

Một vài thuật toán sắp xếp trên mô hình song song

Nguyễn Tuấn Đạt
Đặng Quang Trung

Ngày 11 tháng 1 năm 2017

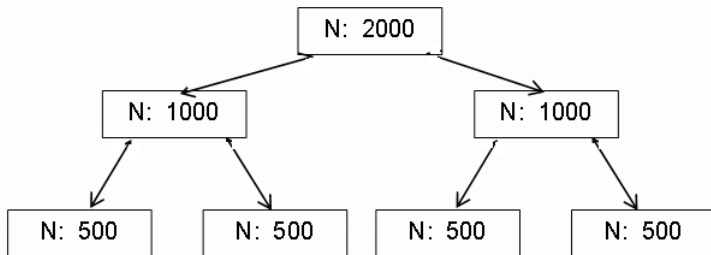
Nội dung



- 1 Giới thiệu MPI
- 2 Các thuật toán sử dụng
- 3 Kết quả
 - Thực hiện
 - Kết quả

- MPI là một thủ viện chuẩn của trao đổi thông điệp giữa nhiều máy tính chạy một chương trình song song trên bộ nhớ phân tán.
- Nó cho phép tính toán song song trên các clusters và các mạng không đồng nhất.
- Được thiết kế cho phép (mở) phát triển các thư viện phần mềm song song.
- Được thiết kế để cung cấp quyền truy cập vào phần cứng song song cho
 - ▶ Người dùng cuối.
 - ▶ Người viết thư viện.
 - ▶ Người phát triển tool.

- **Ý tưởng:** Xây dựng một cây xử lý
 - ▶ Số lượng nút lá của cây bằng số lượng bộ xử lý.
 - ▶ Chiều cao của cây $\log(p)$ (p là số bộ xử lý).
- Mỗi nút lá chứa danh sách các phần tử. Áp dụng giải thuật sắp xếp tuần tự cho mỗi nút lá.
- Kết quả ở mỗi nút lá sẽ được cung cấp về các nút cha (quá trình trộn 2 danh sách để được 1 danh sách mới). Các nút cha lại gửi tiếp.
- Cuối cùng nút gốc sẽ là sự hòa trộn thành dãy đã được sắp xếp.



Hình: Cây xử lý độ cao 3

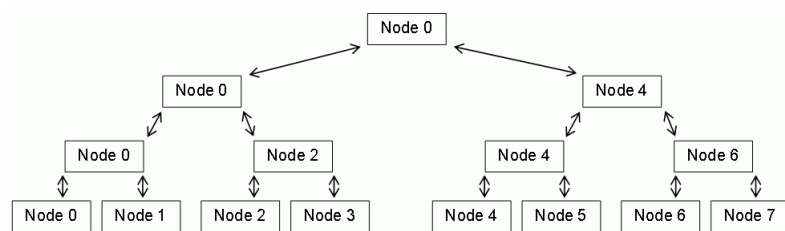
Trộn kết quả các nút



- Mỗi nút trong cây xử lí 1 quá trình riêng
- Ý tưởng trộn mượn ý tưởng binary heap khi nó thực hiện trong 1 mảng với gốc là 0.
- Với mỗi phần tử trong mảng có chỉ số con k có con trái là $2*k + 1$ và con phải là $2*k + 2$ và phần tử cha là $\frac{k-1}{2}$.

$$O(n) + O(n/p * \log p) + O(n/p \log(n/p))$$

Trộn kết quả các nút



Hình: Cây xử lý độ cao 3

Ý tưởng: Sử dụng biến thể của odd-even để mở rộng cho nhiều bộ xử lí

- Chia dữ cho các bộ xử lí (p).
- Mỗi giai đoạn: ($1 \rightarrow p$):
 - ▶ Sắp xếp dữ liệu địa phương trên mỗi bộ
 - ▶ Tìm các đối tác của bộ xử lí dựa trên giai đoạn và rank của chúng.
 - ▶ Nếu bộ xử lí có đối tác:
 - gửi dữ liệu đến đối tác của chúng.
 - nhận dữ liệu từ đối tác của chúng.
 - Nếu rank của chúng nhỏ, thì giữ lại các phần tử nhỏ nhất (dữ liệu của bộ xử lí + dữ liệu của đối tác).
 - Trái lại thì giữ lại các phần tử lớn nhất (dữ liệu của bộ xử lí + dữ liệu của đối tác).

Ý tưởng: Dựa trên nguyên lý chia để trị.

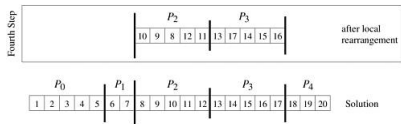
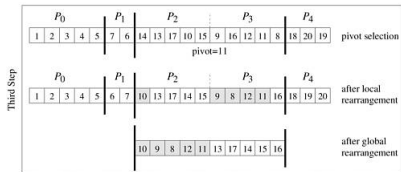
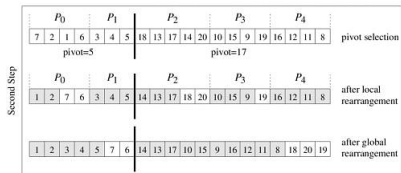
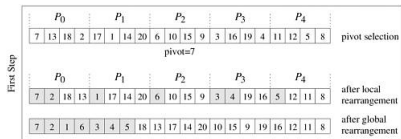
Xét hệ thống có p bộ xử lý.

Thuật toán hoạt động như sau: Tại mỗi bước thuật toán cố gắng chia dữ liệu thành 2 group dựa vào một key chọn sẵn.

