新一代棋牌架构

——彭志浩

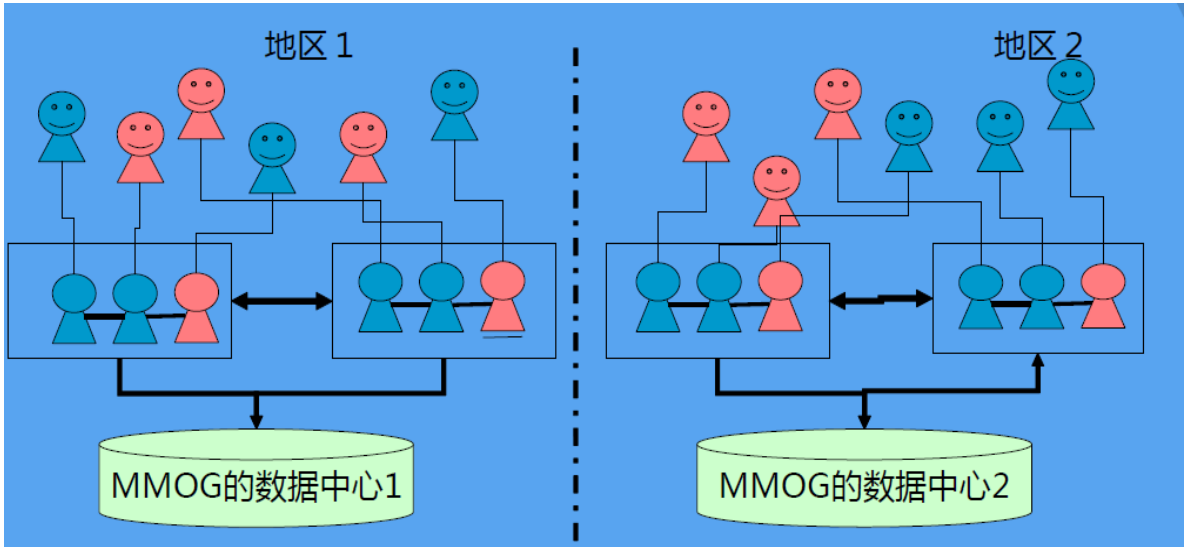

选择游戏
活动中心
充值中心
游戏论坛
点击完善资料
[切换]
金币:700
任务
战队
系统公告
博雅DVI棋牌
三人斗地主
血战到底
二人麻将
川味斗地主
贰柒拾
马 股
血流成河
私人房


zahir
149,343
贰柒拾
在线2308
马 股
最新游戏
川味斗地主
最新游戏
开赛啦！
比赛场
VIP
商城
活动
排行榜
兑奖
牌友
团队赛


zahir
149,343
四川麻将
最新游戏
自建场
商城
活动
排行榜
兑奖
牌友
团队赛

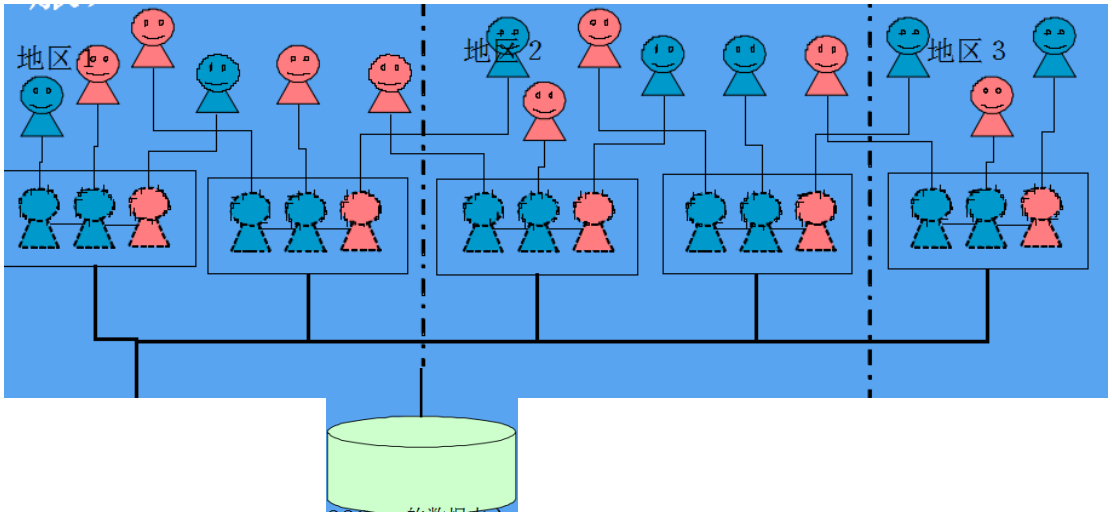
我们在做一个什么样的游戏服务？

高交互的用户/系统模型 (MMOG分区分服)



