# **实时联网游戏后台技术选型和挑战（房间匹配与数据同步）**

作者：[Matchvs](http://gad.qq.com/user/index?id=2257559)

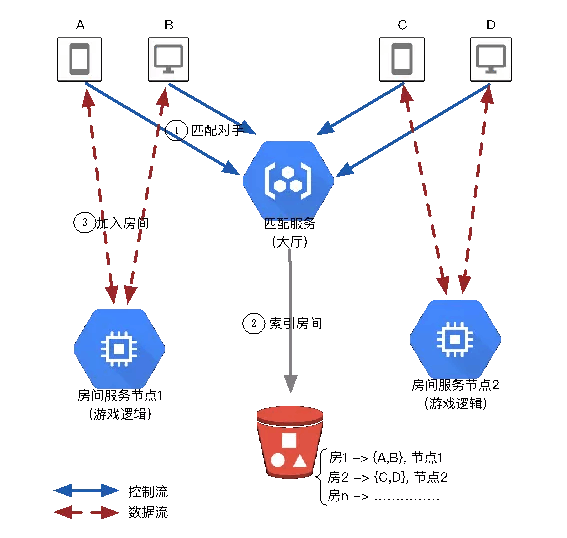
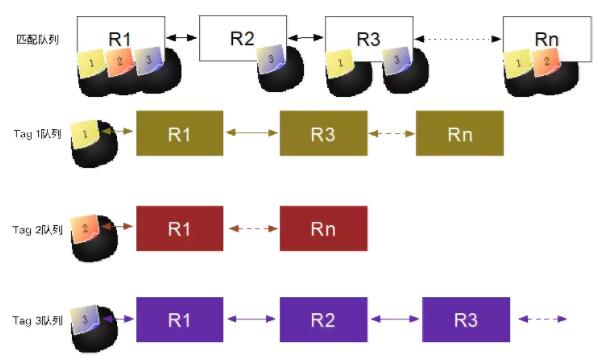
链接：http://gad.qq.com/article/detail/286330

在此前的《[网络接入篇](http://gad.qq.com/article/detail/270717)》中我们介绍了实时联网游戏网络接入相关内容，网络接入层开发考验的是开发者高性能网络编程的功底，即解决C10K甚至C10M的能力。本文开始介绍游戏逻辑层，由于不同游戏玩法千奇百怪，本文不涉及游戏具体的业务逻辑，只探讨在逻辑层实现中经常遇到的房间匹配和数据同步问题。

基于“房间”模式的联网对战游戏，游戏流程可分为匹配（matchmaking）和 对战(gameplay)两个阶段。匹配是指将在线”玩家中满足一定条件或要求的玩家“撮合”到一起进行游戏。对战是指匹配到同一房间的客户端(玩家)彼此建立通讯信道(一般是通过连接到指定房间服务器，并通过房间服务提供的接口)来同步玩家操作、游戏状态等数据以支持同一房间内玩家对战和互动。

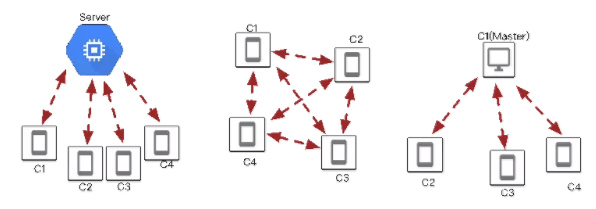
**房间匹配**

一般来说房间匹配服务架构如下图所示，其中和房间匹配相关的通讯数据称之为“控制流”，房间内游戏对战通讯数据称之为“数据流”，显然两种不同类型的流对带宽、延时等指标容忍度不同，实际开发中应分别由不同服务器来承担。

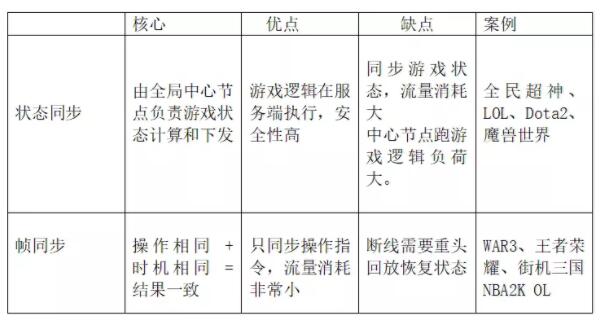
不同游戏的匹配策略可能不太一致，例如常见的基于场次、基于等级、随机匹配等。如果在房间创建时给房间打上一定数量的标签，我们便可以借鉴搜索引擎的倒排索引的思路来实现房间匹配算法。例如待加入的房间有： R1: [T1,T2,T3]，R2:[T3]，R3:[T1,T3]，Rn:[T1,T2]，这些房间不仅会存在与全局匹配列表中，同时还会存在每个标签索引中。假如Tag1表示“竞技场”，要匹配“竞技场”房间只需在Tag1索引队列中匹配，同时为了增加灵活性，匹配还应该支持由AND、OR、NOT、括号操作符组成的逻辑表达式，例如 T1 AND T2 的匹配候选集= INTERSEC(Tag1队列, Tag2队列)，(T1 AND (NOT T2))的匹配候选集=DIFF(Tag1队列, Tag2队列)。

**游戏同步**

在《网络接入篇》中介绍过网游三种常见的网络架构：C/S架构，P2P架构，还有一种C/M架构，分别如下图所示。

网络游戏中通过同步机制来保证各个客户端游戏世界的一致性，主流的同步机制有两种：1 状态同步; 2帧同步。状态同步是指：由客户端负责将玩家的操作发往中心节点 (服务器或master客户端)，由中心节点来负责游戏逻辑计算并将计算结果广播给客户端，再由客户端负责渲染游戏结果。而帧同步的理论基础是：游戏逻辑由操作指令驱动，只要操作序列一致，那么游戏结果就应该一致。

下表我们简要对比了两种同步机制的差异

游戏类型、网络条件是同步机制选择时的首要考虑因素，对于同步频次低，例如休闲回合制游戏，鉴于数据流量小同时对游戏逻辑安全性和以及防外挂能力有较高要求，一般大都采用状态同步的方式。但对于操作要求比较高的，例如运动、赛车类游戏里涉及大量的物理逻辑运算，以及MOBA类对流畅性有更高要求的游戏，帧同步是更好的选择。