

Blender based Rendering-as-a-Service Platform for High Performance Computing Clusters

Milan Jaros
IT4Innovations

VSB - Technical University of Ostrava,
Ostrava, Czech Republic
milan.jaros@vsb.cz

Petr Strakos
IT4Innovations

VSB - Technical University of Ostrava,
Ostrava, Czech Republic
petr.strakos@vsb.cz

Lubomir Riha
IT4Innovations

VSB - Technical University of Ostrava,
Ostrava, Czech Republic
lubomir.riha@vsb.cz

1 INTRODUCTION

Open-source rendering solutions that are capable of utilizing a large number of computational resources are rare. Usually they are made for render farms but not directly for High Performance Computing (HPC) clusters. One of such solutions is Flamenco [web 2018c] that is a render manager based on Blender 3D creation suite [web 2018a]. It supports offline rendering only and because it is based on standard distribution of Blender the parallelization potential of rendering is limited.

We are currently developing a rendering-as-a-service platform that efficiently utilizes HPC resources in the supercomputing centres. HPC cluster can be equipped with classical CPU nodes or accelerated nodes or their combination. We decided to build our solution on Blender since it gathers on popularity and offers realistic renderers and variety of extension possibilities in terms of plug-ins.

2 OUR APPROACH

Our platform is based on Blender renderers and upgrades them with HPC technologies such as: (1) MPI for distributed rendering, (2) OpenMP for parallelization for multi-core CPUs, (3) support for modern architectures such as Intel Xeon Phi in form of accelerator or stand-alone processor which include extended support for wide SIMD units (up to 512 bits), (4) integration of EMBREE library [web 2018b] into Blender Cycles, (5) integration of OpenSWR library [web 2018d] into Blender Eevee. In this way we can offer not only standard offline but also interactive rendering mode which relies on fast HPC interconnecting networks.

In Fig 1 we show the conceptual scheme of the whole rendering-as-a-service platform. Based on this we can divide the platform into two main parts, which is web based or application based front end and HPC rendering back end. The front end serves for the visualization of results while the back end represents the computational part. Within the back end we can distinguish another sub-parts, specifically the HPC task scheduler, offline/interactive renderer and module for the user plug-ins support.

In Fig 1 the already developed parts are highlighted by blue while the parts that are under development are in red.

3 RESULTS

Performance evaluation of the modified version of Cycles in terms of rendering time in offline mode for different scenes (general Blender benchmark scenes) and different processor architectures is presented in Fig 2. The results show promising performance on new Intel Xeon Phi 7250 (68 CPU cores) and Intel Xeon 8160 (24

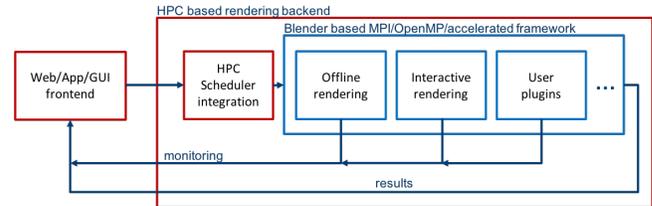


Figure 1: Rendering-as-a-service platform using HPC and Blender

CPU cores) architectures both with 512 bit wide SIMD units. This shows the parallel efficiency of our implementation.

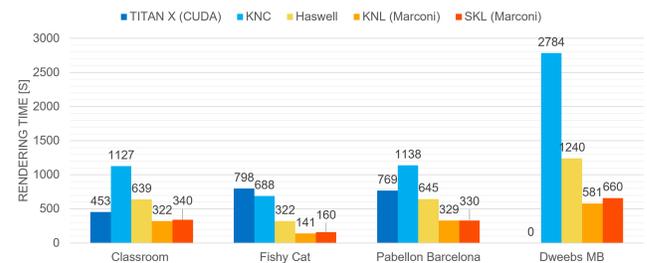


Figure 2: Salomon IT4Innovations: Intel Xeon E5-2680v3 (Haswell), Intel Xeon Phi 7120P (KNC), NVIDIA GeForce GTX TITAN X. Marconi CINECA: Intel Xeon Phi 7250 (KNL), Intel Xeon 8160 (SKL)

ACKNOWLEDGMENTS

This work was supported by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center - LM2015070".

REFERENCES

- 2018a. blender.org - Home of the Blender project - Free and Open 3D Creation Software. (5 2018). <https://www.blender.org/>
- 2018b. Embree. (4 2018). <https://embree.github.io/>
- 2018c. Flamenco. (5 2018). <https://www.flamenco.io/>
- 2018d. OpenSWR. (6 2018). <http://openswr.org/>