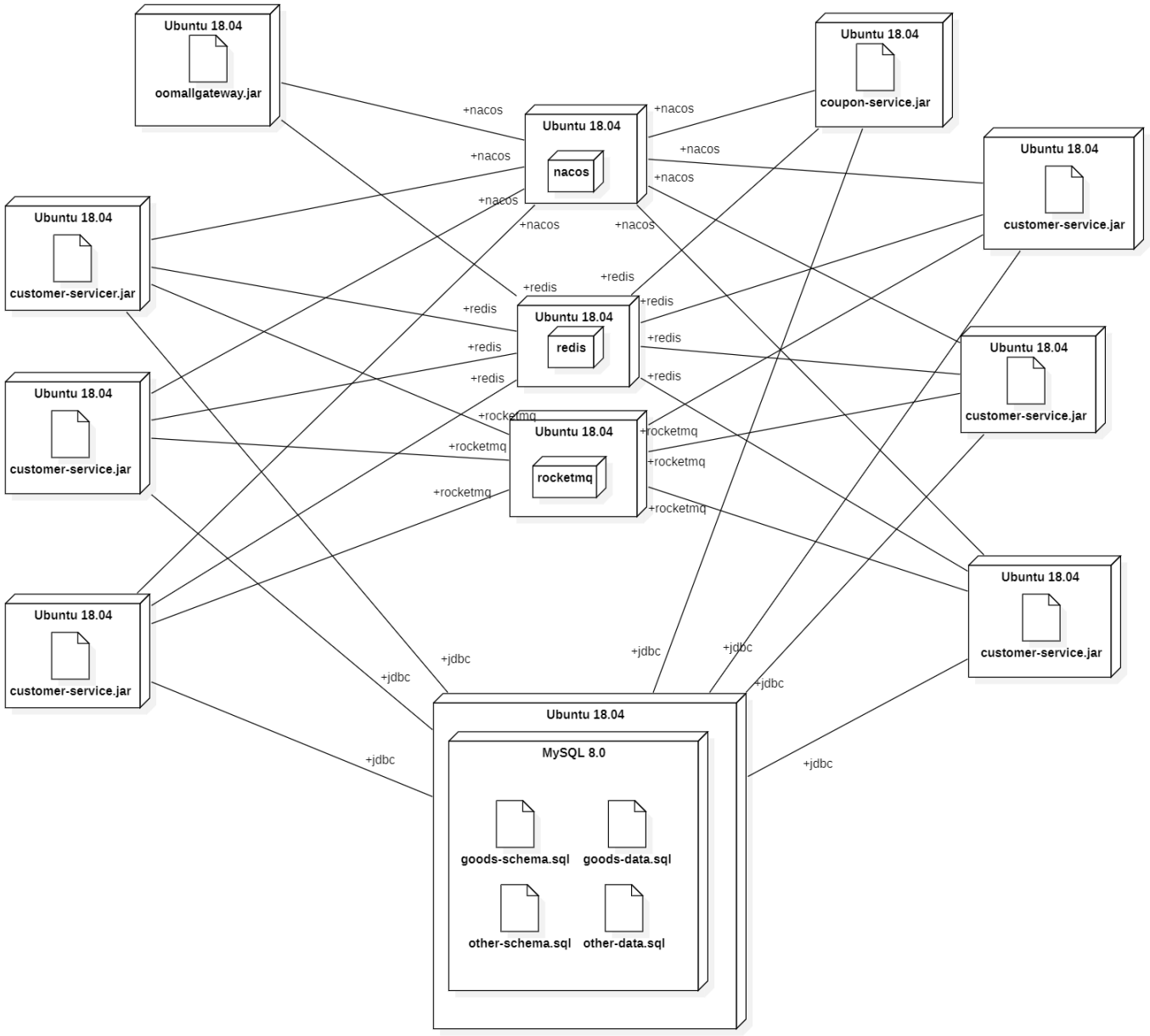
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**4323**	**0**	**0.00%**	**56.02**	**0**	**877**	**0.00**	**1.00**	**630.80**	**805.00**	**130.31**	**8.10**	**2.18**
BeanShell 取样器	3923	0	0.00%	0.11	0	6	0.00	1.00	1.00	1.00	126.35	0.00	0.00
HTTP Request	400	0	0.00%	604.33	136	877	647.50	806.90	839.00	867.99	234.60	157.55	42.38

