istio-proxy性能洞察之路---性能调研的终点、调优之路的出发点

最近配合公司落地 service mesh，整体架构采用了istio 的部署架构，但是最近对envoy的sidecar做了压力测试，sidecar的性能是十分的差

说下istio-proxy是istio社区对envoy做了插件，包装成了istio-proxy,git目录是

```
https://github.com/istio/proxy
```

落地istio之后我们对istio-proxy性能进行了压测，每年技术大会演讲的envoy做sidecar在我们压测下，是那么单薄，显得差强人意，下面公布我们架构组的压测数据，希望给落地istio的朋友一些借鉴

一、envoy包装后的istio-proxy压测数据

我们的配置采用istio1.11官方httpbin的默认配置

下面开始公布我们的调研数据，当然这些调研数据也不全是我的成果，是项目组一起探测落地的

并发数变大时候，envoy 延迟时间变大分析

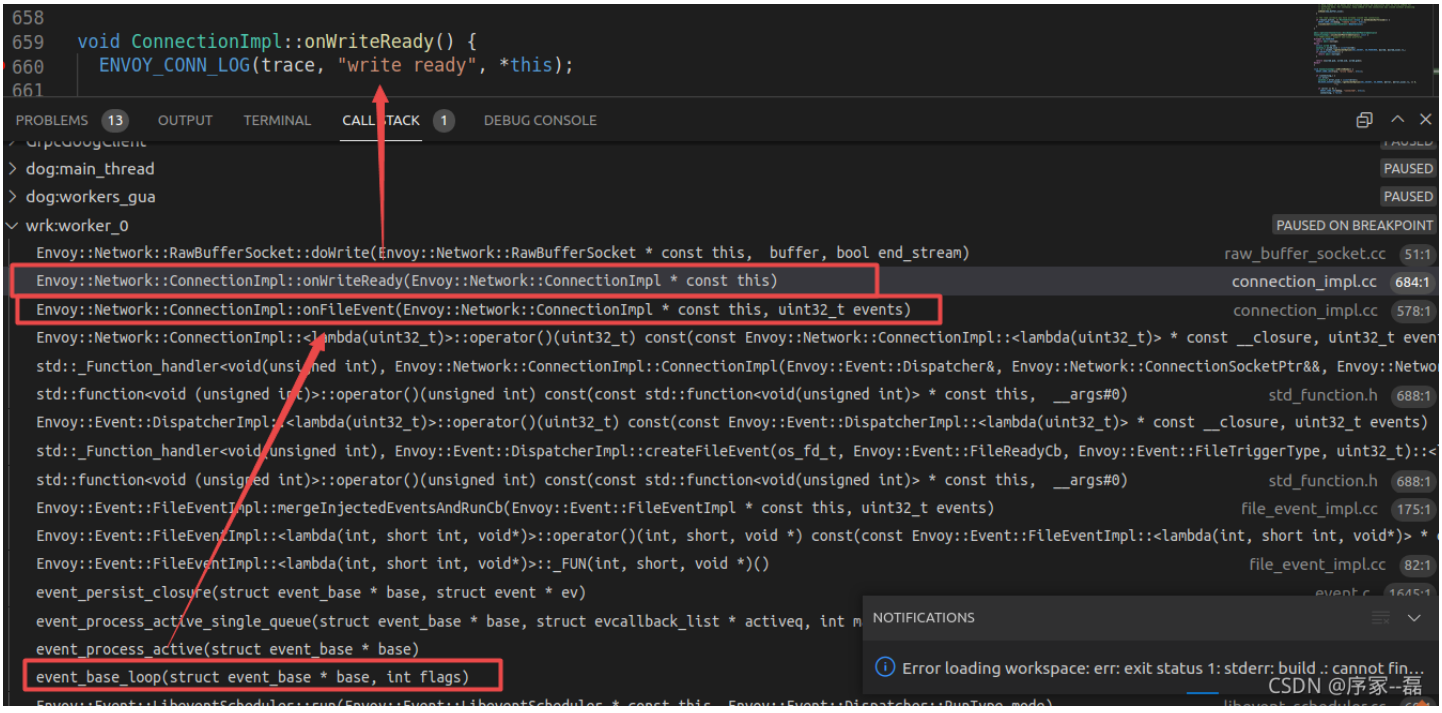
单独测试inbound ，使用ab 直接压测 pod 的ip：port

单个并发下： 请求平均时间 0.88ms

20个并发下： 请求平均时间5ms +

事件循环：日志截图，循环处理不同的socket 事件：

事件循环堆栈截图：




```
C1232 downstream 连接标识符

C988  upstream 连接标识符


```

下图是 长连接下的日志分析：

```
'x-b3-traceid', '10b7c3dd2c26c80c723efb80014f4da4'

2021-11-01T11:51:55.853815Z    trace    external/envoy/source/common/network/raw_buffer_socket.cc:67    envoy connection    [C1232] wri
前一个请求结束到下个请求过来 854531 - 853815 = 0.7ms
2021-11-01T11:51:55.854531Z    trace    external/envoy/source/common/network/connection_impl.cc:551    envoy connection    [C1232] sock
2021-11-01T11:51:55.854531Z    trace    external/envoy/source/common/network/connection_impl.cc:660    envoy connection    [C1232] writ

（854531-854541）10 微妙读取header，（854543-854585）30~40 微妙解析http
854531   [C1232] envoy connection
854536    trace    external/envoy/source/common/network/connection_impl.cc:589    envoy connection    [C1232] read ready. dispatch_buff
854541   raw_buffer_socket.cc:24    envoy connection    [C1232] read returns: 113
+10微妙
854543   raw_buffer_socket.cc:37    envoy connection    [C1232] read error: Resource temporarily unavailable
854566  [C1232] onHeadersCompleteBase
854571  http/http1/codec_impl.cc:1044    envoy http    [C1232] Server: onHeadersComplete size=4
**** header 解析完成30微妙
+40微妙

854576  external/envoy/source/common/network/connection_impl.cc:352    envoy connection    [C1232] readDisable: disable=true disable_co
+45微妙

                ConnectionManagerImpl::ActiveStream::decodeHeaders
854585     debug    external/envoy/source/common/http/conn_manager_impl.cc:857    envoy http    [C1232][S3760040057055989506] request h
+ 54微妙


854586    debug    external/envoy/source/common/http/filter_manager.cc:825    envoy http    [C1232][S3760040057055989506] request end s
+55微妙


32微妙: 854618 - 854586
854618    trace    external/envoy/source/common/http/filter_manager.cc:546    envoy http    [C1232][S3760040057055989506] decode header
854618    trace    external/envoy/source/common/http/filter_manager.cc:546    envoy http    [C1232][S3760040057055989506] decode header
854620    trace    external/envoy/source/common/http/filter_manager.cc:546    envoy http    [C1232][S3760040057055989506] decode header
854627    trace    external/envoy/source/common/http/filter_manager.cc:546    envoy http    [C1232][S3760040057055989506] decode header
'x-request-id', 'e2ba0a92-2e49-9243-8edc-e05fcac6d35d'
'x-b3-traceid', '10b7c3dd2c26c80c723efb80014f4da4'
'x-b3-spanid', '723efb80014f4da4'

854630    trace    external/envoy/source/common/http/filter_manager.cc:546    envoy http    [C1232][S3760040057055989506] decode header
+99微妙

854630   router.cc:443    envoy router    [C1232][S3760040057055989506] cluster 'inbound|9999||' match for URL '/pppp'
854647   external/envoy/source/common/router/router.cc:630    envoy router    [C1232][S3760040057055989506] router decoding headers:

854657    debug    external/envoy/source/common/conn_pool/conn_pool_base.cc:236    envoy pool    [C988] using existing connection
854658Z    debug    external/envoy/source/common/conn_pool/conn_pool_base.cc:175    envoy pool    [C988] creating stream

854661   external/envoy/source/common/router/upstream_request.cc:386    envoy router    [C1232][S3760040057055989506] pool ready

854671Z    trace    external/envoy/source/common/network/connection_impl.cc:474    envoy connection    [C988] writing 299 bytes, end_st

854678   external/envoy/source/common/http/filter_manager.cc:546    envoy http    [C1232][S3760040057055989506] decode headers called:
854678 - 854630 =  48微妙（router filter耗时）
+147微妙
854681   trace    external/envoy/source/common/http/http1/codec_impl.cc:613    envoy http    [C1232] parsed 113 bytes

854681  - 854531 = 0.15ms， 从接收客户端请求，到处理完毕转发

请求发送到可写0.5 ms ()
855179   trace    external/envoy/source/common/network/connection_impl.cc:551    envoy connection    [C1232] socket event: 2

2021-11-01T11:51:55.855180
855180   trace    external/envoy/source/common/network/connection_impl.cc:660    envoy connection    [C1232] write ready

855267Z    trace    external/envoy/source/common/network/connection_impl.cc:551    envoy connection    [C988] socket event: 2
855267Z    trace    external/envoy/source/common/network/connection_impl.cc:660    envoy connection    [C988] write ready
855272 网卡有抓包数据,这里数据已经发到网卡了  GET /ppp HTTP/1.1
855278Z    trace    external/envoy/source/common/network/raw_buffer_socket.cc:67    envoy connection    [C988] write returns: 299

855390 抓包发现这个时间，http 1.1 200 ok 返回数据已经在网卡上面了
855390 - 855272 网卡显示处理时间:  118微妙

855390 - 855267 实际时间 123微妙 ，通过网卡统计时间

程序延迟处理了:  856829 - 855390 = 1439
中间处理了:
```

```
C1242 + C993 + C943 + C1244 + C1236 + C1244 + C1236 + C907 +C992 +  C0 + C1226
```

```
856829 - 855278 业务请求时间：1551
```

```
856829Z    trace    external/envoy/source/common/network/connection_impl.cc:551    envoy connection    [C988] socket event: 3
856831Z    trace    external/envoy/source/common/network/connection_impl.cc:660    envoy connection    [C988] write ready
856832Z    trace    external/envoy/source/common/network/connection_impl.cc:589    envoy connection    [C988] read ready. dispatch_buff
856836Z    trace    external/envoy/source/common/network/raw_buffer_socket.cc:24    envoy connection    [C988] read returns: 179
856842Z    trace    external/envoy/source/common/network/raw_buffer_socket.cc:37    envoy connection    [C988] read error: Resource tem
856842Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:564    envoy http    [C988] parsing 179 bytes
856842Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:843    envoy http    [C988] message begin
856852Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:483    envoy http    [C988] completed header: key=X-B3-Tracei
856854Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:483    envoy http    [C988] completed header: key=Date value=
856854Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:483    envoy http    [C988] completed header: key=Content-Len
856854Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:694    envoy http    [C988] onHeadersCompleteBase
856855Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:483    envoy http    [C988] completed header: key=Content-Typ
856857Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:1264    envoy http    [C988] status_code 200
856859Z    trace    external/envoy/source/common/http/http1/codec_impl.cc:1274    envoy http    [C988] Client: onHeadersComplete size=4
856859 - 856829 response解析时间：30微妙

请求发送到业务再返回约1.7 ms（856863  - 855180  ）
856863   debug    external/envoy/source/common/router/router.cc:1230    envoy router    [C1232][S3760040057055989506] upstream headers
router void Filter::onUpstreamHeaders 花了11微妙，source/common/router/router.cc ： 1228
856874   trace    external/envoy/source/common/http/filter_manager.cc:1099    envoy http    [C1232][S3760040057055989506] encode header

856894Z    debug    external/envoy/source/common/http/conn_manager_impl.cc:1455    envoy http    [C1232][S3760040057055989506] encoding
':status', '200'
'x-b3-traceid', '10b7c3dd2c26c80c723efb80014f4da4'
'content-length', '14'

29微妙 =856903 - 856874 （encode headers）
856903    trace    external/envoy/source/common/network/connection_impl.cc:474    envoy connection    [C1232] writing 296 bytes, end_st

856909Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data
856909Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data
856913Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data
856913Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data
856915Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data
856916Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data
856916Z    trace    external/envoy/source/common/http/conn_manager_impl.cc:1464    envoy http    [C1232][S3760040057055989506] encoding

16微妙 = 856918- 856903  （encode data）
856918    trace    external/envoy/source/common/network/connection_impl.cc:474    envoy connection    [C1232] writing 14 bytes, end_str

520微妙=857438 - 856918
857438Z    trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data


857448   trace    external/envoy/source/common/http/filter_manager.cc:1267    envoy http    [C1232][S3760040057055989506] encode data c
857448 - 856863 = 585微妙， 0.58毫秒

859215 - 857448
等待返回客户端花了1.767ms

859215    trace    external/envoy/source/common/network/connection_impl.cc:551    envoy connection    [C1232] socket event: 2
859215    trace    external/envoy/source/common/network/connection_impl.cc:660    envoy connection    [C1232] write ready
859233    trace    external/envoy/source/common/network/raw_buffer_socket.cc:67    envoy connection    [C1232] write returns: 310

花费总时间 859233 - 854531 = 4.7ms （[C1232] write returns: 310）-（envoy connection    [C1232] socket event: 3）
```

二、outbound + inbound 性能测试

pod1 跟pod2 在k8s 同一node 原因:

一：排除网络干扰

二：不同机器时间戳可能会不同（差几毫秒）

压测工具： ab

测试场景是一个典型的 outbound + inbound 请求：

具体测试数据（长连接，带body）：

大部分业务配置1核即可， 广告业务等qps 高的需要配置2核

默认都使用1核，特殊的可以考虑通过namespace 或者打 label的方式来设置2核

1核（envoy 配置）：

outbound + inbound 性能测试 1核 request body : 1K response body :1K （qps 2000）

outbound + inbound 性能测试 1核 request body : 1K response body :4K （qps 2000）

outbound + inbound 性能测试 1核 request body : 1K response body :8K （qps 1900）

outbound + inbound 性能测试 1核 request body : 1K response body :500K （qps 900） （满足导购qps需求，大body 模仿导购）

2核（envoy 配置）：

outbound + inbound 性能测试 2核 request body : 1K response body :1K （qps 3700）

outbound + inbound 性能测试 2核 request body : 1K response body :2K （qps 3700） （满足广告业务qps需求）

