

南开大学

本科生毕业论文（设计）

中文题目：基于分布式渲染的云游戏系统资源调度

外文题目：Resource Scheduling for Cloud Gaming Systems

Based on Distributed Rendering

学号：2011181

姓名：唐鹏程

年级：2020 级

专业：计算机科学与技术

系别：计算机科学与技术

学院：计算机学院

指导教师：李雨森

完成日期：2024 年 5 月

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律 responsibility 由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

关键词:

Abstract

Key Words:

目 录

摘要	I
Abstract	II
目录	III
第一章 绪论	1
第二章 文献综述	2
第三章 问题阐述	3
第一节 定义与符号表示	3
第二节 问题描述	3
第三节 NP 难解性	4
第四章 算法设计	5
第一节 基础算法	5
4.1.1 最低组合价算法 Lowest-Combined-Price	5
4.1.2 最低延迟算法 Lowest-Latency	5
第二节 启发式算法	6
4.2.1 Hill-Climbing Assignment	6
4.2.2 Simulated-Annealing Assignment	6
第五章 实验评估	8
第六章 结论	9
致 谢	X

第一章 绪论

第二章 文献综述

第三章 问题阐述

第一节 定义与符号表示

常量:

传输延迟阈值: Lat

带宽阈值: BW

工作负载阈值: WL

对于客户端 i , 数据中心中心 j , 定义:

- 单位传输价格: $cost_i^j$
- 传输延迟: l_i^j
- 带宽: b_i^j

客户端集合: $\{c_1, \dots, c_n\}$

客户端的渲染工作负载阈值为 h_i

客户端的总数据量为 d_i , 则每个客户端需要传输至服务器进行渲染的数据量为 $d_i - h_i$

每个客户端被分配到的数据中心 id 和服务器 id 记为 p_{sc_i}, ps_i

数据中心中心集合: $\{sc_1, \dots, sc_m\}$

对于数据中心 sc_j , 租用服务器数量为 num_j , 服务器租用价格为 pr_j 服务器集合 $\{s_j^1, \dots, s_j^{num_j}\}$,

对于服务器 s_j^k , 当前工作负载记为 w_j^k

第二节 问题描述

本文提出的新型渲染架构中, 可以将渲染负载在云端和客户端进行任意分配。客户端将满足自身渲染负载的部分放在本地渲染, 其余部分使用服务器资源进行云渲染。本文的服务器分配目标是在满足带宽阈值的情况下优化云游戏的两个核心问题——延迟和成本, 即最小化云端压力。具体而言, 是从云游戏服务提供商的角度来最小化云服务器总成本和服务器与客户端延迟的加权和。

每个客户端分配后的开销分为传输开销和服务端开销：

$$TransCost = \sum_{i=1}^n (d_i - h_i) * cost_i^{psc_i}$$

$$ServerCost = \sum_{j=1}^m num_j * pr_j$$

总成本即为：

$$Cost = \sum_{i=1}^n (d_i - h_i) * cost_i^{psc_i} + \sum_{j=1}^m num_j * pr_j$$

客户端分配后的与服务器的响应延迟为 $l_i^{psc_i}$ ，总响应延迟为：

$$Latency = \sum_{i=1}^n l_i^{psc_i}$$

对数据进行归一化处理，则

$$Cost' = \frac{Cost - Cost_{min}}{Cost_{max} - Cost_{min}}$$

$$Latency' = \frac{Latency - Latency_{min}}{Latency_{max} - Latency_{min}}$$

本文的问题可表示为，满足带宽阈值约束 $\forall i \in [1, n], j \in [1, m], b_i^j < BW$ 前提下，对服务器进行分配，使得：

$$\min(\alpha * Cost' + \beta * Latency')$$

其中 $\alpha + \beta = 1$

第三节 NP 难解性

第四章 算法设计

第一节 基础算法

本实验选取两种极端情况下的分配作为基准算法：

4.1.1 最低组合价算法 **Lowest-Combined-Price**

当问题定义中成本权重 α 较大时，在满足延迟约束的条件下忽略降低延迟目标，以最低组合价为分配目标，即：

$$\min(TransCost + ServerCost)$$

客户端每发送一个 MCG 会话请求，服务器端对该会话进行摊销成本计算，分配至最低成本的数据中心，步骤如下：

- 遍历数据中心，筛选满足延迟阈值约束的数据中心集合 D
- 将集合 D 按 $cost_i^j$ 升序排序
- 遍历集合 D ，若当前数据中心 sc_j 有已开启且剩余工作负载足够的服务器，直接分配至该服务器
- 若不满足条件，计算 $(d_i - h_i) * (cost_i^{j+1} - cost_i^j) - pr_j * (\frac{d_i - h_i}{WL})$ ，若结果为负，则租用新服务器的均摊开销小于分配至下一个数据中心的传输成本增量，在当前数据中心开启新服务器并分配即可。反之则不在当前数据中心进行分配，继续遍历。

