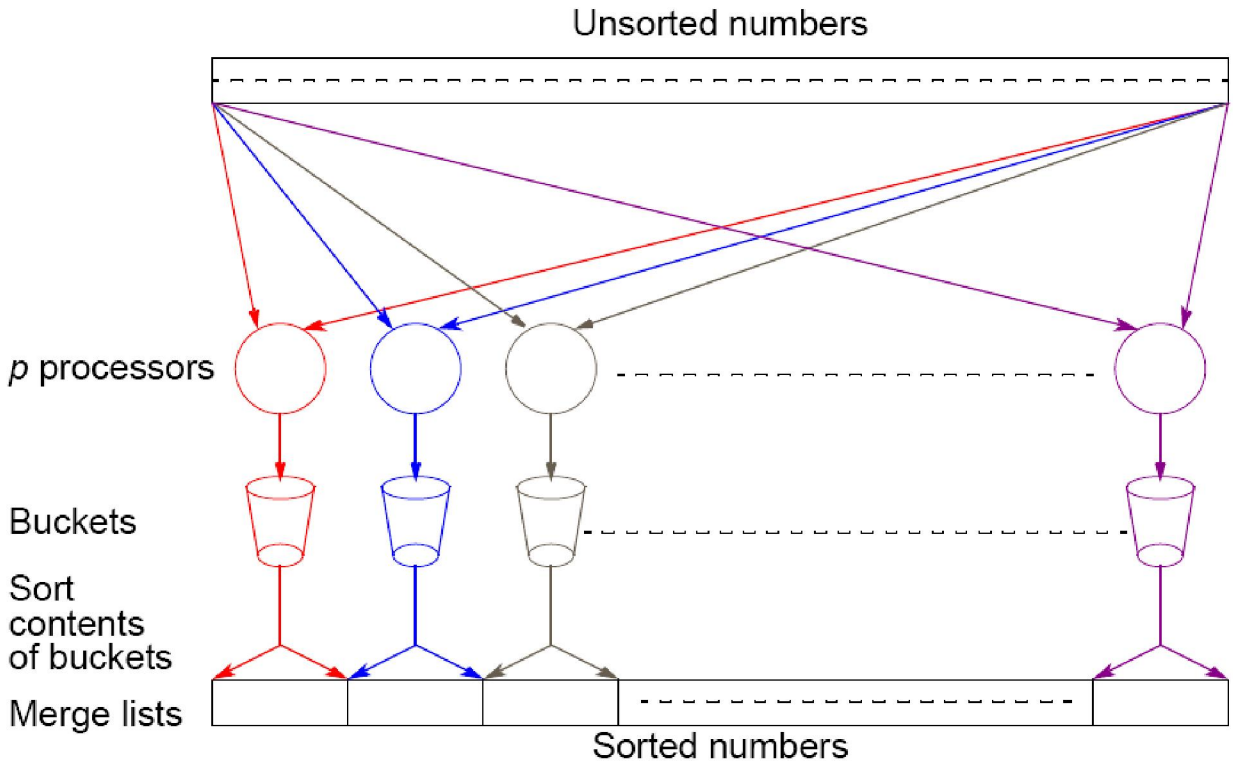
Xử Lý Tính Toán Song Song – Bài tập 2

Sắp xếp mảng theo cơ chế song song, sử dụng Bucket sort.

Phân tích bài toán: Để sắp xếp một mảng theo mô hình bucket sort có 2 đề xuất như sau:

1/ Thuật giải 1:



Giả sử để sort mảng gồm các số trong khoảng [0,999] sử dụng 1 process master và 10 processes tính toán.

Như vậy ta sẽ phân chia: process P_1 đảm nhận sort các số nằm trong đoạn giá trị [0,99], process P_2 đảm nhận các số trong đoạn [100,199],…, process P_{10} đảm nhận sort các số trong đoạn [900,999].

