

Язык программирования E_q

— Аннотация —

Зайченков П.О.¹ и Шинкаров А.Ю.²

¹ Московский физико-технический институт, Кафедра информатики и
вычислительной техники

² University of Hertfordshire, Hatfield, Hertfordshire, AL10 9AB, United Kingdom

Большинство современных естественнонаучных проблем тесно сопряжено с огромным количеством вычислений. Класс вычислений может варьироваться от решения систем уравнений и моделирования физических процессов до обработки данных, полученных с телескопа, и исследования организации геномов. На данный момент, за редким исключением, большинство подобных вычислений реализовано на языках программирования подобных Фортрану. На то существует несколько причин: во-первых, обратная совместимость – не секрет, что метеорологические программы насчитывают миллионы строк кода, а первые версии появились в тот момент, когда Фортран был одним из самых передовых языков программирования; во-вторых, производительность – языки высокого уровня, такие как MatLab, Python, Java предоставляют высокий уровень абстракций, но редко могут состояться в скорости с языками низкого уровня; и, наконец, сложность разработки – Фортран оказался оптимальным компромиссом между скоростью разработки и скоростью выполнения готовой программы – более сложные оптимизации обычно требуют от программиста нетривиальных знаний об архитектуре и тонкостях того или иного языка.

Очевидно, что программы о которых идет речь выше, требуют огромной вычислительной мощности. На сегодняшний день тенденции в производстве компьютеров сменились с гонки за тактовой частотой на увеличение количества ядер процессора. Именно этим фактом обусловлен рост интереса к параллельному программированию. Немаловажную роль играет появление на рынке графических ускорителей GPGPU, предоставляющих еще большие возможности для увеличения скорости программ, однако требующие серьезного изменения парадигмы программирования.

В данной статье мы представляем язык программирования E_q, который позволит разделить усилия ученого, заинтересованного исключительно в результате определенных вычислений и программиста, разрабатывающего компилятор, соответствующий ситуации на современном рынке компьютеров. Основой синтаксиса для E_q является текстовый процессор L^AT_EX, являющийся стандартом для верстки научных публикаций. Основное преимущество выбранного подхода состоит в универсальности синтаксиса E_q: с одной стороны программа понимается текстовым процессором и имеет стандартный графический оттиск, с другой стороны, эта же программа компилируется на большинстве современных архитектур.

Для того, чтобы выполнить некоторые участки кода параллельно, необходимо либо указать на них явным образом, такой подход, к примеру, используется в библиотеках MPI и OpenMP, либо посредством некоторого анализа компилятор может выявить эти участки самостоятельно. В случае E_q используется второй подход, однако, в качестве подсказки, в синтаксисе имеется две важные конструкции – рекуррентное и параллельное выражение. Рекуррентное выражение – аналог нити исполнения (thread) в операционной системе, когда имеется некоторое окружение и последовательность действий, которая не может быть нарушена. Параллельное выражение – набор действий, который не имеет зависимостей, и все действия могут быть выполнены одновременно. Далее мы продемонстрируем, что двух данных конструкций, ветвления и атомарного выражения достаточно, чтобы, во-первых, записать любую программу на Фортране, во-вторых, принять решение о параллельности той или иной операции.