УДК 004.4

РАЗНОРОДНЫЙ БЛОЧНО-ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ УЧИТЫВАЕТ ОСОБЕННОСТИ МНОГОЯДЕРНОЙ АРХИТЕКТУРЫ

Прихожий А.А., Карасик О.Н.
Белорусский национальный технический университет

Алгоритм Флойда-Уоршелла решает задачу поиска кратчайших путей между всеми парами вершин взвешенного графа, однако не ориентирован на параллельную реализацию и эффективное использование иерархической памяти многоядерной системы. Задача о кратчайших путях широко применяется для решения реальных научно-технических задач, включая область искусственного интеллекта, компьютерных сетей, микроэлектроники и многих других, однако время ее решения алгоритмом Флойда-Уоршелла становится нереально большим для больших размеров графов. Вследствие этого в работе [1] предлагается блочный алгоритм, а в работе [2] – разнородный блочный алгоритм решения этой задачи.

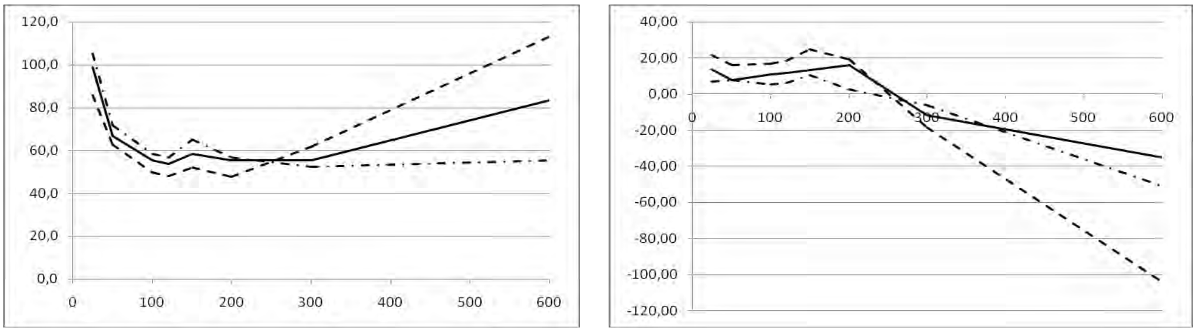
Псевдокод блочного алгоритма Флойда-Уоршелла показан на рис.1, где M – число блоков в строке матрицы кратчайших путей. Его функционирование описывается циклом, на каждой итерации которого выполняется упорядоченный однократный пересчет блоков алгоритмом *CalcBlock*. Разнородный блочный алгоритм различает четыре типа блоков (D0, C1, C2 и U3), реализуемых отдельными алгоритмами. Он ориентирован на параллельную реализацию на многоядерных системах. Все блоки C1 и C2 вычисляются взаимно параллельно, но последовательно с блоком типа D0. Блоки U3 также вычисляются взаимно параллельно, но последовательно с блоками C1 и C2. Всего блочный алгоритм вычисляет M^3 блоков.

Для графа из 9600 вершин рис.2а описывает зависимость процессорного времени от размера блока для трех алгоритмов: блочно параллельного Флойда-Уоршелла v0 (сплошная), разнородного версии v1 (пунктирная) и разнородного версии v2 (штрих пунктирная). Версии v1 отличаются алгоритмами вычисления блоков C1, C2 и U3 и работой с кэш. Алгоритм v1 дает наименьшее время работы до 200 строк в блоке, с увеличением размера блока лучшим становится v2. Рис.2б показывает выигрыш в процентах одного алгоритма у другого при вариациях размера блока. При малых размерах v1 выигрывает у v0 до 16.8% и выигрывает у v2 до 26.3%. Рост размера блока резко меняет картину. Алгоритм v2 начинает выигрывать у v1 более 100% и выигрывать у v0 более 50%. Устойчивый тренд показывает ожидаемый дальнейший рост выигрыша для v2 с увеличением размера графа.

Табл.1 сравнивает для четырех размеров графа алгоритмы v0, v1 и v2 по наименьшему на размерах блока времени выполнения. Алгоритм v1 выиграл у v0 до 12.48% и выиграл у v2 до 16.58%, однако с ростом размера графа выигрыш v1 у v2 устойчиво падает, и v2 становится лучшим.

```
Algorithm Blocked_FW {
    for  m=0…M-1 {
        CalBlock (B_{m,m},B_{m,m},B_{m,m});        // type D0
        for i=0…m-1 {
            CalBlock (B_{i,m},B_{i,m},B_{m,m});    // type C1
            CalBlock (B_{m,i},B_{m,m},B_{m,i});    // type C2
        };
        for i=m+1…M-1 {
            CalBlock (B_{i,m},B_{i,m},B_{m,m});    // type C1
            CalBlock (B_{m,i},B_{m,m},B_{m,i});    // type C2
        };
        for i=0…m-1 {
            for j=0…m–1
                CalBlock (B_{i,j},B_{i,m},B_{m,j});    // type U3
            for j=m+1…M–1
                CalBlock (B_{i,j},B_{i,m},B_{m,j});    // type U3
        };
        for i=m+1…M-1 {
            for j=0…m–1
                CalBlock (B_{i,j},B_{i,m},B_{m,j});    // type U3
            for j=m+1…M-1
                CalBlock (B_{i,j},B_{i,m},B_{m,j});    // type U3
        };
}}
```

Рисунок 1. Блочный алгоритм Флойда-Уоршелла



а) б)

Рисунок 2. Зависимость а) времени выполнения и б) ускорения в % одного алгоритма по сравнению с другим от размера блока для графа 9600 вершин

Таблица 1. Сравнение наилучших результатов для трех алгоритмов и четырех размеров графа

Алгоритм	Без привязки потоков к процессорам				С привязкой потоков к процессорам			
	1200	2400	4800	9600	1200	2400	4800	9600
v0 (сек)	0.114	0.861	6.789	53.68	0.115	0.858	6.822	53.66
v1 (сек)	0.110	0.801	6.136	47.77	0.113	0.788	6.241	47.71
v2 (сек)	0.128	0.924	6.946	52.18	0.127	0.918	6.806	52.22
v0/v1	3.75%	7.55%	10.64%	12.37%	2.07%	8.99%	9.30%	12.48%
v2/v1	15.90%	15.36%	13.20%	9.22%	12.75%	16.58%	9.05%	9.46%

Литература

Venkataraman, G. A Blocked All-Pairs Shortest Paths Algorithm / G. Venkataraman, S. Sahni, S. Mukhopadhyaya // Journal of Experimental Algorithmics (JEA), Vol 8, 2003, pp. 857-874.

Прихожий, А.А. Разнородный блочный алгоритм поиска кратчайших путей между всеми парами вершин графа / А.А. Прихожий, О.Н. Карасик // Системный анализ и прикладная информатика, 2017, №3, с.68-75.