outbound + inbound 性能测试 2核 request body : 1K response body :8K （qps 3600）

具体测试数据（长连接，不带body）：

单条请求分析： outbound + inbound 请求耗时分析详细

1核：outbound + inbound 性能测试 1核 （qps： 2200+）

2核：outbound + inbound 性能测试 2核 （qps： 4000+）

3核：outbound + inbound 性能测试 3核 （qps： 6000+）

4核：outbound + inbound 性能测试 4核 （qps： 7300+）

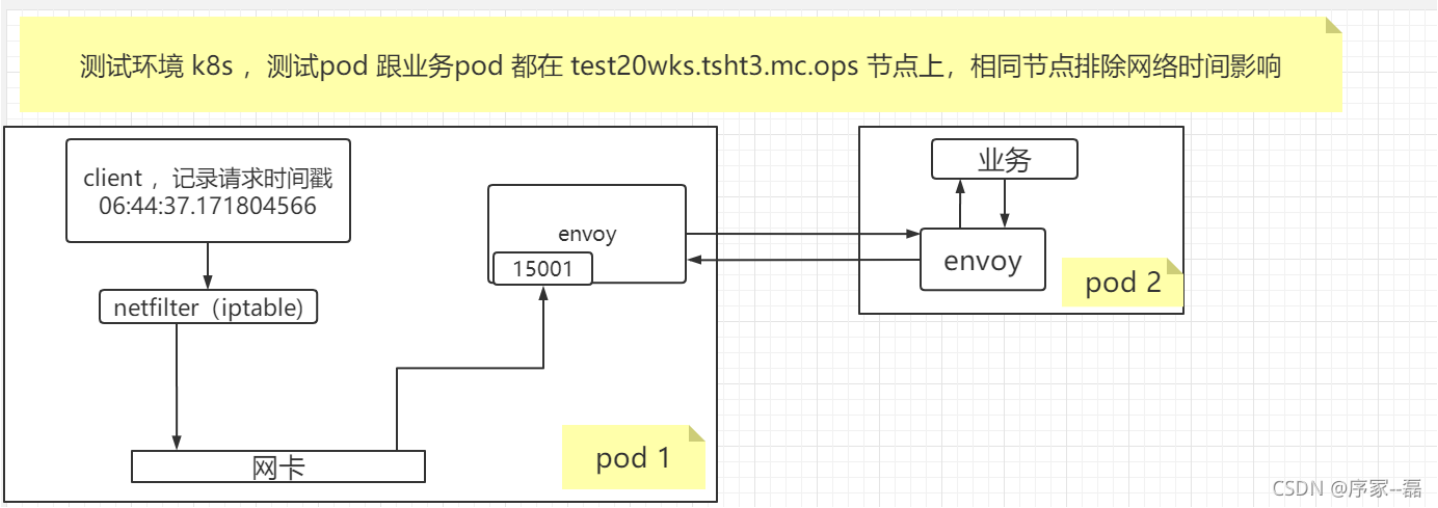
5核：outbound + inbound 性能测试 5核 （qps： 8600+）

6核：outbound + inbound 性能测试 6核 （qps： 9200+）

8核：outbound + inbound 性能测试 8核 （qps： 10000+）

在pod1 内部使用ab 压测pod2 的服务，pod1 与pod2 均有envoy sidecar

pod1 与pod2 均在 测试环境 k8s 的test20wks.tsht3.mc.ops 节点上



通过EnvoyFilter配置： inbound 负载均衡

默认情况下， 多个worker 之间不会做负载均衡，完全靠系统来分配，长连接场景下配置负载均衡，时间数据抖动会小一些

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name:  go-server-6-all-listener-balance
  namespace: zhaozhiyuan
spec:
  configPatches:
    - applyTo: LISTENER
      match:
        context: SIDECAR_INBOUND
        listener:
          portNumber: 15006
      patch:
        operation: MERGE
        value:
```