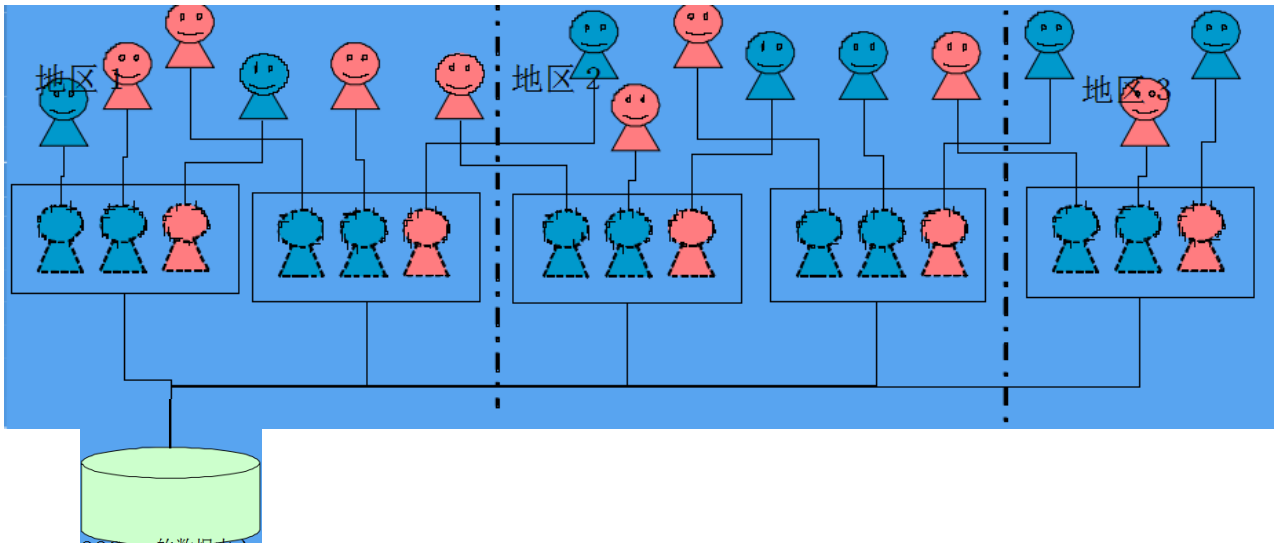
•MMOG（QQ幻想，CF，DNF）的主要用户行为是多人的，MMOG的登陆行为固定，自己选择的。
•MMOG的服务器是地域性的，服务器间是有强联系的，用户的数据是地域化的。
•角色和MMOG服务器间的交互是高烈度的，一个用户的数据发生改变，要广播给很多人知道，单台服务器承载的用户数量是较少的。
•服务器是按照地域，线承载MMOG用户。每个服务器组都构成一个封闭的集群。

中等交互的用户/系统模型
（棋牌游戏全区全服）



✓棋牌游戏的主要用户行为是房间的，用户的的登陆行为是有选择的，同一接入层服务器上的用户之间交互容易。
✓用户数据发生改变，要广播给同一张房间（桌子）的用户。
✓棋牌游戏的接入层服务器是跨越地域的，服务器间是无联系的，但用户的数据是中心化的。
✓棋牌游戏和服务器间的交互是中烈度的，单台服务器承载的用户数量是比较多的。

低交互的WEBAPP
（养成游戏的用户/系统模型）



- 宠物（这儿单指养成部分，或者QQ农场）的主要游戏行为是单人的，宠物的登陆行为是任意的，无选择的。
- 用户的数据改变后无需通知其他用户，某种程度和单机游戏很像。
- 宠物的服务器是跨越地域的，服务器间是无联系的，宠物的数据是中心化的。
- 宠物服务器间的交互是低烈度的，单台服务器承载的用户数量是巨大的。

交互模式的比较

游戏和其他互联网其他业务最大的区别
就是数据变化要（主动）反向通知很多玩家。
交互模式只是业务的一种选择，
其实主要游戏设计希望同屏游戏人有多少。

全区全服从优势在于其可以让全世界的用户一起游戏，全区全服模式的不足在于设计复杂，就近接入性能不好，需要宽带支持。

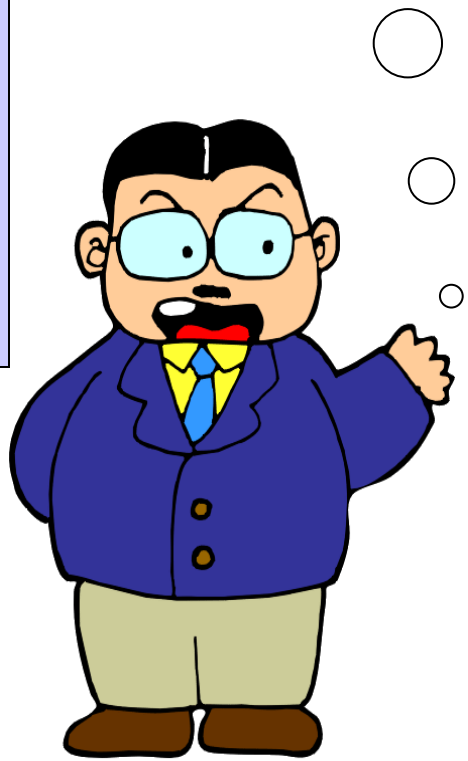
分区分服的优势在于用户就近接入感觉好，实际游戏交互的用户无须那么多。缺点在于用户换一个服务器（组）就要重新练习一个账号，或者换服。

- 思考题：有没有业务综合这些交互模式？未来的游戏会用什么交互模式？交互模式越复杂越好玩吗？

游戏服务的特点

1. 需求变化快，持续运营与改进，永远的beta
2. 系统需要高度的可扩展性
3. 逻辑功能复杂，关联性强
4. 用户增长快数量大，访问量集中
5. 服务器多，物理故障经常出现（使用大量廉价设备）
6. 各种开服需求，迁移需求，合服需求

