

УДК 004.272

МЕТОД АДАПТАЦИИ АЛГОРИТМОВ ДЛЯ ЭФФЕКТИВНОГО ИСПОЛНЕНИЯ НА GPU

Булавинцев В.Г. (ФГАОУ ВО Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики, г. Санкт-Петербург),

Научный руководитель – к.ф.-м.н., доцент Жданов Д.Д.

(ФГАОУ ВО Санкт-Петербургский Национальный Исследовательский

Университет Информационных Технологий, Механики и Оптики, г. Санкт-Петербург)

В работе предложен метод переноса и адаптации алгоритмов для эффективного исполнения на современных графических ускорителях общего назначения (GPU). Метод сформулирован как последовательность шагов представляющих алгоритм в виде графа потока управления, трансформации этого графа в форму дерева и соотнесения уровней дерева с организационными уровнями GPU. Применение метода продемонстрировано на примере адаптации для работы на GPU алгоритма DPLL для поиска с возвратом в задаче определения выполнимости булевых формул.

Введение. Современные графические ускорители общего назначения (GPU) показывают высокую эффективность в решении задач компьютерного зрения и при обработке «больших данных». Эти задачи относятся к классу «чрезвычайно параллельных» (англ. «embarrassingly parallel»), и хорошо соответствуют используемой в GPU архитектуре «одна команда, много данных» (ОКМД). Однако, адаптация алгоритмов к архитектуре ОКМД зачастую является нетривиальной задачей. И хотя на сегодняшний день в научной литературе можно найти описания успешных переносов различных классов алгоритмов на GPU, как правило, тематические исследовательские статьи фокусируются на отдельных узких аспектах процесса адаптации того или иного алгоритма. Соответствующий материал в учебных пособиях по программированию GPU зачастую разрознен и не формализован. В настоящей работе авторы предлагают формализацию метода переноса произвольных алгоритмов на GPU. Метод основан на сопоставлении графа потока управления алгоритма архитектуре GPU.

Основная часть.

Основная идея метода состоит в том, чтобы представить адаптируемый алгоритм в виде иерархии вложенных циклов и эффективно сопоставить ее уровни уровням организации вычислительных блоков GPU. Метод состоит из следующих шагов:

1. Построить для адаптируемого алгоритма граф потока управления.
2. Представить полученный граф в виде дерева циклов.
3. Для каждого узла дерева попытаться подобрать существующую библиотечную реализацию для GPU.
4. Для оставшихся узлов выделить циклы с независимыми итерациями. При необходимости избавиться от зависимостей между итерациями применить эквивалентные преобразования решеточной модели.
5. Сопоставить уровни дерева организационным уровням вычислительного устройства. Начинать сопоставление «снизу-вверх», чтобы минимизировать расход памяти отдельными потоками. При этом учитывать специфические аппаратные ограничения, накладываемые конкретным уровнем устройства (например, выполнение условных переходов в ОКМД, загруженность шины памяти).
6. Если алгоритм использует двойную точность, после реализации попытаться перейти к одинарной точности.

Выводы. Предложенный метод применен к алгоритму для поиска с возвратом DPLL. Адаптированный для GPU алгоритм DPLL показывает при решении регулярных задач многократное преимущество в скорости перед тем же алгоритмом, запущенным на CPU.

Булавинцев В.Г. (автор)

Подпись

Жданов Д.Д. (научный руководитель)

Подпись