Optimizing parallel rendering on distributed cloud

Xingyuan XUE
Ensae Paris Tech
Paris, France
xingyuan.xue@ensae.fr

Yanik Ngoko
Qarnot Computing
Montrouge, France
yanik.ngoko@qarnot-computing.com

***Abstract*—**

***Keywords*-Parallel rendering; Load balancing; Workload estimation**

I. Introduction

II. Model

A. The Qarnot rendering platform

Qarnot cloud (Q.rads, Q.ware, Q.render)

B. Ray tracing rendering

theoretical formulation
Existing implemenation in Q.render

C. Optimization problem

We only have square decomposition. We want to compute more balanced decompositions in exploring rectangular decompositions. Add illustrations with different splitting runtime (examples you have)

III. An adaptive approach

IV. Principle

Explain that we will make a workload estimation and then we will compute the balanced decomposition
Picture that summarizes the approach

A. Implementation model

Explain how the principle is implemented with Q.ware. Mention software components to develop.
Explain the limits of the model (the scheduling will not necessarily be optimal ...)

B. Workload estimation

Define the problem (given a piece of an image, we want to know what is the runtime of the rendering)
Define the method to solve the problem (Fine grain decomposition of the image and then make a noisy rendering to estimate the duration)
Say why the method could work (Monte Carlo theory ...)
Say what are the limits:
- what is the noise level we consider for rendering? This is the
old dilema between exploration and exploitation. - What is the right decomposition grain?

C. Load balancing

Define the problem (we have an estimation of a fine grain decomposition, what is the best partitioning in N rectangles?)
Recall that it is a classical NP hard problems
Say that there exist several algorithms. But they are not efficient to capture irregular decompositions.
Present the Tetris algorithm as a solution for more irregular decompositions
Put the pseudo code

V. Experimental evaluation

Present the dataset
Results of the file prepared for the Blender conf
Analyze the strong and weak points of Tetris
Strong: Explain that in general tetris is more interesting when we have more rectangles or when the optimal decomposition has a too irregular shape. Show some shapes (decomposition) that cannot be captured by the other algorithms.
Weak points: Show that we tend to have smaller rectangles at the top. Illustrate on an example.
Recommend the perfect image for Tetris...

VI. Related works

Existing works on workload estimation.
Existing works on partitioning.

VII. Conclusion