

SWOOLE

Yang Chao
2019.11.07

目录

- 原生php中的swoole
- swoole结合yii2
- yii2-swoole扩展实战
- 问题和总结
- tobe?

原生swoole

Swoole?

-- Swoole是一个为PHP用C和C++编写的基于事件的高性能异步&协程并行网络通信引擎 <https://github.com/swoole/swoole-src/>

协程?

-- Swoole4或更高版本拥有高可用性的内置协程。可以将协程理解为超轻量级的线程，可以非常容易地在一个进程中创建成千上万个协程。

与go中协程的异同?

-- go语言内置协程 实际上是多线程 所以协程之间可以共享数据

-- swoole是php语言扩展 实际上是多进程 协程之间互相独立 需要使用额外的方法来进行数据共享 如pipe/文件等

原生swoole - demos

示例代码位置 `project`仓库下

`console/controllers/swoole-demo`里面 有可运行的代码和本人开发过程中的注释

- WebServer – simple http Server
- coShell – 展示一下协程
- coChan – 协程之间pipe通信
- AsyncMysqlClient – 异步mysqlClient
- mysqlPool – mysql池

Swoole和yii2的结合

-- 有两种方式 参照<https://www.jianshu.com/p/9c2788ccf3c0>

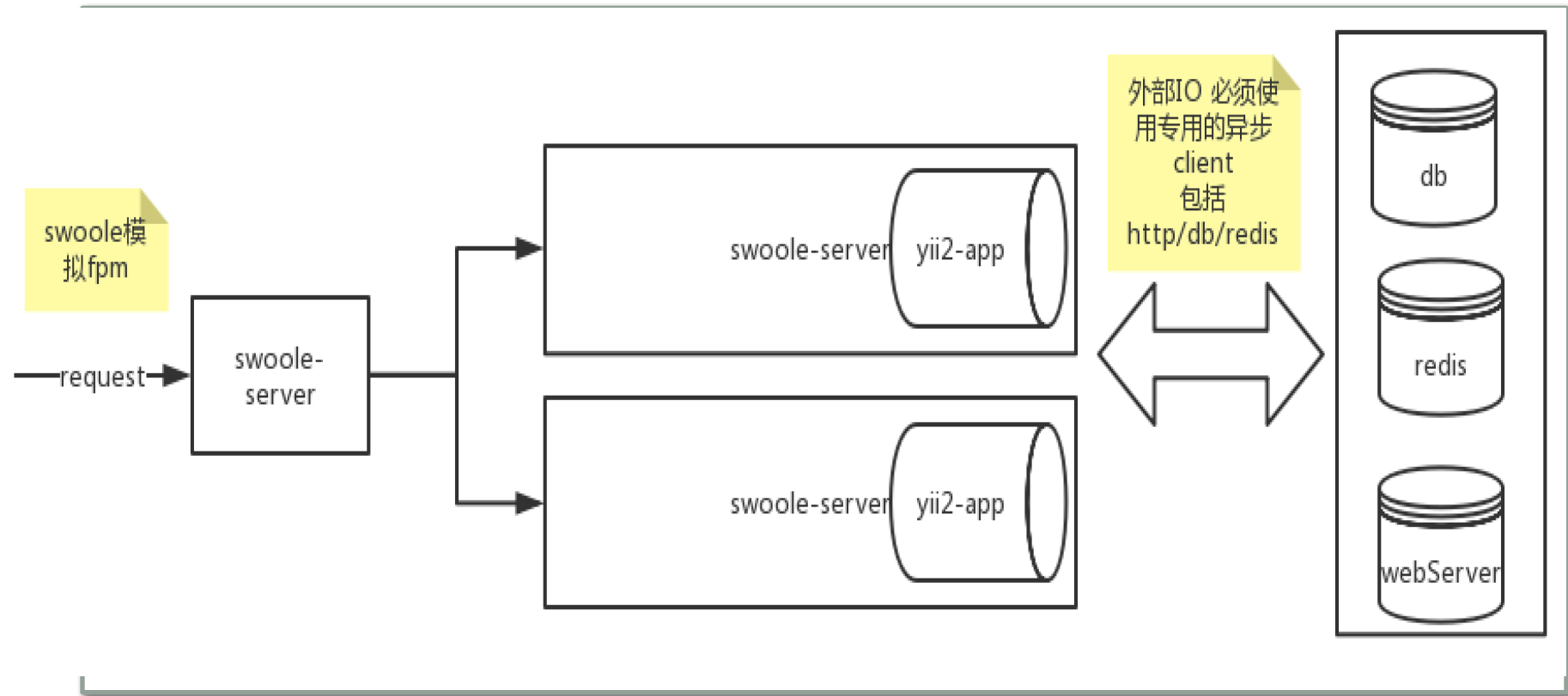
1. yii2中调起swoole

1. 这种就相当于yii2是个壳 框架还是走的yii2 然后每次request调起swoole 没太多价值

2. Swoole-server中调起yii2 app

1. 这种是推荐的方式 类似于swoole写了一个fpm的功能 然后运行的时候里面处理request的是yii2框架的application
2. 要做到这点需要改写相应的组件 request/response/errorHandler等

