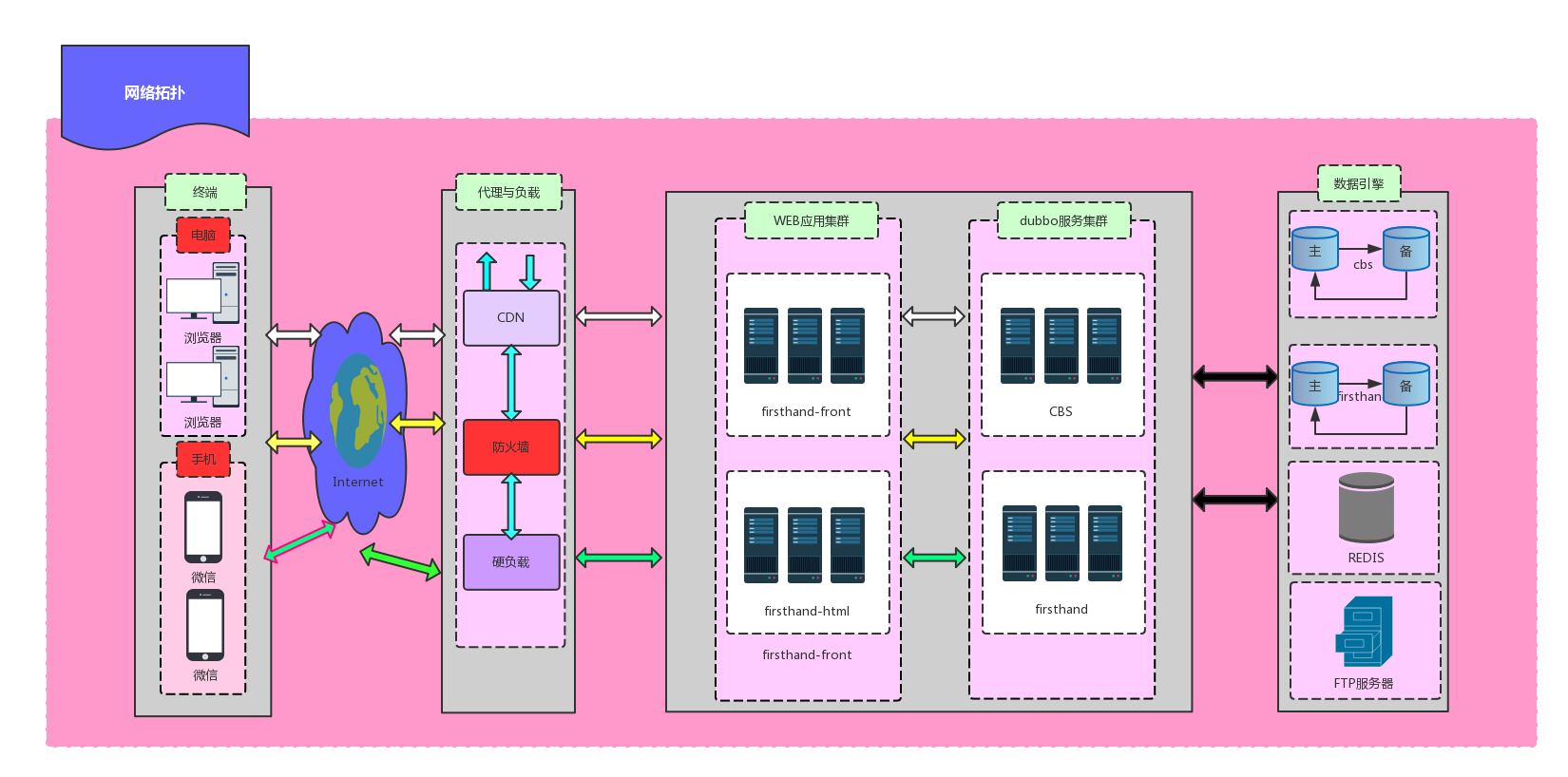
催收系统入口和出口高可用方案



## 一、高可用、系统稳定性要求

1：业务集中某一个职场办公

2：业务系统强依赖系统，有一点问题就会在群里反馈，每天系统可以为公司创造较大的受益

3：公司门户网站，一旦因为网络其他原因造成服务挂，会对公司造成不要的影响

……

待解决问题：核心业务单机房，单出口，单入口；一旦机房出口、入口出现故障，对业务影响致命

入口被堵：当北京机房出现网络拥堵、端口拥堵,造成CDN到北京机房不通的情况下全国各地职场坐席用户请求无法从外部正常访问至业务系统的问题。由于这种问题无论从运维的角度还是从研发系统的角度都比较难于监控和准确定位，如果采用手工去切换的方案需要比较长的问题排查和确认时间，且需要研发和运维之间沟通确

