

Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin

Parallel Merge Sort

CS112 - Phân tích và Thiết kế Thuật toán

Nhóm: 05

Lớp: CS112.N21.KHTN

MSSV: 21520029, 21521845

Ngày 18 tháng 3 năm 2023

1 Tổng quan

Báo cáo này được thực hiện nhằm mục đích thực thi thuật toán sắp xếp Merge Sort theo phương pháp lập trình song song (parallel computing), đồng thời tiến hành so sánh với thuật toán Merge Sort thông thường.

2 Thuật toán Merge Sort cổ điển

Merge Sort (sắp xếp gộp) là một thuật toán sắp xếp có so sánh kinh điển, thuộc lớp thuật toán sắp xếp có độ phức tạp $\mathcal{O}(n \log n)$ với n là kích thước của dãy cần sắp xếp. Merge Sort có đặc trưng là sử dụng phương pháp *chia để trị* kết hợp với kỹ thuật lập trình đệ quy.

Tư tưởng cơ bản của thuật toán Merge Sort như sau:

Gọi dãy $A = a_1, a_2, \dots, a_n$ là dãy ta cần sắp xếp.

Không mất tính tổng quát, ta sẽ sắp xếp dãy này tăng dần về giá trị của các phần tử.

Với $n = 1$, mặc nhiên dãy đã được sắp xếp.

Với $n \geq 2$, ta chia dãy thành hai phần, mỗi phần có kích thước $m = \frac{n}{2}$ phần tử.

Gọi $L = l_1, l_2, \dots, l_m$ là dãy a_1, a_2, \dots, a_m đã được sắp xếp tăng dần (1)

Gọi $R = r_1, r_2, \dots, r_m$ là dãy $a_{m+1}, a_{m+2}, \dots, a_n$ đã được sắp xếp tăng dần (2)

Việc thực hiện (1) và (2) có thể được thực hiện thông qua các thủ tục gọi đệ quy cơ bản.

Sau khi đã có được dãy L và dãy R , ta có tiến hành thu được dãy a thông qua việc lấy phần tử nhỏ nhất thuộc một trong hai dãy và gán vào dãy a . Do L và R đã được sắp xếp tăng dần từ trước nên tổng chi phí cho công đoạn này chỉ mất $\mathcal{O}(n)$.

Ngoài ra, do kích thước của dãy chỉ còn một nửa sau mỗi lần gọi đệ quy, nên mỗi phần tử trong dãy chỉ được xử lý tối đa $\log n$ lần. Từ đây ta có được độ phức tạp tổng của giải thuật là $\mathcal{O}(n \log n)$.

Mã giả của giải thuật có thể được trình bày như sau:

```
MERGE-SORT( $A, p, r$ )
1. if  $p < r$ 
2.    $q = \lfloor (p + r) / 2 \rfloor$ 
3.   MERGE-SORT( $A, p, q$ )
4.   MERGE-SORT( $A, q+1, r$ )
5.   MERGE( $A, p, q, r$ )
```

Hàm Merge-Sort sắp xếp dãy A từ vị trí p đến vị trí r

```
MERGE( $A, p, q, r$ )
1.   $n_1 = q - p + 1$ 
2.   $n_2 = r - q$ 
3.  let  $L[1..n_1 + 1]$  and  $R[1..n_2 + 1]$  be new arrays
4.  for  $i = 1$  to  $n_1$ 
5.       $L[i] = A[p + i - 1]$ 
6.  for  $j = 1$  to  $n_2$ 
7.       $R[j] = A[q + j]$ 
8.   $L[n_1 + 1] = \infty$ 
9.   $R[n_2 + 1] = \infty$ 
10.  $i = 1$ 
11.  $j = 1$ 
12. for  $k = p$  to  $r$ 
13.     if  $L[i] \leq R[j]$ 
14.          $A[k] = L[i]$ 
15.          $i = i + 1$ 
16.     else  $A[k] = R[j]$ 
17.          $j = j + 1$ 
```

Thao tác Merge dãy A từ hai dãy L và R
Nguồn: stackoverflow

3 Thuật toán Merge Sort song song

Ta nhận thấy việc thực thi (1) và (2) là hoàn toàn độc lập với nhau. Do đó, ta có thể tách chúng thành hai tiểu trình con xử lý song song với nhau nhằm giảm thời gian, tăng độ hiệu quả của giải thuật. Việc này về lý thuyết thì tương đối đơn giản, nhưng để hiện thực trên các ngôn ngữ lập trình đòi hỏi các kiến thức về sử dụng tiểu trình trong lập trình kết hợp với lập trình song song.

4 Thực nghiệm Giải thuật

4.1 Môi trường, dữ liệu và mã nguồn

Môi trường:

- Hệ điều hành: Ubuntu 22.04 LTS
- Trình biên dịch: Python 3.8 64 bit
- CPU: Intel Core i5 - 8250U @ 1.6-3.4 GHz

- GPU: nVidia GeForce MX150 2GB
- RAM: 12GB

Dữ liệu:

Mỗi bộ test dùng cho việc thực nghiệm là một dãy số nguyên không âm có 10^6 phần tử, trong đó mỗi phần tử có giá trị không vượt quá 10^9 .

Dữ liệu bao gồm mười bộ test, bộ test thứ nhất được sắp xếp sẵn theo thứ tự tăng dần, bộ test thứ hai được sắp xếp sẵn theo thứ tự giảm dần, các bộ test còn lại có thứ tự sắp xếp là ngẫu nhiên.

Mã nguồn của giải thuật có thể được xem tại link github môn học của nhóm.

4.2 Kết quả thực hiện

	Merge Sort	Parallel Merge Sort
test0	4622	2337
test1	4573	2513
test2	6415	2978
test3	6982	2768
test4	6347	2755
test5	6039	2798
test6	6050	2753
test7	6063	2907
test8	6081	2862
test9	6087	3118
Chậm nhất	6982	3118
Nhanh nhất	4573	2337
Trung bình	5926	2779

Bảng ghi nhận thời gian thực thi (ms)

5 Kết luận

Trong toàn bộ các bộ test được thử nghiệm, việc cài đặt Merge Sort theo kỹ thuật lập trình song song đều cho kết quả khả quan hơn. Tuy nhiên, để đạt được hiệu suất làm việc tốt nhất, thuật toán này đòi hỏi kỹ năng lập trình nhất định của người cài đặt, đồng thời cần được điều chỉnh linh động cho phù hợp với hệ điều hành và phần cứng máy tính.