

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 естественнонаучных проблем сопряжено с огромным количеством вычислений, которые варьируются от решения систем уравнений до обработки данных, полученных с телескопа. Большинство подобных вычислений реализовано на языках программирования подобных Фортрану. На этот счет есть несколько причин. Во-первых, обратная совместимость – 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 не могут состязаться в скорости с языками низкого уровня; и, наконец, сложность разработки – Фортран оказался компромиссом между скоростью разработки и скоростью выполнения готовой программы. 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.

Язык программирования для вычислительных задач должен обеспечивать высокую производительность. Ключевой момент при вычислениях – разделение кода на два типа. Первый тип – независимые друг от друга блоки, которые могут выполняться параллельно. Другой тип – блоки, которые имеют зависимости, что означает, что инструкции должны выполняться в строго определенной последовательности, чтобы не нарушить зависимости между инструкциями. 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.