```
16          connection_balance_config:
17              exact_balance: {}
```

1.outbound + inbound 性能测试 1核 request body : 1K response body :1K

测试url： http://go-server-6-one-cpu-body-change.zhaozhiyuan.svc.cluster.local/

request body : 1K

response body :1K

测试命令：

./ab -n 10000 -c 1 -k -p ./1024 -H "Resp_size: 1024" http://go-server-6-one-cpu-body-change.zhaozhiyuan.svc.cluster.local/

Resp_size 调整response body 大小为1K

并发数	qps	平均时间	平均时间（所有并发平均值）	99线：分布 时间（毫秒）	99线：分布 时间 数量	Transfer rate
1	737.31	1.356	1.356	50% 1 66% 1 75% 2 80% 2 90% 2 95% 2 98% 2 99% 2 100% 5 (longest request)	69.740000% 1 6974 29.900000% 2 2990 0.290000% 3 29 0.050000% 4 5 0.020000% 5 2	963.23 [Kbytes/sec] received 894.99 kb/s sent 1858.22 kb/s total
2	1070.25	1.869	0.934	50% 2 66% 2 75% 2 80% 2 90% 2 95% 3 98% 3 99% 4 100% 10 (longest request)	21.850000% 1 2185 69.630000% 2 6963 7.370000% 3 737 0.660000% 4 66 0.250000% 5 25 0.090000% 6 9 0.100000% 7 10 0.020000% 8 2 0.020000% 9 2 0.010000% 10 1	1398.19 [Kbytes/sec] received 1299.14 kb/s sent 2697.33 kb/s total
3	1485.11	2.020	0.673	50% 2 66% 2 75% 2 80% 2 90% 3 95% 3 98% 3 99% 4 100% 11 (longest request)	11.160000% 1 1116 76.050000% 2 7605 11.680000% 3 1168 0.820000% 4 82 0.170000% 5 17 0.090000% 6 9 0.010000% 7 1 0.010000% 8 1 0.010000% 11 1	1940.23 [Kbytes/sec] received 1802.72 kb/s sent 3742.95 kb/s total
4	1609.36	2.485	0.621	50% 2 66% 3 75% 3 80% 3 90% 3 95% 3 98% 4 99% 4 100% 9 (longest request)	1.450000% 1 145 55.140000% 2 5514 38.830000% 3 3883 4.000000% 4 400 0.440000% 5 44 0.080000% 6 8 0.030000% 7 3 0.010000% 8 1 0.020000% 9 2	2102.52 [Kbytes/sec] received 1953.55 kb/s sent 4056.07 kb/s total
5	1737.00	2.879	0.576	50% 3 66% 3 75% 3 80% 3 90% 3 95% 4 98% 4 99% 4 100% 7 (longest request)	0.070000% 1 7 23.680000% 2 2368 66.750000% 3 6675 8.570000% 4 857 0.770000% 5 77 0.120000% 6 12 0.040000% 7 4	2269.30 [Kbytes/sec] received 2108.48 kb/s sent 4377.79 kb/s total
6	1792.99	3.346	0.558	50% 3 66% 4 75% 4 80% 4 90% 4 95% 4 98% 5	0.040000% 1 4 4.880000% 2 488 58.840000% 3 5884 33.380000% 4 3338 2.540000% 5 254 0.260000% 6 26 0.060000% 7 6	2342.44 [Kbytes/sec] received 2176.45 kb/s sent 4518.89 kb/s total

				99% 5 100% 7 (longest request)		
8	1871.33	4.275	0.534	50% 4 66% 4 75% 5 80% 5 90% 5 95% 5 98% 6 99% 6 100% 10 (longest request)	0.150000% 2 15 8.880000% 3 888 58.770000% 4 5877 28.820000% 5 2882 2.850000% 6 285 0.400000% 7 40 0.070000% 8 7 0.050000% 9 5 0.010000% 10 1	2444.77 [Kbytes/sec] received 2271.55 kb/s sent 4716.32 kb/s total
10	1904.48	5.251	0.525	50% 5 66% 5 75% 6 80% 6 90% 6 95% 6 98% 7 99% 8 100% 14 (longest request)	0.280000% 3 28 10.140000% 4 1014 60.110000% 5 6011 25.880000% 6 2588 2.310000% 7 231 0.840000% 8 84 0.150000% 9 15 0.070000% 10 7 0.060000% 11 6 0.070000% 12 7 0.040000% 13 4 0.050000% 14 5	2488.08 [Kbytes/sec] received 2311.78 kb/s sent 4799.86 kb/s total
15	1994.25	7.522	0.501	50% 7 66% 8 75% 8 80% 8 90% 8 95% 9 98% 9 99% 10 100% 17 (longest request)	0.050000% 3 5 0.030000% 4 3 0.500000% 5 50 6.630000% 6 663 43.460000% 7 4346 41.090000% 8 4109 7.020000% 9 702 0.900000% 10 90 0.170000% 11 17 0.060000% 12 6 0.020000% 13 2 0.020000% 14 2 0.020000% 15 2 0.010000% 16 1 0.020000% 17 2	2605.34 [Kbytes/sec] received 2420.76 kb/s sent 5026.10 kb/s total
20	1999.18	10.004	0.500	50% 10 66% 10 75% 11 80% 11 90% 11 95% 12 98% 12 99% 13 100% 22 (longest request)	0.010000% 4 1 0.030000% 5 3 0.250000% 6 25 0.760000% 7 76 3.990000% 8 399 24.150000% 9 2415 44.170000% 10 4417 21.170000% 11 2117 4.250000% 12 425 0.690000% 13 69 0.270000% 14 27 0.130000% 15 13 0.030000% 16 3 0.030000% 17 3 0.010000% 18 1 0.020000% 19 2 0.020000% 20 2 0.010000% 21 1 0.010000% 22 1	2612.11 [Kbytes/sec] received 2426.74 kb/s sent 5038.85 kb/s total

2.outbound + inbound 性能测试 1核 request body : 1K response body :4K

测试url： http://go-server-6-one-cpu-body-change.zhaozhiyuan.svc.cluster.local/

测试命令：

./ab -n 10000 -c 1 -k -p ./1024 -H "Resp_size: 4096" http://go-server-6-one-cpu-body-change.zhaozhiyuan.svc.cluster.local/

Resp_size 调整response body 大小为4K

并发数	qps	平均时间	平均时间（所有并发平均值）	99线：分布 时间（毫秒）	99线：分布 时间 数量	Transfer rate
1	729.32	1.371	1.371	50% 1 66% 1 75% 2 80% 2 90% 2 95% 2	0.080000% 0 8 73.590000% 1 7359 25.660000% 2 2566 0.530000% 3 53 0.100000% 4 10 0.020000% 5 2	3150.20 [Kbytes/sec] received 885.30 kb/s sent 4035.50 kb/s total

				98% 2 99% 2 100% 9 (longest request)	0.010000% 7 1 0.010000% 9 1	
