资源混用智能占用

1. 概述
2. 背景
   1. 会议能力能由上层自定义控制，满足各种用户场景。
   2. 会议能力包含主流、双流、音频、会议码率。
   3. 平台需要支持多类型外设资源的混用。
   4. 终端接收能力希望做到不同码率不同接收。
3. 引入问题
   1. 多资源环境下，如何选择外设。
   2. 适配、合成编码参数以及路数如何确认。
   3. 终端及老模版创会的兼容问题（只下最大能力）
4. 基础概念
   1. Vmp/Bas编解码能力固定，通道数量有上限，但远远大于业务需求。
   2. 能编H265的不一定能编H264，HP和BP同理，不过一般芯片都兼容。
   3. 目前科达终端的解码能力，都是支持向下兼容的，H265支持H264BP/HP。
   4. 自定义模版：会议能力参数完全自定义，外设编码完全按照用户配置。
   5. 全适配模版：会议能力只下最大能力，其它外设编码由平台自行计算。
5. 能力描述



* 1. 解码能力一般都固定，编码能力由多个芯片组一起完成。
  2. Vicp能力以1080p60为基准，能力值为长x宽x帧率，不区分视频格式。
  3. 单个芯片的能力可以**拆分**，多个芯片的能力无法**组合**。
  4. X86的媒体服务器，如果每路输出固定，可以将其看成多个芯片的组合。如果每路不固定，总能力固定，则看成单芯片。
  5. Xmpu为12个Netra，1个Netra为3个Vicp，Vicp为编解码共同占用。一个Bas/Vmp占用3个Netra。
  6. 详细Vicp能力见附录。Vicp只是用来抽象媒体能力的单位，不对应任何物理形态。

1. 逻辑预设
   1. 当媒体能力超过外设能力，不进行自动归档，直接拒绝。
   2. 为了保证适配能力，4k2k/1080p60适配不出，由业务做特殊策略让其收最合适的码流。
2. 上层设计
3. 媒体能力归档表



* 1. 媒体能力归档表，是一个抽象的表，用于表示全适配模式下，平台外设输出码流的规则，会管和CSS可以通过该表的描述处理自身逻辑。
  2. 平台存储从H265 4K2K到 H264BP1080P30的全适配媒体归档。（存储方式、获取方式由CSS决定）
  3. 媒体归档表中，码率是该档位的最小码率，当会议码率低于该码率，该档为无效档位。
  4. 可选格式表示，多种格式中，只有一种格式可用，需要根据会议码率进行选择。

1. 界面呈现



* 1. 当用户添加一种格式，勾选了全适配，会管根据媒体能力归档表，进行显示。
  2. 可选格式，需要根据用户当前的码率进行隐藏。
  3. 会管需要预判不同格式下的最低码率，避免用户勾选出现低码率高分辨率的情况。
  4. CSS遇到低码率高分辨率的情况，自动拒绝。
  5. 当用户编辑该模式的时候，会自动变成自定义模版。自定义模版情况下，隐藏的可选格式删除。

1. CSS逻辑



* 1. 创会接口中加入customizemedia，用于标识是否为自定义媒体能力。终端创会不带该字段，自动为全适配媒体能力。
  2. 裁剪 = 比较所有能力，能力大于会议码率的去掉。
  3. 过滤 = 过滤可选格式，保证只出一路。
  4. 过滤流程如下。



* 1. 最终到业务的会议能力参数均为固定能力值。

1. 业务设计
2. 流程调整



* 1. 业务不再根据外设型号进行下参。
  2. 合成的前适配逻辑调整到媒体对象创建成功后。
  3. MediaClient的能力匹配方式调整。

1. 适配逻辑
   1. MediaClient回复消息中，只有三种类型。 不支持（能力不匹配），完全支持（所有路数都能出），不完全支持（需要标识哪一路不支持）。
   2. 业务码流输出（4k2k 2.5m终端发言场景）
   3. Bas不输出的路数，降档（考虑实际场景中码率不同的终端少，降源会导致多数终端接收质量降低且级联场景下，降源会导致接收也降）



