Computer Science and Application 计算机科学与应用, 2018, 8(1), 43-48
Published Online January 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.81007

Hans汉斯

Design of Distributed Rendering Management System

Changyin Xing, Kanjie Zhang, Xuan Liu

Jilin Animation Institute, Changchun Jilin
Email: 63080180@qq.com

Received: Jan. 3rd, 2018; accepted: Jan. 16th, 2018; published: Jan. 29th, 2018

Abstract

In the animation production process, rendering is the most important part. However, the rendering process will take a lot of time, which also affects the efficiency of animation production. Distributed rendering is a great way to solve these problems. The distributed rendering system can allocate the rendering task reasonably and manage the rendering operation to improve the efficiency of rendering. This paper mainly designs a distributed rendering task management system. It can achieve user management of rendering tasks. The system provides a user-friendly management interface that can effectively improve the rendering speed and quality.

Keywords

Render, Distributed Rendering, Management System

分布式渲染管理系统的设计

行长印，张堪杰，刘　旋

吉林动画学院，吉林　长春
Email: 63080180@qq.com

收稿日期：2018年1月3日；录用日期：2018年1月16日；发布日期：2018年1月29日

摘　要

在动漫的制作过程当中，渲染是其最重要的环节。然而渲染过程会耗费大量的时间，也因此影响了动漫的制作效率。分布式渲染是解决上述问题的绝佳途径。分布式渲染系统能对渲染任务进行合理的分配，并对提交的渲染作业进行管理，以提高渲染的效率。本文主要设计了一种分布式渲染任务的管理系统。它可以实现用户对渲染任务的管理。本系统提供了对用户友好的管理界面，可以有效的提高渲染的速度

文章引用: 行长印, 张堪杰, 刘旋. 分布式渲染管理系统的设计[J]. 计算机科学与应用, 2018, 8(1): 43-48.
DOI: 10.12677/csa.2018.81007

与质量。

关键词

渲染，分布式渲染，管理系统

1. 引言

在网络、新媒体以及数字技术飞速发展的大环境下，影视动漫产业应运而生。在动漫的制作过程当中，渲染是其最重要的环节。然而渲染过程中，由于会产生大量的计算，普通计算机往往会耗费大量的时间，影响着动漫的制作效率。而大型的图形工作站费用高昂，中小企业及个人无法承担，也因此渲染的发展出现了瓶颈[1]。分布式渲染是解决上述问题的绝佳途径。分布式渲染系统能对渲染任务进行合理的分配[2]，并对提交的渲染作业进行管理，以提高渲染的效率。

由于市场上存在的管理软件中大部分不能实现用户自己对渲染任务的管理功能，只能通过管理员对任务进行分配。本文主要对分布式渲染任务的管理系统进行了设计。它可以完成用户对渲染工作的管理操作。本系统设计了与用户进行友好交互的 Web 门户，选用 OpenPBS 作业管理系统，并对其进行二次开发。系统从 Web 前台接收到用户提交的渲染任务，将其分割为多个子任务，这些子任务被管理系统分配到可用的渲染节点上并行处理。用户可对渲染任务进行管理，包括对每个渲染任务的启动，分配，监控以及终止等等。使得在分布式渲染系统中，各渲染节点可以发挥最大功效，以提高渲染的速度和质量。

2. 分布式渲染管理系统的架构设计

分布式渲染指的是一种网络渲染技术。它利用了集群中的计算机资源，通过并行的 3D 渲染软件，对动画中的复杂场景进行大量的计算，把渲染任务分布到多台计算机上进行渲染，最后再收集这些渲染后的图像[3]。利用分布式渲染技术可以极大地提高动画的渲染效率，提高了设备利用率。

本分布式渲染管理系统的架构包含：与用户良好交互的 Web 前台，对渲染任务进行分割与收集的分布式渲染管理器，分布式渲染管理软件 OpenPBS，渲染节点、存储节点。

1) Web 前台

Web 前台是连接用户与管理服务的桥梁，编写该系统的 Web 前台使用的语言是 Java。用户通过前台页面提交渲染任务，系统接收后发送给下一级。同时它可以把从下级得到的信息反馈回来，以便用户查看。Web 前台可实现的功能有：管理用户信息，渲染作业上传，监控渲染任务，管理渲染节点等。

