PMS概要设计

1. 概述
2. 功能说明
   1. 监控灾备平台域状态
   2. 提供查询、通知平台域状态接口
   3. 控制平台域启停流程
3. 接口依赖
   1. 服务发现客户端可查询到跨域公共服务器信息
   2. BMC机房配置下发和查询
4. 场景分析



* 1. 主机房可设置主机房或同平台域的副机房作为灾备机房
  2. 主主就是跨 域灾备，主副/副副就是同域资源共享
  3. 主副/副副机房之间由CSS控制（会议无法召开，自动模式就使用备机房）
  4. 主主机房之间由PMS控制（链路中断，恢复会议）
  5. 手动模式，主主所有业务都感知。主副/副副仅APS和CSS感知

1. 平台对外模型



* 1. 平台使用haproxy和nginx作为反向代理服务器
  2. 会议中的终端，不会处理APS登录不了逻辑
  3. 终端离会，会马上登录APS，若登录失败，会尝试登录其它APS
  4. PAS没有主动踢终端逻辑

1. 设计
2. 模型



* 1. 部署会将本平台域所有静态信息上报到服务发现服务器（Mq/Mysql/Redis/Tomcat/Nginx/Haproxy/PAS/CSS/APS/MS/DSS/CMS等）
  2. 组件代理会将一些公用组件动态信息上报到服务发现服务器（Mq/Mysql/Redis/Tomcat/Nginx/Haproxy）
  3. 业务启动后，会将业务信息上报到服务发现服务器（PAS/CSS/APS/MS/DSS/CMS）
  4. PMS通过服务发现客户端可以动态感知主平台域心跳(断链)和主平台域启停状态(Watch节点)

1. 灾备流程



* 1. 备PMS和主服务发现服务器链路中断
  2. 将主平台域moid加入到托管列表
  3. 阀值为主备板卡切换时间(可以通过查询是否存在主备板卡减少切换时机)

1. 手动停用平台域流程



* 1. 封锁入口（需要服务发现明确实现逻辑）
     1. 停止Nginx本平台域Http代理（动态配置修改reload，或发送http请求给Nginx）
     2. ~~停止Haproxy本平台域的Tcp代理(同上)~~
  2. 停用基础云平台（接口基础云平台提供）
     1. 停止Mqsql、Redis写（如果做不到，就等待业务ACK）
     2. 重新启用的时候，Mysql会同步一次
  3. 清除业务
     1. CSS结会（不清除数据）
     2. ~~会管剔除用户~~
     3. APS不再用于登录
  4. 修改服务发现平台域启停状态
  5. 备平台域处理
     1. 备PMS接收到节点变化，告知备业务主平台域加入到服务列表
     2. CSS恢复会议
     3. APS接收非本平台域用户登录
     4. 会管允许非本平台域用户登录

1. PMS启动流程
   1. 数据库查询灾备信息
   2. 服务发现客户端查询跨域服务发现地址
   3. 检测当前数据是否同步完成（基础云平台提供接口）
   4. 读取数据库查看平台域启停状态，将状态注册到服务发现服务器
   5. 动态修改nginx和haproxy(两个组件启动时，不对外提供服务)
   6. 广播当前平台域状态
2. 对外接口
3. 获取平台域状态

接口: monitor.req.q:moid(平台域)

类型: RPC

服务对象:平台域业务

请求内容(json)

|  |
| --- |
| {  msg:”PLATFORM\_STATE\_REQ”,  moid:”123456789”  } |

回复内容(json)

|  |
| --- |
| {  msg:” PLATFORM\_STATE\_NOTIFY”,  moid:”123456789”,  state:1 // 1=启用 2=停用  delegate\_plamforms: // 托管的平台域  [  {moid:”123”},  {moid:”456”}  ]  } |

1. 平台域状态广播

接口: monitor.notify.ex monitor.plamform.k

类型: 广播

服务对象: 平台域业务

广播内容(json)

|  |
| --- |
| {  msg:” PLATFORM\_STATE\_NOTIFY”,  moid:”123456789”,  state:1 // 1=启用 2=停用  delegate\_plamforms: // 托管的平台域  [  {moid:”123”},  {moid:”456”}  ]  } |