用Python架设大型多人在线游戏服务端

王健

应用领域

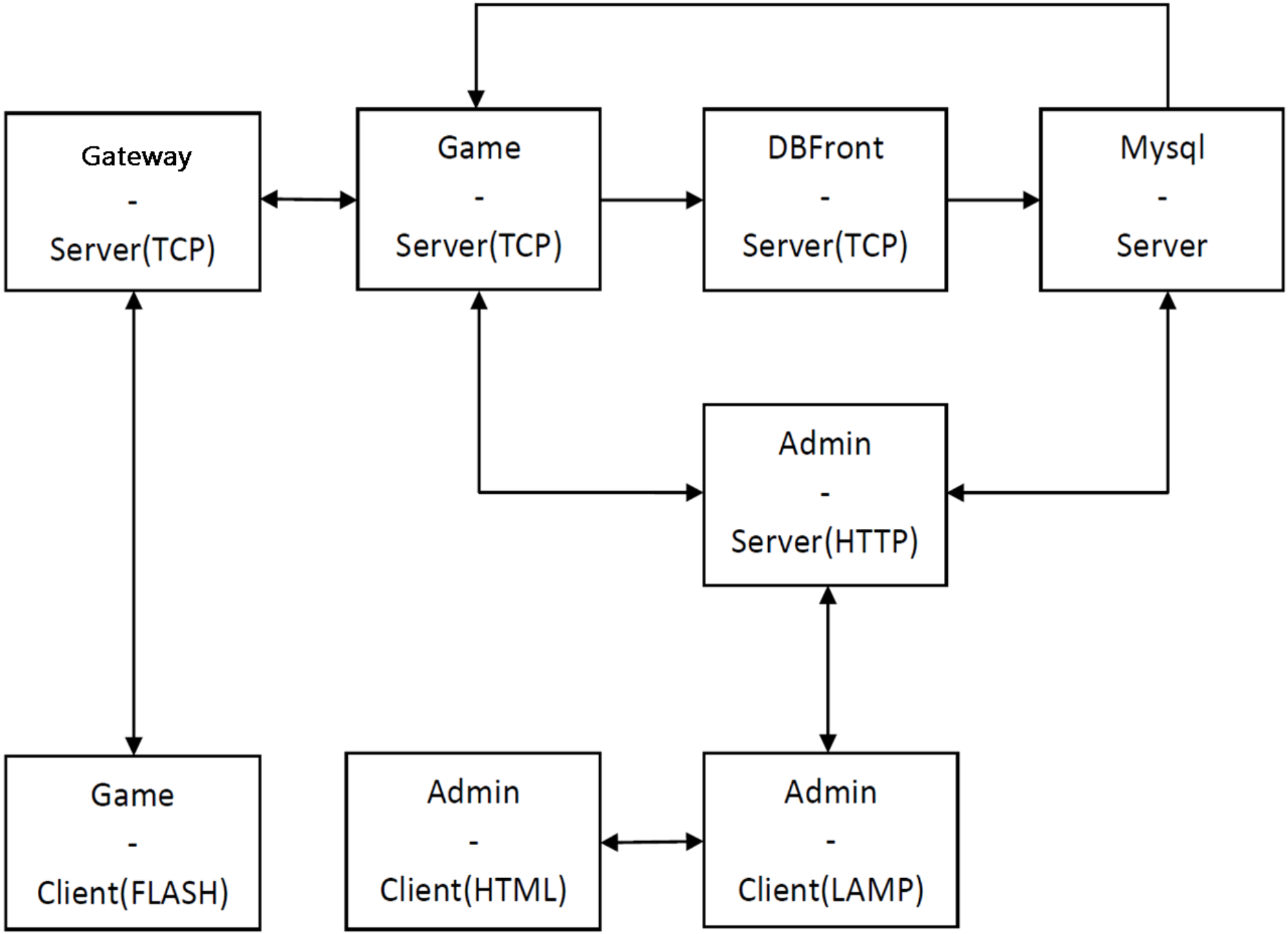
- 战网：暗黑破坏神2/3
- 社区：QQ宠物，TNT，宝石总动员

- 特征：
  - 多个线路
  - 数据共享
  - 线路通信
  - 线路切换
  - 全服商店（拍卖行）
  - 全服社交（好友，邮件，聊天）

需求概述

- 海量用户分布式数据库设计
- 海量用户数据缓存机制
- 线路负载均衡机制
- 线路通信事件机制

回顾2011届Pycon介绍的架构


Gateway
-
Server(TCP)
Game
-
Server(TCP)
DBFront
-
Server(TCP)
Mysql
-
Server
Admin
-
Server(HTTP)
Game
-
Client(FLASH)
Admin
-
Client(HTML)
Admin
-
Client(LAMP)

