Sắp xếp nội và sắp xếp ngoại

So Sánh ?

Sắp xếp nội

- Sắp xếp dữ liệu trên ram
- Tốc độ truy xuất ngẫu nhiên
- Dựa trên hoán vị
- Dữ liệu nhỏ

Sắp xếp ngoại

- Sắp xếp dữ liệu trên file
- Tốc độ truy xuất tuần tự cao(vẫn thấp hơn ram)
- Dựa trên pp trộn
- Dữ liệu lớn

External merge sort

Ý Tưởng :

- Chia nhỏ dữ liệu từ tập dữ liệu ban đầu thành từng phần
- Sắp xếp từng phần dữ liệu đó
- Lưu từng phần dữ liệu đã sắp xếp vào các file tạm
- Trộn các file kết quả để ra kết quả cuối cùng

External merge sort

Ví dụ cụ thể

Tập dữ liệu DL 100GB. Và ta chỉ có 4GB trong quá trình sắp xếp.

1. Đọc 4GB từ dữ liệu đầu vào, tải lên RAM.
2. Thực hiện sắp xếp cho phần dữ liệu mới tải lên Ram. Có thể dung quick sort, merge sort hay bất kỳ thuật toán nào bạn tự tin nó chạy tố.
3. Ghi kết quả sau khi sắp xếp trên ô cứng.
4. Lặp lại bước 1,2,3 cho dến khi hết toàn bộ dữ liệu. Như vậy ta sẽ có, 100/4=25 files dữ liệu đã được sắp xếp cục bộ.

External merge sort

Đến đây ta có 25 danh sách đã được sắp xếp trước.

Ta thực hiện trộn các file thành danh sách đã sắp xếp.
Lưu ý : Ở mỗi file, chỉ tải lên một phần của file dữ liệu

Để tối ưu quá trình ta có thể sử dụng kỹ thuật **k-way merging**

External merge sort

K-way merging

- **Bước 1**: Tìm phần tử nhỏ nhất trong số các phần tử đầu tiên của các danh sách.
- **Bước 2**: Đẩy phần tử tìm được vào danh sách kết quả và xóa nó khỏi danh sách hiện tại

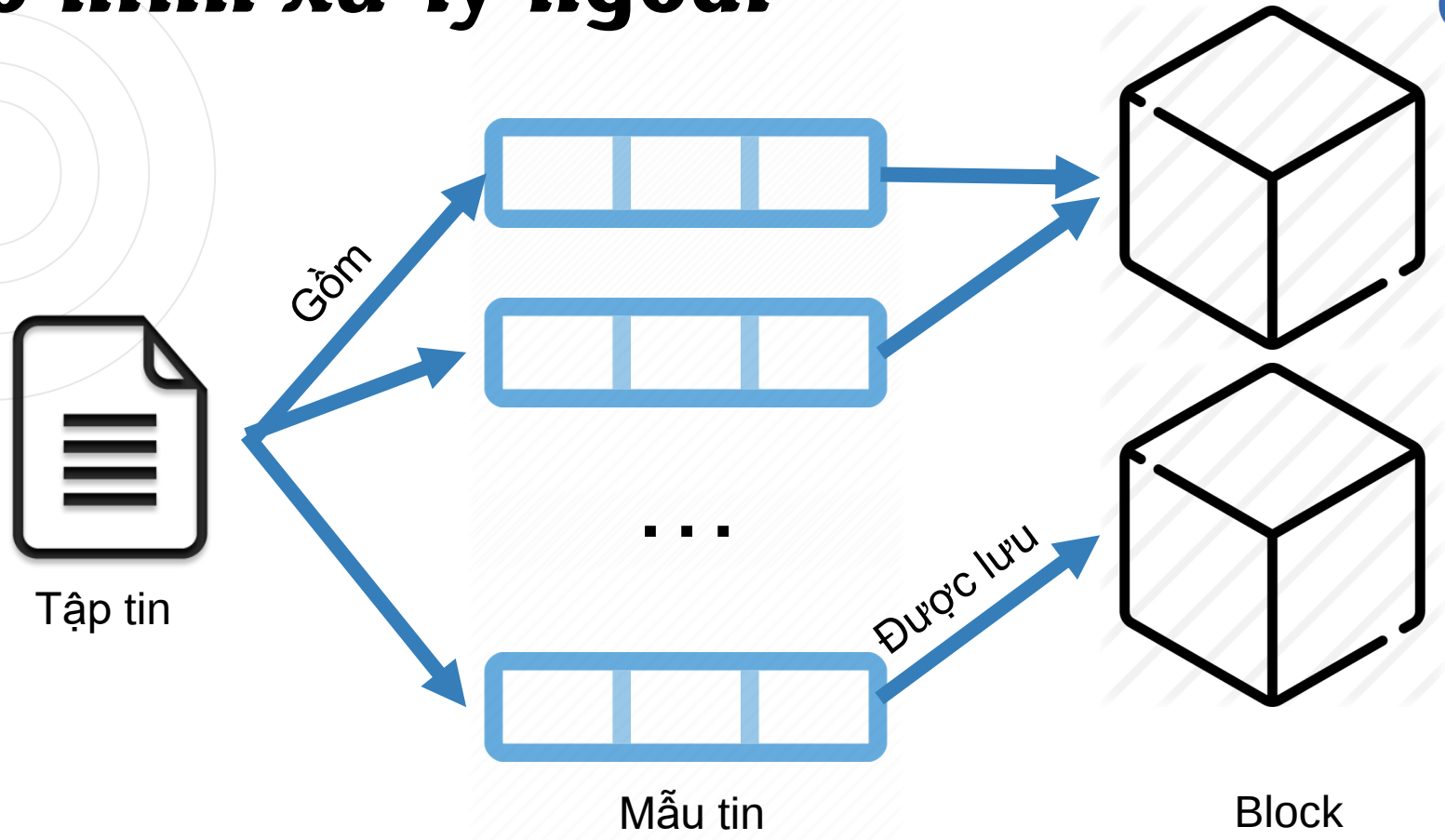
Thực hiện vòng lặp cho tới khi tất cả danh sách đều trống thì dừng lại

