

Решение задач глобальной многоэкстремальной оптимизации с использованием графических процессоров

С.И. Бастраков, Д.Б. Бражкин, А.Н. Половинкин

Многие практические задачи могут быть поставлены в форме задач глобальной многоэкстремальной оптимизации [1]. Алгоритмы, предназначенные для решения таких задач, являются вычислительно трудоемкими, поэтому актуальной является разработка параллельных методов поиска глобально-оптимальных решений. В данной работе предлагается параллельная реализация индексного метода [2], адаптированная под архитектуру графического процессора, с использованием смешанных CPU-GPU вычислений.

Реализация параллельной схемы работы связана с особенностями архитектуры графического процессора, а именно, разделением всех вычислительных ядер на группы по 8 ядер, называемых мультипроцессорами [3].

На каждом шаге итерации индексного метода распараллеливание реализовано схожим образом, поэтому рассмотрим в качестве примера один из шагов алгоритма – вычисление характеристик R_i для каждого интервала (x_{i-1}, x_i) , $1 \leq i \leq k+1$. В общем виде R_i представляет собой функцию $R(x_i, x_{i-1}, z_i, z_{i-1}, \mu_v)$, где x_i, x_{i-1} – координаты границ интервала, z_i, z_{i-1} – значения целевой функции в этих точках, а μ_v – нижняя оценка константы Липшица для функции либо одного из ограничений. Нетрудно видеть, что вычисления значений характеристики R_i происходят для каждого интервала независимо. Это позволяет применить достаточно простую схему распараллеливания: разделение всей области поиска минимума $(0, I)$ на несколько неперекрывающихся подобластей

$$\left(\frac{id}{I}, \frac{id+1}{I} \right), \quad (1)$$

где id – номер потока, а I – общее число потоков. Для эффективного использования особенностей архитектуры графического процессора при разделении на подобласти необходимо также учитывать номер блока потоков [3] (выполняемого на одном мультипроцессоре). Таким образом, схема принимает вид:

$$\left(\frac{b_{id} + \frac{t_{id}}{N}}{M}, \frac{b_{id} + \frac{t_{id}+1}{N}}{M} \right), \quad (2)$$

где t_{id} – номер потока в блоке, N – общее число потоков в блоке, b_{id} – номер блока, M – общее число блоков.

После вычисления характеристик R_i для всех интервалов (x_{i-1}, x_i) , $1 \leq i \leq k+1$ необходимо найти интервал с максимальной характеристикой. Поиск происходит также параллельно, причём сначала каждое вычислительное ядро выполняет поиск на подобласти (2), затем среди всех значений, полученных в блоке потоков, выбирается максимальное, затем аналогично ищется максимум среди всех блоков. Таким образом, применяется двухуровневая схема распараллеливания.

Литература

1. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. М.: Наука, 1978.
2. Strongin R.G., Sergeyev Ya.D. Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht, 2000.
3. CUDA 2.0 Programming Guide
[http://developer.download.nvidia.com/compute/cuda/2_0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf]