

РАЗРАБОТКА МЕТОДА ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ НА ОСНОВЕ ШАБЛОНА ПРОЕКТИРОВАНИЯ ENTITY COMPONENT SYSTEM

Тугушев Т.Р. (Университет ИТМО)

Научный руководитель – доцент Загарских А.С.
(Университет ИТМО)

Введение. В настоящее время компьютерные игры стали неотъемлемой частью современной массовой культуры, завоевав сердца сотни миллионов игроков по всему миру. Однако, с возрастающей популярностью видеоигр семимильными шагами возрастают и требования к вычислительным ресурсам. Игроки жаждут наилучшего игрового опыта: более реалистичной графики, высочайшей частоты кадров, отсутствия каких-либо тормозов и максимального погружения в игровые вселенные.

По мере развития компьютерных технологий, постепенно совершенствовались принципы и методы написания наиболее эффективного и производительного программного обеспечения. Одним из них является шаблон проектирования Entity Component System, позволяющий жестко разделять данные и поведение между собой путем композиции. ECS применяется как в игровой индустрии для реализации сложной и в то же время гибкой и расширяемой внутриигровой логики, так и во множестве других областей. Множеством независимых исследований было доказано превосходство подходов, лежащих в основе архитектурного паттерна ECS, над классическими подходами как по производительности, так и по использованию ресурсов центрального процессора [1, 2, 3]. В то же время, его применение в целях реализации вычислений на графическом процессоре, в том числе и для компьютерной графики, еще не так хорошо изучено.

Реализация эффективного и высокопроизводительного метода вычислений с учетом специфики работы современных графических процессоров с использованием шаблона проектирования ECS довольно актуально, поскольку применение данного шаблона проектирования уже позволяет как повысить производительность, так и упростить разработку, тестирование и поддержку высокопроизводительных вычислений.

Основная часть. Существует два основных, классических метода реализации вычислений на графическом процессоре: shader (или шейдер) предназначен для изменения поведения отдельных этапов графического конвейера, в то время как kernel (или ядро) – для реализации неспециализированных вычислений на графических процессорах (тех, что напрямую не связаны с компьютерной графикой).

Важнейшими критериями эффективности данных методов в контексте высокопроизводительных вычислений являются latency (или задержка) и throughput (или пропускная способность). В то же время, измерение computational & memory complexity с применением Threaded Many-core Memory (TMM) model позволяет определить ключевой фактор, ограничивающий производительность конкретного алгоритма, будь то вычислительная мощность графического процессора либо же пропускная способность глобальной памяти графического процессора [4]. Еще одним немаловажным критерием является ossurance – отношение числа выделенных варпов (набора потоков, выполняющих одну и ту же инструкцию над разными наборами данных) на максимальное число варпов, которое теоретически может выделить графический процессор [5].

Одними из преимуществ использования шаблона проектирования Entity Component System являются четко определенный порядок выполнения отдельных систем и разделение данных от поведения, что позволяет достичь слабой связности программного кода. Путем использования большего числа информации о потоке исполнения систем и зависимостях между ними становится возможным оптимизировать распределение систем между очередями команд графического процессора с целью повышения производительности вычислений.

Выводы. Проведен анализ существующих методов высокопроизводительных вычислений на графическом процессоре, а также программных решений, которые используют шаблон проектирования ECS для организации подобных вычислений. Разработан метод для реализации высокопроизводительных вычислений на GPU с использованием шаблона проектирования ECS.

К тому же, были затронуты особенности и проблемы, возникшие при реализации данного метода, а также выявлены его достоинства и недостатки по сравнению с уже устоявшимися методами вычислений. Полученные результаты не только обобщают наше понимание об особенностях реализации высокопроизводительных вычислений на графических процессорах, но и предоставляют новые возможности для дальнейшего развития методов для реализации подобных вычислений.

Список использованных источников:

1. D. Wingqvist, F. Wickström, S. Memeti. Evaluating the performance of object-oriented and data-oriented design with multi-threading in game development // 2022 IEEE Games, Entertainment, Media Conference (GEM), St. Michael, Barbados. 2022. P. 1-6.
2. Walid Faryabi, Geir Mathisen. Data-oriented Design approach for processor intensive games // NTNU, 2018.
3. Fedoseev K., Askarbekuly N., Uzbekova A. E., Mazzara M. Application of Data-Oriented Design in Game Development // Journal of Physics: Conference Series. 2020. V. 1694. № 1. P. 12035.
4. Lin Ma, Roger D. Chamberlain, Kunal Agrawal, Chen Tian, Ziang Hu. Analysis of classic algorithms on highly-threaded many-core architectures // Future Generation Computer Systems, 82 – 2018. – P. 528 – 543.
5. Occupancy explained - AMD GPUOpen [Электронный ресурс]. – 2023. – URL: <https://gpuopen.com/learn/occupancy-explained/> (дата обращения 25.08.2024).