

БІБЛІОТЕКИ РОЗБИТТЯ ГРАФІВ METIS I PARMETIS

Шепель О.І., Святний А.В.

Донецький національний технічний університет

В цій доповіді розглядаються базові алгоритми бібліотек METIS і PARMETIS. Ці дві бібліотеки були обрані серед аналогічних як найбільш розповсюджені на сьогоднішній час. У цих програмних продуктах реалізовано клас так званих багаторівневих алгоритмів, які показали дуже хороші результати за швидкістю їх роботи та якості розділів, які вони створюють, навіть у порівнянні зі спектральними алгоритмами, які доволі довгий час були лідерами за всіма цими параметрами.

1. Постанова задачі розбиття.

Ми маємо граф $G=(V,E)$; $|V|=n$, де V – це множина вершин графа, а E – множина його ребер, n - кількість вершин. Треба розбити граф на k частин V_1, V_2, \dots, V_k , щоб $V_i \cap V_j$ при $i \neq j$. $|V_i|=n/k$, $\cup_i V_i=V$.

Важливою умовою є мінімізація кількості ребер між розділами графа. Результатом розбиття являється вектор $P[1..n]$, де $P[v]$ приймає значення від 1 до k , $v \in V$. Тобто значення елементів цього вектора вказують до якого розділу належить поточна вершина. Треба зауважити, що при рішенні більшості графових задач у яких треба застосовувати розбиття ми маємо справу з доволі великими графами (кількість вершин більш ніж 100000, а кількість ребер може досягати 1000000 та навіть більше).

2. Описання алгоритму роботи бібліотеки METIS

Бібліотека METIS [1] базується на k -way-algorithm розбиття графів. Він поділяється на три етапи: мінімізація графа до декількох сотень вершин (coarsening - стадія), якщо це потрібно; далі граф розбивається на k частин і третя, остання, стадія – це збільшення розбитого графу до первісного розміру (uncoarsening - стадія), при цьому на кожному рівні збільшення графа відбувається мінімізація кількості ребер між розділами. Мінімізація робиться на uncoarsening - стадії тому, що при поетапному збільшенні графа збільшується і ступень свободи кожного розділу, що дає більше варіантів для мінімізації.

Під час мінімізації графа ми послідовно отримуємо менший похідний граф $G_i=(V_i, E_i)$, з первинного графу $G_0=(V_0, E_0)$, причому $|V_i| < |V_{i-1}|$, де i – рівень мінімізації. Принцип мінімізації графа полягає в наступному: набір суміжних вершин об'єднується в одну при наступному рівні мінімізації. Причому вага мінімізованої вершини дорівнює сумі ваг вершин, які об'єдналися у мінімізовану. Зв'язність мінімізованої вершини теж дорівнює сумі зв'язностей первісних вершин. Мінімізація графа завершується, коли кількість вершин у похідному графі досягає заданого значення.

Є декілька алгоритмів мінімізації графа: Random Matching (RM) та Heavy Edge Matching (HEM). Розглянемо кожний з цих алгоритмів:

Алгоритм Random Matching (RM) полягає у наступному: об'єднанню підлягають дві інцидентні вершини які, вибираються випадково, якщо кожна з них не була обрана раніше.

Алгоритм Heavy Edge Matching (HEM). Як і у попередньому випадку, ми випадково вибираємо вершину, але іншу вершину, яку ставимо у відповідність першій ми не шукаємо також випадково, а перебираємо усі вершини, які інцидентні обраній, та

вибираємо ту, зв'язність якої з першою є максимальна. Таким чином досягається оптимізація по зв'язності.

Фаза розбиття графа найпростіша з усіх трьох фаз алгоритму. При розбиванні мінімізованого графа треба виконати всього три умови:

Кількість розділів повинна відповідати заданому значенню;

Сумарна вага вершин розділу повинна бути приблизно однакова;

Кількість зв'язків між розділами графа повинна бути приблизно однакова, та по можливості мінімальна.

Для розбиття графа на частини використовують алгоритм під назвою KwayRecursiveBisection, іншими словами багаторівневий рекурсивний алгоритм поділу пополам. Граф розбивається навпіл, потім ці дві половини б'ються ще навпіл, і так доки ми не отримаємо потрібну кількість розділів.

Далі настає етап збільшення графа (uncoarsening - фаза), у процесі якого виконується розгортання його вершин у розділах $P_m: G_{m-1}, G_{m-2}, \dots, G_1$, що і зумовлює його збільшення до первісних розмірів. При цьому на кожному етапі збільшення використовується клас алгоритмів локального покращення. Ці алгоритми базуються на Kernighan-Lin (KL), який переміщує вершини між розділами графа G_i , покращуючи якість розбиття, доки не буде досягнуто локальний мінімум графа G_i . На цьому покращення на i -му рівні закінчується, та виконується перехід на наступний $i-1$ рівень, і т.д.

3. Описання алгоритму роботи бібліотеки PARMETIS

Алгоритм роботи даної бібліотеки [2] будується на паралельній варіації k-way алгоритму під назвою multilevelkway parallel algorithm. З трьох етапів послідовного k-way алгоритму мінімізація та розгортання займають 95% обчислень. Саме паралельним виконанням цих етапів і відрізняється даний алгоритм від послідовного аналогу.

Попередньо етапу мінімізації виконується додатковий етап розділу графу на частини для їх розподілення по процесорам. Він базується на алгоритмі Лубі, який об'єднує у розділи по можливості інцидентні вершини. У розділах кількість вершин повинна бути приблизно однаковою. У кінці цього етапу розділи графу розсилаються по процесорах.

Етап мінімізації виконується на кожному процесорі паралельно. Перевага при згортанні вершин надається вершинам, які належать одному й тому ж самому процесору. Фаза мінімізації завершується, коли кількість вершин у похідному графу дорівнює кількості процесорів.

Фаза розбиття розподіляє вершини на процесорах, таким чином, що на кожному процесорі лишається лише по одній вершині мінімізованого графу.

На етапі збільшення графа (uncoarsening - фаза), виконується його розгортання з використанням алгоритмів локального покращення аналогічно тому, як і у бібліотеці METIS на даному етапі.

Література

- [1] George Karypis and Vipin Kumar, METIS - A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices
- [2] George Karypis, Kirk Schloegel and Vipin Kumar, ParMETIS - Parallel Graph Partitioning and Sparse Matrix Ordering Library