认，到最后手工修改并生效的时间周期比较长。对业务正常作业有较大的影响

出口被堵：无法通过应用访问第三方服务的，造成系统核心功能无法造成，严重影响生产

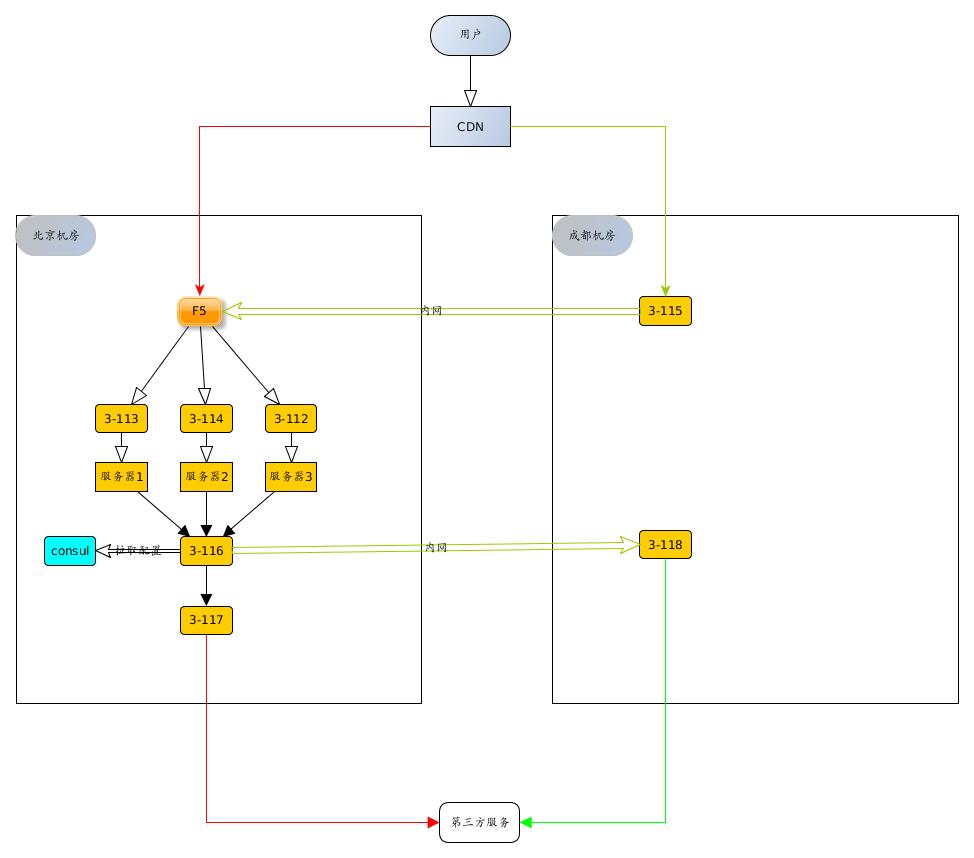
异常截图：





### 解决方案

基于成都和北京机房内网是专线互通；在成都机房新增Nginx，采用HTTP协议通过 Nginx 请求 …….…….com、 访问状态正常。



## 相关业务时间统计

测试环境：

研发中心网络

使用命令：

time curl https://{访问入口IP}/ -H "Host:域名" -k -vosa

time curl https://{访问入口IP}/ -H "Host:域名" -k -vosa

访问入口IP分别为：主源北京idc入口IP和备源成都idc入口IP

测试数据：

用时数据单位均为：秒

|  |  |  |
| --- | --- | --- |
| …….…….com | 主源 | 备源 |
| 时间1 | 0.069 | 0.206 |
| 时间2 | 0.041 | 0.200 |
| 时间3 | 0.047 | 0.205 |
| 时间4 | 0.030 | 0.209 |
| 时间5 | 0.042 | 0.212 |
| 平均 | 0.0458 | 0.2064 |

访问…….…….com时

北京与成都访问的延迟之差在0.1606秒左右，可以接受

## 产品、业务通过绑定host验证正常可以作业没有问题

测试条件：

1：产品机器绑定host可以正常操作，打开页面正常，暂且没有问题