常见问题
用户量增大，系统运行缓慢需求变化快，系统无法扩展bug增多系统崩溃求变化快，系统无法扩展bug增多系统崩

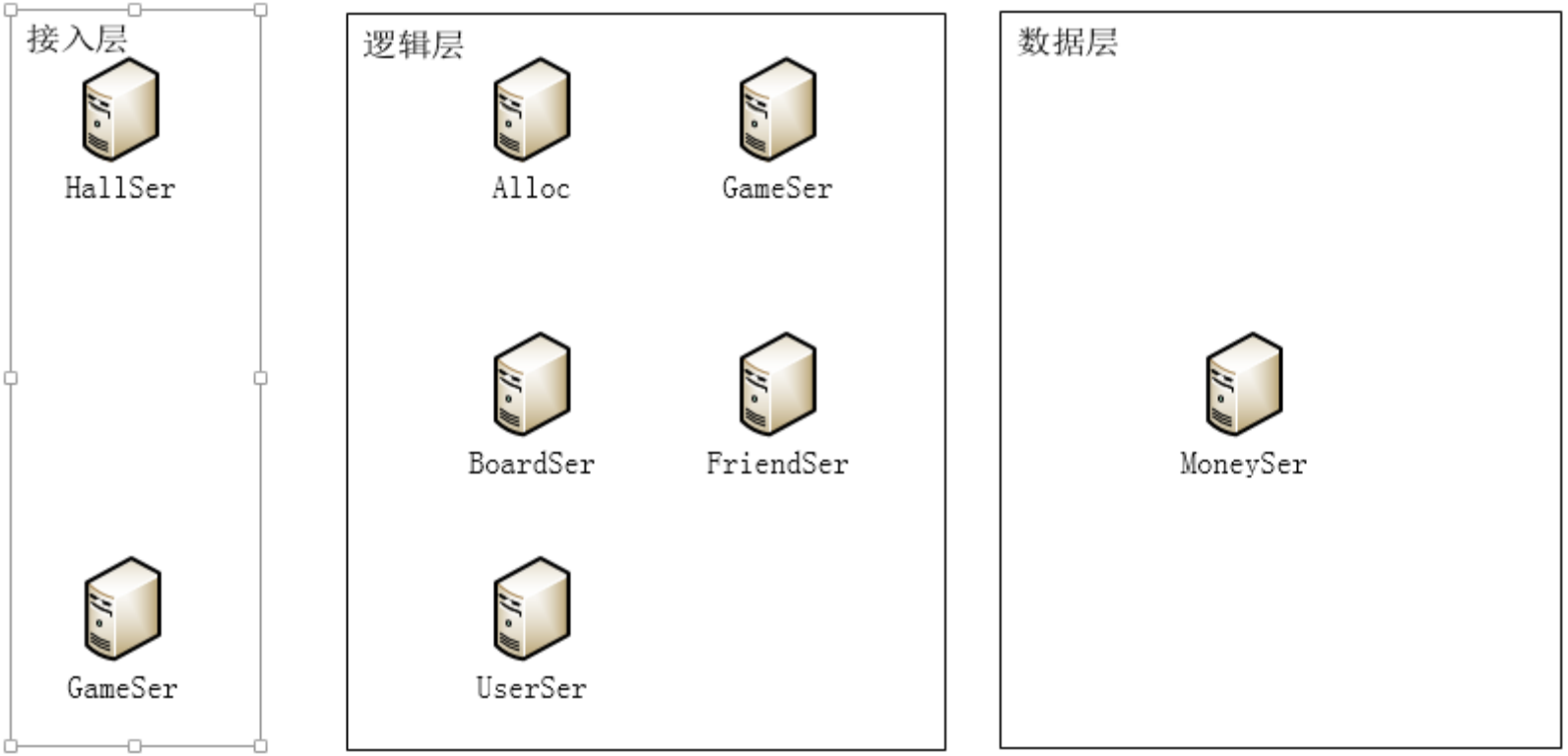


架构从哪里来？

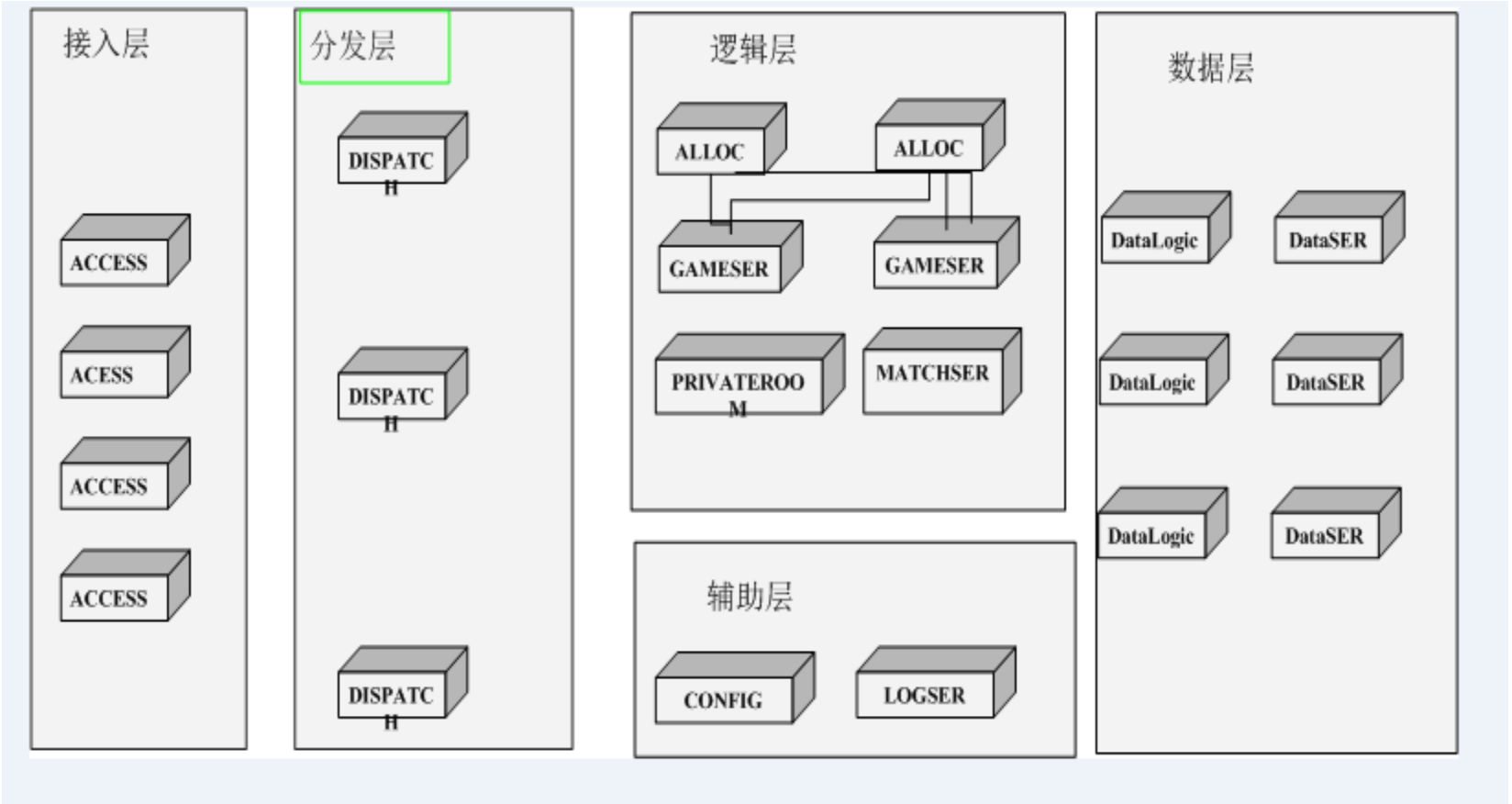
- 策划 提供高可扩展性的系统
- 运营 提供高可运维性的系统
- 玩家 提供高可用性的服务
- 开发 提供高可维护性的代码

架构本天成 妙手偶得之

上一代棋牌架构


接入层
HallSer
GameSer
逻辑层
Alloc
GameSer
BoardSer
FriendSer
UserSer
数据层
MoneySer

新一代棋牌架构


接入层
ACCESS
ACESS
ACCESS
ACCESS
分发层
DISPATCH
DISPATCH
DISPATCH
逻辑层
ALLOC
ALLOC