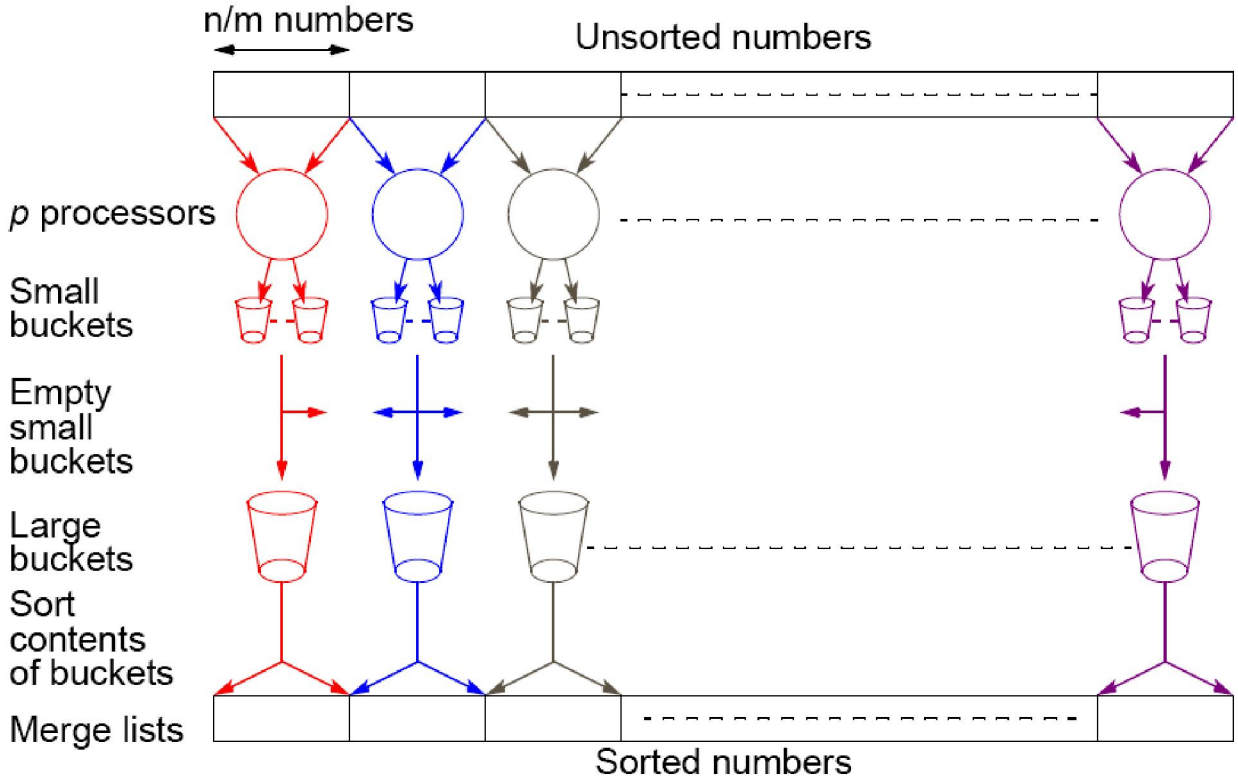
Process master P_0 đọc lần lượt từng số trong mảng dữ liệu, sau đó phân loại nếu số nằm trong đoạn do process nào đảm nhận sẽ gửi cho process đó. Ví dụ nếu đọc được số 60 sẽ gửi cho process P_1, nếu đọc được số 194 sẽ gửi cho process P_2. Mỗi process sau khi nhận đủ dữ liệu từ master sẽ bắt đầu sắp xếp phần dữ liệu mình nhận được (gọi là một bucket).

Sau khi các process con sắp xếp xong sẽ gửi dữ liệu về cho process master. Process master ghép dữ liệu theo đúng thứ tự của từng process con sẽ nhận lại được mảng đã sắp xếp. Để tiết kiệm thời gian, master có thể phân loại hết tất cả các số sau đó mới gửi cho các process con một lần duy nhất, không send từng số riêng lẻ.

Có thể tổng quát hóa bài toán: sort các số nằm trong đoạn [x,y], sử dụng 1 master và p process tính toán tương tự theo mô tả bên trên. Mỗi process đảm nhận sắp xếp cho một đoạn có chiều dài khoảng $\frac{y-x+1}{p}$ số.

Thuật giải 1 có ưu điểm là dễ cài đặt, nhược điểm là master phải đảm nhận công việc lớn (phân loại từng số) và các process con có thể đảm nhận khối lượng công việc không đều nhau khi các số phân bố trong miền giá trị không đều.

Thuật giải 2:



Đầu vào của dữ liệu tương tự như thuật giải 1. Tuy nhiên ở đây công việc của master P_0 được giảm tải. Giả sử có 1000 số cần sort, chia đều cho 10 processes như vậy mỗi process sẽ nhận 100 số. Master P_0 không cần phân loại, đọc 100 số đầu tiên gửi cho P_1, 100 số kế tiếp gửi cho P_2, lần lượt cho đến 100 số cuối cùng gửi cho P_{10}.

Quá trình phân loại diễn ra tại các processes con: Process P_1 sẽ phân loại các số nó nhận được, nếu nằm trong khoảng [0,99] nó sẽ giữ lại và tự sắp xếp, nếu nằm trong khoảng [100,199] sẽ gửi cho P_2, trong khoảng [900, 999] sẽ gửi cho P_{10}. Tương tự với P_2, P_3, … , P_{10}. Mỗi process sẽ phân chia dữ liệu nó nhận được vào 10 đoạn giá trị, gửi đoạn giá trị cho process tương ứng đảm nhận.

Sau khi nhận được dữ liệu lần 2 (là dữ liệu trao đổi nội bộ giữa các process con), mỗi process sẽ sort đoạn dữ liệu tương ứng của mình và gửi cho master.

Tương tự như thuật giải 1, master nối các đoạn dữ liệu nhận được từ các process con theo đúng thứ tự sẽ được mảng đã sắp xếp.

Ưu điểm: Master được giảm tải, quá trình phân loại dồn về các processes con nên thuật giải chạy "song song" được nhiều hơn và phần nào công việc ở các processes con đều hơn.

Nội dung bài tập 2:

Đọc vào file input.txt (đường dẫn file do người dùng nhập) gồm n số, phân bố trong đoạn giá trị [-65536,65535]. Sắp xếp nội dung mảng theo thuật giải 2, số process do người dùng nhập thông qua câu lệnh mpiexec (động, không biết trước). Kết quả mảng đã sắp xếp lưu vào file output.txt cùng thư mục với file input.txt

Cấu trúc file input .txt gồm 2 dòng, dòng đầu cho biết số n, dòng sau là danh sách n số cần sắp xếp, các số cách nhau khoảng trắng. Ví dụ:

5

1 2 3 4 5

Cấu trúc file output.txt tương tự file input.txt.

Yêu cầu: Các process con sử dụng thuật toán Quick Sort để sắp xếp.

Nội dung bài nộp: Source code, file thực thi và Báo cáo chi tiết các làm cụ thể.

Deadline: Hết ngày chủ nhật, 25 Oct 2009