PMS概要设计

1. 概述
2. 功能说明
   1. 为平台其它业务模块屏蔽服务发现
      1. 提供平台域、机房状态查询、状态推送
      2. 提供跨域平台公共组件查询、推送
   2. 控制平台启停流程
      1. 正常平台启停
      2. 灾备平台启停
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



* + 1. CMS下的临时节点，用于监控逻辑
    2. 所有业务在启动的时候，都会在各自业务永久节点下创建临时节点
    3. 平台公共信息存储于平台config子节点中
  1. BMC机房配置下发和查询

1. 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. 所有程序都是由服务发现脚本启动，并将状态上报到服务发现服务器（ZK节点）
  2. PMS通过服务发现客户端可以动态感知节点
     1. 订阅平台域status子节点，用于监听灾备平台域启停状态
     2. 查询平台域config子节点，用于获取平台域公共配置信息
     3. 订阅业务启动临时节点，用于查询机房状态

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. Redis数据灾备控制（具体查询 跨级建云-Redis灾备.docx）
   5. 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
   6. 广播当前平台域状态
2. 灾备流程



* 1. 备PMS和主服务发现服务器链路中断
  2. 备PMS感知到服务发现节点状态变化(services/domain-moid)
  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)
  4. 将主平台域moid加入到托管列表，广播状态

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. 对外接口
2. 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  master\_plamforms: // 主平台域  [  {moid:”123”,http:”0.0.0.0”,state:0},  {moid:”456”,http:”0.0.0.0”, state:1}  ],  slave\_plamforms: // 备平台域  [  {moid:”123”,http:”0.0.0.0”, state:0}  ]  } |

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  master\_plamforms: // 主平台域  [  {moid:”123”,http:”0.0.0.0”,state:0},  {moid:”456”,http:”0.0.0.0”, state:1}  ],  slave\_plamforms: // 备平台域  [  {moid:”123”,http:”0.0.0.0”, state:0}  ]  } |

1. 平台域机房状态广播

接口: monitor.notify.ex monitor.server\_room.k

类型: 广播

服务对象: 平台域业务

广播内容(json)

|  |
| --- |
| {  msg:” SERVER\_ROOM\_STATE\_NOTIFY”,  moid:”123456789”, // 平台域  rooms: //  [  {moid:”123”,state:0,type:0}, // type 0=主 1=副  {moid:”456”,state:1,type:1}  ]  } |

1. 平台域机房状态查询

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:”SERVER\_ROOM\_STATE\_REQ”,  moid:”123456789” // 平台域  } |

回复内容(json)

|  |
| --- |
| {  msg:” SERVER\_ROOM\_CONFIG\_NOTIFY”,  moid:”123456789”, // 平台域  rooms: //  [  {moid:”123”,state:0,type:0}, // type 0=主 1=副  {moid:”456”,state:1,type:1}  ]  } |

1. 平台域公共信息查询

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\_CONFIG\_REQ”,  moid:”123456789”  } |

回复内容(json)

|  |
| --- |
| {  msg:” PLATFORM\_CONFIG\_NOTIFY”,  moid:”123456789”,  MQ:{ IP:””,PORT:””},  Redis: // 为每个平台域分配的Redis  [  {moid:””,ip:””,port:””,db:””},  {moid:””,ip:””,port:””,db:””}  ]  } |