

Иерархический параллелизм – новая парадигма программирования

Л.Б. Соколинский

Южно-Уральский государственный университет, Челябинск

e-mail: sokolinsky@acm.org

Широкое распространение многоядерных процессоров, кластерных систем и технологии Grid обуславливает необходимость пересмотра парадигмы параллельного программирования. Традиционный подход к параллельному программированию был ориентирован на однородные многопроцессорные системы, представленные, главным образом, двумя основными классами: симметричные многопроцессорные системы с общей памятью (SMP) и многопроцессорные системы с массовым параллелизмом (MPP). Современные многопроцессорные системы имеют явно выраженный иерархический принцип построения. На первом уровне иерархии располагаются многоядерные процессоры. На втором – SMP-модули. Третий уровень представлен кластерными установками, объединяющими большое количество SMP-модулей с помощью высокоскоростной сети. На верхних уровнях иерархии располагаются Grid-системы, объединяющие с помощью локальной сети или Интернет несколько кластеров в одну вычислительную установку. Такой подход позволяет строить экономичные многопроцессорные системы, обладающие практически неограниченной мощностью. Подобные иерархические многопроцессорные системы характеризуются определенной однородностью внутри каждого уровня иерархии, однако различные уровни обладают разной скоростью обмена данными между процессорами (процессорными ядрами). Эта скорость может уменьшаться на несколько порядков при переходе от низких уровней к более высоким. В соответствии с этим возникает необходимость пересмотра традиционной парадигмы параллельного программирования. Действительно, алгоритм, разработанный для SMP-систем, может не учитывать наличие общей кэш-памяти, размещенной в многоядерном процессоре, и такой алгоритм, скорее всего, не будет эффективен для MPP-систем. В свою очередь, алгоритм, разработанный для MPP-системы, не будет эффективно использовать SMP-модули кластерной установки и многоядерные процессоры. Для эффективного использования современных многопроцессорных систем необходимо разрабатывать новые гибридные алгоритмы, структура которых будет отражать структуру много-

процессорных иерархий. Назовем такие алгоритмы *иерархическими*. Иерархический алгоритм должен поддерживать разбиение исходной задачи на иерархию подзадач. Подзадачи различных уровней могут решаться разными методами, каждый из которых в наибольшей степени соответствует определенному уровню многопроцессорной иерархии. В качестве примера рассмотрим метод, ориентированный на MPP-архитектуру (средний уровень иерархии).

Пусть $\varphi(x)$ — произвольное фейеровское отображение относительно множества M , $\varphi \in \{\mathbb{R}^n \rightarrow \mathbb{R}^n\}$. Разобьем вектор $x \in \mathbb{R}^n$ на r подвекторов $x = [x_1, \dots, x_r]$, где $x_i \in \mathbb{R}^{n_i}$ ($i = 1, \dots, r$) и $\mathbb{R}^n = \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_r}$. Обозначив через $\pi_i(x)$ проекцию x на \mathbb{R}^i , определим отображения $\varphi_i(x) = \pi_i(\varphi(x)) + \sum_{j \neq i} \pi_j(x)$ и положим $\bar{\varphi}(x) = \alpha \sum_{i=1}^r \pi_i(\varphi_i^t(x))$, $0 < \alpha < 1$ при некотором фиксированном натуральном t . При определенных условиях отображение $\bar{\varphi}$ будет M -фейеровским. Таким образом, на базе одного фейеровского отображения мы сконструировали другое, обладающее большим ресурсом параллелизма. Действительно, значения $\varphi_i^t(x)$ могут вычисляться независимо друг от друга для различных $i = 1, \dots, r$. При этом мы получаем две степени свободы для регулировки баланса загрузки процессорных узлов. Во-первых, увеличивая t , мы можем повышать вычислительную нагрузку на процессорный узел между двумя соседними итерациями, вычисляющими $\bar{\varphi}(x)$. Во-вторых, мы можем произвольным образом перераспределять координаты вектора x между процессорами.

Указанный метод был использован при разработке нового параллельного алгоритма на базе фейеровских отображений, предназначенного для решения задач линейного программирования на кластерных вычислительных системах. Проведенные вычислительные эксперименты подтвердили эффективность этого подхода [1].

Работа поддержана РФФИ (проект 06-01-00380) и Программой государственной поддержки ведущих научных школ (грант НШ-5595.2006.1).

ЛИТЕРАТУРА

- [1] Шелудько А.С., Соколинский Л.Б. Параллельная реализация S-технологии для решения задач линейного программирования // Параллельные вычислительные технологии: Тр. междунар. науч. конф. (29 января — 2 февраля 2007 г., Челябинск). — Челябинск: Изд-во ЮУрГУ. — 2007. — Т. 2. — С. 277-280.