可以看到响应时间明显延长
此时的 Response Time 分布图如下：

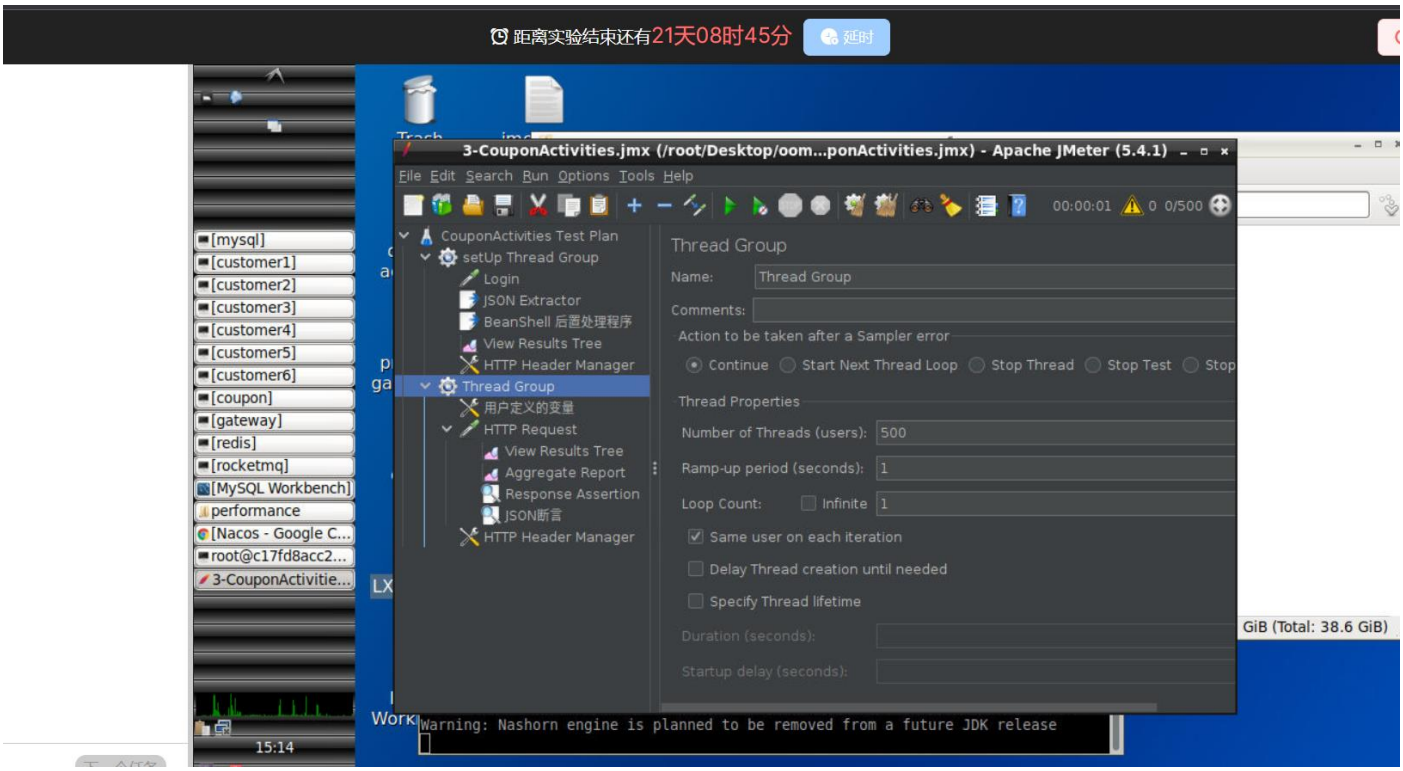


选测部分

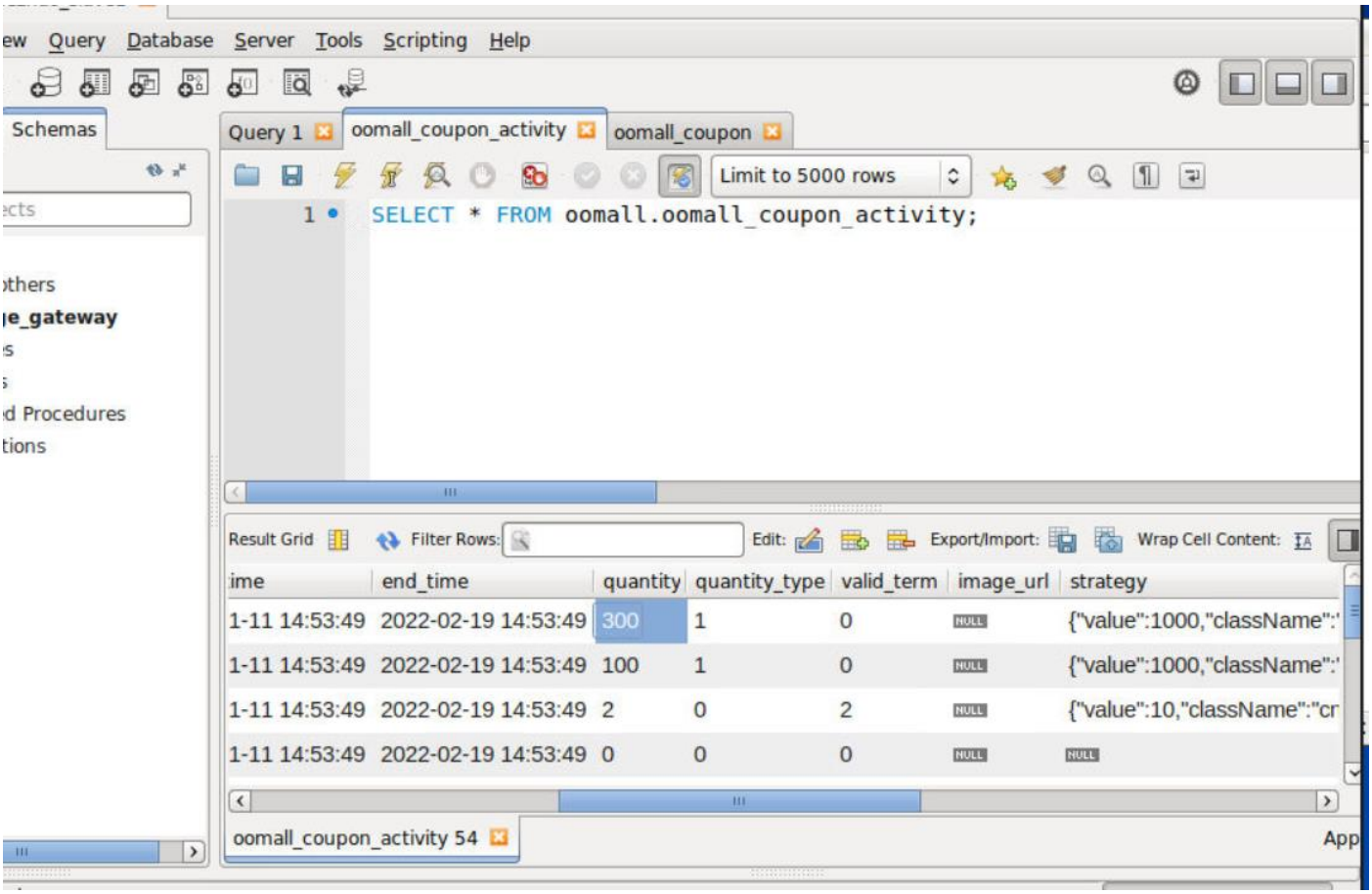
服务器部署图如下：


Ubuntu 18.04
oomallgateway.jar
Ubuntu 18.04
coupon-service.jar
Ubuntu 18.04
nacos
Ubuntu 18.04
customer-service.jar
Ubuntu 18.04
customer-servicer.jar
Ubuntu 18.04
redis
Ubuntu 18.04
customer-service.jar
Ubuntu 18.04
customer-service.jar
Ubuntu 18.04
rocketmq
Ubuntu 18.04
customer-service.jar
Ubuntu 18.04
customer-service.jar
Ubuntu 18.04
MySQL 8.0
goods-schema.sql
goods-data.sql
other-schema.sql
other-data.sql
+nacos
+redis
+rocketmq
+jdbc

线程数设置为 500



将 quantityType 为 1 的活动 1 的库存量设置为 300



使用命令行运行，可以看到错误是 200 个，处理 500 个请求不超过 1s

Nacos | Apache JMeter Dashboard | File | /root/Desktop/oomall/performance/result/index.html | Update

APDEX (Application Performance Index)

