

Задание 3

Отчёт

по реализации и анализу алгоритмов работы с графом в формате CSR

Ши Хуэй shihuicollapsor@gmail.com

1. Постановка задачи

В данном отчёте рассматривается реализация двух алгоритмов работы с графом в формате CSR:

1. Определение вершины с наибольшим суммарным весом инцидентных рёбер, ведущих к вершинам с чётными номерами.

$$Rank(vertex) = \sum_{i=0}^{N_inc_edges} w_{edge_i} * W_{vert_i},$$

где N_inc_edges - число инцидентных вершине $vertex$ рёбер, w_{edge_i} - вес i -го ребра, а вес вершины W_{vert_i} определяется по формуле

2. Определение вершины с наибольшим рангом, где ранг определяется по следующей формуле:

$$W(vertex) = \sum_{j=0}^{N_inc_edges} w_{edge_j} * N_inc_edges_{vert_j}$$

Задача состоит в реализации алгоритмов на основе формата графа CSR, их тестировании на различных графах, а также анализе производительности с помощью инструментов RAPI. Основное внимание уделяется двум событиям:

- RAPI_L1_TCM — количество промахов кэша первого уровня.
- RAPI_L2_TCM — количество промахов кэша второго уровня (если доступно).

Дополнительно для анализа производительности был выбран native event RAPI_TOT_INS, который отражает общее количество выполненных инструкций.

2. Формат командной строки

```
g++ papi_work.cpp -lpapi -o papi_work -I/usr/local/include -L/usr/local/lib
```

3. Спецификация системы

- PAPI version : 7.2.0.0
- Operating system : Linux 6.8.0-45-generic
- Vendor string and code : GenuineIntel (1, 0x1)
- Model string and code : Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz (165, 0xa5)
- CPU revision : 2.0000000
- CPUID : Family/Model/Stepping 6/165/2, 0x06/0xa5/0x02
- CPU Max MHz : 5000
- CPU Min MHz : 800
- Total cores : 12
- SMT threads per core : 2
- Cores per socket : 6
- Sockets : 1
- Cores per NUMA region : 12
- NUMA regions : 1
- Running in a VM : no
- Number Hardware Counters : 10
- Max Multiplex Counters : 384
- Fast counter read (rdpmc): yes

4. Описание алгоритмов

Алгоритм 1: Определение вершины с максимальным суммарным весом

Алгоритм находит вершину, у которой сумма весов рёбер, ведущих к чётным вершинам, максимальна. Псевдокод:

1. Для каждой вершины i в графе:
 - Инициализировать суммарный вес как 0.
 - Для каждой инцидентной вершине j , если номер j чётный, добавить вес рёбер в суммарный вес.
2. Сравнить суммарные веса и выбрать вершину с максимальным значением.

Алгоритм 2: Определение вершины с максимальным рангом

Алгоритм находит вершину, у которой ранг, вычисляемый по формуле, максимален.

Псевдокод:

1. Для каждой вершины i в графе:
 - Инициализировать ранг как 0.
 - Для каждой инцидентной вершине j вычислить сумму весов рёбер, инцидентных j .
 - Умножить вес ребра между i и j на сумму весов рёбер, инцидентных j , и добавить к рангу.

2. Сравнить ранги и выбрать вершину с максимальным значением.

5. Записи экспериментов и результаты

Тестирование проводилось на пяти графах: synt, road_graph, stanford, youtube и syn_rmat. Для каждого теста были измерены и проанализированы показатели производительности.

Algorithm_Performance_Results

Test	Alg	Max_Weight_V ertex	TY PE	DCM	ICM	TCM	PAPI_TOT_INS
synt	Alg1	447	L1	548057	33169	581226	901114960
			L2	728645	16061	744706	
	Alg2	447	L1	493172754	9108798	502281552	336721664182
			L2	371438533	7695174	379133707	
road_graph	Alg1	474251	L1	4528879	259933	4788812	33202357948
			L2	10420529	40040	10460569	
	Alg2	474251	L1	7197938	656214	7854152	75385189006
			L2	11200970	65166	11266136	
stanford	Alg1	28226	L1	4831842	275308	5107150	23552750489
			L2	10788306	78508	10866814	
	Alg2	48966	L1	63610823	4432039	68042862	382598196147
			L2	92910275	1217066	94127341	
youtube	Alg1	382	L1	13293331	430490	13723821	29277940392
			L2	19773321	216095	19989416	
	Alg2	41	L1	834851395	13786181	848637576	738408239048
			L2	1169718724	12512399	1182231123	
syn_rmat	Alg1	20486	L1	5689894	166983	5856877	7091298054
			L2	6926458	113290	7039748	
	Alg2	6931	L1	10639721652	114810017	10754531669	5449668865699
			L2	11520946910	107593017	11628539927	

6. Заключение

Alg2 на всех наборах данных демонстрирует значительно более высокое количество общих инструкций (PAPI_TOT_INS) по сравнению с Alg1. Например, на наборе данных synt общее количество инструкций для Alg2 составляет 336,721,664,182, в то время как у Alg1 — всего 901,114,960. Аналогично, на других наборах данных количество инструкций Alg2 превышает показатели Alg1 более чем на порядок. Это свидетельствует о том, что Alg2 в процессе вычислений задействует больше сложных операций и инструкций.

Что касается промахов кэш-памяти L1 и L2, Alg2 демонстрирует значительно более высокие показатели L1_DCM и L2_DCM по сравнению с Alg1. Например, на наборе данных syn_rmat количество L1 DCM у Alg1 составляет 5,689,894, в то время как у Alg2 оно достигает 10,639,721,652; аналогичная ситуация наблюдается и с L2 DCM — у Alg2 данный показатель составляет 11,520,946,910, а у Alg1 — всего 6,926,458. Такое резкое увеличение количества промахов L1 и L2 у Alg2 может быть связано с его более сложными вычислительными процессами и частыми обращениями к памяти, тогда как у Alg1 паттерн доступа к памяти более простой и локализованный. Если рассматривать общее количество промахов кэша L1 и L2 (L1_TCM и L2_TCM), можно увидеть, что у Alg2 на всех наборах данных данные показатели значительно выше, чем у Alg1. Это указывает на то, что паттерн доступа к памяти у Alg2 менее локализован и приводит к большему числу промахов кэша, тогда как у Alg1 наблюдается более высокая эффективность доступа за счет лучшей кэш-локализации.

На больших наборах данных, таких как stanford и youtube, количество промахов кэша у Alg2 резко возрастает, общее количество промахов L1 и L2 может достигать сотен миллионов и даже десятков миллиардов, что свидетельствует о низкой эффективности управления памятью Alg2 при обработке графов большого масштаба. На всех наборах данных найденные Alg1 и Alg2 номера вершин часто различаются, что свидетельствует о значительных различиях в критериях, используемых алгоритмами для определения «оптимальной вершины». Например, на наборе данных youtube Alg1 находит вершину под номером 382, в то время как Alg2 находит вершину под номером 41. Эти различия отражают использование различных метрик (веса и ранга) для оценки важности вершин.

Таким образом, можно сделать вывод, что Alg1 демонстрирует более высокую эффективность как по доступу к памяти, так и по исполнению инструкций на всех наборах данных.