密级：B

**云办公5.0-镜像模板编辑**

软件设计规格

本文档描述云办公5.0中镜像模板编辑的设计细节

仅供内部使用

|  |  |
| --- | --- |
| **主送对象** |  |
| **抄送对象** |  |

评审者

|  |  |  |
| --- | --- | --- |
| 部门/专业组 | 评审者 | 评审日期 |
|  |  |  |
|  |  |  |
|  |  |  |

[蓝色字体使用时删除]

1. 文档提交评审时需要给评审者预留足够的评审时间。需要哪些人评审依据项目的范围由项目团队界定。
2. 在本文档的撰写过程中，需要与相关利益人进行沟通交流（请勿到开始评审时才与相关利益人进行交流，这样效率低，而且评审效果也比较差），请将撰写过程中交流人员的名单列在附件里面。

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改者 | 修改描述 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[目录 2](#_Toc233699962)

[1. 问题定义 3](#_Toc233699963)

[2. 设计考虑 3](#_Toc233699964)

[3. 主要结构 3](#_Toc233699965)

[4. 系统流 3](#_Toc233699966)

[5. 数据结构 4](#_Toc233699967)

[6. 算法描述 4](#_Toc233699968)

[7. 接口设计 4](#_Toc233699969)

[8. 最终用户接口 4](#_Toc233699970)

[9. 软件限制与约束 5](#_Toc233699971)

[10. 固件限制与约束 5](#_Toc233699972)

[11. 硬件限制与约束 5](#_Toc233699973)

[12. 外部限制和配置 5](#_Toc233699974)

[13. 源代码 5](#_Toc233699975)

[14. 开发单元测试 6](#_Toc233699976)

[15. 专利考虑 6](#_Toc233699977)

[16. 参考资料 6](#_Toc233699978)

[17. 术语表 6](#_Toc233699979)

[18. 附件 6](#_Toc233699980)

[18.1. 评审活动记录 6](#_Toc233699981)

[18.2. 需求可跟踪考虑 7](#_Toc233699982)

[18.3. 本文档协助人员 7](#_Toc233699983)

# 问题定义

## 目标和范围

本文档描述如何在RCDC中进行镜像模板编辑，并使用编辑后的模板创建和启动虚机。

本文档中所描述的镜像中的操作系统均为Windows操作系统，对于Win7和Win10的部分版本，还支持应用分层分发以及Windows正版授权的功能。

对于之前文档中未明确描述的虚机与虚拟网络的关系，也在本文档中进行详细描述。

## 名词解释

|  |  |
| --- | --- |
| 名词 | 描述 |
| 镜像  和  模板 | 镜像和模板在本文档中是同义词。  表示一个已经完成初始配置的、完成应用程序安装的、可直接使用的操作系统的C盘内容，可以理解为一个只包含C盘的虚拟的硬盘。  使用镜像可以派生出多个桌面，从技术角度理解桌面可以当成虚机，从使用角度理解镜像派生桌面的过程可以认为是复制一个C盘，然后将复制后C盘安装到PC中，启动PC的并使用过程。 |
| 镜像和  还原点 | 还原点是镜像的技术概念，后文会详细描述 |
| 虚机  和  桌面 | 桌面是我司为云桌面产品包装的一个商业层面的名词，借助这个名词可以实现Desktop As A Service的商业理念。  从RCDC涉及的技术角度来看，桌面的底层技术实现就是一台KVM虚机，所以在本文档中虚机和桌面是同一个概念。  涉及用户呈现的内容，均使用“桌面”这个名词。  在本文档中，会根据上下文以及语言描述的顺畅程度，混用上述两个名词，读者应注意上述两个名词是同一个含义。 |

## 场景和商业需求

### 场景1：管理操作系统镜像

#### 导入镜像模板

RCDC出厂时没有携带任何客户操作系统，所以首次配置时需要创建操作系统镜像，结束镜像编辑后，用户才可以使用系统。

根据旧版商业需求有如下三种制作镜像的方式：

* 方式1：使用操作系统安装程序（ISO）安装
* 方式2：导入已有的镜像文件
* 方式3：复制已有的镜像

不论使用上述哪种方式进行镜像制作，最终都需要得到一个立即可用的、可以用来派生桌面的镜像模板。

#### 编辑镜像模板

镜像使用一段时间后，通常需要进行一些维护工作，常见的维护工作为：安装新的操作系统补丁、统一安装或更新应用程序。

上述需求要求新更新的补丁、安装或更新的应用程序能立即在所有已派生出的桌面中生效。

出于技术原因，仅有部分操作系统支持上述能力，对于支持上述能力的操作系统称为支持应用集中分发的操作系统，对应镜像模板称为支持应用集中分发的镜像。

#### 其他功能

在RCDC中需要能根据一定条件查询镜像模板，并对镜像模板进行删除，统计和关联查看镜像模板相关的桌面等功能。

上述功能作为商业需求不会一一描述，但作为软件来说是必须功能，会在对应SRS文件中进行描述，本章节不再赘述。

### 场景2：正版Windows激活

对于部分版本的Win7和Win10系统，我司与微软已签订协议可使用开源KMS服务器进行正版激活，云办公5.0中目前已安排对应授权导入和KMS服务器状态管理的功能，本文档中需要实现桌面运行情况下的正版Windows激活功能。

### 场景3：桌面和虚拟网络的关系

云办公5.0中《虚拟网络管理》的详细设计中已经明确如何创建和维护虚拟网络，对于虚机和虚拟网络的关系在本文档中进行描述，包括如何为虚机分配IP地址、如何修改虚机的IP地址等功能。从技术角度来看上述功能会涉及IP池中的IP分配和回收等能力。

# 设计考虑

【帮助1】描述软件设计的逻辑。讲解主要的因素，以及设计软件结构时考虑的设计目的。包括性能考虑，可用性目的，设计折中，以及支持有效的软件测试的软件接口和用户接口。包括当错误发生时，能在内存中保留信息或者提供跟踪手段，以及相应的调试考虑（比如debug命令以及选项等）。

如果是对现有产品的代码进行重写，需要说明为什么现有的代码不符合需求。

评估该功能的复杂性以及与其它软件功能点的交互关系。

【帮助2】

* 设计考虑
  + 功能设计考虑
    - 除按已有的标准、规范或创新自主的标准与规范进行功能设计实现的考虑外，还应评估是否有受影响的功能模块，比如：是否还有其他功能或模块会操作与控制与本设计所涉及的相同的资源，相互间如何进行相容设计。
  + 质量属性设计考虑
    - 可用性设计考虑，包括容错考虑，处理可能的故障，避免故障使整个系统失效，比如：内存不足、硬件表项容量用满时，应如何处理。以及对出错返回应做何处理。
      * 应针对上一级设计分解的模块可用性场景，通常是各种异常场景，分析说明设计对策：
        + 如何检测故障或异常；应明晰故障或异常判断条件。
        + 检测到故障或异常后要如何恢复；
        + 如果无法恢复，如何保障其他模块与功能不受影响。
    - 性能（包括容量）设计考虑：线程优先级，对高优先级的线程（如核心态线程）需要分析原因，线程、中断运行时间预算与系统响应速度、内存与Cache使用效率及与性能的权衡、带宽使用、数据产消模型的处理与调度策略的应用，等。
      * 比如：对于网管如SNMP/MIB的性能与容量需求，要考虑在性能与容量极限情况下的设计实现，不会对系统对其他事件的响应速度产生负面影响。
    - 可修改性设计考虑：
      * 如何应对将来可能的变化，如功能优化升级、增加、删除；关键器件升级；关联模块或组件变化；等，带来的兼容性问题，以及降低为适应变化所作设计修改的工作量。
      * 如何适应不同产品的需求，快速方便的完成版本定制。如A协议首次实现在86等高端设备，后续可能需要工作在S23等低端接入设备，这几类的硬件平台的处理器与内存资源差异比较大，设计提供何种手段以及算法以适应等。
      * 如何适应不同使用场景的需求，快速方便的完成上点部署。
    - 易用性设计考虑，如何帮助用户理解并操作系统，比如：一些对用户使用的限制与约束，除了写在产品相关手册上以外，能否通过系统软件的实现，防止用户掉入“陷阱”，帮助用户完成其想完成的任务；可能的话，对用户的错误操作，如何降低负面影响；还有其他如，对于需要较长时间才能完成的操作，如何让用户确信，用户的操作正在被忠实的执行中。
    - 安全性设计考虑（如果需要），考虑如何抗攻击；检测、识别与记录攻击；并能够从攻击中恢复。
    - 可测试性设计考虑：设计提供测试手段（统计信息、内部状态信息记录等，以及输入/输出方法），方便测试验证功能与质量属性的需求是否得到满足。还有为可能的单元测试提供手段。
  + 设计规则：今后的扩展与修改应遵循的规则，应遵循系统现有的哪些规则，为遵循系统现有的规则，设计应做何考虑。
  + 可能的Errata分析与设计考虑
* 解决方案评估与选择
  + 在上述设计考虑中，可能存在多种解决方案，为评估与选择方案，需要明确选择标准。下表仅为示例，决策评审要素供参考。
  + 标准优先级与权重的确定，主要受四个要素的影响：产品需求、组织目标、技术条件与组织能力。
  + 评分建议：0，不满足；5：满足；10：有优势。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 决策评审要素 | 优先级或权重 | 方案1评估结果或得分 | 方案2评估结果或得分 | 方案3评估结果或得分 |
| 1 | 满足功能需求（功能需求） |  |  |  |  |
| 2 | 开发难度  （技术条件与组织能力） |  |  |  |  |
| 3 | 工作量  （组织目标，如上市时间等） |  |  |  |  |
| 4 | 可用性  （质量属性需求） |  |  |  |  |
| 5 | 性能与容量  （质量属性需求） |  |  |  |  |
| 6 | 可修改性  （质量属性需求） |  |  |  |  |
| 7 | 易用性  （质量属性需求） |  |  |  |  |
| 8 | 可测试性  （质量属性需求） |  |  |  |  |
| 9 | 安全性  （质量属性需求） |  |  |  |  |
| 10 | 其他… |  |  |  |  |
| 11 | 总计 |  |  |  |  |

【裁剪说明】内存/性能/兼容性的设计考虑不可裁剪，如果不用考虑内存/性能/兼容性等因素，也要明确说明为什么不用考虑。

<正文>

## 桌面（虚机）和还原点（快照）

### 概述

在桌面云的商业场景中，用户的桌面是根据镜像派生出来的，当用户桌面删除时，仅丢失用户自身的数据，对原有镜像是没有后影响。

上述需求会引出一个称为“还原点”的抽象概念，即桌面云场景的镜像实际上是一个磁盘的还原点，用户的所有业务操作基于还原点进行操作，如果需要进行桌面还原，对应的动作是丢弃用户所有的个性化操作，从原有还原点重新派生新的系统磁盘。

### 桌面和还原点的关系图

#### 图示



#### 说明

从上图可以看出，RCDC中的镜像是每个还原点，每个还原点经过软件更新和操作系统更新后可以生成新的还原点，新旧还原点都可以派生桌面，不同还原点派生的桌面可并存，不受影响。

RCDC中用于派生新桌面的镜像模板是还原点B，还原点A作为镜像模板编辑的历史记录进行呈现，可以给用户呈现出在整个镜像模板的生命周期中修订了多少次，每次修订都包含哪些内容。

#### 还原点的删除和合并

当某个还原点中没有磁盘或虚机关联时，可以删除，这里的删除不是直接删除还原点，而是将其修改合并到最近的一个还原点中（上图中就是删除A时，将A和B的修改合并），这样可以减少虚机磁盘内部层次，减少磁盘块索引的搜索深度，有利于提升系统性能。

#### 业务限制

从上图可以看出，还原点A实际上还可以继续派生出新的还原点C，但从RCDC自身业务中来看，派生还原点C对于还原点A的管理复杂度会增加很多，所以不使用此功能。

对于基于模板复制新模板的业务来说，HCI层可以从还原点A快速开启虚机，制作还原点C，无需完整复制所有磁盘数据，但这样做会让管理的复杂度变的很高，所以复制模板还是沿用复制所有数据的方案。

### 还原点的实现方案

#### 旧版实现

旧版没有HCI存储池的概念，所有虚机的磁盘实际上是文件系统中的一个文件，通过母盘+差分的方式实现桌面的快速派生和还原。

这里的母盘实际上就是还原点，差分是用户的个性化操作内容在磁盘上的体现，但旧版中并未完全实现上图中的还原点链式结构（技术上可以，但是旧版中云管和桌管业务尚未分离，控制起来非常复杂，所以未实现）。

#### 期望的标准实现

目前业界均以快照这个概念实现还原点，配合HCI层的存储池支持，将磁盘封装为块设备暴露给虚拟化业务。

使用快照方式可以完全实现上图中的还原点链式结构，属于标准实现。

#### HCI层的设计考虑

RCDC仍然提供的概念是镜像模板，HCI层进行封装，可以支持HCI的旧版实现方式，也可以支持HCI的快照实现方式，此差异由HCI适配层进行处理，RCDC的业务无需关注。

### 应用分层和还原点

#### 概述

应用分层主要解决传统还原点方式在统一补丁安装和统一应用软件安装、配置升级方面的局限性。

#### 还原点的局限性

从上图中可以看出虽然还原点B中更新了软件，但已经由还原点A派生的桌面1、2、3无法感知到还原点B中管理员对操作系统做的软件更新和补丁更新等更新的内容。

这个问题在旧体系中变得尤为严重，并且因为此技术限制导致了很多业务上的限制，例如非还原镜像不能编辑之类的限制。

#### 应用分层技术的优势

应用分层技术将两个独立的磁盘通过Windows驱动技术整合成一个逻辑的磁盘，其中一个磁盘作为系统磁盘层（只读）、另一个磁盘作为用户个性应用层（读写）。

每次虚机启动时可以根据业务需要选用系统磁盘层和用户个性应用磁盘层作为用户桌面的系统盘。

上述技术可以解决还原点的局限性，当管理员编辑镜像（创建还原点B）时，如果用户的桌面仍然处于运行状态，则使用还原点A的数据，一旦桌面关闭后，RCDC可以调度将对应虚机的系统磁盘从还原点A切换到还原点B，下次启动后就可以应用到系统管理员对操作系统的统一更新。

#### 应用分层桌面的使用方法

RCDC调度应用分层的桌面时，除了要基于快照派生一个新磁盘外，还需要一个应用分层的磁盘，此磁盘对业务不可见，RCDC内部调度时可见。

#### 还原桌面和非还原桌面

还原桌面和非还原桌面是用户桌面的属性，是业务层面的约束和属性，考虑到还原点和应用分层的技术差异，这里列举出两种技术方案对两种桌面属性的支持情况。

|  |  |  |
| --- | --- | --- |
|  | 还原桌面 | 非还原桌面 |
| 传统还原点 | 关机后根据新的还原点为已经创建的虚机重建系统盘 | * 当系统管理员完成新镜像模板制作后，为已经派生的用户桌面发消息，明确说明应用新模板会导致系统盘个人数据全部丢失的问题，并由用户选择是否进行更新； * 为了避免骚扰用户，上述提示消息需要让用户选择“切换时间”或者永远不切换的选项。 |
| 应用分层 | 还原桌面不采用应用分层模式开启 | * 当系统管理员完成新镜像制作后，为已经派生的用户桌面发消息，引导用户进行切换，并明确说明切换后会导致的安装在系统盘中少数应用不可用、少数操作系统配置会丢失的问题，并由用户选择是否进行更新； * 为了避免骚扰用户，上述提示消息需要让用户选择“切换时间”或者永远不切换的选项； * 目前应用分层技术尚处于完善阶段，等完全完善后提示信息可以减少，引导用户安全的完成应用分层切换。 |

#### 桌面是否开启应用分层

镜像模板中需要记录是否支持应用分层的属性，管理员在创建用户桌面时需要选择是否使用应用分层特性，并且可以由管理员进行修改。

应用分层开启与否的修改会涉及到虚机磁盘的变更，必然涉及用户个性化数据的丢失，所以此操作必须进行二次确认。

当前版本的规划中不对这种切换进行数据安全性支撑，所以需要在页面交互上面引导用户尽量要做这个操作，并写明免责声明，同时需要引导用户和我司售后或渠道反馈切换的原因，以便于我司在未来版本中能给出更好的解决方案辅助解决用户的问题。

### 镜像和还原点的维护

管理员每次编辑镜像成功后，对于HCI层来说会建立一个新的还原点。

从HCI层来说，只要某个历史的还原点中仍然与虚机关联，则一定不能删除此还原点的记录。

所以随着系统的使用，RCDC中的一个镜像会对应HCI中一个还原点的链条，根据镜像新派生的桌面始终使用最新的还原点，而以前派生出的桌面使用历史还原点。

RCDC的业务层无需保留还原点和桌面的关系，这个关系应该维护在HCI中，但HCI适配层出于记录和故障排查的便利性，需要记录还原点和具体桌面的关系，即需要记录业务层的镜像模板的还原点链条和链条每个节点与桌面的关系。

## 镜像管理简述

### 概述

镜像管理包含镜像的创建、修改、使用和删除等业务流程。

### 创建镜像

#### 概述

系统支持三种途径创建镜像，分别为：

* 途径1：通过操作系统安装包（ISO）文件制作镜像；
* 途径2：导入预制镜像文件创建镜像；
* 途径3：复制现有镜像创建新镜像。

上述三种途径的主业务流程类似，本章节将统一考虑相关技术问题和业务问题。

#### 途径1：通过操作系统安装包（ISO）文件制作镜像



#### 途径2：导入预制镜像文件创建镜像



#### 途径3：复制现有镜像创建新镜像



#### 总结

从上述三个流程图可以看出，创建镜像的主流程都是经历文件导入、开启虚机编辑、关机虚机生成镜像的步骤。

### 编辑镜像

#### 流程图



#### 流程说明

编辑镜像的核心流程与创建对象基本类似，唯一区别在于无需复制原有镜像，直接使用现有镜像启动编辑镜像用的虚机即可。

### 删除镜像

如果镜像正在使用中（有从镜像派生的桌面），则不能删除镜像。

如果执行过复制镜像的操作，HCI层可能使用快照方式实现复制镜像的技术支持，此时HCI层中应该不能删除原始镜像，但从RCDC角度来看不关注HCI层的实现，所以HCI层提供的接口需要处理此场景，返回删除成功，但自行维护内部数据的有效性关系。

### 导出镜像

旧版中有导出镜像的功能，考虑到本功能非刚需，所以5.0中暂不实现此功能，留到5.1或以后版本实现。

## 镜像管理关键设计决策

### GuestTool的架构角色

#### 旧版方案

在云课堂考试场景下，国家对考试使用的计算机有软硬件的强制要求，涉及到软件中要求除指定版本操作系统、考试必须软件之外，不能有任何其他第三方软件，所以在此场景下不能安装GuestTool。

目前云办公场景应对的是非保密的行业，更多的是职能人员、门店的简单办公PC场景，在上述场景下客户并不关心是否有其他第三方软件。

#### RCOS 9.0的方案

RCOS 9.0中，在GuestTool的参与下可以实现如下管理能力：

* 虚机关闭的通知
* 在用户可控的范围内让管理员进行远程协助
* 向用户发送自定义消息
* VDI驱动的安装和管理

未来可以实现的管理能力：

* 在线修改虚机的IP地址
* Windows正版授权能力
* 应用分层的驱动安装和辅助支持

如果没有GuestTool则无法实现上述能力，云桌面管理非常受限。

从我司云办公目前商业场景看，还没有进入军队、保密等政府行业的场景，如果上述场景有第三方软件限制要求，再根据实际需求进行设计。

#### 结论

GuestTool为镜像中的刚需，没有安装GuestTool的镜像属于不完整的镜像，不允许根据此类镜像派生桌面。

### 状态变迁和状态管理

#### 状态、状态内部的阶段和状态属性

镜像模板的编辑等状态均属于RPC耗时操作，需要采用“持久态”+“暂时态”的状态机思路进行设计和实现。

镜像模板最耗时的状态是编辑中，当镜像模板处于编辑中时，会经历虚机启动中、管理员编辑镜像内容、镜像保存等内部子状态，这些子状态与“编辑中”的状态之间相比属于父子关系，所以本文档中将这些子状态称为编辑镜像的各个阶段。

同时，镜像必须安装GuestTool才可使用，这些属于又和子状态不同，将其称为状态属性。

一个镜像从编辑中变成可用状态，需要满足以下条件：

* 条件1：所有必须的状态属性必须都Ready（例如GuestTool必须成功安装或升级）；
* 条件2：状态内部的各个阶段的处理均完毕。

上述状态内部阶段和状态属性需要持久化记录，但又不是镜像模板需要长期维护的数据，所以在数据库设计阶段需要将其拆分到另一张表中记录，这张表专门记录镜像编辑过程中的阶段和状态属性的满足情况。

只有当镜像编辑完成后，会将这张表中的某些属性（例如是否支持应用分层）原子的写入镜像模板的主表中，并删除这张镜像编辑过程专用表的对应数据记录。

#### 镜像模板的状态转换



#### 创建中和编辑中的各个阶段



#### 影响镜像模板发布的属性

GuestTool是必须的，所以没有安装GuestTool或者GuestTool与RCDC未交换过数据的镜像模板都认为是不能发布的。

### 镜像编辑过程中是否可继续派生桌面

#### 技术支持情况分析

在课堂场景中，下课时间和周末通常没有上课，所以可以放心的编辑镜像并应用镜像。

在办公场景中，很可能办公桌面一直处于开机状态，同时在日常的镜像内系统补丁维护过程中也会有桌面创建需求，所以不能沿用旧版本业务。

前文已经描述了镜像和还原点的关系，从中可知，镜像如果处于编辑中，桌面可以从之前一个已经编辑完成的还原点派生，不产生冲突。

只有首次导入或编辑镜像时会产生上述冲突，此时镜像应该不可用，所以业务层面根据状态进行控制即可。

#### 当前版本的范围和边界

当前版本因为进度考虑，主要交互沿用旧体系的交互，如下所示：

* 镜像编辑过程中不允许继续派生桌面
* 镜像编辑前关闭所有还原桌面，镜像编辑完成后重建所有还原桌面
* 镜像编辑完成后所有非还原桌面可以看到升级通知，由桌面使用者决定是否升级
* 管理员可以看到镜像编辑的历史记录

### 镜像编辑中途是否可放弃

镜像编辑过程不影响桌面的派生，所以是否放弃对于业务没有影响，只对数据一致性和资源回收有影响。

创建镜像过程中，如果中途放弃，则删除对应镜像模板所有数据和资源；

编辑镜像过程中（非首次创建），如果中途放弃，则镜像保持原有还原点的数据，编辑过程中产生的所有数据和资源都要回收；

这里说的资源包含但不限于虚机、磁盘、IP，如果编码阶段识别除了其他资源，也一样要进行释放。

### 编辑镜像过程如何联网

#### 旧版方案

从DHCP池获取或者用户在编辑过程中手动配置，没有特别考虑过这个功能，同时IP地址用完即回收，不存在长期占用的问题。

#### RCOS 9.0的方案

泽塔不支持直接的编辑镜像的功能，如果需要得到一个泽塔概念中的镜像，需要创建虚机，创建虚机的时候会分配IP地址，所以补丁更新过程可使用虚机所在的IP地址。

同时，RCDC中的镜像模板映射到的不是泽塔中的镜像或虚机，而是虚机的快照，所以虚机会长期占用一个IP地址，这和旧版是完全不同的。

#### 结论

编辑镜像过程中可以临时占用一个IP地址，但不应该长期占用，所以泽塔超融合平台需要对这个需求提供支持。

编辑镜像过程中的IP地址可以由管理员指定，也可以从现有IP地址池中获取，只要IP没有被其他虚机占用，并且在RCDC管理的虚拟网络范围内即可（本需求与虚拟网络管理有关系，即在镜像编辑阶段如果IP被占用，则对应虚拟网络的删除、修改需要进行对应的业务校验，防止IP托管）。

编辑镜像过程中需要超融合平台为对应虚机分配一张网卡，网卡的IP地址如上文描述，在安装操作系统过程中是不会体现IP地址的，当安装完GuestTool后，GuestTool需要和RCDC进行首次通信，获得当前虚机的IP地址，并且向RCDC传递是否支持Windows正版授权和是否支持应用分层等特性，辅助完成其他管理业务。

### 编辑镜像和用户桌面的网络复用

编辑镜像过程管理员可以任意选择一个用户网络进行编辑，如果没有选择RCDC将随机选择一个没有被占用的IP地址进行镜像编辑。

在《虚拟网络管理》的详细设计文档中根据业务需要对于HCI适配层的虚拟网络适配方案以及HCI层的网络API进行了重新设计，所以编辑镜像和用户桌面的网络分配中传递的网络ID在HCI适配层中需要映射为《虚拟网络管理》文档中描述的新网络ID，这个切换过程会在本文档中详细描述。

### 编辑镜像过程中GuestTool的安装和升级如何进行

#### 旧版方案

当前版本的云课堂/云办公中是通过将GuestTool的ISO挂载为CD-ROM的方式启动虚机，在虚机中进行GuestTool的安装和升级。

这样涉及到在启动虚机过程中需要挂载两个CD-ROM，一个作为启动盘，安装操作系统使用，另一个是GuestTool自身的ISO。

#### RCOS 9.0的方案

RCOS 9.0 沿用就把的方案。在RCOS 9.0中，我司已经与超融合平台（泽塔）协商好了GuestTool.iso的文件名和位置，并且也协商好了更新GuestTool的HTTP通信接口，同时也协商好每次虚机启动时都将GuestTool.iso作为CD-ROM挂在给虚机（不论是用户模式还是镜像编辑模式）。

#### 结论

在RCOS 9.0中的上述约定确保了GuestTool可以正常的升级和更新，云办公5.0中保持现状，只是使用ISO启动虚机进行镜像编辑时，要求超融合平台可以同时挂载操作系统安装的ISO和GuestTool的ISO，并且将操作系统的ISO设置为启动盘。

上述需求属于通用需求，不论超融合平台是泽塔还是RCCP，都需要这样启动才能让用户无感知的使用云桌面。

如果未来RCDC对接的某个HCI供应商未直接提供上述能力，至少HCI层提供的启动虚机接口需要传递一个或多个ISO文件名或路径，并且支持指定某个ISO文件为系统盘。

如果虚机中已经存在GuestTool，则可以由GuestTool自动检测版本更新情况并自动升级，具体升级方式（交互升级或静默升级）属于GuestTool的易用性需求，不在本文档中深入讨论。

### 编辑镜像过程中的应用程序安装包如何管理

#### 旧版方案

旧版云办公/云课堂中有应用程序管理的功能模块，在对应功能模块中可以上传应用软件安装包，当虚机在编辑模式启动时，会将上述安装包通过技术手段以磁盘的方式挂载给虚机，但磁盘中的数据是只读的（目前通过擦除编辑过程中的写操作来实现只读，即在编辑过程中可修改对应磁盘中的内容，但编辑模式虚机关机后所有修改被丢弃），除此之外，此方案还带来了镜像编辑过程中无法更新应用程序或者无法实时看到已更新应用程序的信息。

前期旧版尝试过使用CD-ROM挂载应用程序的方案，虽然能实现100%只读，但是在镜像编辑的过程中无法更新应用程序。

#### RCOS 9.0的方案

上两种方案都存在或多或少的弊端，RCOS 9.0中采用samba方式暴露应用程序的数据，通过GuestTool创建虚拟盘的方式向用户呈现需要安装的应用程序列表，支持列表实时更新，支持写权限控制。

#### 结论

以我司目前技术水平来看，RCOS 9.0选用的方案属于弊端最少，易用性、可用性均最好的方案，所以云办公5.0沿用上述方案。

### 管理员如何登入制作镜像的虚机

#### 旧版方案

旧版采用AdminTool登入虚机，AdminTool实际上是一个Spice Client，制作镜像的虚机会暴露一个Spice Port传输虚机中的画面。

#### RCOS 9.0的方案

RCOS 9.0中在泽塔平台使用VNC方式登入虚机，只要浏览器支持websocket即可，无需单独安装客户端。

#### 方案分析

采用VNC远程登入虚机属于事实上的业界规范，但VNC在流媒体播放和视频等体验不如Spice协议。

不过管理员登入虚机的目标是配置和软件安装，而不是看视频、文字编辑等长期使用，所以Spice提供的高级能力反而用不到。

#### 结论

云办公5.0中抛弃旧版的AdminTool，转而采用VNC方式进行镜像编辑。

目前9.0中开发的远程协助属于用户使用过程中的远程协助，整个业务流程除了VNC通信之外还有一系列外围控制的逻辑。

在镜像编辑过程中，无需那么多的外围控制逻辑，只要RCDC从泽塔获取到VNC的URL后就可以通知前端开始建立VNC通信，RCDC和前端之间需要保持一些token之类的安全参数，确保每个远程协助的URL只能使用一次，并且是被正确的管理员使用。

### 编辑镜像过程中的Windows的正版授权

#### 旧版实现

旧版编辑镜像的过程的虚机会当做一个特殊虚机处理，不占用黄金镜像的license数量。

#### RCOS 9.0的实现

无此功能。

#### 结论

当前版本黄金镜像license仅控制开启和关闭，同时编辑镜像过程中GuestTool也会和RCDC进行交互，所以只要是GuestTool上报的正常的虚机信息，都允许进行激活（对于利旧客户端的激活问题，需要后续设计中留有扩展，不在云办公5.0中考虑，具体扩展在Windows正版授权的对应章节进行描述）。

### 编辑镜像的虚机配置

#### 旧版方案

目前旧版云办公、云课堂中制作镜像的虚机配置如下：



#### RCOS 9.0的方案

9.0中没有此功能，镜像是在泽塔上进行编辑。

#### 结论

##### CPU核数和内存大小的配置

编辑镜像开启虚机的CPU、内存的大小配置属于高级配置，通常不需要管理员或运维人员进行手动调整，所以需要像旧版本一样提供默认配置，允许用户进行手动修改。

默认配置参数：

* CPU：2核
* 内存：4GB

##### 磁盘大小的配置

使用ISO制作镜像属于首次制作，所以制作镜像的虚机磁盘大小固定为40G。

但导入已有镜像时，制作镜像的虚机磁盘大小实际上需要依赖镜像文件的大小，至少不能比它小，而且为了保存OS的软件安装、升级过程中的数据，磁盘大小应该比原始镜像文件大。

编辑镜像阶段系统盘大小默认值、最大值、最小值的计算公式如下：

|  |  |
| --- | --- |
| 类型 | 公式 |
| 默认大小 | Max（原始镜像文件大小 \* 150%, 40GB） |
| 最小大小 | Max（原始镜像文件大小 \* 200%, 100GB） |
| 最大大小 | Min（原始镜像文件大小 \* 120%, 40GB） |

##### CPU类型的配置

旧版支持host、SandyBridge和kvm64三种CPU类型，从当前商业需求来看可以满足，除非商业场景是卖给使用30年历史以上的医院等行业的旧体系HIS系统时，需要支持更早期的CPU架构以运行对应的操作系统或软件（大概是Win95、98这种旧系统）。

目前上述商业需求是否是刚需还不明确，而且CPU类型的选择对于RCDC来说仅是枚举字段的新增，主要工作量不在于RCDC，所以沿用旧版的三种CPU类型，如果确有需求再进行调整。

### 异常场景和数据一致性保障

#### 概述

镜像编辑属于长时间的虚机操作，操作过程中会遇到很多异常问题，有些可以预知、有些不能预知，本章节记录所有能想到的异常场景，根据每种异常场景进行分析和抽象，输出统一的方案进行处理，通过统一的方案可以解决已经想到的问题，并且也可以减少未预期的问题带来的影响。

|  |  |
| --- | --- |
| 问题编号 | 说明 |
| 问题1 | VNC窗口异常关闭或浏览器进程被杀 |
| 问题2 | 服务器掉电 |
| 问题3 | 无法获取到IP地址 |
| 问题4 | IP地址冲突 |
| 问题5 | GuestTool和RCDC的通信中断 |
| 问题6 | 镜像编辑结束关闭虚机后RCDC未正常收到通知 |

#### 问题1：VNC窗口异常关闭或浏览器进程被杀

浏览器或VNC窗口关闭从技术原理上分析不应该影响正在运行的虚机，所以虚机仍然处于正常运行的状态。

RCDC需要记录镜像状态为编辑中，并且允许在此状态下重建VNC连接。

注意：不论VNC从技术原理上是否支持同一个虚机同时开启多个VNC连接，从编辑镜像的业务角度考虑都必须关闭旧的VNC连接，所以在开启新VNC连接之前，如果发现原有VNC连接的记录是存在的，则需要调用HCI层的接口执行强制关闭。

#### 问题2：服务器掉电

##### 概述

服务器掉电不应该影响之前已经完成制作的还原点，但可能影响目前正在制作的虚机磁盘的完整性。

所以此异常情况引出了两个子问题。

##### 子问题1：服务器掉电后是否能继续进行镜像编辑

虚机启动过程中的数据不是实时落盘，所以掉电后能否继续启动虚机并进行镜像编辑也未知。

HCI层应提供功能在此场景下尝试启动虚机，如果无法启动，则通知RCDC上次编辑未保存的数据丢失，需要从新编辑。

##### 子问题2：服务器掉电是否影响RCDC和HCI层的数据一致性

RCDC层采用“记录中间状态”、“执行RPC操作”、“更新最终状态”的RPC场景最佳实践进行容错处理和设计，可以将数据一致性问题降低到最小程度，并且可以在一定时间内自动修复。

#### 问题3：无法获取到IP地址

首次使用ISO安装操作系统时，会没有IP地址，此时需要GuestTool读取ACPI文件获取RCDC的IP地址，并且与RCDC通信，获取真实IP地址。

这要求HCI层能在完成操作系统安装后，将ACPI写入GuestTool要求的位置。

#### 问题4：IP地址冲突

可能产生IP地址冲突的原因如下：

* 原因1：用户手动修改了操作系统中的IP地址；
* 原因2：RCDC分配的IP地址冲突；
* 原因3：手动在HCI上开启了虚机，RCDC无法感知到对应IP占用情况。

对于上述原因，解决方案如下：

* 原因1：如果IP地址是使用GuestTool设置和管理的，则需要GuestTool锁死IP地址配置功能，避免人为修改导致故障；
* 原因2：RCDC自身bug导致，完成bug修改即可；
* 原因3：在HCI上开启虚机本身就是违规操作，在现有云办公场景下不允许此问题发生，如果未来云办公在数据中心场景下部署，也需要有独立的VLAN或VxLAN网段，不能和其他网络或业务共用同一个网段。

#### 问题5：GuestTool和RCDC的通信中断

GuestTool和RCDC之间的通信中断可能原因如下：

* 原因1：Windows网络防火墙禁用GuestTool与RCDC的通信；
* 原因2：GuestTool进程异常退出或被强杀；
* 原因3：RCDC所在虚机与GuestTool之间的通信异常。

不论是上述哪种原因，都会导致镜像的某些状态无法记录或修改，所以一旦出现上述情况，认为镜像未完成编辑，不能投入使用。

RCDC与GuestTool之间的通信设计RCDC-->VMMessage-->GuestTool，上述三个通信节点之间通过IP网络和串口进行通信，如果通道不通可能是由于HCI自身IP网络调度或VmMessage进程故障导致，对应可靠性保证不是RCDC的业务，与首次部署和VMMessage自身的高可用保障有关，不在本文档中赘述。

#### 问题6：镜像编辑结束关闭虚机后RCDC未正常收到通知

本问题会导致RCDC中的状态与HCI层的虚机状态不一致，因为HCI层的管理页面不会暴露给管理员，所以管理员无法知道此时虚机到底是关闭还是没关闭，最稳妥的操作方式是重新开启虚机，再关闭。

如果每次关闭都无法收到通知，则是bug，不应该在客户现场才发现。

所以通常情况下重新开机，再关闭虚机结束编辑已经可以满足需求。

这样需求要求HCI适配层进行容错处理，可以识别出HCI层实际状态，并通过调用合适的HCI层API实现最终一致性的虚机状态保障。

## 编辑镜像的ISO和预制镜像的文件上传

### 需求分析

镜像模板管理的功能需要使用操作系统安装包（ISO文件）通过安装的方式构造镜像供桌面使用，也可以直接导入预先制作好的镜像文件（qcow2格式）的方式构造镜像。

上述两种方式都涉及web界面导入的功能，而且上述文件大小在4GB~10GB之间，属于超大文件，导入过程的主流程类似，可以进行一定的抽象减少重复代码的工作量。

### 旧版方案

#### 导入方案

采用ftp等第三方大文件上传工具上传超大文件，需要服务端开启特殊服务支持、客户端安装软件或插件支持。

#### ISO复用方案

旧版云办公/云课堂对于安装操作系统使用的ISO文件提供的文件管理的功能，上传的ISO文件会归档到对应功能中进行管理，并允许管理员查询和删除不需要的ISO文件。

### 新版期望带来的方案效果

|  |  |  |
| --- | --- | --- |
| 编号 | 描述 | 如何支持 |
| 效果1 | 无需依赖独立的第三方工具上传镜像文件 | 当前SkyEngine框架已支持超大文件分块上传能力，所以无需采用第三方工具支持。 |
| 效果2 | ISO文件的管理易用性加强 | 对于ISO文件根据使用的特性进行文件暂存、暂存文件的复用和过期文件的清理，无需管理员干预。 |

### 上传ISO文件的功能考虑

#### 如何将ISO从RCDC传递到超融合平台

RCDC自身运行在超融合平台提供的虚拟运行时环境中（KVM虚机或docker容器），超融合平台为其提供一定大小的磁盘。

ISO文件的目标是用来在虚机中安装操作系统，虚机管理的职责是超融合平台，所以必然要让超融合平台可以读取对应ISO文件内容，并将其挂载为虚机的一个包含启动能力的CD-ROM。

目前超融合平台（泽塔或我司）均采用存储池的概念管理所有磁盘，所以需要超融合平台提供一个功能，可以将安装操作系统的ISO文件拷贝到超融合平台自身存储池中。

上述功能涉及对超融合平台的接口需求：

* 上传ISO文件：将ISO文件流传递给超融合平台，并记录ISO文件和文件唯一标识的关联关系，实际设计时会拆分三个子接口，一个是开始上传、一个是上传内容、一个是上传结束；
* 删除ISO文件：根据文件唯一标识删除对应文件；
* 使用ISO文件启动虚机：根据文件唯一标识启动虚机

#### ISO文件的删除

RCDC不维护实际的ISO文件，最终文件由HCI层维护，由于和HCI层之间上传文件的设计已实现了调用幂等和重复文件剔除，所以删除ISO文件时，需要同时删除和ISO的MD5相同的基础组件中的上传文件记录，避免出现数据一致性问题。

### 上传预制模板的功能考虑

#### 如何将预制镜像从RCDC传递到超融合平台

本流程与ISO文件上传流程类似，设计HCI适配层对外接口时，需要根据业务独立设计一套复制预制镜像的接口，但HCI适配层的实际实现以泽塔提供的最终接口为准，我司提出的最优的接口需求也是独立设计的接口。

### 上传文件的数据一致性保障

文件上传属于RPC耗时操作，上传时如果中断可能出现数据一致性问题。

所以上传文件采用“暂态”-->RPC-->“终态”的设计思路进行开发，事务补偿阶段根据HCI层文件接收的实际状态进行状态修改或数据清理。

### 质量属性

#### 性能

大文件上传主要性能问题在于IO，需要先记录在我司本地，再复制给超融合平台，复制采用HTTP协议，从传输协议角度来看数据量都是那么大没有太多可优化的空间。

从易用性角度来看，要尽量提升前端的交互体验，就需要尽量多的将同步操作修改为异步操作。

上传ISO文件的流程可分为如下几个步骤：

* 步骤1：SkyEngine接收大文件；
* 步骤2：SkyEngine完成大文件接收，由云桌面管理组件将接收到的大文件剪切到暂存区；
* 步骤3：云桌面管理组件负责将大文件传输给超融合平台，并且处理传输过程中的出错问题。

上述步骤的分析如下：

* 步骤1：目前已经是分块上传，前端体验良好，无需改造；
* 步骤2：业务接收文件和剪切的过程伴随一系列业务校验，这个步骤如果做成异步反而体验会变差，所以沿用目前的同步处理方式；
* 步骤3：这个步骤涉及HTTP IO，性能比磁盘IO差，同时失败概率高，需要改造为异步传输的方式，从用户角度来看，上传ISO页面会近实时的呈现上传状态，并在上传成功后通知前端可以执行其他逻辑。

未来版本考虑将整个分块接收和拼装的业务都交给超融合平台实现，减少各步骤中临时缓存文件和读取缓存文件占用的IO时间。

#### 容量

因为ISO文件很大，并且目前常见的Windows系统版本为WinXp、Win7、Win10，每个版本都有x86和x64两种平台，即常见的ISO数量通常不超过10个。

所以上传ISO时业务层给出数量限制，如果当前系统中已上传的ISO数量达到10个，上传第11个时，会有选择的删除之前的ISO文件，删除规则为删除当前未使用（未开启虚机），并且最后使用时间最早的ISO文件。

同时，系统对于上传的ISO提供定时清理的能力，清理15天以前未使用的ISO文件。

#### 安全性

RCDC接收文件采用HTTPS，没有通信安全的问题。

RCDC向超融合平台传输文件时是内部IO，不涉及通信安全的问题。

向超融合平台传输文件属于RPC操作，传输文件的状态涉及传输中和传输结束，并且需要根据实际情况进行事务补偿（根据超融合平台实际状态进行判断，删除传输记录或者认为传输成功）。

RCDC默认所有向超融合平台传输成功的文件在RCDC未主动调用删除接口前都是可用的，如果违反此规则认为超融合平台有bug。

#### 可扩展性

未来可能扩展为分块直接由超融合平台接收，在状态机的设计和存储结构设计时需要考虑此扩展。

#### 易用性

本功能对于用户来说不直接可见，用户可体验到的是第二次上传同样ISO会很快，但不会为用户提供已上传ISO列表等功能。

## 镜像管理的质量属性

### 性能和容量

镜像管理属于复杂状态交互，涉及流程很长，性能的考虑主要是与前端的交互做的更优化，让用户感受不到长期无交互的等待。

镜像的实际数据在HCI层以快照方式体现，快照方式占用存储资源较少，但快照层次过多会带来性能问题，这要求HCI层可以定时清理或合并没有虚机关联，并且被RCDC标记为可清理状态的快照数据。

### 安全性和可靠性

对于RCDC自身来说主要是数据一致性，在之前章节已描述

对于使用者来说，最大的安全性是自身数据的安全性，对于应用分层和非应用分层场景下应用更新过的镜像模板数据时需要有风险揭示和用户确认，用户确认需要输入自身的登录密码，并且后台需要记录终端操作日志供审计用。

### 可扩展性

HCI层对业务暴露镜像模板的概念，在HCI层中以还原点的概念进行镜像模板的维护工作，可以适配不同的HCI层实现。

后续扩展为IDV镜像编辑时，目前能想到的最好方案仍然是在HCI层中进行编辑，然后将镜像导出到RCDC中，供IDV终端现在，但具体实现方案尚未详细推敲，在5.1中会详细考虑，最终可能使用本方案，也可能采用RCDC嵌套虚拟化方案实现。

### 易用性

在交互设计中体现。

## Windows正版授权

### 概述

本文档中描述Windows正版授权的业务，关于如何导入license，如何驱动license校验的逻辑在《授权导入》的详细设计中已经详细描述，本文档中不再赘述。

Windows正版授权的逻辑涉及Windows操作系统、GuestTool、RCDC和KMS服务四个子系统共同协作完成。

### 开启Windows正版授权

#### 流程图



#### 流程描述

导入license后，需要开启KMS服务，并通知所有处于运行中的桌面执行正版授权的激活动作。

### 关闭Windows正版授权

#### 流程图



#### 流程描述

关闭正版授权时需要关闭KMS服务，并且通知所有已经激活的桌面关闭激活功能，并设置为未激活状态。

### 云桌面启动

#### 流程图



#### 流程描述

云桌面启动时，GuestTool会和RCDC通信，进行客户端的初始化动作，此流程中同时返回是否允许进行Windows正版授权。

### 执行激活

#### 流程图



#### 流程描述

执行激活由GuestTool触发，由Windows进行业务细节处理，由RCDC判断是否允许进行激活。

### 质量属性

#### 性能和容量

对于RCDC来说，只是返回是否允许激活的状态，没有存储其他数据，不存在性能和容量问题。

#### 安全性和可靠性

整个方案没有安全性问题，KMS的可靠性由其自身守护进程保障。

## 应用分层方案带来的关键设计决策

### 桌面的系统盘大小

应用分层方案中，系统盘由系统分层和个性化应用分层组成，但因为系统盘的写入有些是COW的模式，有些不是，所以最终用户可用的系统盘大小比系统分层+个性化分层的总容量小。

应用分层驱动的开发人员也无法给出明确的系统盘大小计算公式，所以借鉴现有云办公、云课堂的系统盘大小计算方案：

最终系统盘大小 = 15GB（固定大小） + 个性化分层的大小。

### 应用分层驱动如何知道哪个磁盘是系统层、哪个磁盘是个性层

磁盘中需要有一些特殊标识来区分是系统层还是个性层，可以选的方案是磁盘序列号或者磁盘中的隐藏文件。

Libvirt可以为磁盘写入序列号，但块设备会受限，无法使用，所以只能选择隐藏文件的方式。

为了避免隐藏文件被用户篡改或删除，需要GuestTool做如下限制：

* 文件不能被用户看到；
* 用户不能创建同名文件进行覆盖；
* 用户不能在其他磁盘创建同名文件。

为了避免和用户文件冲突，文件名命名为：layer\_templete，文件内可以写入一个数字，这个数字表示当前分层的排序，数字越大优先级越高，目前只有一层，文件中的内容默认为1。

## RCDC和HCI层的接口考虑

### 概述

HCI适配层将RCDC对HCI层的技术需求通过接口方式进行封装，使用适配器模式实现不同HCI层的对接，对业务层提供一致的接口。

### 上传ISO文件

#### 业务描述

将RCDC虚机本地磁盘中保存的ISO文件上传到HCI的超融合平台中。

#### HCI层处理建议

ISO文件很大，如果同样的文件内容上传多份没有意义，而且ISO文件上传时间较长，上传途中出错可能导致HCI存储池出现垃圾文件或其他问题。

所以HCI层应该记录文件的MD5值，在上传之前根据MD5判断文件存在性，如果存在则无需再次上传。

同时如果已经有一个文件上传中，则相同MD5的文件再次申请上传时，HCI层应该返回“正在上传中”的提示，并且返回对应文件在超融合存储池中的ID或URL。

#### 对泽塔的接口需求

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 入参 | 处理方式 | 返回值 |
| 根据MD5查询ISO文件是否存在 | 文件MD5 | 根据文件MD5查询文件存在性 | HCI层的文件ID  文件状态：可用/上传中 |
| 上传ISO文件 | 文件内容  文件MD5 | 存储文件内容  生成HCI层的文件ID | HCI层的文件ID |

### 上传预制镜像文件

#### 业务描述

将RCDC虚机本地磁盘中保存的预制镜像上传到HCI的超融合平台中。

#### HCI层处理建议

预制镜像文件较大，如果上传中途失败需要HCI层清理垃圾数据。

#### 对泽塔的接口需求

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 入参 | 处理方式 | 返回值 |
| 上传镜像文件 | 文件内容  文件MD5 | 存储文件内容  生成HCI层的文件ID | HCI层的文件ID |

### 应用新镜像

#### 业务描述

镜像编辑后将新镜像应用给已创建的桌面。

#### HCI层处理建议

只能在关机状态下执行操作，执行的结果是替换虚机的系统盘。

#### 对泽塔的接口需求

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 入参 | 处理方式 | 返回值 |
| 替换虚机的系统盘 | 虚机ID  新的快照ID | 删除原有虚机系统盘  根据新的快照ID创建系统盘并挂载给虚机 | 无返回值  如果虚机不处于关机状态，则报错 |

### 创建个性化应用分层模板

#### 业务描述

创建用户个性化应用分层的磁盘模板。

#### HCI层处理建议

根据个性化应用层磁盘大小初始化块设备，并挂载到自身的一台内部虚机中格式化成NTFS格式，同时在磁盘中增加文件。



#### 对泽塔的接口需求

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 入参 | 处理方式 | 返回值 |
| 创建磁盘模板 | 磁盘大小 | 创建一个系统磁盘并格式化为NTFS格式 | 磁盘ID |

### 根据磁盘模板创建磁盘

#### 业务描述

创建独立的磁盘做应用分层的个人盘。此磁盘内部文件系统必须是NTFS格式，并且携带文件

#### HCI层处理建议

创建一个已经格式化为NTFS的块设备模板，根据块设备模板复制出可用的个性化数据盘，或者根据对应块设备的快照派生出个人盘。

注意系统盘的不同规格对应不同的NTFS块设备模板。

#### 对泽塔的接口需求

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 入参 | 处理方式 | 返回值 |
| 创建磁盘 | 创建磁盘的磁盘模板ID | 根据磁盘模板复制出新磁盘 | 新磁盘ID |

### 创建云桌面（修改）

#### 业务描述

支持应用分层能力、支持虚拟网络参数的传递。

#### HCI层处理建议

N/A

#### 对泽塔的接口需求

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 入参 | 处理方式 | 返回值 |
| 创建云桌面 | 用户个性磁盘大小  虚拟网络ID | 创建虚机并返回虚机ID | 虚机ID |

## RCDC和GuestTool的接口考虑

### 概述

和GuestTool的交互用于对GuestTool进行初始化，并且获取到镜像模板编辑阶段的相关参数。

### 初始化GuestTool

#### 目标

GuestTool首次启动时向RCDC上报当前虚机的信息，同时RCDC向GuestTool返回IP地址以及是否允许进行Windows正版激活等信息。

#### 通信方式

VmMessage

#### 接口描述

|  |  |  |
| --- | --- | --- |
| 接口名 | GuestTool入参 | RCDC的返回值 |
| 初始化GuestTool | 是否支持Windows正版授权  是否支持应用分层 | 虚机的IP、掩码、网关、DNS  RCDC是否开启Windows正版授权  当前桌面使用的镜像模板是否是最新版  虚机运行模式：用户模式/镜像编辑模式  Samba服务的访问地址 |

### 管理员更新镜像模板的通知

#### 目标

管理员更新镜像模板后，向所有运行中的桌面发送的消息通知。

#### 通信方式

VmMessage

#### 接口描述

|  |  |  |
| --- | --- | --- |
| 接口名 | RCDC入参 | VmMessage返回值 |
| 管理员更新镜像模板 | 无 | 无 |

#### 处理建议

提醒用户管理员更新了镜像，询问用户是否进行更新，并向用户揭示更新的风险：

* 未开启应用分层模式：全部用户的操作将丢失；
* 开启应用分层模式：部分软件配置和操作系统配置会丢失。

GuestTool需要给用户提供选项，让用户决策是否进行更新，选项如下：

* 永不更新：不再提示用户
* 立即更新：提示用户保存现有工作，并在2分钟后进行关机，关机前二次确认，用户确认后通知RCDC更新镜像模板，并关机；
* 下次镜像更新时通知

### 应用镜像模板更新的通知

#### 目标

用户决定应用镜像模板的更新。

#### 通信方式

VmMessage

#### 接口描述

|  |  |  |
| --- | --- | --- |
| 接口名 | GuestTool入参 | RCDC返回值 |
| 应用镜像模板更新的通知 | 无 | 无 |

#### 处理建议

RCDC调度当用户桌面关闭后开始执行镜像模板的更新动作。

### RCDC开启/关闭Windows正版授权功能

#### 目标

导入Windows正版授权文件以及试用授权过期时，通知所有运行中的桌面。

#### 通信方式

VmMessage

#### 接口描述

|  |  |  |
| --- | --- | --- |
| 接口名 | RCDC入参 | VmMessage返回值 |
| RCDC开启/关闭Windows正版授权 | 开启/关闭正版授权 | 无 |

### ACPI接口扩展

在ACPI中要告诉GuestTool当前正在使用应用分层的功能。

开启应用分层后，GuestTool需要做到如下几件事：

* 向用户隐藏应用分层的磁盘；
* 隐藏应用分层目录下的layer\_templete文件；
* 不允许用户删除、修改、覆盖layer\_templete文件；
* 不允许用户在其他磁盘新建layer\_templete文件。

## RCDC和KMS服务的接口考虑

保持现有接口

## RCDC和应用分层驱动的接口考虑

采用文件系统根目录下的layer\_templete文件标识当前磁盘开启应用分层功能。

# 主要结构

## 工程和包结构

沿用组件化的开发规范。

## ISO文件类图

### Controller层

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| OsIsoFileController | ISO文件管理的Controller入口 |
| DeleteOsIsoFileWebRequest | 删除ISO文件的请求对象 |

### API定义

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| CbbOsIsoFileMgmtAPI | ISO文件管理的对外API |
| ImportOsIsoFileRequest | 导入请求 |
| ListOsIsoFileRequest | 查询请求 |
| DeleteOsIsoFileRequest | 删除请求 |

### API实现

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| CbbOsIsoFileMgmtAPI | ISO文件对外API |
| CbbOsIsoFileMgmtAPIImpl | ISO文件对外API的实现类 |
| OsIsoFileAdapter | HCI层适配器接口 |
| OsIsoFileValidationService | 业务校验器接口 |

### 校验器实现

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| OsIsoFileValidationService | 业务校验器接口 |
| OsIsoFileValidationServiceImpl | 业务校验器实现 |

### HCI层实现

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| OsIsoFileAdapter | ISO文件上传的HCI层接口 |
| ZetaOsIsoFileAdapterImpl | ISO文件上传的HCI逻辑适配器 |
| ZetaOsIsoFileAdapterImpl | ISO文件上传的HCI接口适配器 |

### 定时任务

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| SyncOsIsoFileStateTask | 定时同步文件状态的任务 |

## SkyEngine简单同步状态机

### 概述

镜像模板编辑的细节流程属于嵌套的多阶段复杂状态迁移，需要状态机的支持，目前SkyEngine没有对应功能，需要开发一个最简版的状态机供使用。

本状态机的功能仅为调度，没有持久化存储能力。

### 对外接口

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| StateProcessResult | 状态机的执行结果，内部有枚举表示返回状态 |
| ResultState | ProcessResult的内部状态枚举 |
| StateMachineHandler | 状态机的业务接口 |
| StateMachineDriver | 框架提供的状态机调度器 |
| StateMachineDriverFactory | 状态机调度器的工厂类 |

### 对外辅助类

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| StateProcessResultBuilder | 构造Result对象的工具类，提供三个工具方法：   * Next：跳转到下个状态，内部状态为CONTINUE； * Retry\*：表示sleep一段时间后重试，内部状态为CONTINUE； * Finish：表示结束状态，内部状态为BREAK。 |

### 实现类

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| SimpleSyncStateMachineDriver | 提供同步能力的状态机调度器 |
| DefaultStateMachineDriverFactory | 状态机调度器工厂类的实现 |

### Spring注册器

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| StateMachineBeanDefinitionRegistryPostProcessor | 注册器的入口，注册到Spring根上下文中 |
| StateMachineBeanNames | Spring中bean名称的常量 |

# 系统流

## ISO文件管理

### ISO文件状态

#### 概述

ISO文件状态表示RCDC业务层的ISO文件状态，这里的状态是以业务方式呈现，屏蔽了HCI层更多的文件状态和处理流程。

#### 状态转换图



#### 状态说明

|  |  |  |
| --- | --- | --- |
| 标题 | 英文名 | 功能 |
| 上传中 | UPLOADING | 表示RCDC已经完成前端上传的文件接收，正在传递给泽塔 |
| 可用 | AVALIABLE | 文件已经成功传递给泽塔 |
| 删除中 | DELETING | 正在删除泽塔中的文件 |

### 上传文件

#### WEB层

##### 序列图



##### 说明

Web层需要生成ISO文件在数据库中的UUID，便于后续进行事务补偿。

#### 正常流程

##### 序列图



##### 说明

将文件传递给HCI层之前，需要添加文件记录并记录文件状态，此时文件状态为：UPLOADING；

当文件传递给泽塔成功后，需要将对应的记录的文件状态修改为：AVALIABLE。

#### 补偿流程

##### 序列图



##### 说明

如果文件已经成功传递给HCI层，则将对应数据库状态设置为：AVALIABLE；

如果文件未成功上传给HCI层，删除数据库中的文件记录。

##### 异常场景

如果HCI层抛出异常，无法判断文件是否存在，则认为文件未成功上传给HCI层，对应数据一致性由HCI层保障。

#### 业务校验

##### 序列图



##### 说明

如果文件已经存在，则不允许重复上传（通过文件名和文件MD5进行判断）。

如果文件处于上传中、删除中的状态，认为文件存在，不允许上传。

### 查询文件

#### WEB层

##### 序列图



##### 说明

基本的分页查询，无需赘述。

#### 正常流程

##### 序列图



##### 说明

上传中、删除中的文件都需要呈现在列表中，避免管理员出现困惑。

### 删除文件

#### WEB层

##### 序列图



##### 说明

WEB页面为批量删除，需要使用批处理框架异步处理每个文件的删除动作。

#### 正常流程

##### 序列图



##### 说明

需要先将数据库记录的状态修改为：DELETING；

然后调用HCI层的接口进行文件删除；

文件删除成功后再删除本条数据库记录。

#### 补偿流程

##### 序列图



##### 说明

如果HCI层对应文件不存在，则删除本地数据库记录；

如果HCI层对应文件存在，则将本地数据库记录状态修改为：AVALIABLE；

##### 异常场景

如果HCI层抛出异常，则认为删除失败，将本地数据库状态记录为：AVALIABLE，并且记录文件不一致的状态，由定时任务进行一致性处理。

#### 业务校验

##### 序列图

****

##### 说明

删除文件需要校验两个点：

1. 当前数据库状态不能是暂态；
2. 文件不能在使用中（使用数据库中的useCount字段进行判断，此字段的正确性由其他业务保障）。

### 数据一致性保障

#### 同步文件状态

##### 序列图



##### 说明

需要查询删除状态与实际状态不一致的数据，如果发现远程数据已经删除，则执行删除操作。

### HCI适配层 to 泽塔

#### 上传文件

##### 序列图



##### 说明

调用HCI接口之前，记录文件映射关系，此时zetaFileId为null

调用HCI接口之后，记录zeta返回的文件ID。

##### 异常场景

记录映射文件之前，需要查询是否有重复记录（残留记录），并删除重复记录。

将文件传递给泽塔之前，也要查询下泽塔是否有重复记录，如果有重复记录，则不执行实际的拷贝，直接记录已有文件的ID。

#### 根据文件名和MD5查询ISO文件是否存在

##### 序列图



##### 说明

如果查询到泽塔中存在文件，则记录泽塔的文件ID；

如果无法查询到泽塔中的文件，则删除本地记录。

#### 删除ISO文件

##### 序列图



##### 说明

先删除泽塔记录，再删除本地文件。

##### 异常场景

#### 根据ID查询ISO文件是否存在

##### 序列图



##### 说明

如果远程文件不存在，则同步删除本地文件。

## SkyEngine简单同步状态机

### 状态机的创建

#### 对外接口的使用方法

##### 序列图



##### 说明

上层业务只需要实现框架要求的StateMacheHandler接口，调用StateMachineDriverFactory.create\*方法，即可创建一个可运行的状态机。

#### 内部处理业务

##### 序列图



##### 说明

状态机驱动注册在Spring中，bean类型为prototype类型，getBean时第一个构造方法参数为对应handler对象。

### 状态机的运行

#### 对外接口的使用方法

##### 序列图



##### 说明

上层业务只要调用Driver的start方法，即开始运行状态机。

#### 主流程

##### 序列图



##### 说明

状态机的运行分为DO阶段和UNDO阶段，DO阶段处理正常业务，UNDO阶段处理回滚业务。

当前处于DO还是UNDO状态由业务进行判断。

##### 异常处理

DO状态如果出现异常，框架调度开始执行UNDO状态的逻辑（图中未画出）；

UNDO状态如果再出现异常，则仅记录LOG，反复进行重试，直到故障解决；

#### DO流程处理

##### 序列图



##### 说明

DO流程是一个while循环，循环结束条件有两个：

* 业务返回的ResultState为BREAK；
* 循环体内代码出现异常。

##### 异常处理

默认框架不会产生异常，所有异常都是业务产生的。

所有异常都需要记录LOG日志，级别是ERROR，所有异常需要打印详细的堆栈信息。

默认业务实现的changeToUndoState的方法不应该抛出异常，如果此方法抛出异常状态机将调度UNDO业务处理，但业务可能因为状态问题导致处理出错。

#### UNDO流程处理

##### 序列图



##### 说明

UNDO流程的逻辑与DO流程类似，不同点在于异常处理时不能继续抛出，需要记录LOG并吞掉所有异常。

### Spring Bean注入

#### 启动流程

##### 序列图



##### 说明

状态机的Factory接口使用BeanDefinitionRegistryPostProcessor初始化，注册到RootSpringContext中。

## 镜像模板管理

### 概述

镜像模板管理的流程包括导入文件、编辑准备、编辑、发布等过程，复杂流程会使用状态机进行逻辑处理。

### 公共工具逻辑

#### 创建状态机

##### 概述

后续流程中会涉及到状态机的创建和启动，这部分的代码属于公共流程，封装到辅助类中统一处理。

##### 序列图



##### 说明

创建镜像模板编辑状态机的工作由ImageTemplateEditStateMachine接口进行处理，ImageTemplateEditStateMachineImpl是其实现类。

ImageTemplateEditStateMachine属于Façade模式，封装所有镜像模板编辑中用到的状态机的构造和启动操作。

所有状态机都以Prototype方式注册到Spring容器中，每次构造状态机时是新实例，避免单例导致故障。

### 查询正在编辑的镜像状态

#### WEB层

##### 序列图



##### 说明

Web层不做复杂业务处理，仅将请求传递给组件接口。

#### 正常流程

##### 序列图



##### 说明

本流程不执行复杂的校验，如果数据库中存在对应记录，则认为镜像处于编辑中，负责需要抛出异常。

##### 异常场景

本接口通常用于异步查询镜像状态，镜像编辑、创建的请求提交后，通常5秒左右可以查询到编辑中的状态，前端如果在提交编辑、创建请求后立即查询此接口，可能因为时序问题得到错误的状态。

### 选中ISO文件

#### WEB层

##### 序列图



##### 说明

请求由前端发起，在Controller层生成新镜像模板的UUID，并调用组件API实现对应业务。

#### 正常流程

##### 序列图



##### 说明

正常流程分两个步骤，第一步先进行业务校验，第二步调用事务层的接口构造出多张表，并记录ISO文件的引用计数。

#### 补偿流程

没有事务之外的RPC或跨组件调用，无需业务补偿。

#### 业务校验

##### 序列图



##### 说明

本流程属于新建流程，无需进行更多业务校验，只需要校验ISO文件状态是否处于可用，只有可用状态的ISO文件才能用于创建镜像模板。

#### 事务处理

##### 序列图



##### 说明

本流程涉及三张表的变动，包括镜像模板主表、镜像模板编辑中的状态记录表、ISO文件记录表。

* 镜像模板主表：使用上层传递的主键添加数据、状态为创建中；
* 镜像模板编辑状态记录表：与主表同主键，关键状态为使用ISO文件创建镜像，并记录ISO文件的ID；
* ISO文件记录表：将引用计数+1。

##### 异常场景

因为前一章节描述的校验和当前数据库记录修改不在同一个数据库事务中，可能出现校验时ISO文件状态记录正确，但在事务前其他线程执行了ISO文件的删除动作，所以本流程中需要校验ISO文件是否处于可用状态，否则抛出业务异常，通知管理员。

此异常场景的数据记录表在同一个事务中进行变更，不涉及外部回滚动作。

#### 未来可能的变化

##### ISO文件记录表事务处理的可能变化

这里的流程中ISO文件记录表和镜像模板的相关数据库表在同一个数据库中，可以使用上述数据库事务的方式保证数据一致性。

如果未来逻辑变更，ISO文件记录表的变更与镜像模板数据库表的变更无法纳入同一个数据库事务进行管理时，需要在ISO文件记录表中记录引用计数和引用者，并且通过补偿流程保证数据一致性。

##### 镜像名称的记录

上述流程未涉及镜像名称的记录，如果需要记录则应在ISO选择的请求中同时携带镜像名称，在事务处理中记录镜像名称，对其他业务无影响。

### 上传预制镜像文件

#### WEB层

##### 序列图



##### 说明

管理员选中预制镜像文件后会在前端执行分块上传，分块上传由SkyEngine框架接收，等所有分块上传完成并校验通过后，将控制权交给ImageTemplateController进行处理。

ImageTemplateController生成新的镜像模板ID后，调用组件API处理实际的业务。

##### 交互体验

本业务属于耗时操作，需要使用批处理框架尽快向管理员反馈状态，减少操作等待的体验。

#### 正常流程

##### 序列图



##### 说明

上传镜像模板时文件的完整性由SkyEngine框架负责，考虑到文件较大，MD5校验完整性耗时会很长，所以组件API处理时不进行二次校验。

业务处理时分为如下三步执行：

* 步骤1：创建数据库记录；
* 步骤2：将文件传递给HCI平台，并在HCI平台创建一个完整的可派生虚机的镜像记录；
* 步骤3：将镜像状态设置为：等待配置临时虚机参数。

#### 补偿流程

##### 序列图



##### 说明

补偿流程中会校验HCI层中是否已经成功上传并构造了镜像，如果有则更新本地状态为等待配置临时虚机参数，否则删除已创建的镜像模板数据。

##### 异常场景

HCI适配层中上传镜像文件和构造镜像数据可能需要调用不同的HCI层API，期间也涉及多个状态的流转，此场景下的数据一致性和状态流转由HCI适配层自身状态机负责处理，与业务层无关。

#### 事务处理

##### 序列图



##### 说明

上述流程包含创建和删除的流程：

* 在创建流程中，使用上层传递的UUID构造镜像模板主表和状态表的数据，主表状态为创建中，状态表中记录当前状态表记录数据是由上传预制镜像产生的，并且将状态设置为镜像上传中；
* 在删除流程中只关注主表和状态表的数据清理，不关心其他表的删除（其他逻辑如果复用本API，需要注意本操作是直接删除，不执行任何业务层面的校验）。

#### 未来的可能变化

##### 镜像名称的记录

上述流程未涉及镜像名称的记录，如果需要记录则在上传文件同时应携带镜像名称，事务处理中记录镜像名称，对其他业务无影响。

### 复制已有镜像文件

#### WEB层

##### 序列图



##### 说明

与其他创建流程一样，新模板的ID在Controller层生成。

##### 交互体验

本业务属于耗时操作，需要使用批处理框架尽快向管理员反馈状态，减少操作等待的体验。

#### 正常流程

##### 序列图



##### 说明

组件API的主流程分为：校验、前置数据添加、HCI层RPC调用、本地状态修改四个阶段。

#### 补偿流程

##### 序列图



##### 说明

在补偿流程中，需要查询本地和远程的数据状态，如果远程模板已添加，则将本地状态设置为等待配置临时虚机参数，否则需要删除本地数据。

##### 异常场景

HCI层中可能出现本地没有、但远程有数据的情况出现，此问题由HCI层统一进行数据一致性修正。

HCI层中复制已有模板可能执行多个子步骤，对应状态管理和状态机的流转由HCI适配层考虑，与业务层无关。

#### 业务校验

##### 序列图



##### 说明

如果原有策略模板没有处于可用状态，不能进行复制。

#### 事务处理

##### 序列图



##### 说明

本流程需要添加镜像模板主表和状态表，其中主表的状态为创建中，状态表需要记录当前模板由哪个模板Clone而来，并且将状态设置为等待HCI层Clone镜像。

对于Clone的来源需要进行引用计数的+1动作，避免在Clone过程中原有模板被删除或修改。

##### 异常场景

Clone的来源可能在进入事务之前状态会发生变化，所以在事务中需要二次校验其状态是否可用，避免出现并发问题。

### 准备编辑已有镜像

#### WEB层

##### 序列图



##### 说明

Controller层只是简单将请求转发给组件API，不进行其他业务处理。

##### 交互体验

本业务属于耗时操作，需要使用批处理框架尽快向管理员反馈状态，减少操作等待的体验。

#### 正常流程

##### 序列图



##### 说明

正常流程分为以下几步：校验、添加本地数据、启动状态机操作远程数据。

#### 补偿流程

##### 序列图



##### 说明

补偿流程中启动状态机，由状态机接管业务处理。

#### 业务校验

##### 序列图



##### 说明

开始编辑镜像之前，需要镜像处于可用状态，否则不可编辑。

#### 状态机接口

##### 序列图



##### 说明

状态机的接口仅调用自身私有方法创建并启动一个状态机。

#### 状态机实现

##### 序列图



##### 说明

上述状态机实现只是简图，流程是根据镜像模板ID查询出所有还原桌面，关闭所有还原桌面，并记录状态。

实际上桌面的状态很多，考虑到性能，需要分批多次循环进行并行处理：

* 对于休眠状态的桌面，执行关机（对应桌面的业务已经处理过）；
* 对于处于运行中的桌面，执行关机；
* 对于唤醒中、启动中、重启中的桌面，等待其状态变更为运行中后再执行关机；
* 对于处于迁移中、恢复中的桌面，等待其状态变为稳定态后再根据稳定态判断如何处理；
* 对于错误状态的桌面，忽略。