GAMESER
GAMESER
PRIVATEROOM
MATCHSER
辅助层
CONFIG
LOGSER
数据层
DataLogic
DataSER
DataLogic
DataSER
DataLogic
DataSER

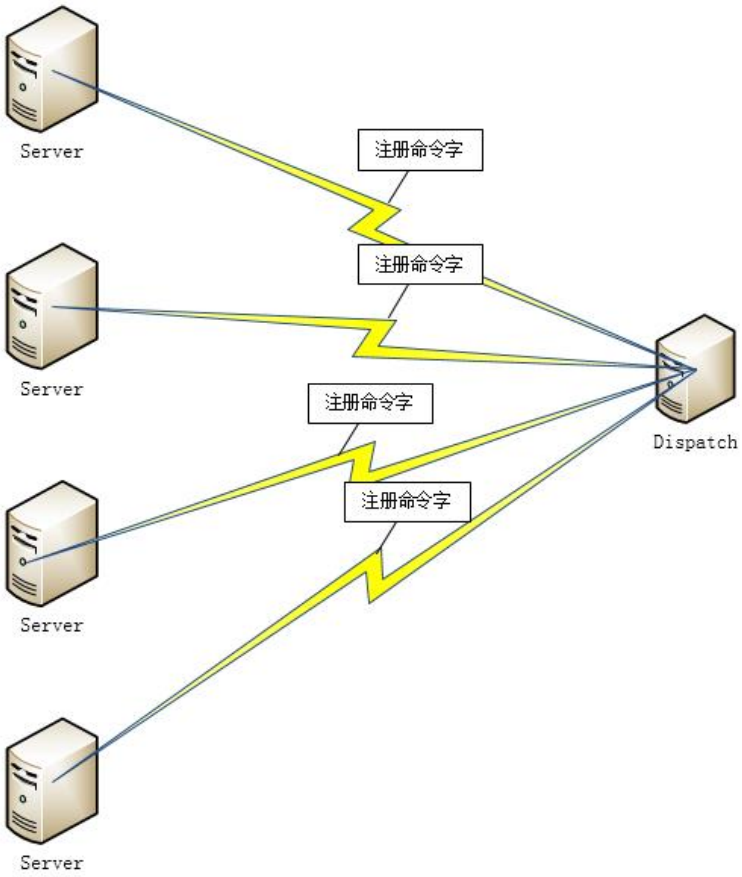
新一代棋牌架构特点

- 层次型设计，结构清晰，便于组件的复用与沉淀。
- 无缝升级，平滑扩展
- 单进程异步处理框架保证了服务器的高效处理性能，复杂业务采用状态机处理。
- 时间复杂性高的业务逻辑交由辅助线程处理，处理完交还给主线程。
- Lockless 设计。
- 所有内存对象采用内存池分配，减少内存分配时间以及内存碎片。内存数据分为在线数据和离线数据。在线数据分为持久化数据和非持久化数据。
- DB负载监控与合并写入机制。
- 完善的日志系统。
- 尽可能的保持系统的简单。只有少数进程，方便部署维护。

