doi:10.20396/revpibic.v0i0.id

Using CPU-GPU Cloud Clusters as an OpenMP Accelerator

Matheus Mortatti, Hervé Yviquel, Guido Araújo

Abstract
Rendering an image from a 3D scene requires a large amount of computation and designers need to use high-performance computer systems to keep the rendering time acceptable. They usually rent computing power directly from cloud service providers (e.g. AWS and Azure). However, integrating them within the standard workflow of modeling softwares can become a complex task. This project uses an extension of OpenMP 4.X [1-2] to eliminate any major interactions with the end-user, minimizing the complexity of cloud integration and optimizing the workflow. It applies such approach to a ray-tracing application, a simplified version of the engines used by professional 3D modeling software (e.g. Blender [3]). It automatically offloads the rendering process from the user computer to computer cluster within the Microsoft Azure cloud, brings the resulting images back after the computation ends and displays them directly on the screen of the user computer, thus providing a transparent programming model and good speed-ups over local execution.

Key words:
Rendering, OpenMP, Cloud

Introduction

The technology behind production-grade animated films, such as the ones produced by Pixar Animation Studios, is available to the public, but it usually requires high-performance computer systems, called render farms, to generate high quality scenes in reasonable time. Cloud providers, such as Amazon AWS, offer dedicated services to setup custom render farms within their datacenters. However, the workflow required to integrate the cloud is still far from being user-friendly, making it hardly accessible to the common graphic designer. To address such problem, we have previously developed a novel and yet compatible extension of OpenMP 4.X, called OmpCloud [1-2], that can be used to perform automatic computation parallelization and cloud offloading to render farms from 3D modelling software. Using a ray-tracing algorithm, it was possible to emulate and test the rendering work-flow, validating the use of OmpCloud as a solution to achieve easy offloading and great speed-ups over local rendering.

Results and Discussion

The Ray-Tracer was tested with two different configurations: the standard multi-threaded OpenMP implementation executed on the local computer, and the OmpCloud implementation which offloads the computation from the laptop to the remote cluster. To properly compare the performance between local and cloud rendering, we created three sets of benchmarks, each increasingly heavy on the rendering device (named Benchmin, BenchMid, and BenchMax respectively).

The execution times of the different dataset for all implementation are presented in Chart 1. For the serial execution, *BenchMin*, *BenchMid* and *BenchMax* took about 8 min, 44 min and 91 min respectively. Result shows that the serial execution is always slower than the cloud offloading one, even including the extra overhead due to the communication and the initialization of Spark.

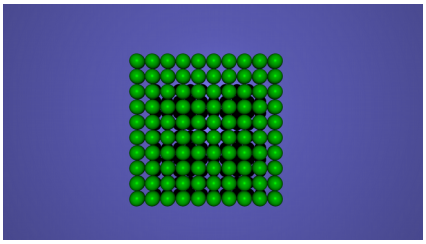
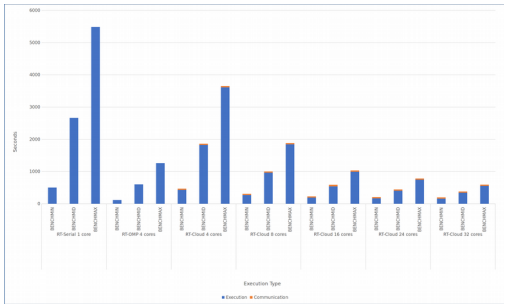


Image 1. Ray-Tracer Image Result.

Chart 1. Execution Time



Conclusions

Upon analyzing the results and having the experience of implementing OmpCloud support in a real world situation, it is safe to say that, OmpCloud does provide a simple workflow for the arguably hard existing solutions for cloud offloading. With only a few lines of code and no more required experience than knowing the basics of OpenMP, it was indeed possible to achieve great speedups on a graphical heavy application, coming closer to a future when cloud resources could be easily programmable.

---

[1] H. Yviquel and G. Araujo, "The Cloud as an OpenMP Offloading Device," inThe 46th International Conference on Parallel Processing (ICPP), 2017.
[2] H. Yviquel, L. Cruz, and G. Araujo, "Cluster Programming using the OpenMP Accelerator Model," ACM Transactions on Architecture and Code Optimization (TACO) – Accepted, 2018.
[3] Blender: free and open source 3D creation suite." [Online]. Available: https://www.blender.org/