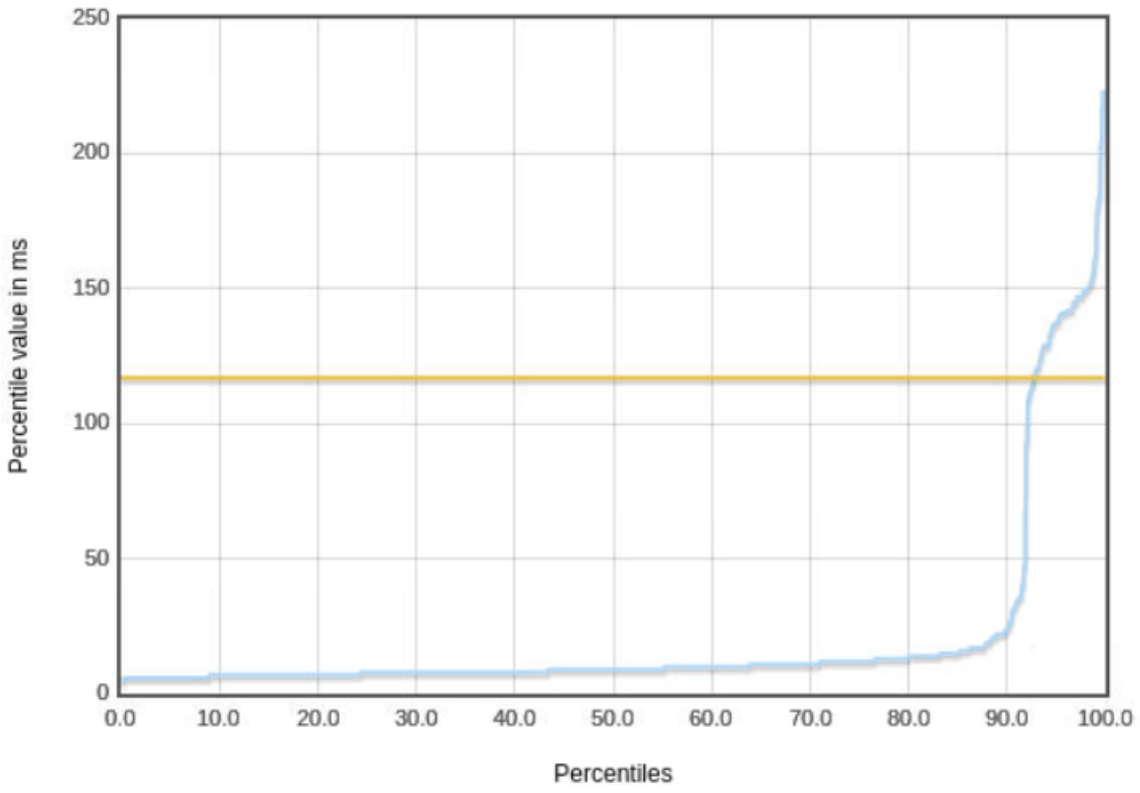
Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.601	**500 ms**	**1 sec 500 ms**	**Total**
0.600	500 ms	1 sec 500 ms	HTTP Request
1.000	500 ms	1 sec 500 ms	Login

Requests Summary

Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**501**	**200**	**39.92%**	**12.71**	**6**	**174**	**9.00**	**20.00**	**29.90**	**78.56**	**305.30**	**134.54**	**165.29**
HTTP Request	500	200	40.00%	12.47	6	174	9.00	20.00	29.00	57.00	350.14	154.14	189.77
Login	1	0	0.00%	134.00	134	134	134.00	134.00	134.00	134.00	7.46	4.96	1.85

Response Time Percentiles

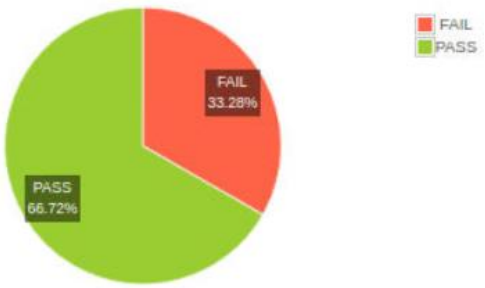


增加线程数至 600，测试结果如下：

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.640	**500 ms**	**1 sec 500 ms**	**Total**
0.639	500 ms	1 sec 500 ms	HTTP Request
1.000	500 ms	1 sec 500 ms	Login

