



PolarisMesh开发者例会

2022-01-11

- 生态对接经验分享--GoFrame
- PolarisMesh社区分组介绍
- PolarisMesh推送机制设计讨论
- Q&A

生态对接经验分享--GoFrame

主讲人：智刚老师

材料归档：<https://github.com/polarismesh/polaris/discussions/261>

PolarisMesh社区分组介绍

背景：

PolarisMesh涉及多个语言，多个组件，不同组件以及不同语言存在不同的设计以及知识栈，分组后便于维护人员可以更聚焦更深入去了解相关组件

Polaris-Server

Polaris-Java

Polaris-Go









Polaris-CPP

github.com/polarismesh/polaris
github.com/polarismesh/polaris-controller
github.com/polarismesh/polaris-console

github.com/polarismesh/polaris-java
github.com/polarismesh/grpc-java-polaris
github.com/polarismesh/spring-boot-polaris
github.com/polarismesh/polaris-java-agent

github.com/polarismesh/polaris-go
github.com/polarismesh/grpc-go-polaris
github.com/polarismesh/polaris-sidecar

github.com/polarismesh/polaris-cpp
github.com/polarismesh/grpc-cpp-polaris
github.com/polarismesh/nginx-polaris
github.com/polarismesh/polaris-lua
github.com/polarismesh/polaris-php

polaris-cpp-commiter		1 member	0 teams
polaris-cpp-maintainer		4 members	0 teams
polaris-go-commiter		2 members	0 teams
polaris-go-maintainer		1 member	0 teams
polaris-java-commiter		2 members	0 teams
polaris-java-maintainer		2 members	0 teams
polaris-server-commiter		2 members	0 teams
polaris-server-maintainer		2 members	0 teams

社区会定期置顶发布新手任务，并且推送到开发者的googlegroup，大家可以及时认领更新

googlegroup: <https://groups.google.com/g/dev-polaris>

Label issues and pull requests for new contributors
Now, GitHub will help potential first-time contributors [discover issues](#) labeled with `good first issue`

Combine polaris with surrounding ecological components
#163 opened on 29 Nov 2021 by chuntaojun
Open 22

Document all the polarismesh related projects
#188 opened 28 days ago by andrewshan
Open 2

Polaris code quality optimization
#223 opened 4 days ago by chuntaojun
Open

Filters is:issue is:open

Labels 22 Milestones 1 New issue

22 Open 60 Closed

Author Label Projects Milestones Assignee Sort

support eureka api access `apiserver` `enhancement` `keystone` `service`
#228 opened 2 hours ago by andrewshan v1.5.0

Support dynamic config service `enhancement` `keystone`
#225 opened 4 days ago by lepdou v1.5.0 1

Polaris code quality optimization `code quality` `enhancement`
#223 opened 4 days ago by chuntaojun 8 tasks

Failed to build polaris on macOS `bug` `documentation` `good first issue`
#218 opened 6 days ago by GuangmingLuo 5

Use Concurrentmap in CacheProvider to reduce lock granularity `enhancement` `need discuss`
#205 opened 19 days ago by liu-song 1 5

support auto create namespace `enhancement`
#204 opened 20 days ago by andrewshan

add benchmark document `documentation`
#202 opened 21 days ago by andrewshan v1.5.0 1

Document all the polarismesh related projects `documentation` `enhancement` `help wanted`
#188 opened 28 days ago by andrewshan 12 of 36 tasks 2

PolarisMesh推送机制设计讨论

背景：

PolarisMesh需要支持动态配置能力，配置的发布及变更事件需要推送给各个客户端：<https://github.com/polarismesh/polaris/issues/225>

PolarisMesh服务需要支持推送功能，以支持实时数据变更到客户端的需求

实现方案：

方案1：单向stream通知

接口： `rpc subscribe(Request) returns(stream Response) {}`

具体实现：客户端新建一个stream，发送订阅请求后，server会启动一个协程去处理订阅请求，返回订阅成功。并且当订阅的数据有变更后，通过这个协程往客户端不断的推送变更应答。

优点：实现简单，取消订阅无需额外的请求，只需要断开stream即可。

缺点：每种类型订阅需要一个stream，当客户端数量比较多且单个客户端监听的数据类型比较多时，server端需要额外维护很多协程，加重CPU负载

方案2：双向stream通知

接口： `rpc subscribe(stream Request) returns(stream Response) {}`

具体实现：客户端新建一个stream，通过这个stream发送不同类型的订阅请求，server端启动2个协程，协程A负责从stream收消息，然后推送到chan1，协程B负责select chan1读取请求，然后增加订阅监听，并返回应答。当数据出现变更时候，变更数据推送给chan2，协程B负责select chan2读取变更数据，并推送给客户端。

优点：多个类型订阅可以复用一個stream，server端协程数可控（协程数=客户端数*2），CPU负载可控

缺点：编码实现相对负载，用户需要发送额外的取消订阅请求才可以进行取消订阅

Q&A