缺陷阐述

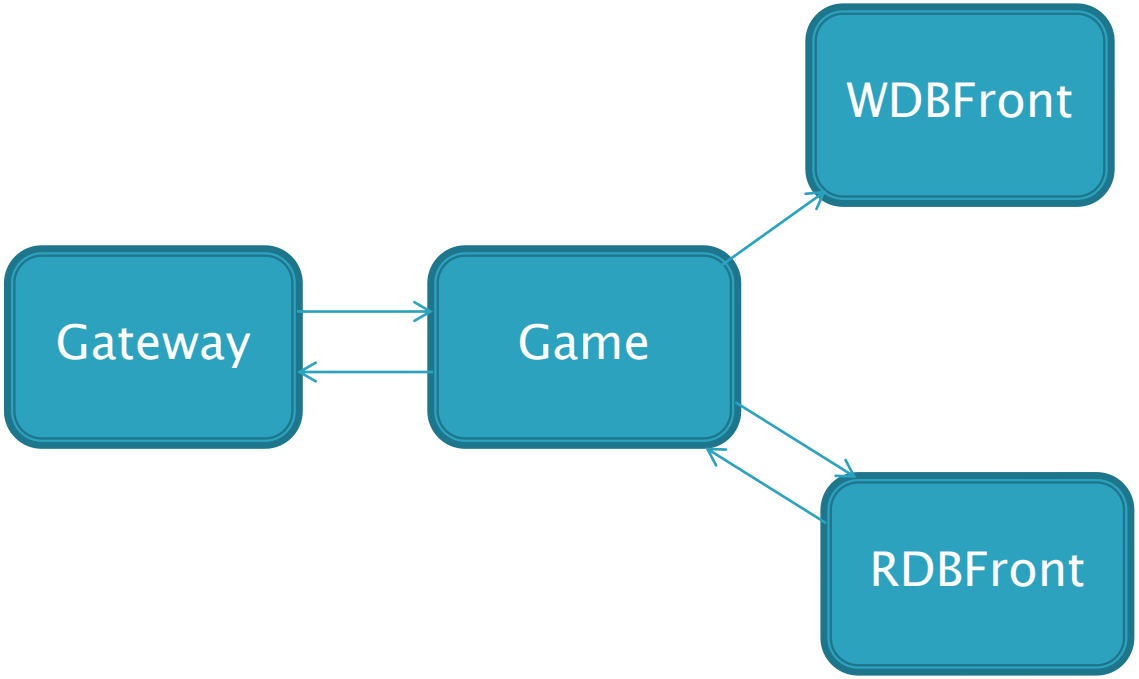
- Game无足够内存加载Mysql所有用户数据
- Mysql不支持分布式
- Gateway不支持海量客户端并发请求
- Game，Admin更难以承担重负
- ……