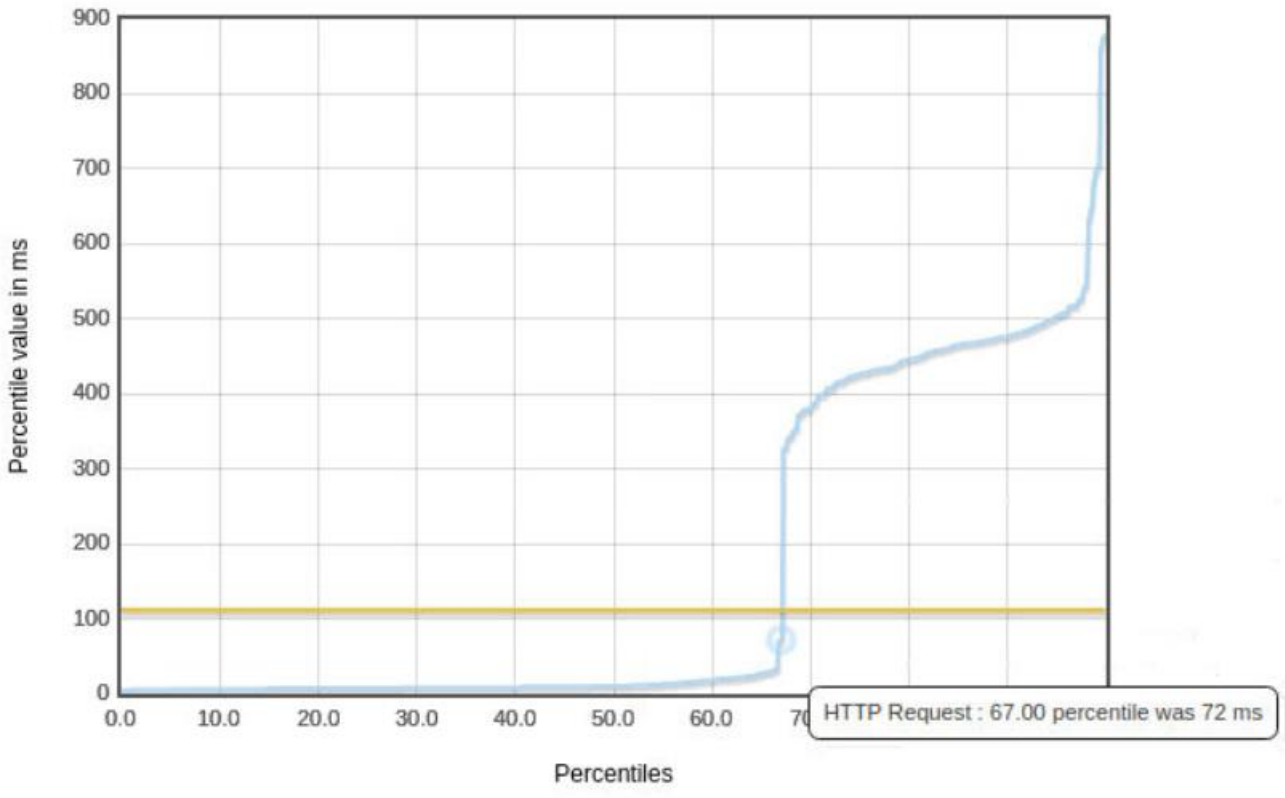
Requests Summary



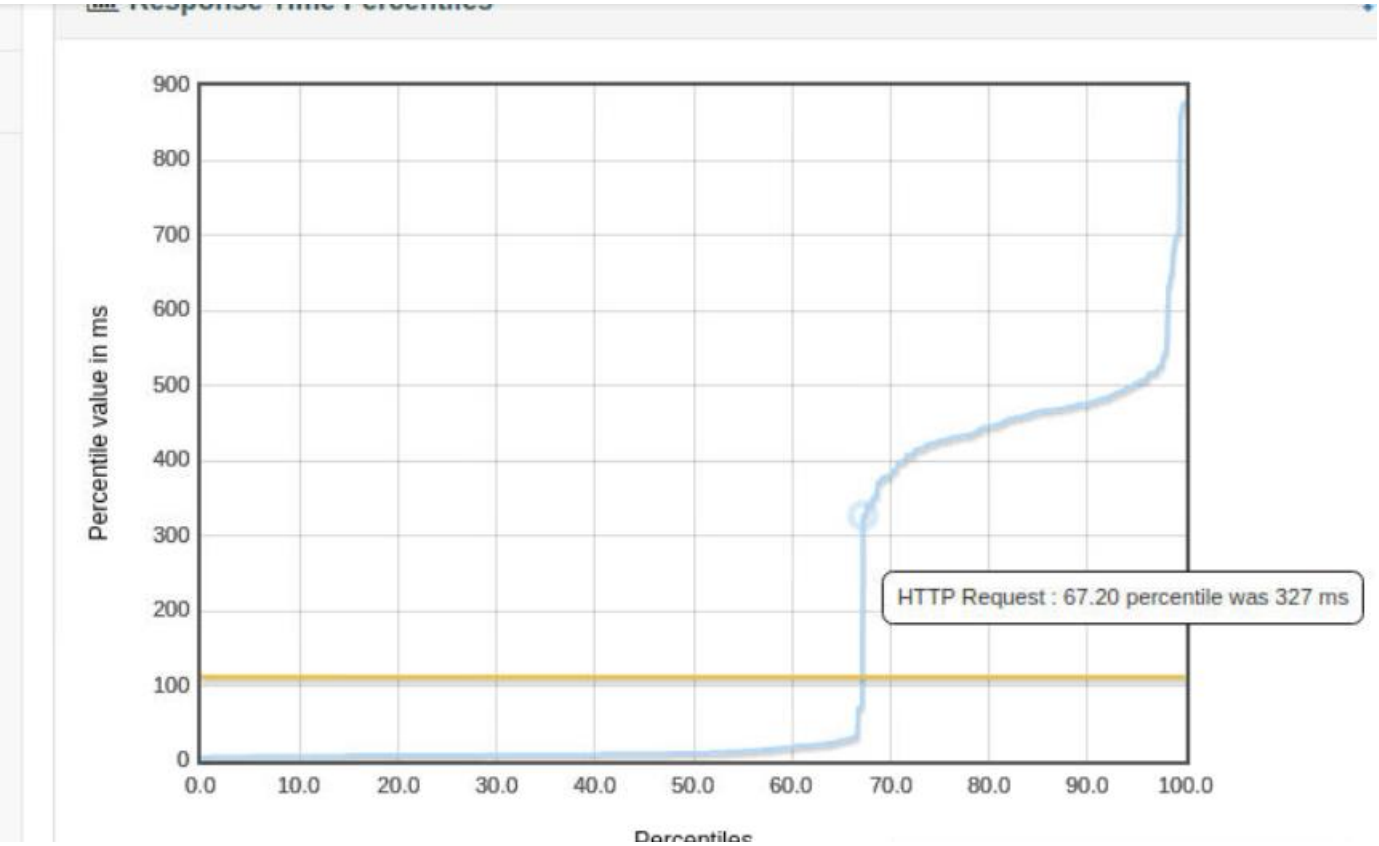
Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**601**	**200**	**33.28%**	**160.97**	**5**	**878**	**11.00**	**476.80**	**504.00**	**700.94**	**343.62**	**154.60**	**186.07**
HTTP Request	600	200	33.33%	161.05	5	878	11.00	476.90	504.00	700.97	386.85	173.90	209.67
Login	1	0	0.00%	112.00	112	112	112.00	112.00	112.00	112.00	8.93	5.94	2.21

Response Time Percentiles



Zoom :



可以看到，此次测试发生明显阻塞，在约 67%的线程时响应时间突然上升

阻塞分析:

在此之前，我们考虑了代码中可能阻塞的情况，并且尽量排除，比如使用 RocketMQ 的异步消息调用优惠券表的插入以及跨模块调用 Coupon 模块的扣库存。之后，考虑服务器原因，我们将 Customer 部署在 6 台服务器上，进行负载均衡，redis、RocketMQ、mysql、nacos 等辅助工具各自单独一台服务器。考虑到 coupon 的压力不大，因为我们只是使用 rocketmq 去通知它扣库存，所以 coupon 只分配一台服务器。oomallgateway 是商城网关，负责进行消息的转发。在进行以上配置，以及对 jvm 进行充分的预热之后，500 个线程基本上可以轻松解决，而 600 个线程则会在 60%的进度时就发生阻塞。经过分析，我们认为可能的原因在于负载并不仅限于 Customer 模块，应该对 nacos、oomallgateway 也进行负载均衡。因为我们在运行过程中对 nacos 所在服务器进行内存检测，发现其只剩 70MB 左右空闲内存；同时，网关在进行消息转发时也可能有极限值，但我们的测试需要向网关发送请求，再由网关进行消息的转发，因此为网关增设服务器不太现实，可以考虑为网关提供更高性能的服务器