2) 分布式渲染管理器

分布式渲染管理器是衔接 Web 前台与 OpenPBS 的重要接口。由于 OpenPBS 系统不提供系统的最终功能以及操作页面。利用这种优势，开发人员根据渲染工作的需求，可以采用命令方式对系统进行二次开发。分布式渲染管理器就是通过这种方式实现的，它接收到 Web 前台的渲染任务信息，根据用户的需求，完成对渲染任务的初次分配操作。例如提交的渲染任务总共包含 24 帧动画，用户设置的子任务大小为 2 帧，那么每个子任务都包含 2 帧动画。它还将渲染过程中的各种信息反馈给前台，以便用户及管理者的查看。

3) OpenPBS 集群调度软件

PBS 作为一款历史悠久的集群调度软件，拥有齐全的功能。它可以保证系统中的节点在规定的时间内完成指定的工作。与其他集群管理软件相比，PBS 具有许多优势。它提供易于配置的统一接口，该接口可面向任何资源，这便于对系统的二次开发。并且它可以实现自动的负载均衡[4]。

OpenPBS 作为该系统的开源版本，它由命令行、调度器、服务端以及执行端四部分组成。其中服务端的工作是接受任务，执行端的职责是将任务变为执行状态，调度器对任务的执行时间、运行节点以及是否运行等问题进行控制与分配。

在本系统中，OpenPBS 将分布式渲染管理系统初次调度后的子任务分配给系统当中的可用的渲染节点，执行渲染操作，实现对渲染任务的二次调度。

4) 渲染节点、存储节点

计算节点的主要功能是执行计算，它是整个系统的计算核心，本系统的渲染客户端就需要在计算节点上运行。

渲染节点上需要安装 Maya、3ds Max 等渲染软件，处理由 OpenPBS 分配下来的渲染任务，它是执行渲染任务的终端。

由于在动画渲染的进程中，会产生巨大的数据信息，因此对数据的读写及传输十分重要。系统中的存储节点就用来存储渲染进程中的数据信息。

下面根据图 1 分析分布式渲染系统的工作流程。

1) 用户登录系统后，通过 Web 前台提交渲染任务，并设置子任务大小等相关参数。

2) 分布式渲染管理器得到任务信息后，根据子任务大小将提交的渲染任务进行分割，成为多个渲染子任务，并传递给 OpenPBS，实现初次调度。

3) OpenPBS 将渲染子任务分配给系统中的渲染节点进行渲染，实现对渲染任务的二次调度，并返回渲染过程中的各种信息。

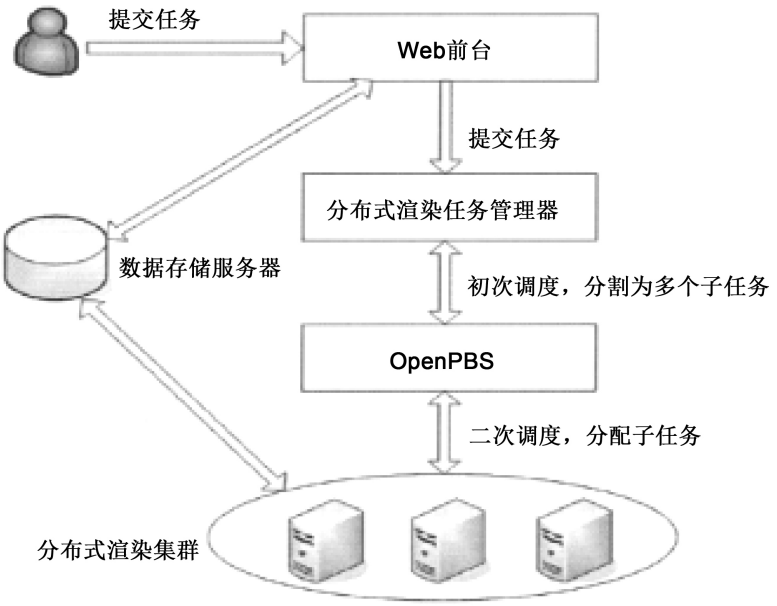

Figure 1. Distributed rendering management system architecture and workflow diagram
图 1. 分布式渲染管理系统架构及工作流程图