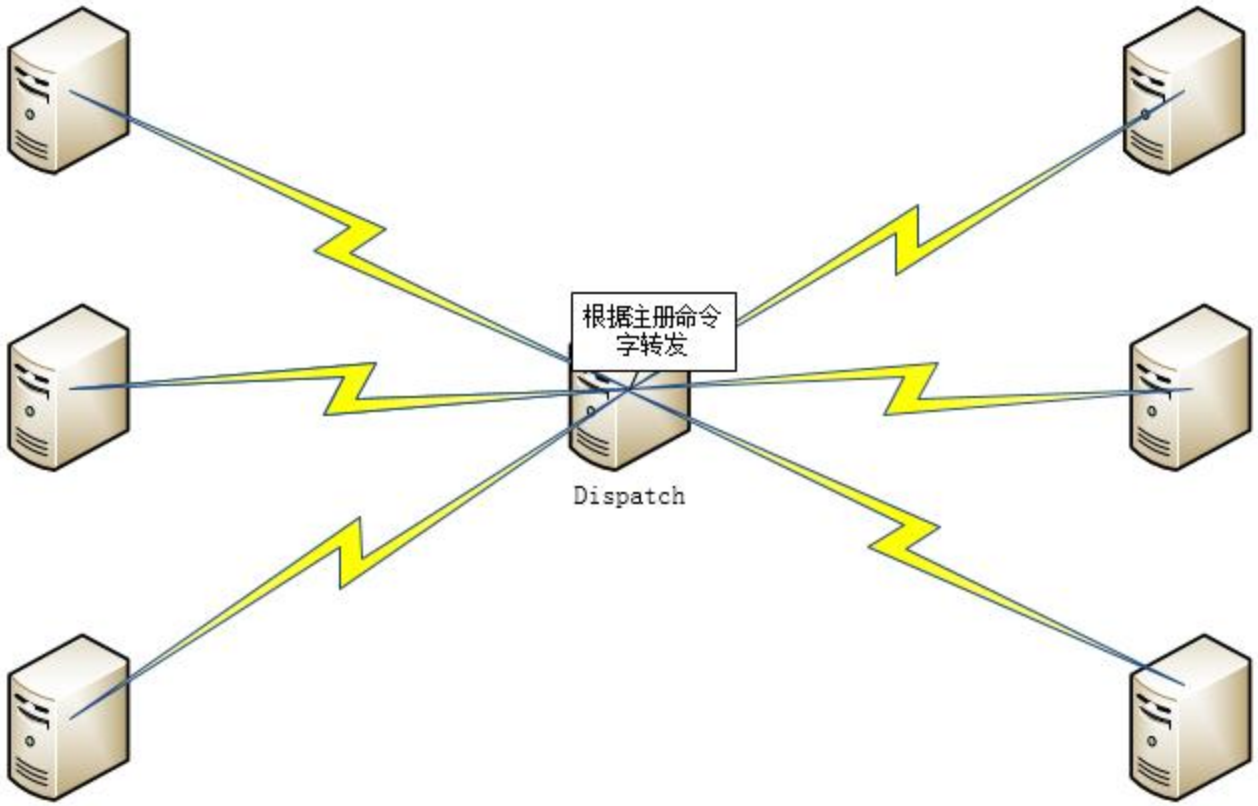
复用与组件化

- 层次化设计，注册机制降低耦合
- 公共逻辑的整合，高度复用

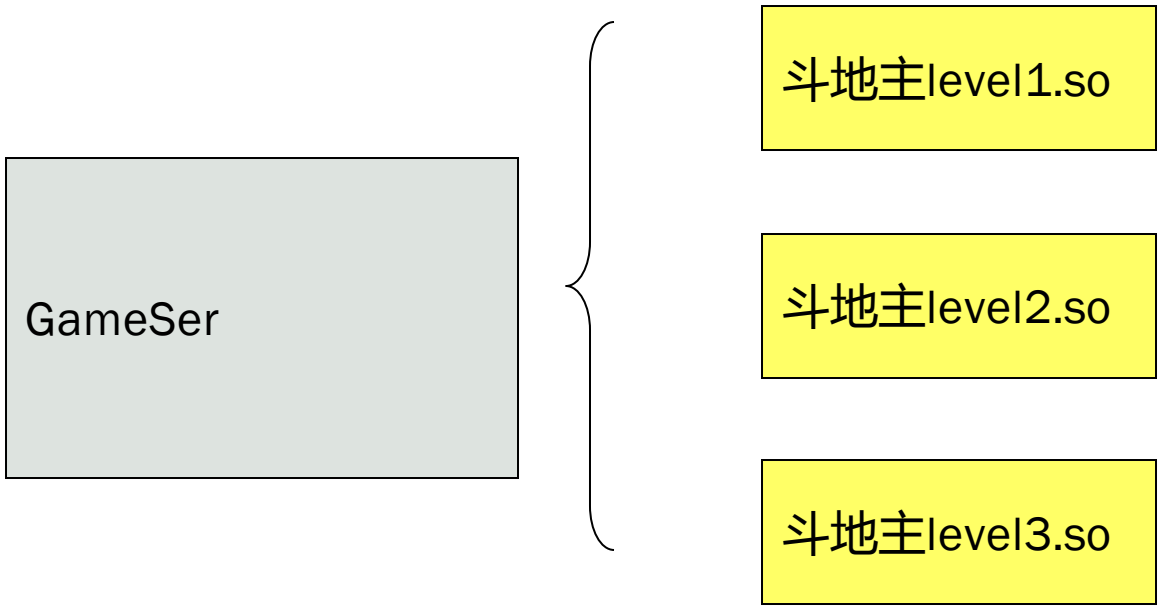
注册机制


Server
注册命令字
注册命令字
Server
注册命令字
Dispatch
注册命令字
Server
Server

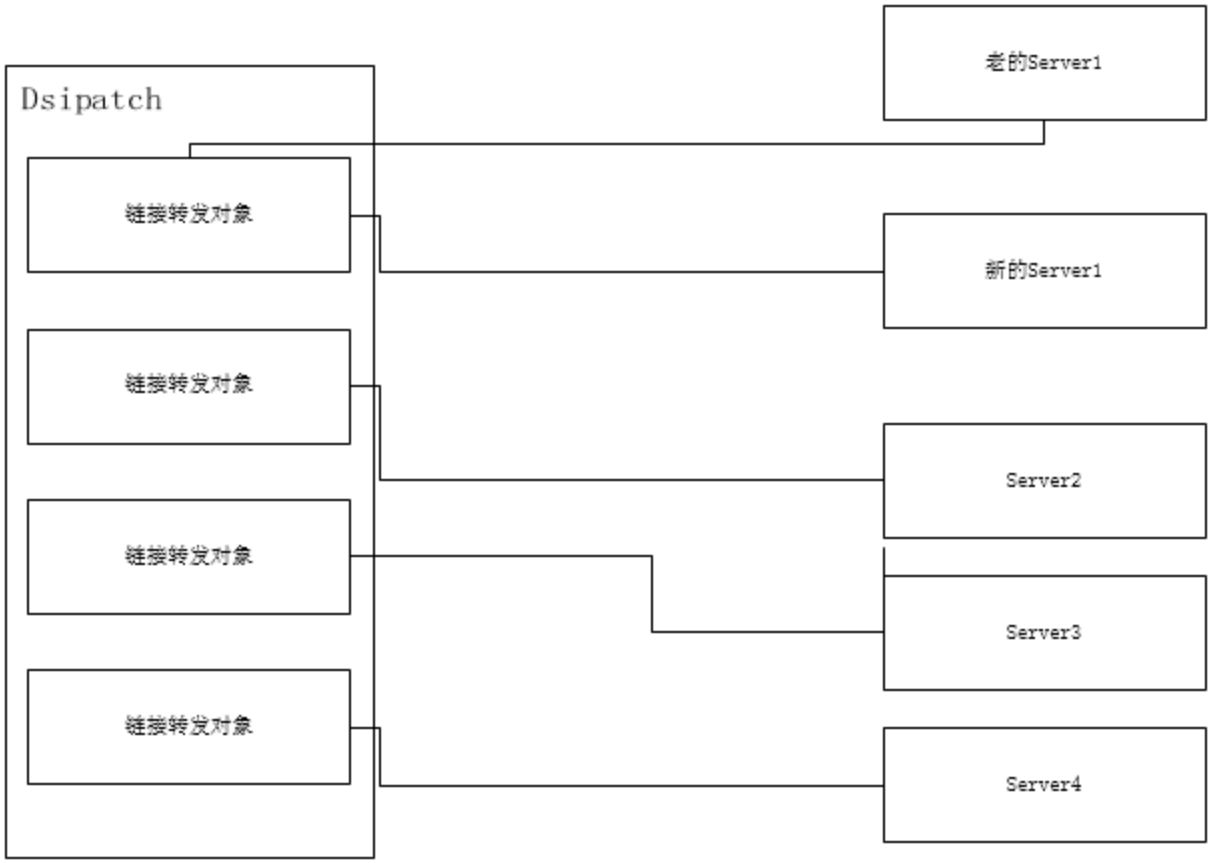
转发机制



针对桌子的游戏编程



无缝升级
利用Dispatch的链接替换



高可用性服务

- 高性能服务
- 全方位监控
- 高精度日志查询

进程模型

多进程模型
多线程模型
以上模型都消耗大量上下文切换的时间

单进程多连接更适用于大容量服务器

- 单进程+异步IO 使得系统运行在最佳的性能状态下
- 减少CPU额外工作量
- 需要保证核心(高负载)进程/线程数 < CPU 数量

千里之行始于足下--异步IO

- 网络通信和业务逻辑分离（剥离网络IO）
- 业务逻辑和存储分离 （剥离磁盘IO）
- 业务逻辑和日志分离 （剥离磁盘IO）
- 复杂业务和主业务分离 （运算量分离）

