# 渲染系统概要设计

文档历史

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 作者 | 描述说明 |
| 2019.4.2 | V0.01 | 祁宇 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 文档最后一次修改时间 | | 2019-04-08 14:14:00 | |

## 设计目标

渲染系统是一个分布式计算系统，计划分两步来实现这个系统。

第一步：

在局域网环境下渲染3dmax图片，系统由一台服务器和数十台客户端组成，这一步实现的是一个demo，用来测试渲染系统的所有业务流程，比如，分割源文件，大文件传输，渲染分割后的文件以及错误处理等等。这个demo非常重要，它是后续大规模分布式计算系统开发的坚实基础。

第二步：

在互联网环境下渲染3dmax图片，系统由多台服务器和成千上万的客户端组成。

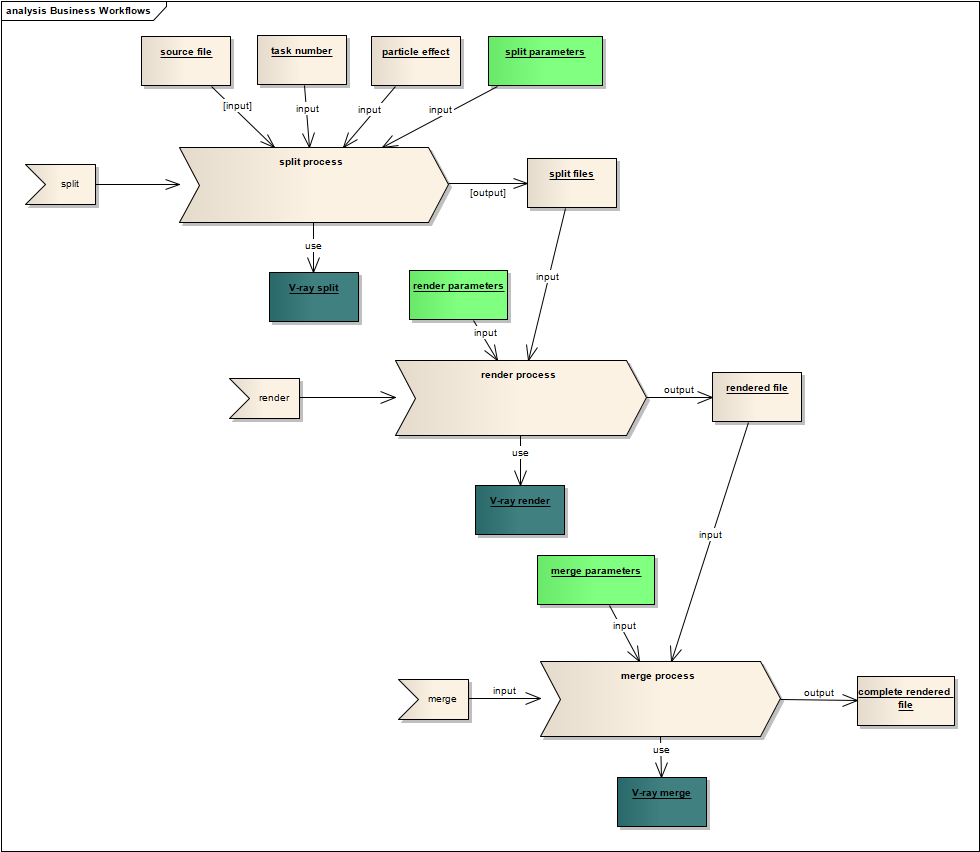
因此本文档是关于第一步的设计，只考虑局域网环境下的3dmax图片渲染。

对于大文件传输的问题在局域网环境下不是一个问题，我们仅仅需要关注业务方面的问题，比如如何定义渲染系统需要的输入参数。

## 业务流程分析

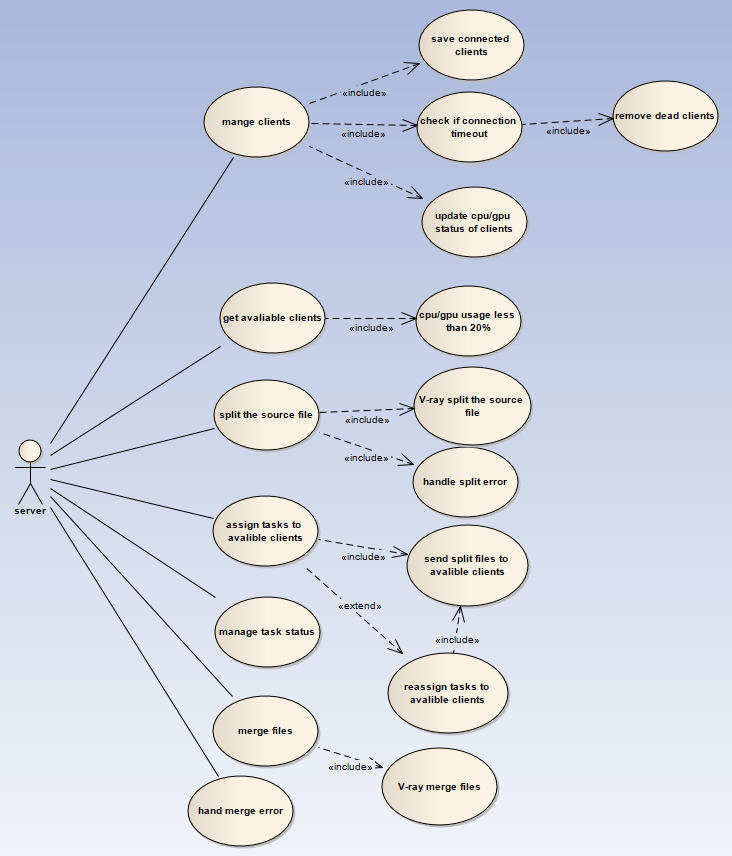
渲染系统的业务流程主要有三个：分割源文件流程、渲染流程和缝合流程。

