SWOOLE入门

马雄飞

Swoole的一些概念

PHP部分局限性：

1.PHP是单进程的，没法在一个程序块中使用多进程来处理一个复杂的逻辑

2.PHP不支持异步,在处理高并发高负载的请求时会力不从心

3.PHP不适合做TCP，Socket之类的长连接。

类似这种Unix系统编程、网络通信编程、异步io，大部分PHPer不懂。PHP界也确实没有这样的东西。Swoole开源项目就是为了弥补PHP在这些方面的缺陷诞生的。与Swoole实际上是一个网络通信和异步io的引擎，一个基础库，能让PHP很好的支持异步、异步、简单的使用TCP/UDP/Socket服务.

PHPer可以基于Swoole去实现过去PHP无法实现的功能。swoole为PHPer打开了通往另一个世界的大门

Swoole的安装

1.pecl安装：

```
pecl install swoole
```

2.源码安装：

```
git clone https://github.com/swoole/swoole-src.git
cd swoole-src
phpize
./configure
make && make install
```

Swoole的结构

1.Master 进程：Swoole的主进程，处理Swoole核心的事件驱动，包含若干Reactor子线程，每个Reactor子线程运行epoll函数的实例，进行客户端事件监听，如客户端连接

2.Manage进程：管理进程，用于创建管理下一层的Worker进程和Task Woker进程

3.

Worker进程：

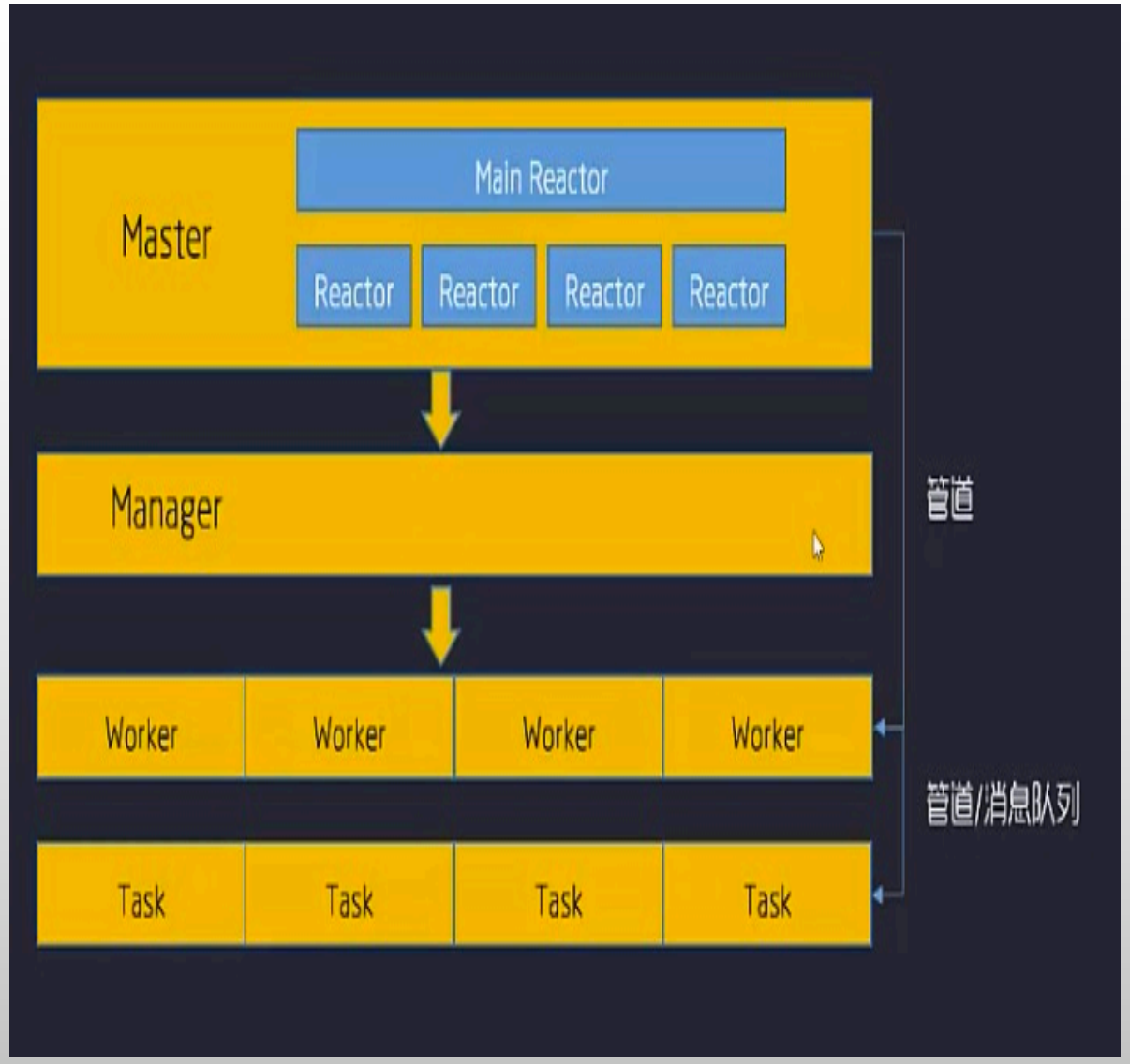
主逻辑进程，处理来自客户端的请求

Task进程：

异步工作进程，用于处理耗时较长的任务

https://wiki.swoole.com/wiki/page/163.html

Swoole的运行流程



请求到达 Main Reactor
|
|
Main Reactor根据Reactor的情况，将请求注册给对应的Reactor
(每个Reactor都有epoll。用来监听客户端的变化)
|
|
客户端有变化时，交给worker来处理
|
|
worker处理完毕，通过进程间通信(比如管道、共享内存、消息队列)发给对应的reactor。
|
|
reactor将响应结果发给相应的连接
|
|
请求处理完成

Swoole的重要特性

1.纯C语言编写，是一个扩展

2.swoole_server: 强大的TCP/UDP Server框架，多线程，EventLoop，事件驱动，异步，Worker进程组，Task异步任务，毫秒定时器，SSL/TLS隧道加密。协议上还支持HTTP和websocket

3.swoole_client: TCP/UDP客户端，支持同步并发调用，也支持异步事件驱动

4.swoole_event: EventLoop API，让用户可以直接操作底层的事件循环，将socket，stream，管道等Linux文件加入到事件循环中

5.swoole_async: 异步IO接口，提供了异步文件系统IO，异步DNS查询，异步MySQL等API。支持异步毫秒定时器和异步文件操作接口

6.swoole_process: 进程管理模块，可以方便的创建子进程，进程间通信，进程管理。

Swoole的实例

1. tcp Server 和 tcp Client

2. udp Server 和 udp Server

3. webscoket

4. 异步mySql
   node.js之类的语言可以实现异步的数据库查询功能，执行SQL语句之后不必等待数据库返回结果。继续去执行其他的代码，当数据库返回结果是再对数据进行处理，如渲染页面，并将HTML页面发送给客户端。这样应用程序完全不需要阻塞等待。这种方式运行效率非常高。

5. 异步导入用户demo

Swoole的一些资料

https://wiki.swoole.com

https://www.zybuluo.com/phper/note/494239

http://www.bewithyou.me/archive/detail/65