Swoole和yii2的结合



yii2-swoole扩展

















预先思考问题

1 swoole是console下运行 而web服务在其他文件夹下 比如frontend 那么日志输出在哪里？如何实现？

2 swoole是常开的进程 如何保证服务进程常在？

3 思考一个问题 假如yii2中的常用写法 无法或者很难在swoole的协程异步客户端中支持 那么是否还有使用swoole的意义？

yii2-swoole扩展-目录

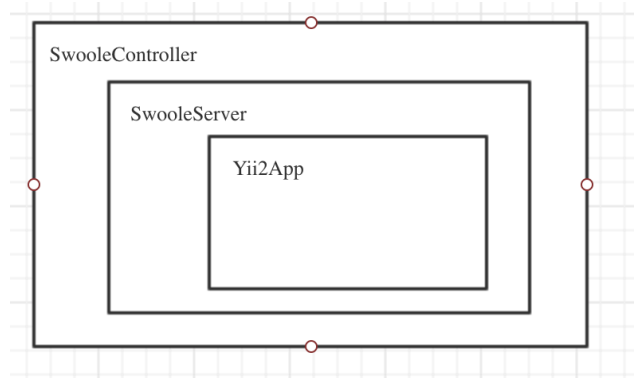
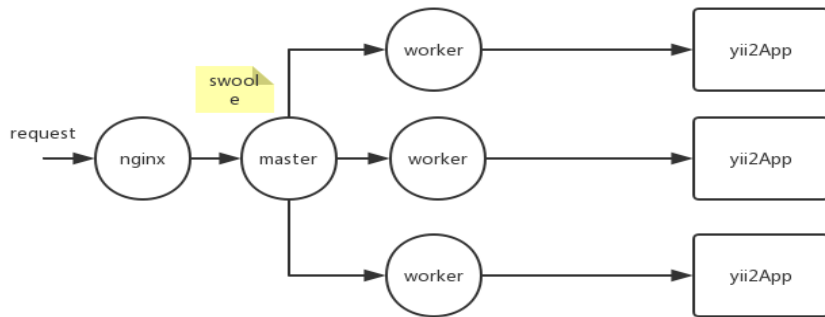
- ▼  src
 - ▼  console
 -  SwooleController.php
 - ▼  debug
 - ▶  panels
 -  Module.php
 - ▼  swoole
 -  SwooleServer.php
 -  Util.php
 - ▼  web
 -  Dispatcher.php
 -  ErrorHandler.php
 -  Logger.php
 -  Request.php
 -  Response.php
 -  Session.php

yii2-swoole扩展-目录

目录名	文件:作用
console	SwooleController: 运行swooleServer 并且在里面 加载yii2config配置和组件并且启动yiiApp
debug	就是对yii2-debug的扩展 改了一下view
swoole	Request/Response: 主要是一些参数解析的传递 如cookie/request/response的参数 在yiiApp和swooleServer之间传递 Dispatcher: 就是加了一个beginTime 来方便打印调试 否则的话 时间就永远是启动console的时间 Session: 基于文件来共享session ErrorHandler: 错误处理, 有error就VarDump处理
web	SwooleServer.php: 封装swoole_server 绑定处理方法 将request的参数放到php超全局变量(如\$SESSION)里 Util.php: 提供一些helper工具

yii2-swoole-运行结合源码

```
server {  
    charset utf-8;  
    client_max_body_size 128M;  
  
    listen 80; ## listen for ipv4  
  
    server_name yc-swoole-baike.dev.ethercap.com;  
    include /etc/nginx/default.d/*.conf;  
  
    set $web /home/yangchao/project/wiki/web;  
    root $web;  
    location ~* \.(ico|gif|bmp|jpg|jpeg|png|swf|js|css|mp3)$ {  
        root $web;  
    }  
    location / {  
        # 反向代理到 9998 端口  
        proxy_pass http://127.0.0.1:9997;  
        add_header Access-Control-Allow-Origin *;  
        proxy_http_version 1.1;  
        proxy_set_header Connection "keep-alive";  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```



总结

- 优点
 - 可以在保持原yii2业务不变的基础上 在有性能瓶颈的地方 使用 **swoole** 中的协程和异步客户端 来有效提高并发和响应速度
- 问题
 - 原有的许多组件需要用异步客户端改写
 - 原有的许多功能和写法可能不支持或者需要改写
 - **reload** 热启动需要自己实现

tobe?

- golang – 为协程而生的语言
 - <https://golang.org/>
- swift – 基于swoole的异步框架
 - <https://github.com/swift-cloud/swift>
 - <https://segmentfault.com/blog/swift>

Q & A

- Thanks & More?