1. 限流的实现

接下来，我们配置三层微服务的限流，有两个配置文件：rate_limit_rule.yml 和 recommendation_rate_limit_handler.yml。

在 rate_limit_rule.yml 配置文件中，定义了如下配置：quota rule（mixer 端）、Quota 实例（mixer 端）、QuotaSpec（客户端）、QuotaSpecBinding（客户端）。内容如下：

```
apiVersion: "config.istio.io/v1alpha2"
kind: quota
metadata:
  name: requestcount
spec:
  dimensions:
    source: source.labels["app"] | source.service | "unknown"
    sourceVersion: source.labels["version"] | "unknown"
    destination: destination.labels["app"] | destination.service |
"unknown"
    destinationVersion: destination.labels["version"] | "unknown"
---
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: quota
  namespace: istio-system
spec:
  actions:
  - handler: handler.memquota
    instances:
    - requestcount.quota
---
```

```
apiVersion: config.istio.io/v1alpha2
kind: QuotaSpec
metadata:
  creationTimestamp: null
  name: request-count
  namespace: istio-system
spec:
  rules:
  - quotas:
    - charge: 1
      quota: RequestCount
---
apiVersion: config.istio.io/v1alpha2
kind: QuotaSpecBinding
metadata:
  creationTimestamp: null
  name: request-count
  namespace: istio-system
spec:
  quotaSpecs:
  - name: request-count
    namespace: istio-system
  services:
  - name: customer
    namespace: tutorial
  - name: preference
    namespace: tutorial
  - name: recommendation
    namespace: tutorial
```

可以看到在上述配置中定义了如下内容：

- Quota 中定义了名为 requestcount 的配额实例，实例中定义了 source、sourceversion、destination、destinationversion。
- QuotaSpec 中定义了配额实例的名称为 requestcount，每次请求消费的 quota 实例数量为 1 个。
- QuotaSpecBinding 定义了：将 QuotaSpec 与 tutorial 项目中的三个微服务：customer、preference、recommendation 进行绑定。
- Rule：在 rule 中执行了配额实例使用的限流 handler 为 memquota。

在 recommendation_rate_limit_handler.yml 配置文件中，定义了如下配置：memquota（mixer 端）。内容如下：

```yaml
apiVersion: "config.istio.io/v1alpha2"

kind: memquota

metadata:

  name: handler

spec:

  quotas:

  - name: requestcount.quota.istio-system

    # default rate limit is 5000qps

    maxAmount: 5000

    validDuration: 1s

    # The first matching override is applied.

    # A requestcount instance is checked against override dimensions.

    overrides:

    - dimensions:

        destination: recommendation

        destinationVersion: v2

        source: preference

      maxAmount: 1

      validDuration: 1s
```

在配置文件中定义了 memquota handler：从 preference 到 recommendation v2 的请求，最多每秒一次调用。

应用所有的配置：

```
# oc create -f recommendation_rate_limit_handler.yml
# oc create -f rate_limit_rule.yml
```

对三层微服务发起压力测试，观测结果：

```
# while true; do curl http://istio-ingressgateway-istio-system.apps.example.com/ ; sleep .1; done
customer => preference => recommendation v1 from '58fcd486f6-m42lh': 21760
customer => 503 upstream connect error or disconnect/reset before headers
customer => preference => recommendation v1 from '58fcd486f6-m42lh': 21761
customer => 503 preference => 429 RESOURCE_EXHAUSTED:Quota is exhausted for: RequestCount
```

可以看到，出现了 429 RESOURCE_EXHAUSTED:Quota 的报错，说明限流起到了效果。