2：业务配合安排5个坐席，绑定host后不间断按照日常的操作进行作业，业务反馈正常，通过后台的日志监控，通过备源nginx进来的请求符合负载均衡的策略，会打到三台对应的后台服务器上，证明线上配置策略无问题，生效。

## CDN切换说明

### 1、手工修改CDN主备源并生效的时间

设置CDN改为主备按照比例访问的方式，或者设置CDN为完全走备源的方式，从要求 CDN厂商修改配置，到CDN反馈已经部署完成，大概花了15分钟。

本次主要进行了两次次演练验证，

第一次：按照地区（贵阳）业务请求流量1：9，少部分流量分发到备源（成都新的nginx节点）后，在接下来10分钟内通过业务反馈，有坐席页面不间断的提示多次登录的页面情况，不能正常作业，多个业务同时反馈，随后通知CDN把所有的流量都切换主源（北京），待CDN切完生效后，业务逐步反馈正常



分析整个原因CDN按照部分流量切换，现在的公司部署的架构方式：

备源采用：nginx的route方式进行路由规则；而北京主源这边采用的是F5，是通过jessionid进行负载均衡策略，在部分流量通过由于用户的一系列操作中不同的操作，请求到北京、成都，分发到不同的服务器上，主备策略不一致，无法查询到对应的登录信息，验证无法通过，cookie失效，不断提示用户登录的情况。

为了进一步确认问题，通过研发本地已经模拟出相关策略下会话失效，发送302跳转，确认策略不一致导致问题的原因。

结论

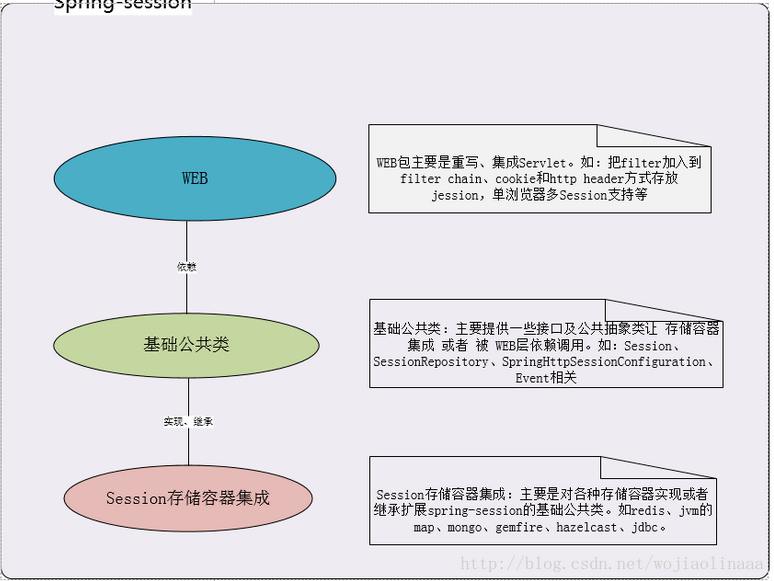
本方案解决了当北京机房出现网络或者端口被堵，业务请求无法访问至催收系统的问题。目前是先采用手工方式验证整个方案是否可行，

