

# go hive skynet\_MMORPG游戏服务器技术选型参考-Go语言中文社区

原创

[weixin\\_39908948](#) 于 2020-12-22 17:37:16 发布 238 收藏 1

文章标签: [go hive skynet](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/weixin\\_39908948/article/details/111858976](https://blog.csdn.net/weixin_39908948/article/details/111858976)

游戏服务器一般追求稳定和效率, 所以偏向于保守, 使用的技术手段也是以已经过验证、开发人员最熟悉、能 HOLD 为主要前提。

## 1、典型按场景分服设计

开发语言: C++

数据库: mysql

架构:

多个网关: 维持与玩家间的 SOCKET 连接, 可处理广播、断线重连等逻辑。

一个或多个账号登陆验证服务器: 处理登陆、排队等逻辑。

多个场景服务器: 处理在本地图上能解决的逻辑, 如: 打怪、玩家间战斗、接任务、完成任务等各种不需要跨地图的逻辑。一个场景服内可放置多张地图, 一般是单线程处理逻辑, 也有多线程(如: 《天龙八部》《武侠世界》是一个线程处理一个地图组, 一个场景服可以有多个线程)。

一个全局服务器: 处理不能在一张地图上处理的逻辑, 如跨地图聊天、跨地图交易。

可选:

一个专门与数据库打交道的 DB 代理服务器

日志服务器:

跨服逻辑处理服务器: 支持跨服 PK

共享内存服务器: 将游戏内实体数据放置在另一个进程内, 场景服务器出现宕机时能及时存储数据, 并恢复。

示例如下:

## 2、云风写的开源架构《skynet》

使用开发语言: C 语言、LUA

使用数据库: 内置 redis、mysql 访问接口, lua 的 table 能方便地转化为字符串进行存储

结构: 单进程、多线程

执行结构: skynet 主执行程序, 支持库动态链接库、服务动态链接库、LUA 脚本文件或已编译的 lua 的脚本文件

特点：ACTOR模式、游戏逻辑以多个服务存在，服务间以消息进行通讯(可以是同步，也可以是异步)，LUA服务对每个请求生成一个协程进行处理

个人感受：skynet主要以lua作为逻辑的实现，C与LUA的结合特别深。如果对LUA本身了解程序不够，不容易上手。进行同步消息处理时，没有超时机制。

### 3、开源服务器引擎《kbengine》

《kbengine》在结构上和《bigworld》完全一致，没有动态负载功能。

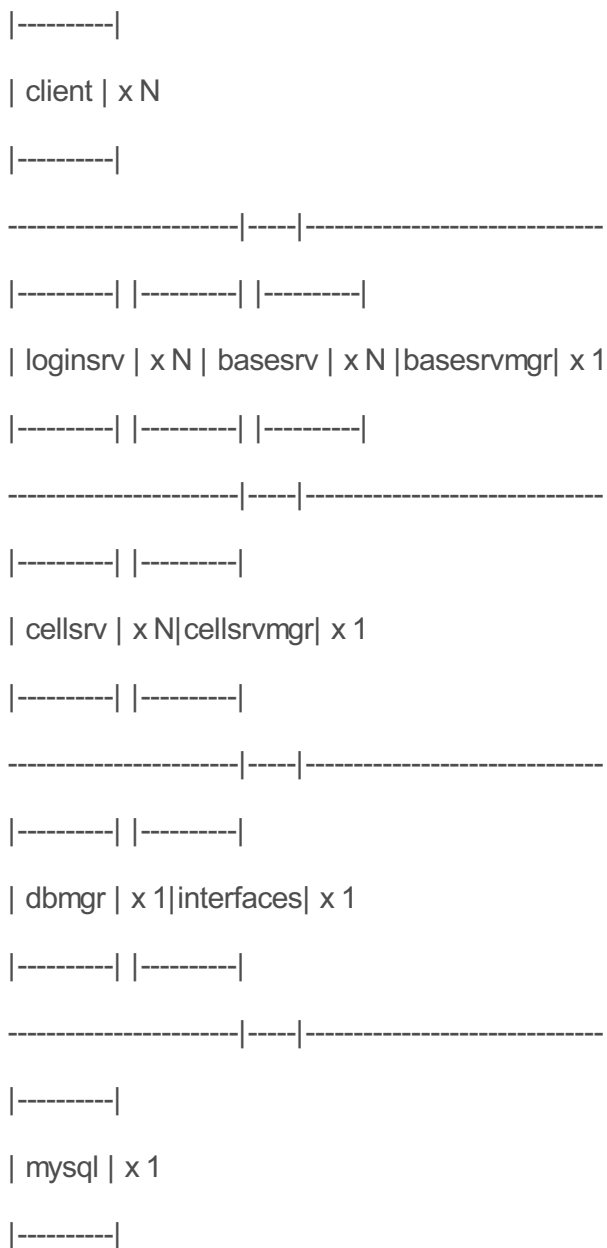
开发语言：c++、python

数据库：mysql

特点：远程调用机制rpc使用开发过程中的消息传递特别方便。baseserver网关与部分功能合为一个服务器的设计极大减小了消息在服务器间的传输，方便了玩家自己的逻辑处理。