下面是业务流程图。

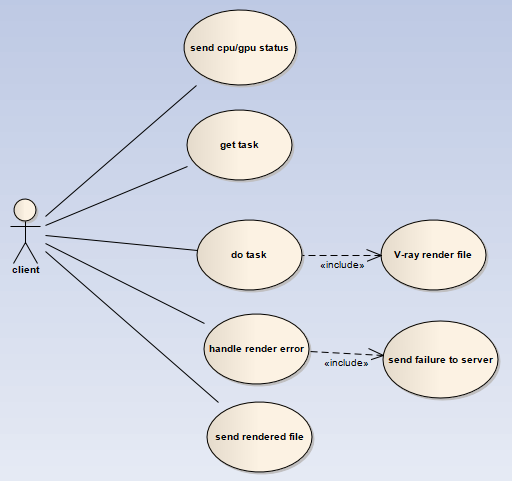


## 系统用例

### 服务器用例



### 客户端用例



### 用例说明

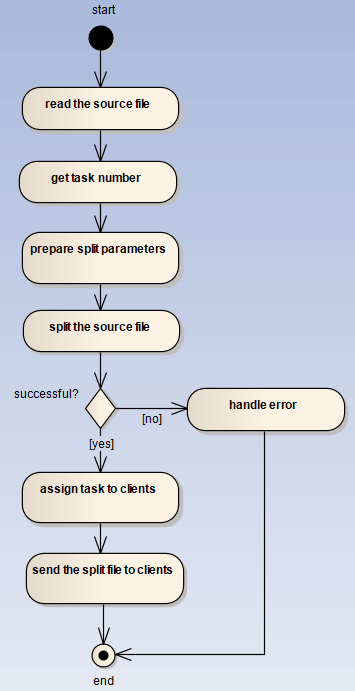
|  |  |
| --- | --- |
| 用例编号 | 001 |
| 用例名称 | 分割源文件 |
| 执行者 | Server, V-ray |
| 用例简介 | 描述如何分割源文件 |
| 前置条件 | 所有的客户端都连接到服务器了  获得了源文件的路径  获得了任务数  分割源文件需要的参数 |
| 基本事件流 | 1. 获取任务数（可用客户端数）  2. 获取源文件路径  3. 准备分割源文件的参数  4. 调用V-ray 的分割源文件的API  5. 获取分割后的文件路径  6. 分配渲染任务  7. 读取和发送分割后的文件到渲染客户端  8. 将原始文件发送到渲染客户端  9. 维护任务状态 |
| 异常事件流 | 非法参数:  记录日志  分割发生错误:  更新日志并退出程序  某个任务失败:  将失败的任务分配到其它 |
| 后置条件 | 1.成功完成  所有的任务都发到了客户端.  2.失败  更新日志. |

|  |  |
| --- | --- |
| 用例编号 | 002 |
| 用例名称 | 渲染文件 |
| 执行者 | Client, V-ray |
