# 背景

《厅弹》是一款基于网页Flash上的休闲竞技类游戏。以下为该产品的服务器架构说明书

# 术语

## 术语和定义

|  |  |
| --- | --- |
| 术语 | 解释 |
| GameServer | 游戏逻辑服务器 |
| FightServer | 战斗服 |
| Request（web） | 辅助服 |
| Flash(web) | 客户端资源服务器 |

# 系统概述

## 运营环境

### 服务器软件环境

|  |  |  |
| --- | --- | --- |
| 软件名称 | 版本 | 备注 |
| CentOS | 6.0以上 | 内核心基于linux 的操作系统 |
| MySQL /腾讯分布式CDB | 5.5 | Mysql数据库 |
| Java | 1.7 | 服务器Java 运行环境 |
| Tomcat | 7 | Java Web项目容器 |
| Nignx | 1.9 | WEB前端反向代理 |

### 服务器硬件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务器名称 | 型号 | CPU | 内存 | 硬盘 |
| 游戏服务器  中心服 | VC3 | 8核心 | 7G | 300 |
| 战斗服  DB 服务 | 自定 | 8核心 | 8G | 500 |
| Request(WEB)  Flash(资源) | VC2 | 4 | 3.5G | 200 |

根据实际测试情况，分配服务器硬件参数。

## 系统功能

### 产品功能需求

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| QQ用户接入 | 注册 | 登陆 | 等级 | 技能 | 时装 | 装备 |
| 宝石 | 卡牌 | 宠物 | 图腾 | 单人副本 | 多人副本 | 好友 |
| 公会 | 聊天 | 结婚 | 师徒 | 名人堂 | 跨服战斗 | 挂机系统 |

# 系统设计

## 系统整体介绍

|  |  |
| --- | --- |
| 游戏分区模式 | *分区服* |
| 前后台通讯类型 | *HTTP 与 Socket* |
| 连接类型 | *长连接 与 短连接* |

## 游戏架构

玩家

Request(web)

辅助服

FightServer

战斗服1...n

DB

FLash资源

支付

QQ平台

中控

GameServer

RoomServer

房间服

## 游戏模块分析

1. 游戏服GameServer

维护在此服的客户端与服务器的TCP连接（服务器与玩家的交互通道），玩家列表，

主要为玩家提供基本的游戏功能，如玩家信息，任务，道具，背包，装备等具体逻辑相 关的功能; 维护游戏服（GS）与（FS）战斗服的连接通道，房间列表镜像，提供帐号状

态查询服务，维护登录账号列表。

1. 房间服 RoomServer

维护玩家创建的房间列表

1. 战斗服FightServer

主要作用是实现战斗逻辑功能，跨区服战斗功能（PVP），PVE。实现队伍匹配功能，战斗逻辑功能。处理战斗服转发过来的玩家发出的战斗指令

1. 辅助服Request

辅助服是外部系统与游戏服交互的门面，平台用户登陆，玩家充值，中控接入，

# 系统设计要点

## 负载均衡策略

1游戏采用分服策略，当此导入用户量达到单服设定上限，加开新服起到分流的作用。

2 架构上的均衡略，每个区服，可以同时有多个GameServer游戏逻辑服，游戏运行中如果发现游戏逻辑服的压力过大，动态的启动游戏逻辑服连接到中心服上，完成一个区服的扩展，达到动态扩容的能力。

3 腾讯TGW -接入层的负载均衡策略(通过TGW接入来调整每台游戏服的权重)-此方式受限于腾讯云服务器，考虑是否采用，如何应用于现有的架构中。

## 容灾策略

GameServer 与 FightServer 按照分布式设计，其中单个游戏服出问题，不会影响到其它游戏服的玩家。

## 过载保护策略

游戏登陆采应用排队机制。

限制玩家登陆上限。

根据同时在线玩家数量，采用合适的定时存储策略。

## 模块功能划分

服务器架构以游戏功能，以及业务逻辑压力均衡为前提拆分服务器功能模块，

游戏核心主要战斗系统，需实现跨游戏服战斗与跨区战斗，把这些功能整合为FightServer,  
 游戏逻辑服(GS)主要为战斗功能提供辅助支持，PVE 。

## 数据扩容

数据库表分片设计，用户ID取模进行分表。

把大数据表进行水平拆分，如道具表，任务表...

不用参于数据分析列压缩后存储。

## Cache机制

游戏服（GS）增加缓存机制，缓存玩家基本游戏数据，加速玩家登陆过程，降低数据库压力。

辅助服（request）缓存排行榜数据生成xml文件,为客户端生成的配置文件。

# 业务流程

## 典型业务流程

登陆流程



PVE流程



PVP流程



# 用户模型分析

## 用户模型

|  |  |  |
| --- | --- | --- |
| *用户规模估算* | *单日活跃用户数(ACU)* | *3w* |
| *最高同时在线规模(PCU）* | *8000* |
| *注册用户数* | *10w* |
| *游戏基础数据分析* | *平均在线时长(AvgSessiontime)* | *30Min* |
| *平均每秒登录次数(AvgLoginPerSec)* | *AvgLoginPerSec=PCU/AvgSessiontime=8000/1800=5* |
| *平均单局时长(AvgMatchTime)* | *1Min* |

## DB负载分析

|  |  |  |  |
| --- | --- | --- | --- |
| *表名(Table Name)* | *细分情况* | *读请求(Req/s)* | *写请求(Req/s)* |
| *用户基础数据表\*20*  *可以产生变动表\*10* | *登录请求* | *AvgLoginPerSec\*20*  *=100* | *0* |
| *保存用信息* | *0* | *PCU\*10/WritePeriod*  *=8000\*10/2min*  *=670/s* |
| *总计* | *100* | *670* |
| *系统功能相关信息表 \*30* | *保存功能信息* | *0* | *tableAmount/ WritePeriod*  *=20/1=20/s* |
| *总计* | *100* | *690* |