

## К сравнению многопоточного и автоматного параллельных подходов

В.С. Любченко, Ю.А. Тяжлов

В противовес многопоточному программированию существуют более эффективные параллельные программные технологии. В работе приведено сравнение многопоточного программирования с автоматным. О последнем, несмотря на относительно малый опыт его применения, можно сказать, что проектирование в целом и программирование в частности изменятся в лучшую сторону, если автоматы заложить в архитектуру процессоров, внедрить в операционную среду, сделать базисом языка программирования.

Сравнение проведено на примере задачи нахождения  $n$ -го числа ряда Фибоначчи. В этих целях рассматриваются программы на языке МС#, который представляет многопоточное программирование, и программами на С++, расширенном параллельной автоматной моделью. При этом программы на С++ используют виртуальную машину, реализующую параллельную автоматную модель.

На самом деле программирование на МС# достаточно слабо связано с многопоточной моделью. Просто потоки выбраны в качестве основы для реализации параллельных алгоритмов на данном языке. Однако такой выбор сказывается на реальных возможностях программ, написанных на МС#. Они получаются медлительными и имеют ограничение на число порождаемых параллельных процессов. И хотя формально программирование на языке МС# ничем не ограничено, но реально можно столкнуться с серьезными проблемами даже при решении относительно простых параллельных алгоритмов.

В работе описан переход от алгоритмов, представленных в форме обычных блок-схем, к алгоритмам в автоматной форме. И это может быть, в том числе, и основой для последующей аппаратной реализации алгоритмов. При этом здесь могут найти применение методы синтеза автоматов, которые известны и активно используются достаточно давно.

Сравнительное тестирование программ, созданных на рассматриваемых языках, убеждает насколько элегантно с точки зрения программиста рекурсивное решение может быть не эффективно по скорости работы в сравнении с обычным процедурным подходом. Поэтому нужно с осторожностью относиться к построению параллельных вычислений на базе модели функционального типа. Во-вторых, сравнивая эквивалентные алгоритмические решения, но базирующиеся на разных моделях вычислений, наглядно можно убедиться насколько многопоточное решение медленнее автоматного. Для рекурсивного алгоритма вычисления чисел Фибоначчи многопоточная модель при вычислении, например, 17-го числа более чем в 60 раз медленнее автоматной модели.

Автоматное программирование определяет парадигму, которая качественно отличается от используемой ныне в подавляющем большинстве случаев блок-схемной модели. Понять все возможности автоматной парадигмы и ощутить ее выгоды в полной мере можно даже на примере такой простой задачи, которая рассмотрена в работе. Полученные результаты, особенно сравнение по скорости работы, убеждают в необходимости поиска альтернативы довольно медленной многопоточной модели. В работе в качестве такой альтернативы рассматривается параллельная модель автоматного типа на базе классических автоматов Мили (заметим, что другие модели часто базируются на модели автомата Мура, что по нашему мнению не очень верный выбор). Последнее очень важно, т.к. позволяет использовать хорошо отработанную и проверенную временем теорию конечных автоматов и для рассматриваемой автоматной модели параллельного типа. А с учетом рассмотренных эквивалентных преобразований автоматов в блок-схемы и обратно и на модель блок-схемного типа.

Автоматное программирование в силу специфики базовой модели представляется в большей степени специализированным, чем программирование на МС#. Но многое определяется привычками. К примеру, отказ от потоков равносильно преодолению привычных стереотипов в программировании. Но, освоив автоматы, мы получаем в их лице надежную и технологичную модель параллельных вычислений.