- ➊ Bước 1(i lần): Group j sẽ được chia làm 2 phần dựa key (key được chọn ngẫu nhiên từ tiến trình chủ trong group). $i \in [1..\log(p)]$
- ➋ Bước 2: Sử dụng thuật toán sắp xếp tối ưu để sắp xếp tại mỗi tiến trình.
- ➌ Bước 3: Gửi mảng đã sắp xếp về tiến trình root.

$$O(n) + O(n/p * \log p) + O(n/p \log(n/p))$$

Mô tả



SampleSort

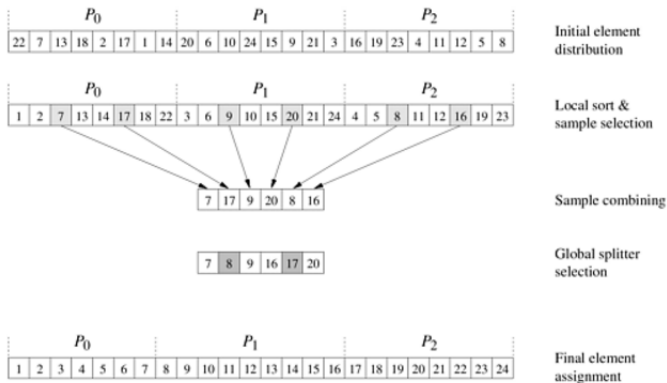
Ý tưởng: Dựa trên ý tưởng của thuật toán bucketSort thuật toán sampleSort sẽ chọn ra một tập mẫu để quy định các bucket cho bucketSort.

Hoạt động:

- Bước 1: Sắp xếp mảng cục bộ tại mỗi tiến trình bằng thuật toán tối ưu.
- Bước 2: Tại mỗi tiến trình chọn ra $p-1$ số cách đều trong dữ liệu đã sắp xếp và gửi về tiến trình root.
- Bước 3: Tiến trình root tổng hợp và chọn ra $p-1$ số mẫu để quy định khoảng cho các bucket và gửi dữ liệu này đến mỗi tiến trình.
- Bước 4: Mỗi tiến trình chia dữ liệu vào các bucket và gửi dữ liệu ấy đến tiến trình ứng với bucket đó.
- Bước 5: Mỗi tiến trình sắp xếp dữ liệu tại bucket của mình.
- Bước 6: Mỗi tiến trình gửi dữ liệu đã sắp xếp về tiến trình root.

Độ phức tạp

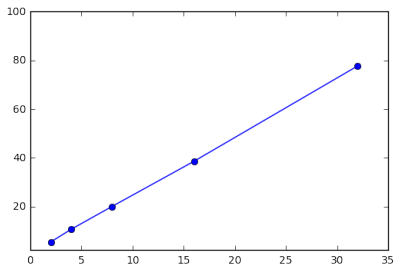
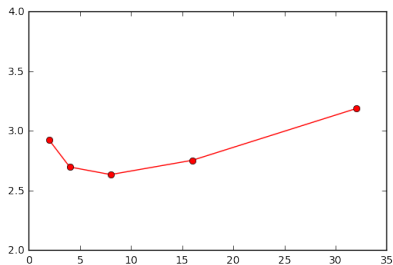
$$O(n/p \log(n/p)) + O(p^2 \log p) + O(p \log(n/p)) + O(n/p) + O(p \log p)$$



Hình: Hoạt động của thuật toán sample sort

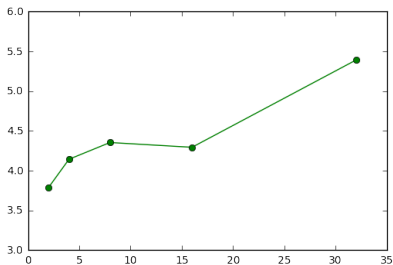
- Thực hiện trên 2 máy bộ xử lý core i5.
- Dữ liệu sử dụng 5 bộ với $n=1000, 10000, 100000, 1000000, 1000000$.
- Thực hiện đo 10 lần với mỗi trường hợp sau đó tính trung bình.

Biểu đồ với số lượng bộ xử lý



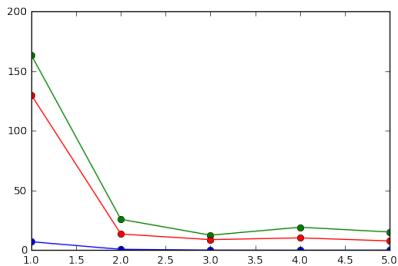
Hình: Biểu đồ trên số lượng bộ xử lý

Biểu đồ với số lượng bộ xử lý



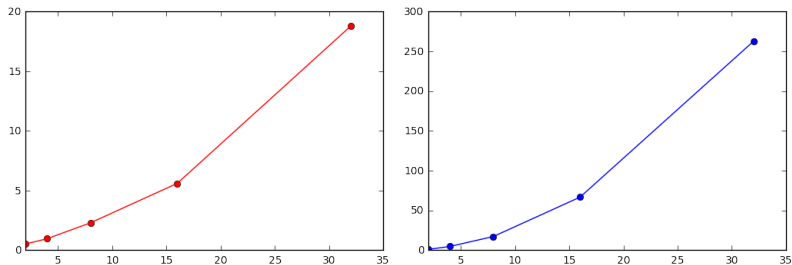
Hình: Biểu đồ trên số lượng bộ xử lý

Biểu đồ speedup



Hình: Biểu đồ speedup

Biểu đồ thể hiện thời gian truyền



Hình: Biểu đồ thời gian truyền

Tài liệu tham khảo



- <http://parallelcomp.uw.hu/ch09lev1sec4.html>
- <http://parallelcomp.uw.hu/ch09lev1sec5.html>
- Slide bài giảng lập trình song song thầy Nguyễn Tuấn Dũng