2	1247.86	1.603	0.801	50% 2 66% 2 75% 2 80% 2 90% 2 95% 2 98% 2 99% 3 100% 7 (longest request)	0.070000% 0 7 39.220000% 1 3922 59.480000% 2 5948 1.100000% 3 110 0.100000% 4 10 0.020000% 5 2 0.010000% 7 1	5388.35 [Kbytes/sec] received 1514.74 kb/s sent 6903.09 kb/s total
3	1545.59	1.941	0.647	50% 2 66% 2 75% 2 80% 2 90% 2 95% 3 98% 3 99% 3 100% 4 (longest request)	0.020000% 0 2 13.310000% 1 1331 77.950000% 2 7795 8.480000% 3 848 0.240000% 4 24	6679.31 [Kbytes/sec] received 1876.14 kb/s sent 8555.44 kb/s total
4	1628.53	2.456	0.614	50% 2 66% 3 75% 3 80% 3 90% 3 95% 3 98% 4 99% 4 100% 7 (longest request)	1.860000% 1 186 55.980000% 2 5598 38.870000% 3 3887 3.040000% 4 304 0.170000% 5 17 0.050000% 6 5 0.030000% 7 3	7039.84 [Kbytes/sec] received 1976.82 kb/s sent 9016.66 kb/s total
5	1709.22	2.925	0.585	50% 3 66% 3 75% 3 80% 3 90% 4 95% 4 98% 4 99% 5 100% 7 (longest request)	0.010000% 0 1 0.180000% 1 18 21.460000% 2 2146 66.300000% 3 6630 10.660000% 4 1066 1.020000% 5 102 0.280000% 6 28 0.090000% 7 9	7390.16 [Kbytes/sec] received 2074.76 kb/s sent 9464.92 kb/s total
6	1775.33	3.380	0.563	50% 3 66% 4 75% 4 80% 4 90% 4 95% 4 98% 5 99% 5 100% 8 (longest request)	0.020000% 1 2 5.190000% 2 519 56.160000% 3 5616 34.970000% 4 3497 3.310000% 5 331 0.320000% 6 32 0.020000% 7 2 0.010000% 8 1	7675.19 [Kbytes/sec] received 2155.02 kb/s sent 9830.21 kb/s total
8	1858.07	4.306	0.538	50% 4 66% 5 75% 5 80% 5 90% 5 95% 5 98% 6 99% 6 100% 9 (longest request)	0.010000% 1 1 0.140000% 2 14 8.630000% 3 863 56.860000% 4 5686 30.380000% 5 3038 3.590000% 6 359 0.330000% 7 33 0.050000% 8 5 0.010000% 9 1	8035.32 [Kbytes/sec] received 2255.45 kb/s sent 10290.77 kb/s total
10	1797.31	5.564	0.556	50% 5 66% 6 75% 6 80% 6 90% 7 95% 7 98% 8 99% 9 100% 14 (longest request)	0.050000% 2 5 0.220000% 3 22 6.610000% 4 661 47.460000% 5 4746 34.900000% 6 3490 6.260000% 7 626 3.050000% 8 305 1.020000% 9 102 0.240000% 10 24 0.090000% 11 9 0.060000% 12 6 0.010000% 13 1 0.030000% 14 3	7772.54 [Kbytes/sec] received 2181.69 kb/s sent 9954.23 kb/s total
15	1814.75	8.266	0.551	50% 8 66% 8 75% 8	0.020000% 3 2 0.040000% 4 4 0.330000% 5 33	7848.74 [Kbytes/sec] received 2202.87 kb/s sent 10051.60 kb/s total

				80% 9 90% 9 95% 9 98% 10 99% 12 100% 111 (longest request)	4.070000% 6 407 27.940000% 7 2794 45.460000% 8 4546 17.470000% 9 1747 3.070000% 10 307 0.430000% 11 43 0.250000% 12 25 0.070000% 13 7 0.020000% 14 2 0.050000% 15 5 0.100000% 16 10 0.060000% 17 6 0.060000% 18 6 0.110000% 19 11 0.020000% 22 2 0.020000% 23 2 0.040000% 24 4 0.010000% 25 1 0.040000% 26 4 0.010000% 27 1 0.010000% 29 1 0.010000% 106 1 0.050000% 108 5 0.130000% 109 13 0.090000% 110 9 0.020000% 111 2	
20	1984.46	10.078	0.504	50% 10 66% 10 75% 11 80% 11 90% 11 95% 12 98% 12 99% 13 100% 20 (longest request)	0.010000% 4 1 0.170000% 6 17 0.560000% 7 56 4.350000% 8 435 23.730000% 9 2373 40.980000% 10 4098 22.460000% 11 2246 5.920000% 12 592 0.930000% 13 93 0.360000% 14 36 0.200000% 15 20 0.150000% 16 15 0.080000% 17 8 0.060000% 18 6 0.030000% 19 3 0.010000% 20 1	8583.15 [Kbytes/sec] received 2408.87 kb/s sent 10992.02 kb/s total

3.outbound + inbound 性能测试 1核 request body : 1K response body :500K

测试url： http://go-server-6-one-cpu.zhaozhiyuan.svc.cluster.local/

./ab -n 10000 -c 1 -k -p ./1024 -H "Resp_size: 512000" http://go-server-6-one-cpu-body-change.zhaozhiyuan.svc.cluster.local/

Resp_size 调整response body 大小为500K

模拟导购业务:

request body : 1K

response body :500K

并发数	qps	平均时间	平均时间（所有并发平均值）	99线：分布 时间（毫秒）	99线：分布 时间 数量	Transfer rate
1	499.13	2.003	2.003	50% 2 66% 2 75% 3 80% 3 90% 3 95% 3 98% 4 99% 4 100% 10 (longest request)	0.050000% 0 5 32.750000% 1 3275 39.460000% 2 3946 25.210000% 3 2521 2.440000% 4 244 0.060000% 5 6 0.010000% 6 1 0.010000% 7 1 0.010000% 10 1	124903.80 [Kbytes/sec] received 606.86 kb/s sent 125510.66 kb/s total
2	712.58	2.807	1.403	50% 3 66% 3 75% 4 80% 4 90% 4 95% 5 98% 5 99% 6 100% 9 (longest request)	0.110000% 0 11 13.090000% 1 1309 31.270000% 2 3127 28.110000% 3 2811 17.960000% 4 1796 8.160000% 5 816 1.110000% 6 111 0.170000% 7 17	178336.32 [Kbytes/sec] received 866.37 kb/s sent 179202.70 kb/s total

					0.010000% 8 1 0.010000% 9 1	
3	727.26	4.125	1.375	50% 4 66% 5 75% 6 80% 6 90% 7 95% 7 98% 8 99% 9 100% 13 (longest request)	0.580000% 0 58 11.340000% 1 1134 14.260000% 2 1426 15.010000% 3 1501 15.890000% 4 1589 15.700000% 5 1570 13.500000% 6 1350 8.840000% 7 884 3.040000% 8 304 1.190000% 9 119 0.330000% 10 33 0.170000% 11 17 0.130000% 12 13 0.020000% 13 2	182003.97 [Kbytes/sec] received 884.22 kb/s sent 182888.18 kb/s total
