会议记录

会议主持人：杨总

会议时间：2012年12月5日星期三

会议主要内容：讲解新版msgsrv及其客户端app的通信机制及实现原理。

会议要点：1.新版MsgSrv程序要组成一个网状结构。

2.Msgsrv 的客户端即app要与多个msgsrv建立连接。这样防止了其中一个物理链路或则msgsrv崩溃导致的服务中断。

3.新的app full name 即

AppName . fd .ns .msgsrvname 代表一个唯一的app的物理地址。

“.” 一个点 还是代表app连接的msgsrv的地址.。

…msgsrv 代表一个msgsrv的地址。

APPname…msgsrv 代表一个msgsrv上登录的app的物理地址 。

大致的通信过程：

App1 要想和App2 通信，首先要知道网络上有多少个App2以及它们的full app name ，以便从之选择一个App2通信（解决负载均衡问题），

1. App1 通过它的每一条与msgsrv的连接发送一个报文，格式如下

. SetSameApp App2 fullname1 fullnam2 fullnam3 ….

可以选择省略发送本条链路连接上的fullappname，

即 在发送这个报文的连接上的app那么为fullname1，则可不发送这个名字即：

. SetSameApp App2 fullnam2 fullnam3 ….

1. msgsrv 群负责转发这几个报文到连接到他们的App2，一个msgsrv上可能有多个相同app的实例，所以，msgsrv负责选取一个app实例用于转发这个报文（可以采用轮询的方法，实现以下负载均衡）。
2. 收到报文的app2负责回复这些报文，其内容为自己的每个fullappname。
3. 这样App1接收到了多个来着app2的回复，从中选择一个频数较高的app2的fullname作为要通信的实例。App1可以选择缓冲这一个实例地址到超时。

在选择了要通信的fullname之后，app1在它的每条链路上发送要传递的报文到app2，通过报文选项中的ns序列号来区分同一个请求。在app2 收到一个报文后，立即进行逻辑操作。并记录这个报文的ns序列号，并在它连接的每个链路上返回数据，完成请求。当下次在收到报文时，通过判断ns序列号来决定是否是同一个报文。

Msgsrv保留的字符 回车字符：’\n’，空格字符:’ ‘

其中要在单引号 ‘ ,双引号之间支持任意的字符。通过添加转义符实现。

Msgsrv之间的通信方式，1:可以使用点对点的连接，2:也可以自己实现路由协议。

另外：我觉得现在实现的这个架构使用3倍或则更多的冗余数据报发送来弥补可能由于单个或则多个msgsrv崩溃，造成的服务中断的情况。但是，msgsrv崩溃的概率是有的，但为了这点概率，导致了几倍的流量。另外，在物理链路不增加的情况下，一个app如果在一台物理主机上，这台主机或则它所连接的网络故障将导致它所有的连接故障，如当网关故障时，这种问题在增加连接数的情况下是无法处理的。