Assignment 4

Duc Minh Le (Tom Le)

ID: 7871324

University of Manitoba

COMP 4190 – Artificial Intelligence

Dr. Cuneyt G. Akcora

# Introduction and Motivation

Currently, computer graphics are developing at a very fast pace, faster than what our hardware are capable of, unfortunately. Technologies such as real time ray tracing are becoming more prominent in video games. These per frame expensive ray tracing calculations on complex scenes would no doubt hamper the performance of even the most powerful GPU available. To maintain performance while also not compromising quality, usage of real time deep learning in graphics upscaling are more referable. Such examples are Nvidia’s [Deep Learning Super Sampling (DSLL)](https://www.nvidia.com/en-us/geforce/news/graphics-reinvented-new-technologies-in-rtx-graphics-cards/#dlss), open-source [FidelityFX Super Resolution](https://www.amd.com/en/technologies/radeon-software-fidelityfx) standard pioneered by AMD, and the upcoming [XeSS (Xe Super Sampling)](https://www.intel.com/content/www/us/en/architecture-and-technology/visual-technology/arc-discrete-graphics/xess.html) by Intel. This report will be focusing on Nvidia’s DLSS, though the problems would be in common.

# Problem Statement

Not all games support DLSS, pre-exist training data of a game need to be manually fed to Nvidia’s AI platforms. Along with this, real time data of said games are collected and fed to the user’s local GPU. As such to support more games, extensive training is needed, this report aims to improve the training time of custom-designed AI platforms such as Nvidia’s with a reinforcement learning based GPU scheduler. Minimizing some challenges of the standard EIU’s cost effective DL platform such as inaccurate prediction model and heavy overhead of offline characterization/profiling.

# Solution

We will take a look at the [RIFLING](https://search.lib.umanitoba.ca/discovery/fulldisplay?docid=cdi_scopus_primary_2014617353&context=PC&vid=01UMB_INST:UMB&lang=en&search_scope=MyInst_and_CI&adaptor=Primo%20Central&tab=Everything&query=any,contains,Deep%20learning%20GPU&facet=searchcreationdate,include,2020%7C,%7C2022&offset=0) GPU scheduler proposed in this article.

Reinforcement learning (RL) has been proven as a general tool for intelligent decision-making in various application fields, and has been extended to the field of system designs. The nature of RL is to allow an agent (i.e., the scheduler in this article) to model the environment (i.e., GPU cluster)by trial and error, make action decisions (i.e., task scheduling and placement) dynamically and learn from its own behaviour based on feedback (i.e., task performance) from the environment, with the aim of achieving the maximum reward(i.e., scheduling objectives: system efficiency first, average task performance as best-effort delivery). As the running time increases, the RL agent improves the modeling environment and makes progressively better action decisions. Q-learning algorithm, which is a typical method, has the rapid speed of convergence and high sample utilization efficiency. Therefore, we adopt Q-learning as our scheduling framework, main components include the state space, action space, reward function, and updating scheme.

# Experimental Results

Table 2 shows the specifications of the DL platform use in this test,

Graphical user interface, text, application

Description automatically generated with medium confidence

1. Evaluation  
   The RL algorithm consists of the exploring phase and the exploiting phase
   1. Exploring phase  
      Chart

      Description automatically generated  
      From this figure, RIFLING without grouping algorithm has both a lower learning efficiency and taskcompletion speed than RIFLING, we can conclude that the grouping algorithm plays an important role in thisRL-based scheduler.
   2. Exploiting phase  
      Chart, scatter chart

      Description automatically generated  
      Compared with randomized policy and MinCut, GENIE and RIFLING achieve higher task completion speed due to their prior knowledge of DL workloads. As the results show, RIFLING can achieve an improvement of up to 19.6% in average normalized processing rate and a 12.9% reduction in make span relative to the best baselines.
2. Large-scale simulation  
   Chart, bar chart

   Description automatically generated  
   Compared with the low task density trace (density=10 tasks/h), RIFLING, which makes full use of the limited resource, is much more competitive and powerful when processing Philly Trace (density≈=35 tasks/h). Relative to other baselines, RIFLING achieves an a reduction of 47.8% in make span and improvement of up to 17.7% in average normalized processing rate. According to the evaluation results, the RL-based scheduler demonstrates the good scalability on larger distributed GPU platforms.  
   Graphical user interface

   Description automatically generated
3. Sensitivity analysis  
     
   *w* is a threshold of centroid difference to decide whether to merge two groups. Setting a too high or too low threshold may lead to an inappropriate state space mapping, *v* is a interval of the number of workloads between the two regrouping. That is, v represents the frequency of regrouping. The results from figure 11 show that optimal combination is w = 5% and v = 20.

𝜃 is the weight factor used to adjust the impactsof resource occupancy concentration in the entire scheduling phase. In real DL scenario, setting it too high or too low, may cause resource fragmentation. From figure 12, 𝜃=0.4 is a optimal choice

Graphical user interface, diagram

Description automatically generated

# Conclusions and further works

The RL-based online GPU scheduler RFLING has been proving to be a lightweight, scalable, able to achieve high task performance and improved system performance. Thus, it should be a great fit for large scale deep learning platform such as Nvidia’s. Performance and effectiveness are critical as the more bet we can render computer graphics the closer we are to an actual “metaverse”.

# References

[RIFLING: A reinforcement learning‐based GPU scheduler for deep learning research and development platforms](https://search.lib.umanitoba.ca/discovery/fulldisplay?docid=cdi_scopus_primary_2014617353&context=PC&vid=01UMB_INST:UMB&lang=en&search_scope=MyInst_and_CI&adaptor=Primo%20Central&tab=Everything&query=any,contains,Deep%20learning%20GPU&facet=searchcreationdate,include,2020%7C,%7C2022&offset=0)

[Nvidia's Deep Learning Super Sampling (DSLL)](https://www.nvidia.com/en-us/geforce/news/graphics-reinvented-new-technologies-in-rtx-graphics-cards/#dlss)

[NVIDIA DLSS 2.3](https://www.nvidia.com/en-us/geforce/news/nvidia-image-scaler-dlss-rtx-november-2021-updates/)

[AMD's FidelityFX Super Resolution](https://www.amd.com/en/technologies/radeon-software-fidelityfx)

[Intel's XeSS (Xe Super Sampling)](https://www.intel.com/content/www/us/en/architecture-and-technology/visual-technology/arc-discrete-graphics/xess.html)