

## **Схема адаптации классических методов поиска экстремумов к задачам оптимизации GPGPU-программ**

Кривов М.А.<sup>(1,2)</sup>, Притула М.Н.<sup>(1)</sup>, Иванов П.С.<sup>(1,2)</sup>

<sup>(1)</sup>ООО «ТТГ Лабс»

<sup>(2)</sup>МГУ им. М.В. Ломоносова

При разработке программ для графических ускорителей, известных как GPU, проблемой становится не только специфичность реализованной в них массивно-параллельной архитектуры, но и необходимость проведения качественной оптимизации создаваемой программы.

Действительно, так как выбор в сторону использования GPU, в том числе в составе суперкомпьютерных систем, в подавляющем большинстве случаев делается, в первую очередь, ради существенного снижения времени расчётов, то вопросам оптимизации уделяется достаточно много внимания. С другой стороны, из-за упрощённой по сравнению с центральными процессорами архитектуры, а также более «молодого», и, как следствие, менее совершенного компилятора, любой недочёт программиста может существенно понизить скорость расчётов. Исходя из этих двух наблюдений, можно утверждать, что задача оптимизации GPGPU-программ является отдельной довольно серьёзной проблемой, к решению которой следует подходить обстоятельно.

К сожалению, в настоящий момент процесс оптимизации GPGPU-программ является крайне неформализованным. Обычно разработчик, исходя из собственного опыта или же руководствуясь рекомендациями из сторонних источников, каковыми могут быть документация от производителя GPU или известные Интернет-ресурсы, составляет список относительно стандартных действий, которые применимы к созданной программной реализации и осуществление которых способно повысить её скорость. Далее, он последовательно вносит требуемые изменения как в сам алгоритм, так и в его реализацию, после чего достигает определенного ускорения, величина которого зависит прежде всего от квалификации разработчика, но отчасти – и от простого везения.

В работе предпринята попытка частично формализовать данный процесс, сведя к минимуму влияние интуиции разработчика на результат его работы, а также немного уменьшив зависимость качества проведённой оптимизации от его опыта. С этой целью время работы GPGPU-программы рассматривается как некий функционал, для минимизации которого может быть применён любой классический метод оптимизации. Естественно, данная идея заставляет совершенно иначе взглянуть на процесс оптимизации программ, а также разработать схему однозначного, и в то же время адекватного отображения реальных свойств и характеристик программы в параметры математической модели, один из вариантов которого и будет представлен в докладе.