信令系统

# 概述

信令系统主要是为了处理用户登陆，在用户间传递消息，控制Conference Call的各种状态。 考虑到美国的互联网环境，信令系统初期设计不需要太多考虑机房间通信和机房间同步，考虑在高可靠的核心机房部署信令服务，以提高初期系统整体稳定性和可靠性。

一旦初期系统搭设完成，信令系统需要进行多机房扩展，在机房间搭设路由系统，部署跨机房的同步和通信方案。

因此，信令体系的建设将分为2个过程完成：

1. 实现机房内信令控制，能完成用户登陆，会议指令的传递。目标是支持快速实时可靠的信令系统，支持同时在线容量：1000万。
2. 实现机房间路由机制，能完成跨机房的信息同步和指令传递，并提供高可用的核心库复制备份和容灾方案。

Router

Router

Push

Dispatcher

Dispatcher

Dispatcher

Login

Login

Login

Login

Login

Login

Login

Login

Login

Mongo

Pull

Dispatcher

# 消息分发系统(Dispatcher)

消息分发子系统的任务，是完成同一个集群内部，登陆到不同Login Proxy的用户之间的消息分发任务。结合到业务层，需要提供好友关系状态，和呼叫控制指令的发布。

消息分发(Dispatcher)以3个层次完成消息分发

1. 如果Login检测到消息接收人登陆到同一个Login进程，Login直接把消息转发给接收方。否则Login把消息送给Dispatcher.
2. 如果Dispatcher 检测到消息接收人登陆到本Dispatcher下令一个Login进程，Dispatcher把消息推给相应的Login Proxy, 然后由Login Proxy完成消息分发。
3. 如果Dispatcher发现消息接收人在另一个Dispatcher下，则通过Router进程把消息推送到另一个Dispatcher, 然后经历Dispatcher=>Logi=>User流程完成分发。

大致方案如下图，单dispatcher下预计可以支持的用户量为：100万。

Dispatcher

Login

Login

Login

msg

msg

此构架最大的问题在于，Dispatcher进程的单点依赖，一旦Dispatcher下线，整个系统将不可用。因此必须进行扩展，在同一个机房内部署多份Dispatcher=>Login的架构。各个Dispatcher之间保持连结性。

Dispatcher需要知道另一个用户登陆在了哪个Dispatcher之下，因此需要有一个online 进程保存实时的用户登陆信息。这个online进程的实现方式多样，而且目前不是非常紧迫的内容，先暂缓实现。短期实现方案可以让Dispatcher进程间做多播。

# 登陆系统(Login)

登陆系统负责用户登陆和长链接保持，并作为用户所有控制信令的入口点。登陆系统保存用户登陆信息，并向Dispatcher和Online库上报。Dispatcher根据Login提供的用户登陆信息，来实现消息在用户间传递。

Login需要实现的功能:

1, 用户登陆和长链接。从核心库查询用户密码进行认证，并保持与用户

# 核心库(udb)

核心库udb，用户保存用户的个人信息，包含mobile, passport, password, billing信息等。登陆进程需要从udb获取信息来验证用户登陆信息。

初步想法基于mongo架设，单master模式，主从结构。所有Login前端只读，注册进程负责写。如果需要多机房部署，保持中心节点(Master)读写，其它节点都是只读, 以确保数据的一致性。

核心库以uid为hash key，初期考虑到用户量级，可以不进行分片。

# Push消息队列(push)

Push消息队列的存在，一定程度上是为了解决router之间，也就是机房大区之间的消息同步。考虑到大区数目，机房数目都是未知数，任意一个机房要想另一个机房发送消息，有2种可行方案：

1. 任意2个大区之间建立链接
2. 建立中心节点，所有大区链接到该中心节点。

第一个方案，任意2个大区之间建立链接，是比较直观的做法，也没有单点失败的问题。但是复杂度会很高，任何时候调整某个节点，会导致所有节点要做配置管理，或者实现完善的节点发现机制。

第二个方案，是消息队列方案，配置管理简单。添加或者删除节点，不需要其它节点做调整。但是可能有性能瓶颈，所有数据必须经过统一的中心节点。因此折中一点的方案是，部署多个中心节点，让每个节点可以连接多个中心节点。中心节点数据按照某个算法做hash(比如uid)。

容量估计：

1000万用户

每个login进程8万用户, 125台Login前端机器。