因此经过线上测试和验证，对于业务方访问  …….…….com的切换是目前是通过CDN技术人员配合人工切换来完成的，时间较长。基本符合预期结果

## 方案规划

5.1建议由架构组统一协调，一起讨论梳理登录信息的存储的位置，来解决不同地区，不同切换策略中出现重复登录的问题，同时为今后AB测试打下基础

考虑对热点数据的单点影响，自己业务系统实现，使用springsession实现的session信息存入redis实现会话信息共享



具体实施方案：

1、本周主要使用spring-session实现了将前端服务器用户登录session保存到redis服务器中，以解决负载均衡时出现session无效的问题。

    由于系统基于sso单点登录，校验session的逻辑在sso提供的过滤器中，子系统只能配置，无法修改逻辑。最终决定使用spring-session解决session共享问题，原因如下：

    1）spring-session使用简单，通过配置redis服务器连接信息，并添加一个filter即可实现，该过滤器通过重写HttpServletRequest对象中获取、保存以及置失效等session相关方法，将默认的session保存在服务器中转移到保存到redis中，实现了session共享。

    2）使用spring-session可以避免配置应用服务器贡献session，完全应用实现，与服务器无关。

    经测试，使用spring-session可以解决nginx负载均衡时session失效的问题：开发环境使用两台前端服务器，一台nginx负载均衡器，将浏览器的请求转发到两台前端服务器上。通过观察日志，发现请求转发到任意一台服务器上登录均验证通过，并成功返回页面，说明session共享成功。同时查看redis服务器上保存了session相关信息，包括sessionId和对应的session对象，以及失效时间。

    测试时将浏览器中的nginx地址的cookie删除，仍能正常访问，因为前端在cas登录信息尚未失效时仍会向cas重新获取登录session，符合预期效果。

    只删除redis中的session信息时，测试结果和上述一直。

    如果删除cas ticket，保留nginx cookie，仍能正常访问，因为前端服务器的session尚未失效。

    如果同时删除cas和nginx的cookie信息，再次访问时要求重新登录，符合预期结果。

    点击退出功能退出系统后，再次访问系统功能要求提示重新登录，符合预期结果。

    由于cas中通过登录名验证如果当前用户并非当前session中的登录名，表示当前访问无效，将要求重新登录，因此不会出现退出重新登陆后当前用户使用之前登录用户信息的情况。

5.2与CND厂商沟通确认自动主从节点方案策略，提高问题出现时缩短对业务的影响

### CDN主备源自动切换并生效的时间，CDN会每两分钟探测一次请求是否失败，当发现一次失败后，会减少探测时间为30s一次，再失败，第二个30s再探测，三次失败切到备源。因此从主源切换到备源最长三分钟，最快一分多点切换。当主源出现异常以后，每分钟会检测一次服务是否恢复，一旦恢复即切换回主源

## 六：总结演练结果：

CDN配置自动切换策略的情况下

  禁用主源IP（主切备）：

1分钟之后，请求开始打到备源；

3分钟之内，所有请求打到备源，服务恢复正常，业务操作系统正常，无问题反馈 。

持续时间：本周四 16:00-18:00在备源，业务无问题反馈

恢复主源IP（备切主）：

1分钟之内，请求开始打到主源；

3分钟之内，所有请求打到主源，服务恢复正常，业务操作系统正常，无问题反馈 。

  入口灾备方案共经过5次演练暂无问题，效果达到预期，可以满足系统入口被堵灾备需要。