4) 用户可以在 Web 页面当中对相关信息进行管理和查询，得到最终的渲染结果。

3. 系统的关键技术

在设计分布式渲染管理系统时，存在着以下几点关键问题。

3.1. 与用户良好的交互

目前市面上存在着许多使用不同分布式渲染系统管理软件的渲染平台，然而这些平台中大多数不能实现用户对渲染作业的管理，只能通过管理员对任务进行分配。本设计利用对 OpenPBS 的二次开发，可以实现用户对自身渲染任务的各种管理需要。

然而，如果用户想访问 OpenPBS 的接口，需要使用命令行的方式。这对一般用户来说具有一定的难度。因此，我们可以使用 Java 语言编写一个对用户友好的 Web 前台页面，方便用户对该系统的使用。

3.2. 渲染任务的管理

1) 渲染任务的划分

解决该问题的思路是将用户提交的渲染任务分配给多台计算机执行。在本设计中，我们根据 OpenPBS 分布式系统管理软件，开发出了用来分配渲染任务的接口，即上文中提到的分布式渲染管理器。它可以根据用户需求，将渲染任务进行分割，达到加快渲染速度的目的。

实现该功能的主要代码如图 2 所示，其中参数 startframe 为子任务的起始帧，参数 packetsize 代表子任务的大小：

2) 渲染任务的状态信息

系统可以通过调用 OpenPBS 管理系统的 qstat 命令对渲染任务的状态信息进行查询。qstat 命令可以查询到的状态信息有：系统所有作业、指定作业信息、列出运行当中的作业、列出分配给指定作业的节点等等[5]。

在 OpenPBS 系统中，渲染任务的状态主要有四种。如果系统中有子任务在运行，则该渲染任务是运行状态；如果所有子任务都已成功完成，则该渲染作业为完成状态；如果渲染任务中的子任务都没有运行，则该渲染作业为排队状态；当子任务在运行过程中发生错误时，该渲染作业为出错状态。渲染任务的这些状态应保存到任务状态信息表中，以供用户查看。

```
division
int j = startframe;
int k = 1;
while{
    ……
    j = startframe + packetsize - 1;
    Render;
    Startframe = startframe + packetsize;
    k + 1;
    ……
}
```

Figure 2. Main code of partition rendering tasks
图 2. 划分渲染任务的主要代码

3.3. 系统的容错性处理

在执行渲染的过程当中，有可能会出现网络错误、渲染引擎故障、源文件出错等等突发状况，这些状况通常会导致渲染的任务出错。出错的渲染任务通常不会正常结束。我们可以设计出相应的容错机制，方便用户对出现错误的任务进行解决，并恢复系统的运行工作。

该容错机制的主要流程是：首先，得到出现错误的渲染任务信息。然后将该任务重新调度到两个不同的节点执行渲染操作。接着，比较两个任务的执行时长。最后得出结论，假如重新调度后该任务能够顺利完成，则出错的原因可能为渲染引擎故障或者是节点故障；如果该任务没有执行完成，而是仍旧报错，则出错的原因可能为源文件异常。

4. 系统的测试结果及分析

最后，对本文设计的分布式渲染管理系统进行了测试。使用该管理软件分别在系统中的 1 个节点、4 个节点以及 16 个节点上对场景进行渲染。图 3 为在 16 个节点上对场景进行渲染的效果截图，表 1 为三种测试情况下渲染时长的对比。

