

Применение параллельных вычислений для решения задачи анализа качества программного кода

С.В. Звездин

Разработка программного обеспечения является сложным процессом, на который человеческий фактор имеет большое влияние. Поэтому характеристики качества удобно выражать лингвистически. В этом случае для построения модели оценки качества кода используется математический аппарат нечеткой логики и нечеткие продукционные модели на основе нейронной сети Ванга-Менделя [1].

Одним из режимов работы модуля по оценке качества является анализ программного кода "на лету". Поэтому работа алгоритма может быть оптимизирована с точки зрения быстродействия путем распараллеливания алгоритма для работы на центральном и графических¹ процессорах.

Поскольку система анализа программного кода реализована на платформе Microsoft .NET 4.0, то для использования параллельных вычислений применяется библиотека TPL [2]. Для вычислений на графических процессорах были использованы библиотеки CUDA.NET, GPU.NET и Accelerator [3]. Сравнение эффективности двух алгоритмов производилось на испытательных площадках, оборудованных центральными процессорами Intel Core 2 Duo P8600 (2x2.4GHz) и Intel Core 2 Quad Q6600 (4x2.4GHz), и видео-адаптером NVIDIA 8800GT (112 потоковых процессоров). Результаты экспериментов приведены на рис. 1.

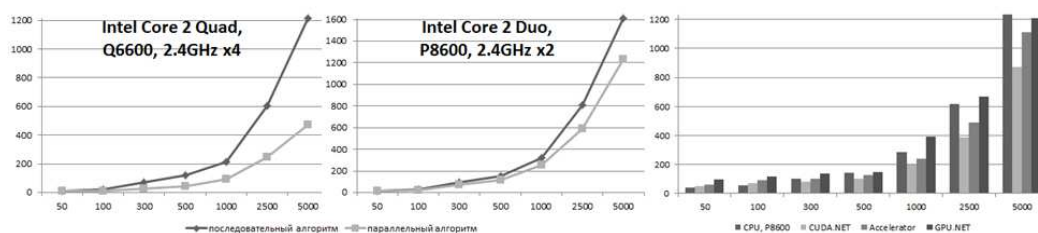


Рис. 1. Затраты времени на анализ программного кода (ось абсцисс - количество оцениваемых программных элементов; ось ординат - время работы алгоритма в мс.)

В результате применения параллельных вычислений на центральном процессоре наблюдается прирост быстродействия на многоядерных процессорах. Применение вычислений на графических процессорах обосновано только в случае анализа очень крупных проектов, содержащих большое количество программного кода (в противном случае можно наблюдать не увеличение, а снижение производительности из-за накладных расходов).

Литература

1. Звездин С.В. Метод определения качественных характеристик программного кода // Научная перспектива. Раздел "Технические науки". №10. 2010. С. 86-88.
2. Leijen D., Schulte W., Burckhardt S. The Design of a Task Parallel Library // ACM SIGPLAN Notices - OOPSLA'09. V. 44. №10. New York: ACM, 2009.
3. Banzhaf W., Harding S. Accelerating evolutionary computation with graphics processing units // GECCO'09 Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers. New York: ACM, 2009.

¹Работа алгоритма на графическом процессоре позволяет также снизить нагрузку на центральный процессор, что положительно влияет на производительность рабочего места в целом.