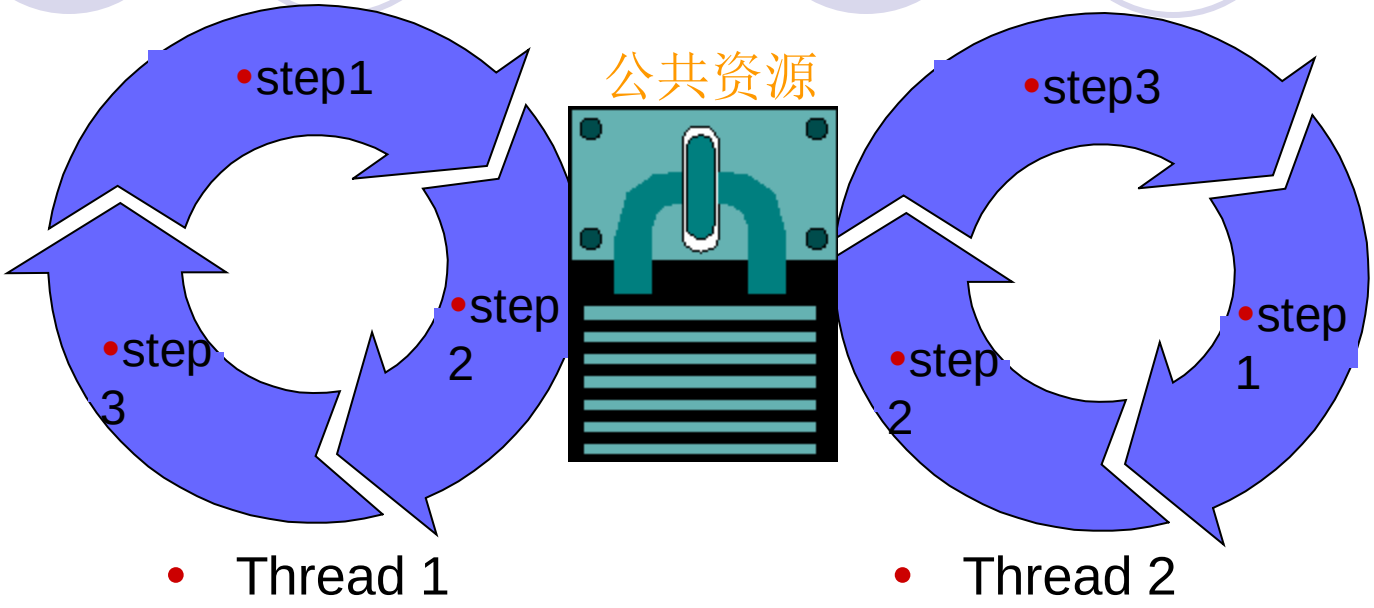
保证主业务不会受到任何阻塞

保证单用户不会掌控你的系统资源

如何避免互斥

- 流水化
- 数据局部化

流水



- **thread1**
- **thread2**
- **thread3**

数据局部化

- 以某种算法将公共的数据分散到各个线程空间中
- 每线程有一份自己的数据
- 每线程对自己的数据有读写权限，对其他线程中的数据只有读权限

异步处理 最高的并行能力

分布式系统下的服务器对请求的处理都可以看做是事务处理。事务模型是后台框架的基本元素，事务的本质就是一个有限状态机，其包括阶段和状态信息。

事务模型在分布系统可以应用在各种环境，加入严格的回滚机制可以开发账务系统，在分布式可以作为各种客户请求的处理机制

事务处理的框架-分布式异步模型

事务框架提供事务的调度，根据事务的阶段和状态进行调度处理。

事务的调度处理就是根据请求，和事务ID处理各种网络消息，每个事务只有在调度的一瞬间占用CPU，事务框架的处理能力远远大于多线程（进程）模型。

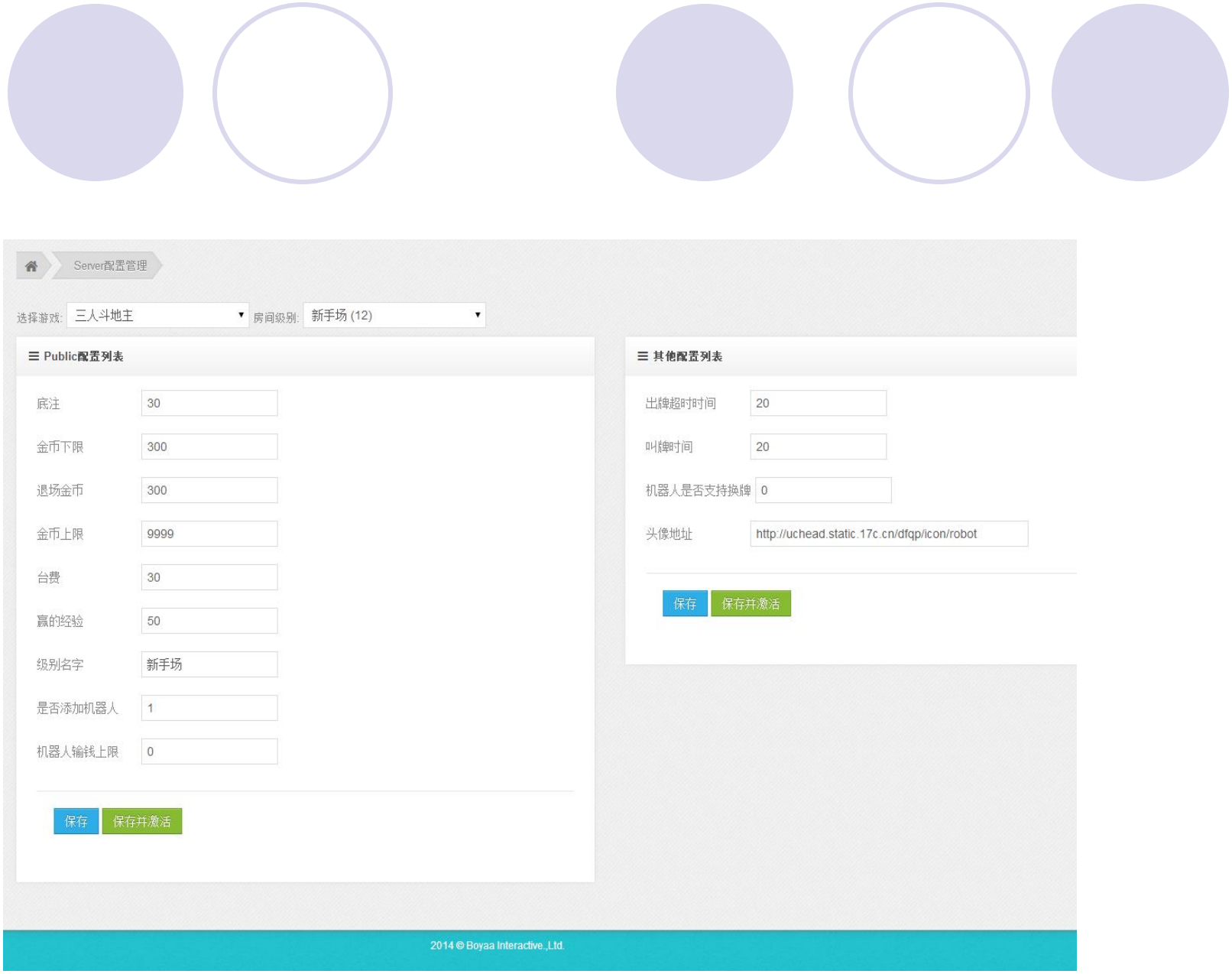
事务可以看做是一个处理一个请求的过程，在事务化的框架下的开发就是开发事务的处理。