| 用例简介 | 描述客户端如何渲染文件 |
| 前置条件 | 客户端接收了分割后的文件和原始文件  获得了用于渲染的业务参数 |
| 基本事件流 | 1. 获取被分割的文件和原始文件路径  2. 准备渲染参数  3. 调用V-ray的渲染参数  4. 向服务器报告渲染过程中的状态  5. 读取并发送渲染后的文件到服务器 |
| 异常事件流 | 非法参数:  更新日志，并报告渲染非法参数的信息和状态到服务器  渲染错误.  更新日志，并报告渲染错误信息的信息和状态到服务器 |
| 后置条件 | 1.成功完成  被渲染的文件成功发送到服务器  2.失败  报告失败信息到服务器  更新日志 |

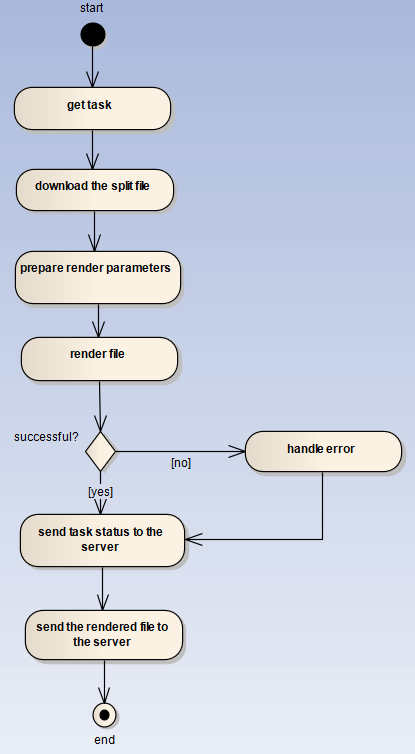
|  |  |
| --- | --- |
| 用例编号 | 003 |
| 用例名称 | 缝合文件 |
| 执行者 | Server, V-ray |
| 用例简介 | 描述如何缝合文件 |
| 前置条件 | 服务器收到了所有的被客户端渲染过的文件  获得了缝合文件需要的业务参数 |
| 基本事件流 | 1. 获取所有的被渲染文件的路径  2. 准备缝合的业务参数  3. 调用V-ray 渲染接口  4. 生成最终的渲染文件 |
| 异常事件流 | 非法参数:  更新日志并退出  缝合错误:  更新日志并退出 |
| 后置条件 | 1.成功完成  用户获得了最终的渲染文件.  2.失败  更新日志. |

## 业务活动

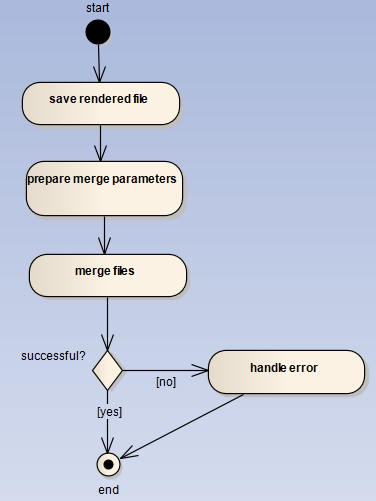
### 服务器分割文件活动图



### 客户端渲染活动图

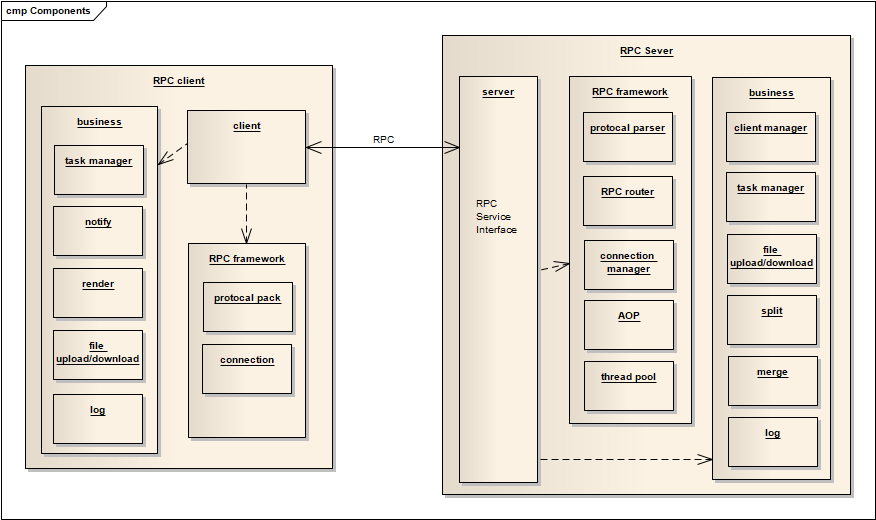


### 服务器缝合文件活动图



## 系统组件

### 系统组件图



### 系统组件说明

服务器和客户端的框架从逻辑上分为三层：

用户接口层：服务器提供RPC服务接口和客户端发起RPC请求的接口。

RPC框架层：RPC协议解析、路由、连接管理、线程池和AOP等。

业务逻辑层：关于渲染系统的具体业务逻辑。

#### 服务器组件说明

RPC服务接口组件：给客户端提供各种服务接口

RPC协议解析组件：负责解析RPC协议

RPC路由组件：负责将请求路由到对应的业务函数

连接组件：处理网络IO事件

连接管理器组件：管理所有的客户端连接

线程池：用于并行计算，提高系统吞吐量

AOP组件：将核心逻辑和非核心逻辑分离

Client管理器组件：管理可用的客户端

Task管理器组件：管理任务分配和调度

文件上传/下载组件：文件上传和下载

分割源文件组件：将源文件分割为多份

缝合文件组件：缝合多个文件成为最终渲染文件

日志组件：记录业务过程中的日志信息

#### 客户端组件说明

客户端RPC服务接口组件：封装了统一的调用RPC服务的接口

RPC协议打包组件：将客户端RPC请求打包为RPC协议

连接组件：处理网络IO事件

任务管理组件：管理渲染任务

通知组件：向服务器发送客户端任务的状态

渲染组件：渲染文件

文件上传/下载组件：文件上传下载

日志组件：记录业务过程中的日志信息

## 问题讨论

因为我们第一步只考虑局域网环境，所以不用担心大文件传输的问题。

假设合作方能提供渲染相关的SDK的前提下，需要讨论以下几个问题：

1. 如何定义渲染系统的业务参数？比如分割源文件、渲染文件和缝合文件的业务参数。

我们需要一个关于业务参数定义的文档，并且还有设置这些参数的测试程序。

1. 如何使用这些SDK？比如分割文件、渲染文件和缝合文件需要用的V-ray SDK.

我们需要一个关于如何使用SDK的文档。