

# 跨App共享限流设计与实现

## 背景

此问题最初主要是为了解决对于跨系统共享资源（如DB）的限流配置问题。未来会通过共享应用解决一些基础的配置，如Pod的基准保护（Pod总QPS、内存使用率、CPU使用率）、动态路由的基准规则（如错误率最大阈值、RTT最大阈值）等。

此文主要讲解实现的基本思路及操作。

## 设计与实现

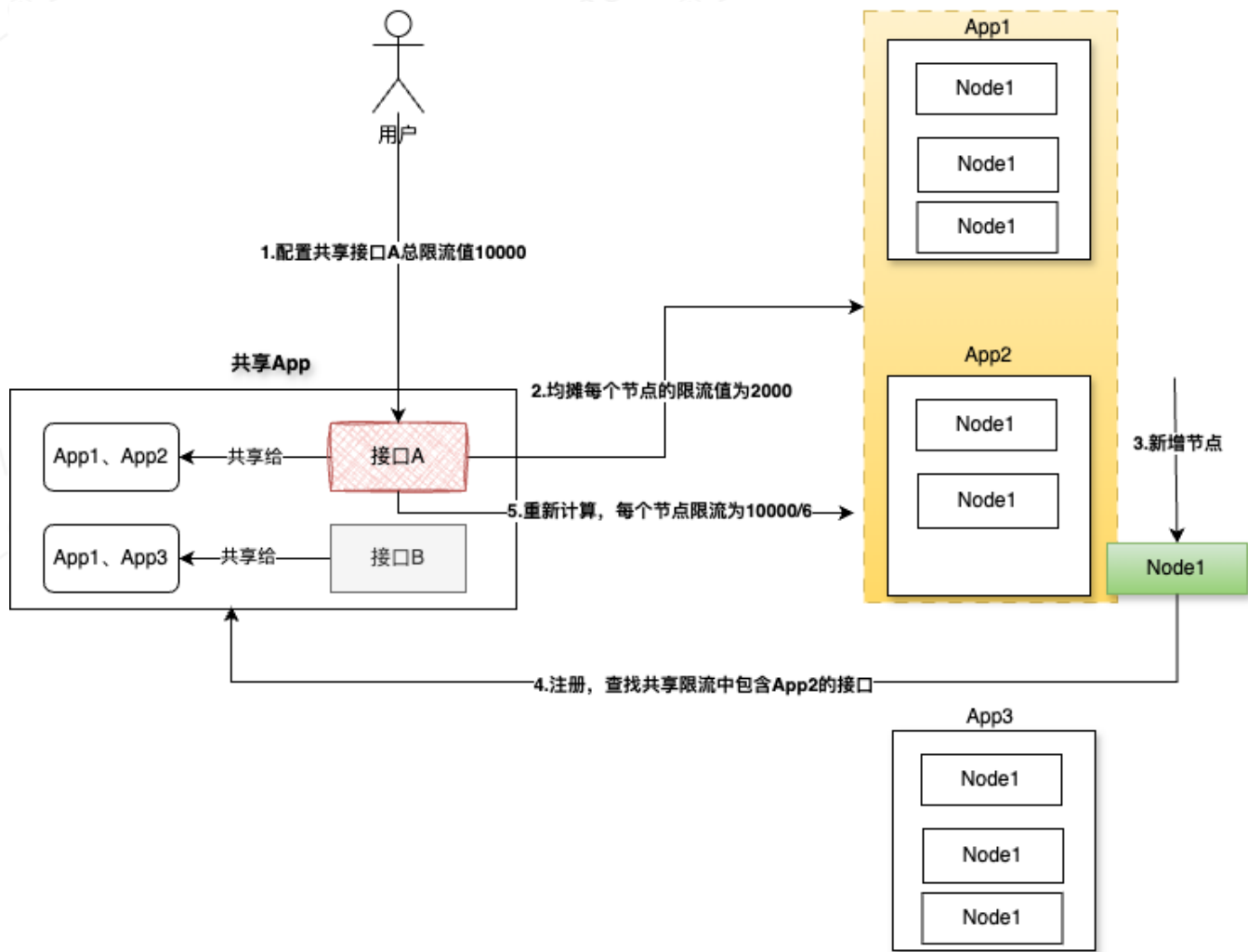


图1 跨App设计与实现思路

Seamiter会为系统建立一个特殊的共享App，在此App下对于限流规则的设置，可以设置共享到具体的关联应用中，如图接口A被共享给了App1、App2。

注意：共享App不能和Seamiter-SDK的应用重名，因此在应用的命名上需要注意规避。

被共享的规则会在Node启动时，除拉取自身规则外还会拉取共享给自己的应用。

对于限流规则，如果选择的全局均摊，则此接口配置的限流值会被所有共享应用下的节点均摊。

举例如下：接口A限流值为10000，被共享给了App1、App2，App1+App2的总节点数为5，则每个节点的限流值为2000。

当共享应用中的节点发生变化时，会触发接口A的均摊值重新计算。

举例如下：某时刻App2新增一个节点，此时会触发共享接口中有App2的且为全局均摊的限流值的重新计算，此处重新计算后单个node的限流值10000/6

# 配置

1、配置共享应用，备注只有SuperAdmin才可以创建共享应用

ID	应用名称	应用类型	分组	Grpc接口	机器地址	在线	备注	操作
35	SHARE_APP_FOR_Credit	共享	fast-escrow-group			离线		实例查看 实时监控 方法列表 编辑 删除

2、设置共享限流

新增流控规则

应用名称

SHARE\_APP\_FOR\_Credit

资源名称

accountService.AccountService.Query

来源应用

多个App使用逗号(,)分割，默认不限制

阈值类型

QPS

单机阈值

200

集群

☒ 是 ☐ 否

集群策略

☒ 节点均摊 ☐ 全局限流

全局阈值

100000

流控模式

☒ 直接 ☐ 关联

流控效果

☒ 快速失败 ☐ 排队等待

Token策略

☒ Direct ☐ Warm Up ☐ 内存自适应

授权应用

fast-escrow-recon-dev,fast-escrow-transaction-ghy

开关

关

# 备注

如果使用共享限流时，没有使用均摊方式，那么相当于把规则以单机的方式下发到了各个业务，这对于一些基准设置依然是有用的