4	770.33	5.193	1.298	50% 5 66% 7 75% 7 80% 8 90% 9 95% 10 98% 11 99% 12 100% 17 (longest request)	1.000000% 0 100 13.020000% 1 1302 10.260000% 2 1026 6.480000% 3 648 10.260000% 4 1026 12.570000% 5 1257 11.680000% 6 1168 11.560000% 7 1156 9.950000% 8 995 6.860000% 9 686 3.260000% 10 326 1.690000% 11 169 0.840000% 12 84 0.270000% 13 27 0.130000% 14 13 0.090000% 15 9 0.070000% 16 7 0.010000% 17 1	192728.66 [Kbytes/sec] received 936.59 kb/s sent 193665.25 kb/s total
5	806.32	6.201	1.240	50% 6 66% 8 75% 9 80% 10 90% 11 95% 12 98% 14 99% 15 100% 19 (longest request)	0.610000% 0 61 16.080000% 1 1608 10.290000% 2 1029 3.420000% 3 342 3.670000% 4 367 7.160000% 5 716 10.550000% 6 1055 9.710000% 7 971 8.570000% 8 857 8.370000% 9 837 7.540000% 10 754 6.010000% 11 601 3.350000% 12 335 2.190000% 13 219 1.310000% 14 131 0.660000% 15 66 0.330000% 16 33 0.130000% 17 13 0.040000% 18 4 0.010000% 19 1	01744.78 [Kbytes/sec] received 980.34 kb/s sent 202725.12 kb/s total
6	844.94	7.101	1.184	50% 7 66% 10 75% 11 80% 12 90% 14 95% 15 98% 17 99% 18 100% 31 (longest request)	0.640000% 0 64 22.390000% 1 2239 13.370000% 2 1337 1.600000% 3 160 1.260000% 4 126 2.020000% 5 202 3.360000% 6 336 5.500000% 7 550 5.380000% 8 538 5.750000% 9 575 6.840000% 10 684 8.130000% 11 813 7.370000% 12 737 5.700000% 13 570 3.800000% 14 380 2.670000% 15 267 1.580000% 16 158 1.170000% 17 117 0.680000% 18 68 0.360000% 19 36 0.150000% 20 15 0.090000% 21 9	211408.09 [Kbytes/sec] received 1027.30 kb/s sent 212435.39 kb/s total

					0.070000% 22 7 0.050000% 23 5 0.030000% 24 3 0.020000% 25 2 0.010000% 28 1 0.010000% 31 1	
8	839.52	9.529	1.191	50% 10 66% 15 75% 16 80% 17 90% 19 95% 21 98% 23 99% 25 100% 32 (longest request)	0.850000% 0 85 21.520000% 1 2152 19.520000% 2 1952 1.180000% 3 118 0.330000% 4 33 0.520000% 5 52 0.530000% 6 53 1.010000% 7 101 1.290000% 8 129 1.730000% 9 173 2.310000% 10 231 2.230000% 11 223 2.580000% 12 258 3.260000% 13 326 4.280000% 14 428 5.680000% 15 568 7.140000% 16 714 6.470000% 17 647 4.960000% 18 496 3.410000% 19 341 2.630000% 20 263 2.240000% 21 224 1.380000% 22 138 1.180000% 23 118 0.610000% 24 61 0.500000% 25 50 0.380000% 26 38 0.120000% 27 12 0.050000% 28 5 0.040000% 29 4 0.020000% 30 2 0.020000% 31 2 0.030000% 32 3	210482.06 [Kbytes/sec] received 1020.71 kb/s sent 211502.77 kb/s total
10	922.03	10.846	1.085	50% 13 66% 19 75% 20 80% 20 90% 22 95% 23 98% 25 99% 26 100% 73 (longest request)	1.270000% 0 127 29.880000% 1 2988 18.270000% 2 1827 0.360000% 3 36 0.060000% 4 6 0.020000% 5 2 0.020000% 7 2 0.010000% 8 1 0.020000% 9 2 0.040000% 10 4 0.040000% 11 4 0.010000% 12 1 0.010000% 13 1 0.070000% 14 7 0.250000% 15 25 0.690000% 16 69 2.140000% 17 214 5.900000% 18 590 11.950000% 19 1195 11.780000% 20 1178 7.090000% 21 709 3.670000% 22 367 2.290000% 23 229 1.630000% 24 163 1.070000% 25 107 0.590000% 26 59 0.400000% 27 40 0.200000% 28 20 0.060000% 29 6 0.070000% 30 7 0.030000% 31 3 0.010000% 32 1 0.010000% 68 1 0.050000% 69 5 0.020000% 72 2 0.020000% 73 2	231011.30 [Kbytes/sec] received 1121.02 kb/s sent 232132.33 kb/s total
15	906.98	16.538	1.103	50% 9 66% 29	0.320000% 0 32 30.950000% 1 3095	226922.73 [Kbytes/sec] received 1102.73 kb/s sent

				75% 31 80% 32 90% 35 95% 38 98% 41 99% 43 100% 59 (longest request)	17.640000% 2 1764 0.910000% 3 91 0.110000% 4 11 0.020000% 5 2 0.010000% 6 1 0.040000% 7 4 0.010000% 9 1 0.010000% 17 1 0.030000% 20 3 0.080000% 21 8 0.110000% 22 11 0.380000% 23 38 0.720000% 24 72 1.090000% 25 109 2.170000% 26 217 3.550000% 27 355 4.720000% 28 472 4.620000% 29 462 5.280000% 30 528 5.160000% 31 516 4.480000% 32 448 3.580000% 33 358 3.020000% 34 302 2.010000% 35 201 1.870000% 36 187 1.590000% 37 159 1.200000% 38 120 1.120000% 39 112 0.930000% 40 93 0.720000% 41 72 0.430000% 42 43 0.410000% 43 41 0.150000% 44 15 0.130000% 45 13 0.170000% 46 17 0.040000% 47 4 0.060000% 48 6 0.010000% 49 1 0.020000% 50 2 0.030000% 51 3 0.020000% 52 2 0.020000% 53 2 0.020000% 54 2 0.010000% 55 1 0.020000% 56 2 0.010000% 59 1	228025.46 kb/s total
20	888.86	22.501	1.125	50% 11 66% 41 75% 42 80% 43 90% 48 95% 51 98% 54 99% 56 100% 81 (longest request)	0.110000% 0 11 29.910000% 1 2991 18.550000% 2 1855 1.320000% 3 132 0.080000% 4 8 0.020000% 5 2 0.010000% 6 1 0.010000% 11 1 0.010000% 23 1 0.010000% 25 1 0.010000% 29 1 0.030000% 30 3 0.040000% 31 4 0.190000% 32 19 0.140000% 33 14 0.270000% 34 27 0.640000% 35 64 0.930000% 36 93 1.650000% 37 165 2.650000% 38 265 3.600000% 39 360 4.670000% 40 467 5.590000% 41 559 5.510000% 42 551 4.700000% 43 470 3.400000% 44 340 2.400000% 45 240 1.780000% 46 178 1.520000% 47 152 1.700000% 48 170 1.380000% 49 138 1.410000% 50 141	222405.00 [Kbytes/sec] received 1080.70 kb/s sent 223485.69 kb/s total