相似架构：《暗黑战神》服务器使用的是相似的架构，只是脚本语言选择的是lua

结构：



服务端组件描述

· loginapp:

登录验证、注册、接入口。

可在多台机器部署多个loginapp进程来负载。

· dbmgr:

高性能多线程的数据存取。

默认使用Mysql作为数据库。

· baseappmgr:

协调所有baseapp的工作，包括baseapp负载均衡处理等。

· baseapp:

客户端与服务端的交互只能通过loginapp分配的baseapp来完成。

定时写entity的数据到数据库、baseapp数据相互备份、灾难恢复。

可在多台机器部署多个baseapp进程来均衡负载。

脚本层通常会选择在baseapp上实现如：社交系统、广播聊天、排行、游戏大厅、等等逻辑系统。

· cellappmgr:

负责协调所有cellapp的工作，包括负载均衡处理等。

· cellapp:

处理游戏与空间和位置有关的逻辑，如：AOI、Navigate、AI、战斗等等。

可在多台机器部署多个cellapp进程来动态均衡负载。

· client:

客户端我们将提供基础框架，这个框架不包括渲染部分和输入输出部分的具体实现，

我们将提供一个lib文件和一套API接口，开发者可以选择使用自己比较适合的图形渲染引擎与输入输出控制部分。

Unity3D, HTML5, Cocos2d等技术我们提供了相关插件，能够快速的服务端对接。

· machine:

抽象出一个服务端硬件节点(一台硬件服务器只能存在一个这样的进程)。主要用途是接收远程指令处理本机上的组件启动与关闭，

提供本机上运行组件的接入口以及收集当前机器上的一些信息，

如：CPU、内存等。这些信息会提供给一些对此比较感兴趣的组件。

支持快速接入第三方计费、第三方账号、第三方数据，快速与运营系统耦合。

· guiconsole:

这是一个可视化的图形界面控制台工具，可以实时的观察服务端运行状态，实时观测不同Space中Entity的动态，

并支持动态调试服务端Python逻辑层以及查看各个组件的日志，启动服务端与关闭等。

· logger:

收集和备份各个组件的运行日志。

#### 4、开源架构《firefly》

《firefly》是九秒社区开发一个开源的框架，代码量比较少。

开发语言：python

使用数据库：mysql

官方介绍：

Firefly是免费、开源、稳定、快速扩展、能“热更新”的分布式游戏服务器端框架，采用Python编写，基于Twisted框架开发。它包括了开发框架和数据库缓存服务等各种游戏服务器基础服务，节省大量游戏开发的工作时间，真正做到让使用者把精力放在游戏玩法逻辑上。用它可以搭建自定义的分布式架构，只需要修改相应的配置文件即可。

优势特性

采用单线程多进程架构，支持自定义的分布式架构；

方便的服务器扩展机制，可快速扩展服务器类型和数量；

与客户端采用TCP长连接，无需考虑粘包等问题；

封装数据缓存服务；

可实现实时热更新数据以及游戏逻辑，客户端玩家无感觉；

有几十个基础游戏玩法系统模块提供组装使用(v1.3.0提供)；

#### 5、开源服务器构架pomelo

开发语言： javascript

Pomelo 是基于 Node.js 的高性能、分布式游戏服务器框架。它包括基础的开发框架和相关的扩展组件(库和工具包)，可以帮助你省去游戏开发枯燥中的重复劳动和底层逻辑的开发。Pomelo 不但适用于游戏服务器开发，也可用于开发高实时 Web

应用，它的分布式架构可以使 Pomelo 比普通的实时 Web 框架扩展性更好。

快速、易上手的游戏开发模型和api

高可伸缩的多进程架构，支持MMO的场景分区和其它各类分区策略

方便的服务器扩展机制，可快速扩展服务器类型和数量

方便的请求、响应、广播、服务器通讯机制，无需任何配置

注重性能，在性能、可伸缩性上做了大量的测试、优化

提供了较多扩展组件，包括游戏开发常用的库和工具包

提供了完整的MMO demo代码(客户端html5)，可以作为很好的开发参考

基于socket.io开发，支持socket.io支持的多种语言客户端

## 6、页游erlang构架

erlang基于消息处理的Actor模式，与传统的C++、JAVA、C#等有着非常大的不同。

erlang语言天生是为分步式服务器开发，其中rpc、集群、热更新、分步式数据库天然集成。

开发语言：erlang

数据库：mysql、mnesia

使用erlang的游戏：页游《画皮》、页游《英雄远征》、页游《远古封神》。。。。。。

## 7、基于go语言的架构

GO天然支持的用户进程模型，吸收了很多语言的特点。

使用ERLANG开发游戏服务器的一批人使用GO来进行游戏服务器的开发。

在看了一些使用GO语言写的MMORPG以后，个有感觉是把C++的思路用GO语言翻译了一篇。