

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH
PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN



Vận dụng thiết kế thuật toán: Parallel Algorithms

Nhóm 5 thực hiện

Môn học: CS112.P11.KHTN

Sinh viên thực hiện:

Nguyễn Thiên Bảo - 23520127

Trần Lê Minh Nhật - 23521098

Giáo viên hướng dẫn:

Nguyễn Thanh Sơn

ngày 30 tháng 11 năm 2024

Mục lục

1	Kiểm tra số nguyên tố song song	1
1.1	Kiểm tra số nguyên tố song song	1
1.2	Thuật toán	1
2	Nhân ma trận song song	2
2.1	Nhân ma trận song song	2
2.2	Thuật toán	3

1 Kiểm tra số nguyên tố song song

1.1 Kiểm tra số nguyên tố song song

Xây dựng thuật toán kiểm tra số nguyên tố song song. Input: Một số nguyên X . Output: Xác định X có phải là số nguyên tố hay không. Yêu cầu:

1. Kiểm tra kết quả với phương pháp tuần tự.
2. So sánh thời gian thực hiện giữa phương pháp song song và tuần tự trên các test case:

$X = 10000000000000091$, $X = 10000000000000099$, $X = 10000000000000049$.

1.2 Thuật toán

Tính toán tuần tự:

- Kiểm tra số nguyên tố bằng cách duyệt qua tất cả các số từ 2 đến \sqrt{N}
- Nếu $X \% i == 0$, X không phải số nguyên tố.

Tính toán song song:

- Chia đoạn $2 \rightarrow \sqrt{X}$ thành nhiều phần.
- Mỗi phần được giao cho một luồng (thread) để kiểm tra.
- Nếu một luồng phát hiện $X \% i = 0$, dừng các luồng khác.

```
def is_prime_parallel(X):  
    if X <= 1:  
        return False  
    sqrt_x = int(X**0.5) + 1  
    results = multiprocessing.Manager().list()  
  
    def check_range(start, end):
```

```
    for i in range(start, end):
        if X % i == 0:
            results.append(False)
            break

# Chia đoạn thành nhiều phần
num_threads = multiprocessing.cpu_count()
step = sqrt_x // num_threads
processes = []
for i in range(num_threads):
    start = i * step + 2
    end = (i + 1) * step + 2 if i != num_threads - 1 else sqrt_x
    p = multiprocessing.Process(target=check_range, args=(start, end))
    processes.append(p)
    p.start()

for p in processes:
    p.join()

return False if False in results else True
```

2 Nhân ma trận song song

2.1 Nhân ma trận song song

Xây dựng thuật toán nhân ma trận song song.

- **Input:** 2 ma trận A, B .

- **Output:** Ma trận $C = A \times B$.

Yêu cầu

- Kiểm tra kết quả khi thực hiện tính toán tuần tự.
- So sánh thời gian thực hiện giữa tính toán tuần tự và song song trên nhiều test case với kích thước ma trận nhỏ và lớn.
- Sinh ngẫu nhiên ma trận A, B kích thước 400×400 . Hiển thị thời gian thực hiện cả song song và tuần tự trên các test case này.
- Không sử dụng `numpy.dot()` hoặc các hàm nhân ma trận có sẵn.

2.2 Thuật toán

Tính toán tuần tự:

- Áp dụng công thức nhân ma trận thông thường:

$$C[i][j] = \sum_{k=0}^{n-1} A[i][k] \cdot B[k][j]$$

Tính toán song song:

- Chia các hàng của ma trận A thành các phần, mỗi phần được giao cho một luồng (thread).
- Tính $C[i][j]$ cho các hàng tương ứng của A .

So sánh và đánh giá

- Đo thời gian thực hiện trên nhiều test case.
- Trình bày bảng so sánh hiệu suất giữa thuật toán tuần tự và song song.

```
def matrix_multiply_parallel(A, B):
    n = len(A)
    C = [[0] * n for _ in range(n)]

    def compute_row(row):
        for j in range(n):
            for k in range(n):
                C[row][j] += A[row][k] * B[k][j]

    threads = []
    for i in range(n):
        t = threading.Thread(target=compute_row, args=(i,))
        threads.append(t)
        t.start()

    for t in threads:
        t.join()

    return C
```