4.1.2 最低延迟算法 **Lowest-Latency**

当问题定义中成本权重 α 较小时，我们以降低延迟为目标，将问题简化为：
 $\min(Latency)$

客户端每发送一个 MCG 会话请求，服务器端对该会话进行预期延迟计算，分配至最低延迟的数据中心，步骤如下：

- 遍历数据中心，筛选满足延迟阈值约束的数据中心集合 D
- 将集合 D 按 l_i^j 升序排序

- 遍历集合 D ，若当前数据中心 sc_j 有已开启且剩余工作负载足够的服务器，直接分配至该服务器，否则开启新服务器并分配。

第二节 启发式算法

4.2.1 Hill-Climbing Assignment

记 C_j 为对于数据中心 sc_j 满足约束条件的客户端集合，即可分配至 sc_j 的客户端集合。在每一轮分配过程中，考虑在每一个服务中心 sc_j 开启一个新的服务器，并分配一组客户端 $E_j \subseteq C_j$ 至该服务器。假设 C_j 依据传输延迟升序排序，那么 E_j 应为 C_j 的前 n 项，使得客户端摊销成本最小：

$$n = \operatorname{argmin}_{1 \leq m \leq |C_j|} \frac{pr_j + \sum_{i=1}^m cost_i^j * (d_i - h_i)}{m}$$

摊销成本记为：

$$\alpha_j = \frac{pr_j + \sum_{c_i \in E_j} cost_i^j * (d_i - h_i)}{|E_j|}$$

在每次迭代中，选择每个客户端的最低摊销成本选项，并将选定的客户端组分配给相应的数据中心。该算法一直进行到所有客户端都分配完为止。该算法称为最低摊销成本启发式算法。

- 对每个数据中心 sc_j ，初始化集合 C_j 并按传输数据量升序排序
- 计算满足摊销成本最小的 n 值与相应的摊销成本
- 找到 $sc_k = \operatorname{argmin}_{\{sc_j \in SC, C_j \neq \emptyset\}} \alpha_j$
- 对于每个数据中心，更新 C_j
- 若所有数据中心 $C_j = \emptyset$ ，分配完成，否则循环至第二步

4.2.2 Simulated-Annealing Assignment

每一轮迭代我们选择一个数据中心开启一个新服务器，记 C_j 为对于数据中心 sc_j 而言满足满足延迟阈值与工作负载阈值约束的客户端集合，为了便于理解，我们将分配情况表示为 $|C_j|$ 位二进制数 res ，第 i 为 1 表示客户端 i 分配给当前数据中心。

评价函数为摊销成本和平均传输延迟归一化后的加权和:

$$F = \alpha * \frac{Cost - Cost_{min}}{Cost_{max} - Cost_{min}} + \beta * \frac{Latency - Latency_{min}}{Latency_{max} - Latency_{min}}$$

$$Cost = \frac{\sum_{i=1}^{|CL|} (d_i - h_i) * cost_i^{psc_i} + \sum_{j=1}^m num_j * pr_j}{|CL|}$$

$$Latency = \frac{\sum_{i=1}^{|CL|} l_i^{psc_i}}{|CL|}$$

记已分配的服务器集合为 CL ，归一化时最大和最小值均由基准算法求出。

每一轮迭代需要选择一个数据中心开启一个新服务器，用模拟退火算法来寻找该近似最优解，过程如下：

- 遍历数据中心集合 SC ，对于当前数据中心需要计算近似最优分配
- 选择一个较大的初始温度 T_0 和下降速率 k
- 随机产生一个初始解 x_0 ，计算当前的评价函数 F
- 令 $T = kT$ ，对当前解 x_t 施加扰动，在领域内产生新解 x_{t+1} ，计算 $\Delta F = F(x_{t+1}) - F(x_t)$
- 若 $\Delta F < 0$ ，接受新解作为当前解，否则按概率 $e^{-\frac{\Delta F}{kT}}$ 判断是否接受新解
- 在温度 T 下，重复 L 次扰动和接受过程
- 判断温度是否达到终止温度水平，若是则继续遍历下一个数据中心，否则返回步骤 4
- 在所有数据中心求出近似最优解对应的函数值 F 后，选择值最小的数据中心开启新的服务器，完成当前迭代

第五章 实验评估

第六章 结论

致 谢