数据Server模型

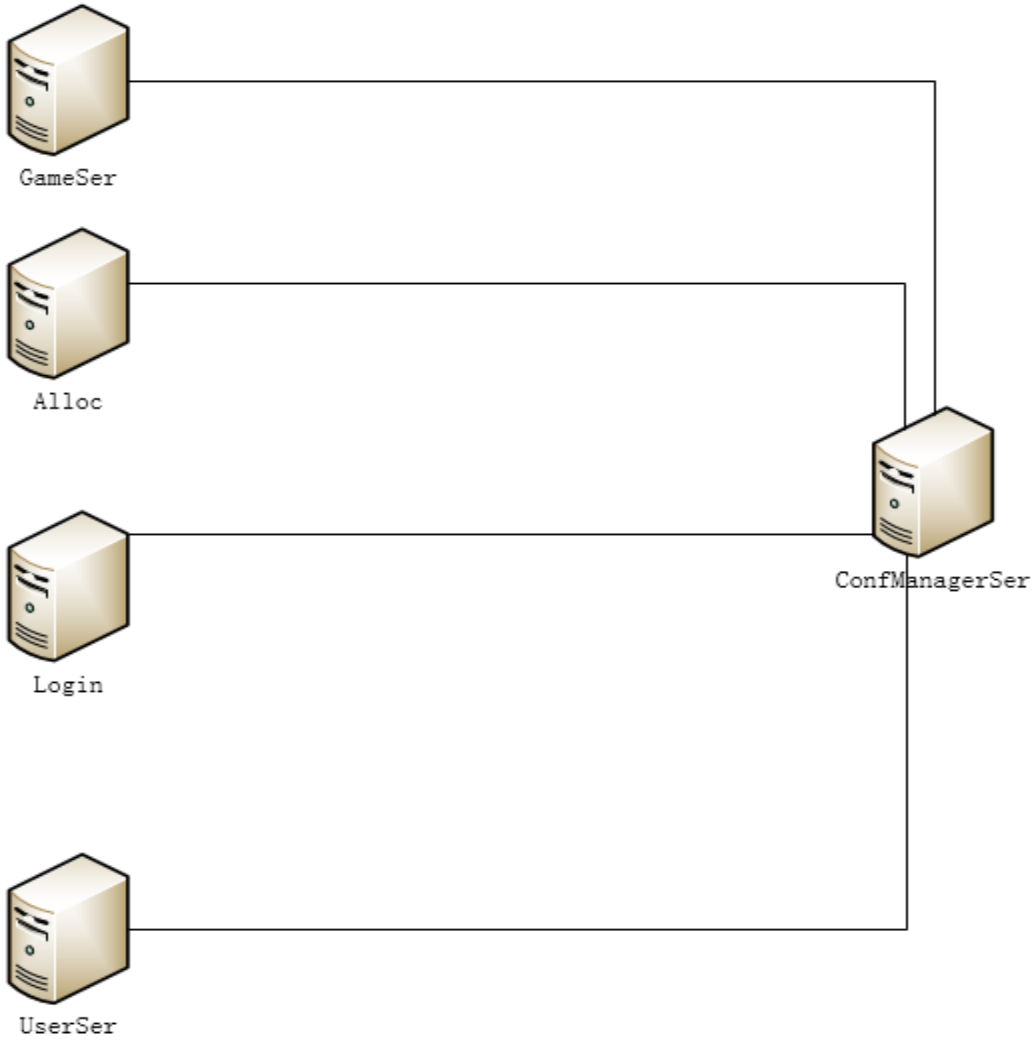

主控线程
处理线程1
处理线程2
处理线程3
处理线程4
处理线程5
……
……
处理线程16
MySQL数据表1
MySQL数据表2
MySQL数据表3
MySQL数据表4
MySQL数据表5
……
……
MySQL数据表16

高运维性

- 配置更新自动化
- 游戏更新自动化


Server配置管理
选择游戏: 三人斗地主
房间级别: 新手场 (12)
Public配置列表
底注 30
金币下限 300
退场金币 300
金币上限 9999
台费 30
赢的经验 50
级别名字 新手场
是否添加机器人 1
机器人输钱上限 0
保存
保存并激活
其他配置列表
出牌超时时间 20
叫牌时间 20
机器人是否支持换牌 0
头像地址 http://uchead.static.17c.cn/dfqp/icon/robot
保存
保存并激活
2014 © Boyaa Interactive.,Ltd.

中心配置策略



未来展望

- 立体监控与过载保护。
- 自动化部署
- 运营数据精确化、实施化

Q & A