					1.310000% 51 131 1.040000% 52 104 0.810000% 53 81 0.820000% 54 82 0.430000% 55 43 0.370000% 56 37 0.200000% 57 20 0.140000% 58 14 0.160000% 59 16 0.050000% 60 5 0.080000% 61 8 0.080000% 62 8 0.020000% 63 2 0.040000% 64 4 0.040000% 65 4 0.020000% 66 2 0.010000% 67 1 0.010000% 69 1 0.030000% 70 3 0.010000% 71 1 0.010000% 72 1 0.020000% 74 2 0.020000% 75 2 0.010000% 76 1 0.020000% 77 2 0.010000% 81 1	

4. outbound + inbound 性能测试 2核 request body : 1K response body :8K

测试url： http://go-server-6-two-cpu-body-change.zhaozhiyuan.svc.cluster.local/

request body : 1K

response body :8K

测试命令：

./ab -n 10000 -c 1 -k -p ./1024 -H "Resp_size: 8192" http://go-server-6-two-cpu-body-change.zhaozhiyuan.svc.cluster.local/

Resp_size 调整response body 大小为8K

并发数	qps	平均时间	平均时间（所有并发平均值）	99线：分布 时间（毫秒）	99线：分布 时间 数量	Transfer rate
1	730.81	1.368	1.368	50% 1 66% 1 75% 2 80% 2 90% 2 95% 2 98% 2 99% 2 100% 5 (longest request)	0.280000% 0 28 72.800000% 1 7280 26.420000% 2 2642 0.410000% 3 41 0.060000% 4 6 0.030000% 5 3	6055.42 [Kbytes/sec] received 887.11 kb/s sent 6942.53 kb/s total
2	1241.49	1.611	0.805	50% 2 66% 2 75% 2 80% 2 90% 2 95% 2 98% 2 99% 3 100% 5 (longest request)	0.220000% 0 22 38.800000% 1 3880 59.770000% 2 5977 1.120000% 3 112 0.080000% 4 8 0.010000% 5 1	10281.72 [Kbytes/sec] received 1507.01 kb/s sent 11788.72 kb/s total
3	1919.78	1.563	0.521	50% 2 66% 2 75% 2 80% 2 90% 2 95% 2 98% 2 99% 3 100% 7 (longest request)	0.260000% 0 26 46.860000% 1 4686 51.250000% 2 5125 1.450000% 3 145 0.150000% 4 15 0.010000% 5 1 0.010000% 6 1 0.010000% 7 1	15875.19 [Kbytes/sec] received 2330.36 kb/s sent 18205.56 kb/s total
4	2204.31	1.815	0.454	50% 2 66% 2 75% 2 80% 2 90% 2 95% 3	0.340000% 0 34 27.090000% 1 2709 63.310000% 2 6331 8.830000% 3 883 0.340000% 4 34 0.090000% 5 9	18242.62 [Kbytes/sec] received 2675.74 kb/s sent 20918.36 kb/s total

				98% 3 99% 3 100% 5 (longest request)		
5	2343.78	2.133	0.427	50% 2 66% 2 75% 3 80% 3 90% 3 95% 3 98% 3 99% 4 100% 7 (longest request)	0.170000% 0 17 21.440000% 1 2144 49.030000% 2 4903 27.400000% 3 2740 1.850000% 4 185 0.100000% 5 10 0.010000% 7 1	19410.56 [Kbytes/sec] received 2845.04 kb/s sent 22255.60 kb/s total
6	2394.59	2.506	0.418	50% 3 66% 3 75% 3 80% 3 90% 4 95% 4 98% 4 99% 4 100% 7 (longest request)	0.140000% 0 14 18.710000% 1 1871 23.190000% 2 2319 47.820000% 3 4782 9.600000% 4 960 0.460000% 5 46 0.070000% 6 7 0.010000% 7 1	19863.22 [Kbytes/sec] received 2906.71 kb/s sent 22769.94 kb/s total
8	2899.69	2.759	0.345	50% 3 66% 3 75% 4 80% 4 90% 4 95% 4 98% 5 99% 5 100% 7 (longest request)	0.140000% 0 14 12.390000% 1 1239 29.310000% 2 2931 30.540000% 3 3054 24.490000% 4 2449 2.930000% 5 293 0.160000% 6 16 0.040000% 7 4	24036.23 [Kbytes/sec] received 3519.84 kb/s sent 27556.07 kb/s total
10	3081.29	3.245	0.325	50% 3 66% 4 75% 4 80% 4 90% 5 95% 5 98% 6 99% 6 100% 12 (longest request)	0.070000% 0 7 4.680000% 1 468 29.840000% 2 2984 19.860000% 3 1986 30.930000% 4 3093 12.380000% 5 1238 1.690000% 6 169 0.260000% 7 26 0.130000% 8 13 0.080000% 9 8 0.020000% 10 2 0.020000% 11 2 0.040000% 12 4	25559.48 [Kbytes/sec] received 3740.28 kb/s sent 29299.76 kb/s total
15	3310.97	4.530	0.302	50% 4 66% 5 75% 5 80% 5 90% 6 95% 6 98% 7 99% 8 100% 13 (longest request)	0.030000% 0 3 0.090000% 1 9 0.830000% 2 83 11.660000% 3 1166 41.070000% 4 4107 33.010000% 5 3301 10.080000% 6 1008 1.950000% 7 195 0.580000% 8 58 0.320000% 9 32 0.190000% 10 19 0.090000% 11 9 0.050000% 12 5 0.050000% 13 5	27527.91 [Kbytes/sec] received 4019.08 kb/s sent 31546.98 kb/s total
20	3619.70	5.525	0.276	50% 5 66% 6 75% 6 80% 6 90% 7 95% 7 98% 8 99% 8 100% 19 (longest request)	0.040000% 0 4 0.060000% 2 6 0.410000% 3 41 8.030000% 4 803 44.530000% 5 4453 36.730000% 6 3673 8.030000% 7 803 1.730000% 8 173 0.190000% 9 19 0.070000% 10 7 0.020000% 11 2 0.040000% 12 4 0.020000% 13 2 0.010000% 14 1 0.020000% 15 2 0.020000% 16 2 0.010000% 17 1	30125.07 [Kbytes/sec] received 4393.83 kb/s sent 34518.90 kb/s total

					0.020000% 18 2	
					0.020000% 19 2	

三、性能洞察

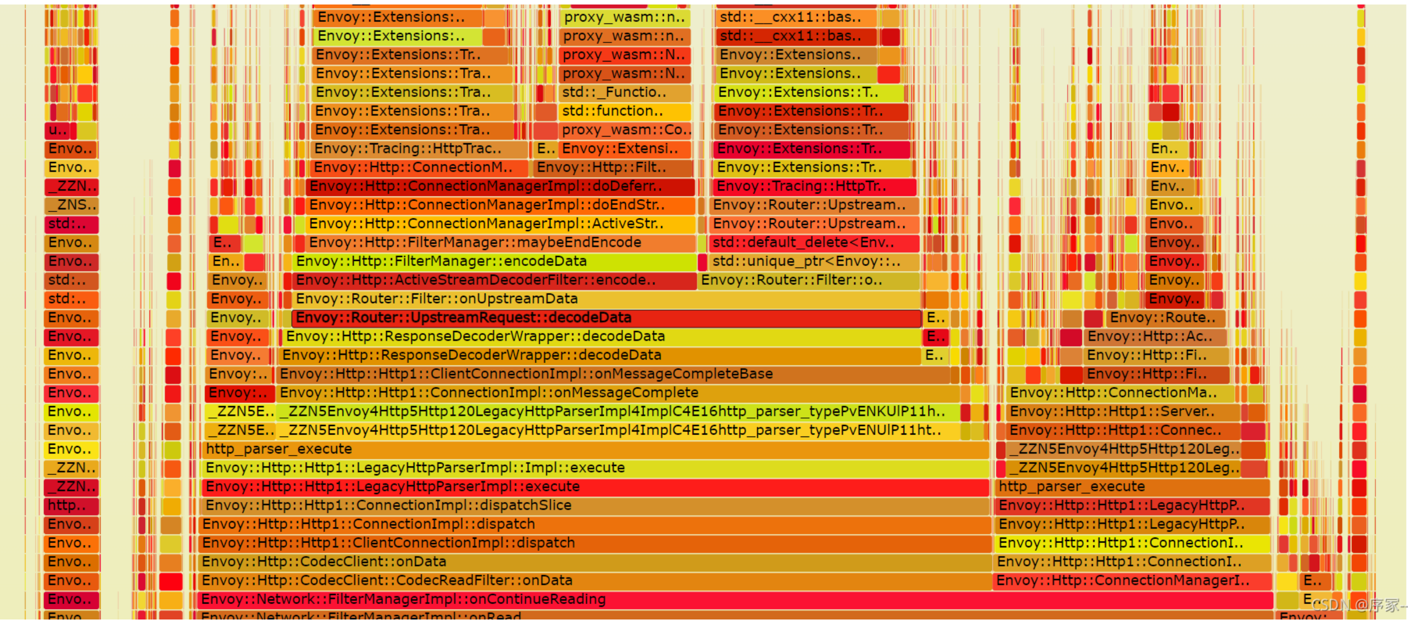
1.cpu性能洞察

我在istio生产环境对 istio-proxy使用perf 进行洞察分析

至于perf 如何用还有如何生成火焰图自己去网上查吧。不细说了