1. 合成逻辑
   1. MediaClient需要提供接口，可以查询创建好的外设的前适配逻辑。
   2. 合成归档，参照适配逻辑。
2. 能力匹配



* 1. 默认4K2K30/1080P60不编（后续外设媒体或特殊场景下放开），待能力比较成功后，直接回复业务，此种格式不支持，业务不管编或不编，都要下参。
  2. 算法从other->BP->HP->265,因为新的格式可以兼容老的格式。
  3. 分辨率计算的时候，从高到低匹配，找最接近的能力（比如当需要0.3时，用0.4而不用0.5，减少碎片）
  4. 此算法适用于主流/双流广播及合成。
  5. 装箱子算法。

1. 外设选择
   1. 当能力匹配后，以资源最优的外设进行选择，当前为负载均衡策略。
   2. 调整mediaclient，支持lua，动态配置分配策略。
   3. 新外设需要上报版本，不上报规格。
2. 外设逻辑
3. 外设能力表（如果有通道限制，可以在能力描述中加入通道数）

|  |  |  |
| --- | --- | --- |
| 外设类型 | 解码能力 | 编码能力 |
| Xmpu-Bas | H264 1080P60 | H264(1.5) Other(0.04)  H264(1.5) Other(0.04)  H264(1)  H264(1) |
| Xmpu-VMP | H264 1080P60 | H264(1+1+1) |
| Xmpu5-Bas | H265 4k2k | H265(1+0.5+0.11) H264(0.5) Other(0.04)  H265(1+0.5+0.11) H264(0.5) Other(0.04)  H265(0.5) H264(0.5)  H265(0.5) H264(0.5) |
| Xmpu5-Vmp | H265 4K2K | H265(2+2+0.22+0.22) H264(1) Other(0.04) |

1. MeidaResource逻辑
   1. 以最小资源占用通道，以句柄销毁作为回收资源的条件。
2. MediaControl逻辑
   1. 申请Bas/Vmp的接口支持设置是否选择兼容芯片，无该选项，当xmpu用完后，再次申请H264，无法在xmpu5上申请。
   2. H264下，XMPU5兼容XMPU的适配能力。
   3. 提供查询当前媒体能力的接口。
3. 兼容问题
4. 终端直接创会或获取模版创会。
   1. 会管的老模版中，媒体能力为数组，老终端只会取数组第一个，所以当前方案不影响老终端。
   2. 创会接口中，老终端不会携带customizemedia字段，默认为全适配模版，只需要在数组中填写一个能力，双流、音频能力均不填写。
5. 附录
6. 各分辨率Vicp理论能力

|  |
| --- |
| 1.VICP实际可用资源按VICP理论可用资源的100%评估。 |
| 2.1个VICP的理论能力计算方法：1920\*1080\*60=124416000 |
| 3.1080P60fps占用的VICP理论值：1920\*1080\*60/1920\*1080\*60=1 |
| 4.1080P30fps格式占用的VICP理论值：1920\*1080\*30/1920\*1080\*60=0.5 |
| 5.720P60fps格式占用的VICP理论值：1280\*720\*60/1920\*1080\*60=0.444 |
| 6.720P30fps格式占用的VICP理论值：1280\*720\*60/1920\*1080\*60=0.222 |
| 7.4CIF25fps格式占用的VICP理论值：704\*576\*25/1920\*1080\*60=0.081 |
| 8.CIF25fps格式占用的VICP理论值：352\*288\*25/1920\*1080\*60=0.02 |
| 9.UXGA60fps格式占用的VICP理论值：1600\*1200\*60/1920\*1080\*60=0.926 |
| 10.SXGA60fps格式占用的VICP理论值：1280\*1024\*60/1920\*1080\*60=0.632 |
| 11.XGA5fps格式占用的VICP理论值：1024\*768\*5/1920\*1080\*60=0.032 |
| 12.SXGA60fps 格式占用的VICP理论值:0.632 |
| 13.SXGA30fps 格式占用的VICP理论值:0.316 |
| 14.SXGA15fps格式占用的VICP理论值： 0.158 |
| 15.SXGA10fps格式占用的VICP理论值： 0.106 |
| 16.SXGA5fps格式占用的VICP理论值： 0.053 |
| 17.UXGA60fps格式占用的VICP理论值： 0.926 |
| 18.UXGA30fps格式占用的VICP理论值： 0.463 |
| 19.UXGA20fps格式占用的VICP理论值： 0.309 |
| 20.UXGA15fps格式占用的VICP理论值：0.232 |
| 21.UXGA10fps格式占用的VICP理论值： 0.154 |
| 22.UXGA5fps格式占用的VICP理论值： 0.077 |