ZNet 的 C4 不同于传统的 CS 和 web 架构：C4 可以做 CS 和 WEB 架构，但 C4 设计思路完全不同。

先聊聊 C4 的设计思路

C4 以服务作为后台项目组成，开发项目就是调度服务，假设我们要开发一个完整的聊天室:那么我们会需要，用户系统，日志系统，房间，三种服务，这三种服务都是独立存在的服务器模型，不可以直接让前端直接使用，因此我们就需要一个调度服务，在 C4 系统中调度服务器就是一种 VM，这个 VM 等同于一个同时调度用户，日志，房间系统的服务器，这台 VM 服务器，可以借助 C4 的专用框架在云端多开，形成 SaaS 云。

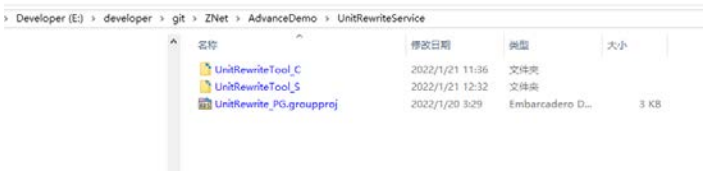
C4 的服务器模型很多，并且是按最苛刻环境和需求设计，例如用户系统可以一个账号有多个可登录 ID，因为绑定微型，手机号都是用户登录的必须功能，并且用户系统还内置了 IM 系统，因为实时消息开发起来非常繁琐和漫长，同时，用户系统也可以支持自定义数据，例如，身高年龄体重，关键字等等，除此之外，用户系统还可以支持千万规模在线的用户后台。我们做 Online 类项目，如果用户系统是现成的服务器模组，何必自己跑去写一个用户系统，这是非常大的时间消耗。

C4 在设计时有个频繁写某种服务的启发，每次做一个项目，总是用一种机制在写后台，长时间下来，这些后台并不可以积累，而且难以维护。试想，如果给出了一种规范，后台直接脱离具体的 CS,WEB 这些规则束缚，实现无穷堆大，轻易复用，分布式。C4 是真正做到了这一点的项目：思路就是把具体服务器需求转换一种可以共享使用的模型，以 SaaS 节点方式存在于 C4 系统中。用 C4 的服务器模型方式做后台是高度集中的，各个项目会很好统一，并且，不会重写，C4 做后台越往后走，越强大。

遇到缺乏技术支持的项目，例如网盘，这时候需要 C4 提供模型和范例，网盘最难解决的问题是虚拟磁盘的数据结构，把这一步搞定，其它地方都是使用 C4 的现成资源。

回到主题，聊聊应用模型吧

C4 的应用模型 1，直接照搬 CS 思路，用 C4 写个服务器，再写个客户端，这种方式可参考，UnitRewriteService，这就是我们常说的：工具云，API 云，轻量通讯。



C4 的应用模型 2，自己写个 VM 负责调度 C4 里面的一堆服务，这些服务，VM 都是分布式的，可以存在于数十台云服务器中，这样干的优点就可以建造出弹性十足的后台，这就是我们常说的，SaaS，分布式后台，HPC 服务器。

C4 的应用模型 3，C4 是按服务器模型组成的项目，它既可以分布式 SaaS 也可以 All In 一个进程，就像一个小型服务器，这种模型的优点是易于分发部署，功能强大又小巧，缺点则是不太容易组网。

By.qq600585

2022-2-4