```
#1.按照 perf
sudo apt update
sudo apt install linux-tools-common
wget http://launchpadlibrarian.net/145025421/linux-tools-3.10.0-3_3.10.0-3.12_amd64.deb
dokg -i linux-tools-3.10.0-3_3.10.0-3.12_amd64.deb
perf record -p 71965 -a -g
perf script -i perf.data &> perf.unfold
stackcollapse-perf.pl perf.data > out.folded
flamegraph.pl out.folded > perf.svg
```

洞察结果，性能瓶颈：



火焰图详细地址：

perf(3).svg-Linux文档类资源-CSDN下载

2.锁性能洞察

由于perf 追踪锁部分耗时，需要重新编译内核很不方便，所以这一部分暂时没有做处理

四、洞察结论

envoy cpu 负担很大，我从perf里得到了下面的结论

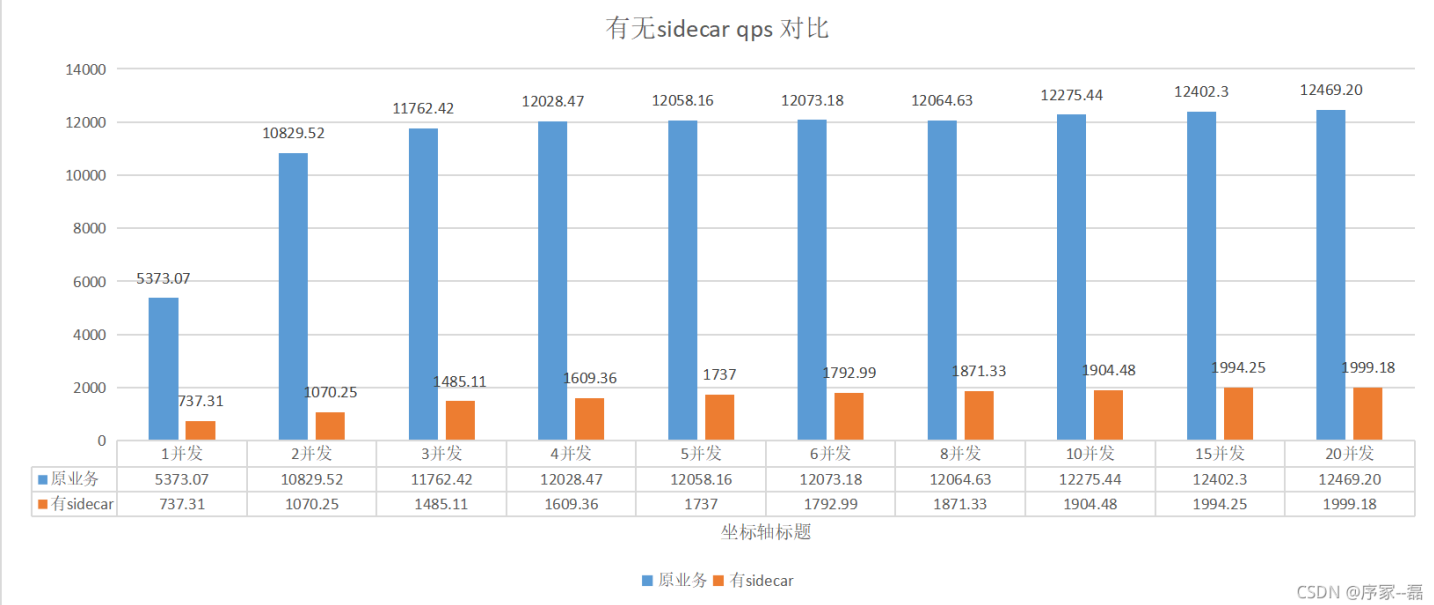
1.envoy 采用的是http_parser库，这个库官方已经不维护了，修改为llhttp库会提升http解析的效率，为cpu减少负

2.istio-proxy 默认的 zipkin + HTTP_JSON在并发很大的时候，会给cpu很大的负担

3.istio-proxy通过wasm 插件给envoy带来了很大的负担，metrics部分也加大了cpu的负担

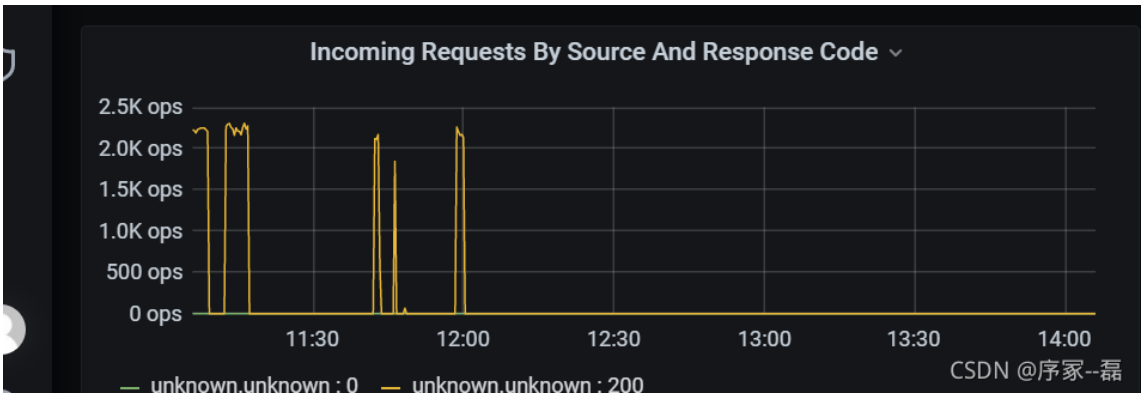
五、排除法论证结论

1.istio默认环境下压测istio-proxy

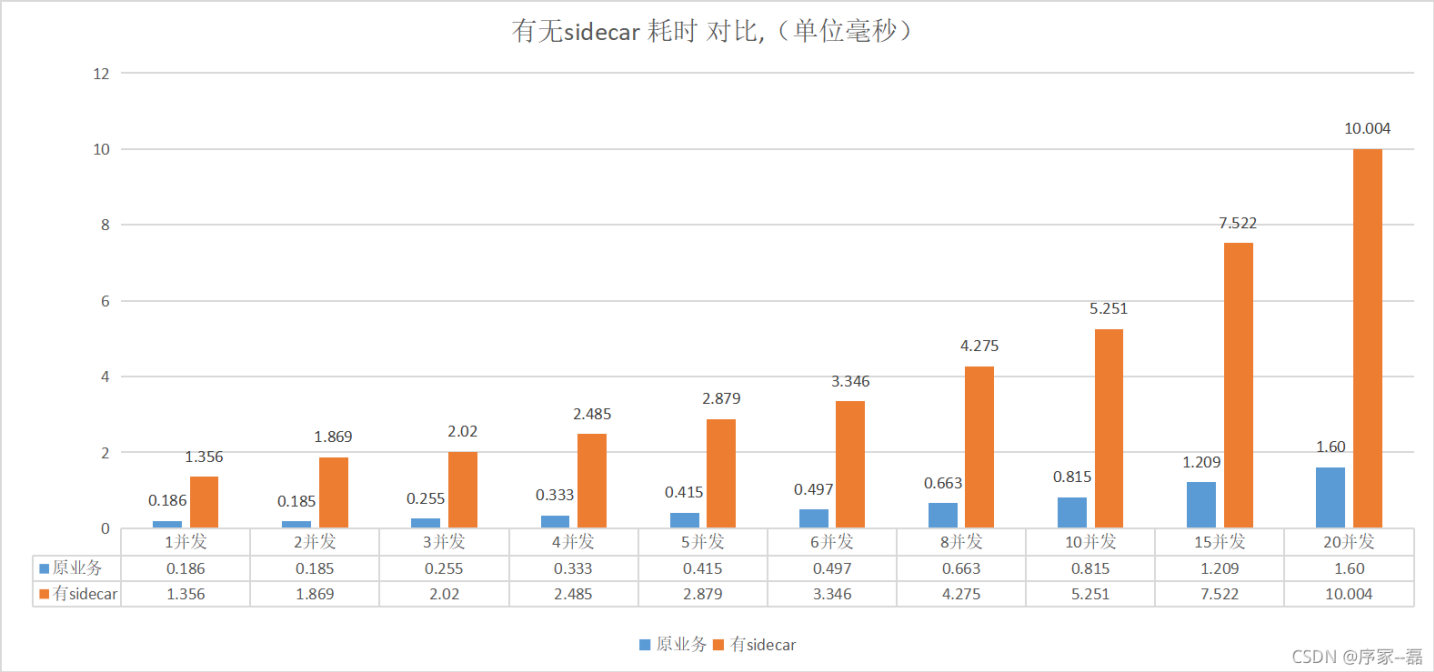


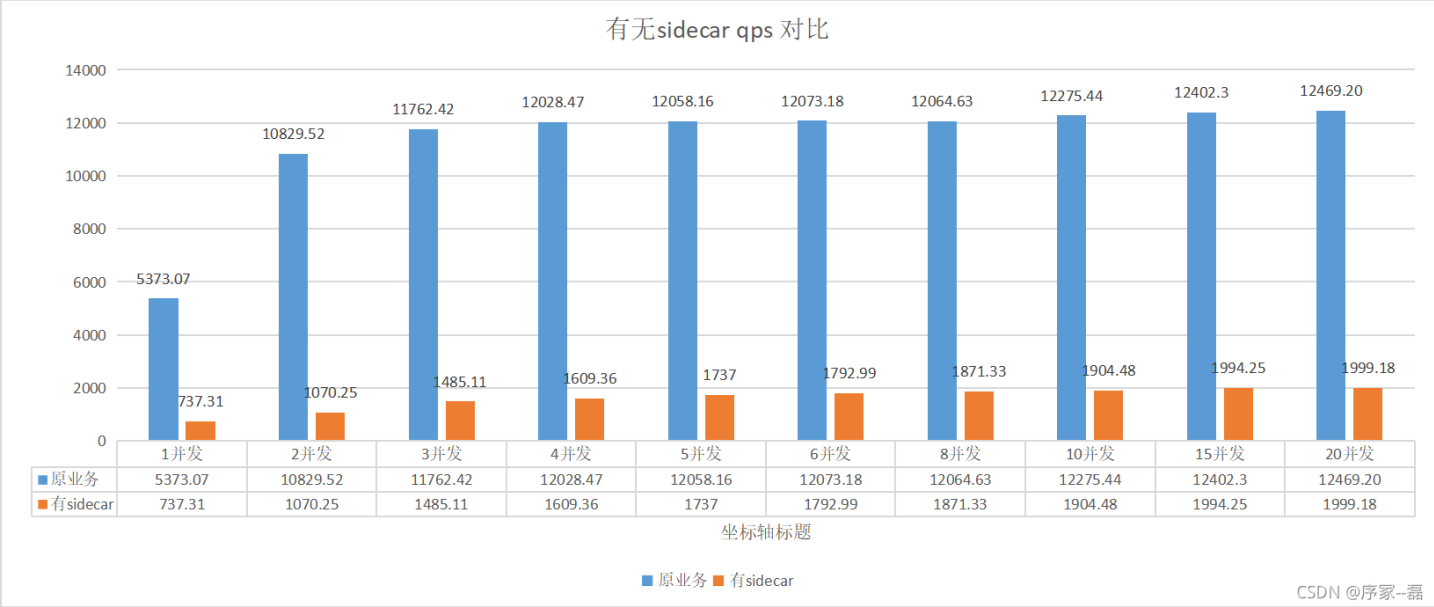
之前istio云原生的默认配置下进行压测