### 配置编辑镜像的虚机参数

#### WEB层

##### 序列图



##### 说明

WEB层只转发请求到组件API，不执行其他处理。

#### 正常流程

##### 序列图



##### 说明

正常流程可以分为业务校验、记录状态、启动状态机三个步骤。

#### 补偿流程

##### 序列图



##### 说明

在补偿流程中重启状态机，确保完成编辑或者回滚到修改前的状态。

#### 业务校验

##### 序列图



##### 说明

修改参数之前，需要执行以下校验：

* 校验1：确保当前状态可进行参数修改（只有处于等待配置参数以及配置参数成功两个状态可以进行参数的修改和配置，其他状态不可执行此配置）；
* 校验2：校验接口中上传的参数，例如CPU核数、内存大小、磁盘大小是否符合要求，这里定义校验规则如下：CPU核数为1~8核、内存大小为2~16GB、磁盘大小请参考第二章的说明；
* 校验3：校验临时虚机的虚机网络是否可用，这里的场景分为三种场景、六种状态分别处理：

|  |  |  |  |
| --- | --- | --- | --- |
| 场景编号 | 场景说明 | 首次编辑 | 再次编辑 |
| 场景1 | 管理员未选用任何虚拟网络，则说明是随机分配 | * 找到系统当前可用的虚拟网络，随机分配一个IP地址； * 如果没有任何可用的虚拟网络，则报错。 | 使用上次编辑的IP地址 |
| 场景2 | 管理员选中了虚拟网络，但未指定IP | * 如果对应虚拟网络中有可用IP，则进行分配； * 如果对应虚拟网络中没有任何可用IP，则报错。 | * 如果虚拟网络ID未变化，沿用上次的配置； * 如果虚拟网络ID发生变化，则先尝试分配，分配成功则进行替换，分配成功不改变原有IP分配，并报错。 |
| 场景3 | 管理员选中虚拟网络，并且指定了IP地址 | * 如果对应IP在虚拟网络的IP池中，并且IP可用，则进行分配； * 否则报错。 | * 如果虚拟网络ID和IP均未变化，则沿用上次配置； * 如果发生变化，则先尝试分配，分配成功则进行替换，分配成功不改变原有IP分配，并报错。 |

#### 状态机接口

##### 序列图



##### 说明

调用私有方法创建并启动状态机。

#### 状态机实现

##### 序列图



##### 说明

对应IP分配策略已经在业务校验章节进行说明，为了避免并发情况下已经校验通过状态在进入事务前被变更，所以事务内需要执行同样的校验，确保数据一致。

# 数据结构

## ISO文件管理

### 枚举定义

#### 类图



#### 说明

在4.1.1章节已明确每个枚举字段含义。

### 实体Bean

#### 类图



#### 说明

|  |  |
| --- | --- |
| 类名 | 说明 |
| OsIsoFileEntity | 保存业务层的ISO文件记录 |
| ZetaUploadFileMappingEntity | 记录和泽塔的ISO文件映射关系 |

### 数据库定义

#### ISO文件记录（t\_cbb\_os\_iso\_file）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 中文名 | 数据类型 | 是否为空 | 备注 |
| ID | 主键 | UUID | 否 |  |
| VERSION | 行级版本号 | int | 否 |  |
| CREATE\_TIME | 添加时间 | datetime | 否 |  |
| FILE\_NAME | 文件名 | Varchar(128) | 否 | 唯一索引 |
| FILE\_MD5 | 文件MD5 | Varchar(64) | 否 | 唯一索引 |
| FILE\_STATE | 文件状态 | Varchar(32) | 否 |  |
| USE\_COUNT | 引用计数 | int | 否 | 默认0 |
| FILE\_SIZE | 文件大小 | bigint | 否 |  |

#### HCI层的ISO文件记录（t\_cbb\_zeta\_upload\_file\_mapping）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 中文名 | 数据类型 | 是否为空 | 备注 |
| ID | 主键 | UUID | 否 |  |
| VERSION | 行级版本号 | int | 否 |  |
| CREATE\_TIME | 添加时间 | datetime | 否 |  |
| RCDC\_FILE\_ID | RCDC中的文件ID | UUID | 否 |  |
| ZETA\_FILE\_ID | 泽塔的文件ID | Varchar(64) | 是 | 如果文件尚未添加成功，可以为空 |
| FILE\_NAME | 文件名 | Varchar(128) | 否 |  |
| FILE\_MD5 | 文件MD5 | Varchar(64) | 否 |  |

# 算法描述

【帮助】详细描述新的和修改的代码模块。鼓励使用伪代码，流程图，以及自然语言。可能的细节描述包括：

·使用的算法

·约束，依赖，限制，或者软件设计中独特的特性

·软件开始运行时需要的条件

·什么条件下控制权传给其他组件

·错误和例外处理

【裁剪说明】伪代码如果用编码语言书写，在本文档中也需遵循我司的编码标准(详见《TEC00S01-编码标准》)。

<正文>

# 接口设计

# 最终用户接口

【帮助】描述最终用户如何使用本产品完成任务。定义用户接口命令，参数，屏幕格式，错误提示信息，以及屏幕快照例子等。也包括用户使用性能(如任务花费时间，成功率等)，安全性考虑，以及适当的命令和错误信息扩展。

对于用户界面上显示的CLI命令中的帮助文字，可以在这里描述。如果这里没有描述，也可以在“源代码”等合适的章节中描述。如果已经在其它的文档中描述，则给出该文档的说明或链接也就可以了。如果都没有描述，则默认采用与Cisco相同的帮助文字。

同时考虑对于使用产品有困难(如残疾人)的人员如何完成任务，特别是用了哪些辅助技术。

产品的外部硬件接口、指示灯、按钮等，也都是用户接口，应根据产品的软硬件接口文档的描述，说明这些外部接口在用户使用过程的功能表现。

【裁剪说明】如果特性需求说明书或者软件功能规格中已经详细描述，这里简单说明一下参见特性需求说明书或者软件功能规格就可以了；如果有细化，则细化的内容在这里描述。

<正文>

# 软件限制与约束

## 大文件上传

1. ISO和预制镜像上传中断后无法断点续传；
2. 如果两个管理员同时上传同一个ISO文件，两个文件都能上传成功，但实际存储到HCI超融合存储池中的文件只会有一个。

# 开发单元测试

【帮助】定义单元测试用例使用的方法，包括：

·定义单元测试通过的标准，通过后才能合并到版本主干开始项目的集成测试

·是否需要特定的测试设备

·与法律法规有关的测试也要考虑

<正文>

【裁剪说明】1．如果项目裁剪掉单元测试活动，本章不适用。

# 参考资料

【帮助】

<文档名>, <文档信息的简单描述>

<文档名>, <文档信息的简单描述>

# 术语表

【帮助】

下面列出本文中英文缩写的词语含义。

<**术语1**>: <简单的定义>

<**术语2**>: <简单的定义>

# 附件

【帮助】给出与本文有关的适当的附件，比如日志信息，图表，有用的研究报告，例子等。

<正文>

## 评审活动记录

【帮助】这里记录文档的评审活动，并跟踪评审问题的关闭。相应的数据有：评审参加者，关注焦点，活动记录等。活动记录包括：问题的描述，提出者，状态（打开或关闭），以及关闭的数据说明等。

作为保存评审活动记录的章节，这里也可以参考外部的评审记录，包括问题的跟踪与关闭等。这些也可以通过相应的评审工具来跟踪。

<正文>

## 需求可跟踪考虑

【帮助】需求可跟踪，可以是手工的跟踪，也可以是软件工具来辅助。

为了可跟踪与可测试，如果系统功能规格中需求描述不够细，这里需要把它继续细化。

所有功能需求需要有唯一的标识，需要依据组织的需求跟踪策略，跟踪产品需求到功能需求。

如果项目有独立的需求跟踪矩阵，请提供到该文档的链接。

在这里，需要描述出软件功能规格中描述的功能元素在这篇文档如何得到体现（如果在其他文档中体现，这里也请说明一下参见的文档）。所谓的功能元素是指：模块、组件、子系统、接口，等等。建议用表格的形式逐条说明。如：

|  |  |
| --- | --- |
| 软件功能规格中的功能元素 | 主要在哪一章中细化描述  （只要给出主要的1章或1节就可以了） |
|  |  |

另外，在本节中说明与本设计有关联的工程文档，关联文档一般指：作为本设计输入的PRD文档、功能规格文档、MIB/LOG文档、对应的配置指南、命令参考、以及有依赖关系的其他设计文档等（关联文档可能识别不全，但不能没有）。

【裁剪说明】如果前面的活动没有输出软件功能规格但有输出SRS（或者SRS也是和软件设计规格一起输出），这里的需求跟踪描述可以改为描述出SRS中的每一条2级规格，在这篇文档的哪个部分分别得到体现（如果在其他文档中体现，这里也请说明一下参见的文档）。建议用表格的形式逐条说明。如：

|  |  |  |
| --- | --- | --- |
| 2级规格说明 | 在哪个功能元素中体现 | 主要在哪一章中说明（只要给出主要的1章或1节就可以了） |
|  |  |  |

<正文>

## 本文档协助人员

【帮助】在本文档的撰写过程中，作者需要与相关利益人进行沟通交流，这个沟通交流应该在发起评审之前进行。这些交流以及交流的人员实际上对本文档的最终完成有促进作用。应尽量避免以评审代替交流。

协助人员

|  |  |
| --- | --- |
| 部门/专业组 | 协助者 |
|  |  |
|  |  |