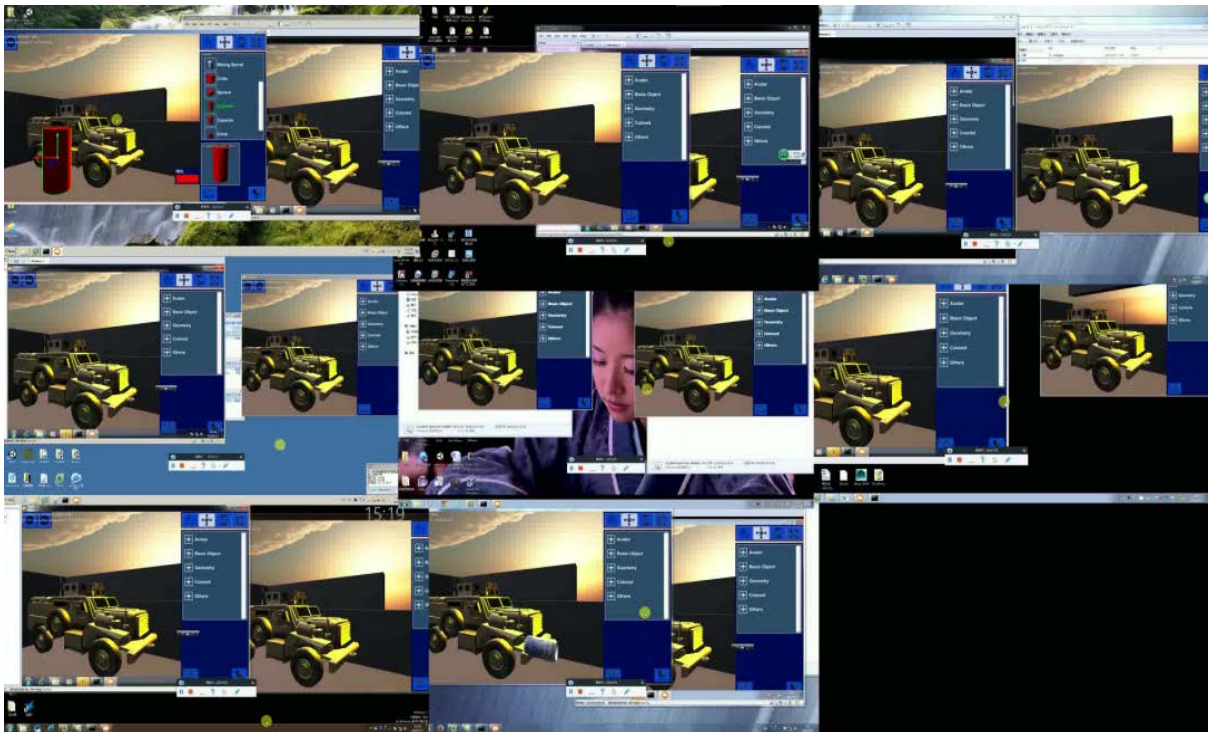

Figure 3. Rendering result screenshot
图 3. 渲染效果截图

Table 1. Rendering time comparison
表 1. 渲染时间对比表

节点数(个)	渲染时间(秒)
1	10,920
4	3052
16	915

通过分析测试结果可知，通过该分布式渲染管理系统，可以有效地减少渲染时长，提高渲染效率。

5. 总结

本文主要设计了一种分布式渲染任务的管理系统。该系统选用了 OpenPBS 分布式任务管理系统，可以实现用户对渲染任务的管理；为用户提供了友好的 Web 前台页面，方便用户的使用。该系统可以有效地提高渲染的速度和质量，有效解决渲染时间过长，图型工作站价格高昂等限制动画渲染发展的瓶颈问题。

基金项目

基于云计算的下一代动画实时渲染技术研究(吉教科合字【2016】第 521 号)。

参考文献 (References)

[1] 耿蕊. 中国动漫产业发展的危机与转机[J]. 湖南社会科学, 2010(1): 216-219.

[2] Rajkumar, B.R. 高性能集群计算: 结构与系统(第一卷) [M]. 郑纬民, 石威, 汪东升, 译. 北京: 电子工业出版社, 2001: 369-376, 415-422.

[3] 李树声. 网络集群渲染在 3D 动画制作中的应用[J]. 广播与电视技术, 2004, 31(9): 63-64.

[4] PBS Grid Works. Render Pro GUI & PBS Web Portal for Animation Segment.

[5] 童端董, 小社, 李纪云, 刘广红. 基于 OpenPBS 的机群作业管理系统的设计与实现[J]. 计算机工程与应用, 2004(13): 123-125.