如何解决？

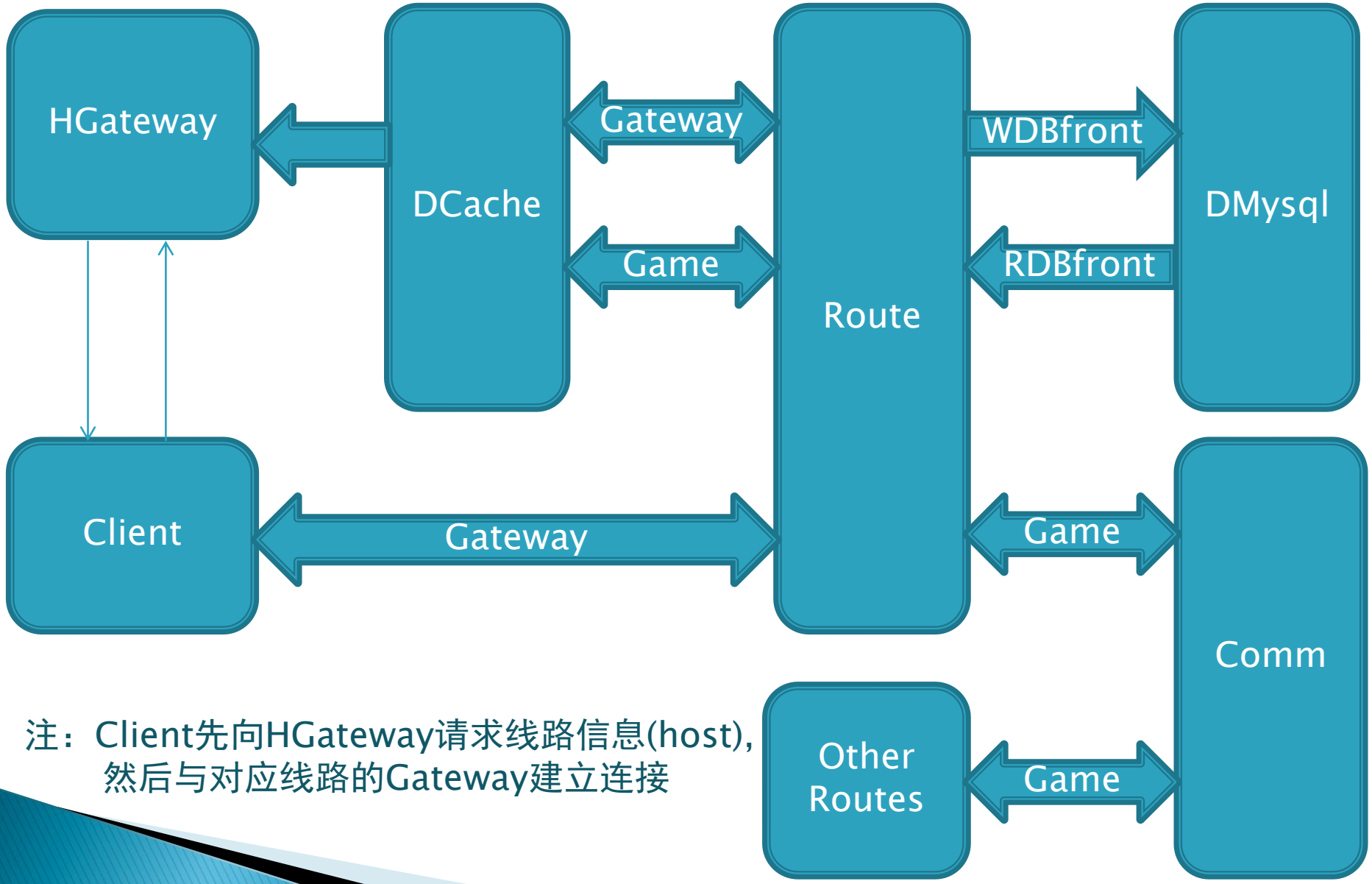
- 分离Mysql，建立分布式数据库集群（DMysql）
- 扩展DBFront为RDBFront和WDBFront，异步读写数据，不再将所有数据驻留内存
- Gateway，Game，RDBFront，WDBFront，Admin组成一个线路，作为单元嵌入新架构
- 新架构其实是线路集群，还要解决线路通信和调度问题

- 缓存机制，采用分布式Redis完成，客户端分布（类似共享内存机制）
- 增加Comm服务，连接所有线路（类似通道机制）
- 缓存记录各线路的运行及负载情况，比如各线路的在线人数
- 增加HGateway，负责线路负载调度，将Client分配到最合适的线路（http协议）

线路单元


WDBFront
Gateway
Game
RDBFront

线路集群与调度



注：Client先向HGateway请求线路信息(host),
然后与对应线路的Gateway建立连接

分布式数据库（DMysql）

- 根据UID范围分库/表
- 分析SQL语句，定位目标数据库/表
- 分布式对用户不可见，与操作单数据库无异

分布式缓存（DCache）

- 按需部署多个Redis服务
- 对key进行hash，定位目标Redis
- 分布式对用户不可见，与操作单Redis无异
- 支持上亿记录，保证性能
- 可应用于UID自增，角色名重复性检测，拍卖行等功能

线路通信（Comm）

- Comm连接所有线路的Game
- 扩展单线路的事件机制，实现远程RPC

事件派发，to为目标线路id，-1代表全服

```
def trigger_event(to, id, evt):
    pass
```

事件回调，from为来源线路id

```
def event_handler(id, evt, from):
    pass
```

- 可应用于全服社交，公告，聊天，好友，邮件等功能

负载均衡调度

- 在DCache中记录各线路的在线人数等信息
- HGateway根据给定策略分析线路负载情况
- Client向HGateway获取负载较低的线路信息，并连接进入游戏
- 或支持游戏中切换线路

未来扩展

如何提高单线路性能？

2011届提到的传统方法

分拆Game

- Game可以分拆为Game-Center和Game-Zone，将整个游戏世界按区域划分，各区域之间通过中心服务器连接：

- Game-Center：
- 游戏中心服务器，负责各场景服务器的连接，以及玩家全局行为处理，例如登陆，聊天，公告等等。

- Game-Zone：
- 游戏区域服务器，负责一个区域内的玩家行为处理，区域可根据各游戏特点灵活划分，例如一般的网页游戏，玩家行为大部分集中在主城，则区域数量不需要太多。

时代在进步，思路需更新

混合编程，erlang/golang

- Game这部分可采用天生分布式的语言重写
- Erlang函数式编程，开发效率低
- Golang有Python的良好基因，是否未来的趋势

关于我

网名：风魔W
QQ：4331436
微信：wangjianbecks
微博：http://weibo.com/wjbeckie
邮箱：charlie.wj@gmail.com

请联系我！

谢谢观赏