普罗米修斯压测情况：



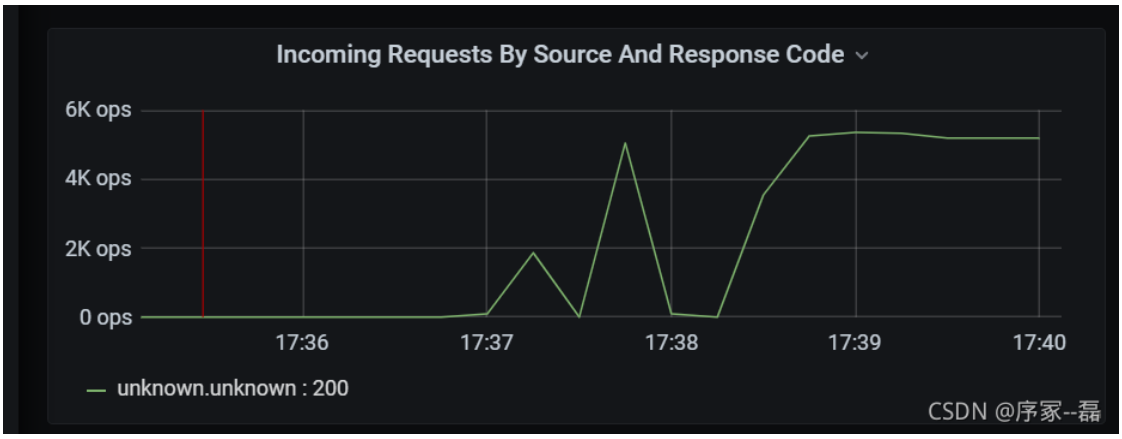
延迟压测结果分析图：





发现性能很差

2.当我们关闭zipkin tracing 再看性能：



性能打到了5500，比之前2000 提升了 250%

3.性能250%意味着给公司带来了什么

之前一个sidecar 容器需要2核，现在只需要1核，如果部署10000个pod，只需要10000核，而不需要20000核，可以给公司降低成本

六、解决方案结论，以及未来展望

我认为良好的解决方式

1.envoy支持多种tracing，zipkin、lightstep、datadog、stackdriver、skywalking、jaeger，我认为jaeger的原生tracing 性能最好，jaeger 的 thrift 协议 是 facebook的二进制协议，性能可以跟protobuf匹敌,网络传输方式上用的udp，也会比其他协议的tcp开销小很多，毕竟tracing 不需要那么完全可靠

2.降低tracing 采样率，不要使用100%采样

3.替换envoy http协议1的解析库，因为我们公司内部大范围使用的是http1,放弃http_parser库，采用llhttp

```
https://github.com/nodejs/http-parser
```

llhttp地址：

```
https://github.com/nodejs/llhttp
```

4.wasm metrics cpu 占用也很高，逐步降低cpu消耗，优化c++代码，如果是同步统计，改为异步，优化性能

5.想办法统计出内核锁的耗时，对加锁代码进行优化，减少临界区

七、性能好了意味着有什么用

尤其是做中间件，是一个公司核心，节约内存核cpu使用是基本素养，如果一个3000 qps的项目需要一个2核的机器，会给公司造成很大的开销

做技术一定要有极客精神，每一行代码，每一次内存拷贝，每一次io都是要讲良心的，必须要对自己写的程序负责，做最优质的程序