Cuối cùng ta sẽ có được duy nhất 1 danh sách kết quả chứa các phần tử đã được sắp xếp

Mục tiêu:

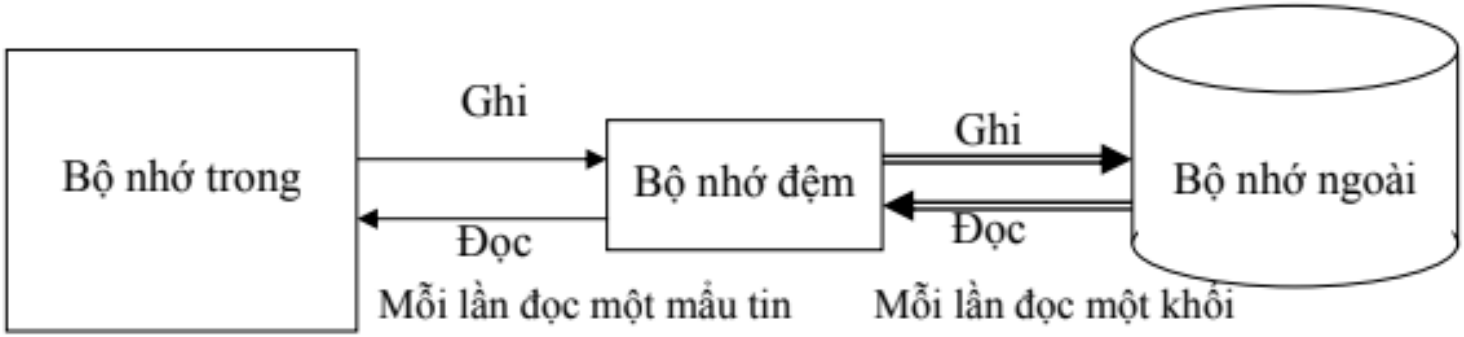
- Bài toán kinh điển
- Đặc trưng của SX trên file: Bài toán trộn
- TT tìm kiếm cơ bản: Tuần tự, nhị phân
- TT sắp xếp
- Đánh giá thuật toán

Mô hình xử lý ngoài



Mô hình xử lý ngoài

Mô hình giao tiếp giữa bộ nhớ trong, bộ nhớ ngoài và vùng nhớ đệm



- Thời gian tìm một khối để đọc vào bộ nhớ trong là rất lớn so với thời gian thao tác trên dữ liệu trong khối đó.
- Kiểu dữ liệu tập tin là kiểu thích hợp nhất cho việc biểu diễn dữ liệu được lưu trong bộ nhớ ngoài. Hệ điều hành chia bộ nhớ ngoài thành các khối (block) có kích thước bằng nhau, kích thước này thay đổi tùy thuộc vào hệ điều hành nhưng nói chung là từ 512 bytes đến 4096 bytes.

Vì vậy, chúng ta tập trung vào việc xét số lần đọc khối vào bộ nhớ trong và số lần ghi khối ra bộ nhớ ngoài

=> Merge sort là phù hợp nhất

Các Phương pháp sắp xếp ngoại khác

- Phương pháp trộn Run
- Phương pháp trộn tự nhiên
- Phương pháp trộn đa lối cân bằng
- Phương pháp trộn đa pha

Phương pháp trộn Run

Khái niệm Run: Run là một dãy phần tử được sắp xếp theo thứ tự

Ví dụ về Run: 2 4 7 12 50 40 60

2	4	7	12	50	40	60

L=5 L=2

Phương pháp trộn Run

Việc tạo ra Run mới từ 2 run ban đầu người ta gọi là trộn (merge).

2	4	7	12	50	40	60

L=5 L=2

Phương pháp trộn Run

Mô tả bài toán

Dữ liệu vào: tập tin f0 cần sắp xếp
Dữ liệu ra: tập tin f0 đã được sắp xếp
F1, f2 là hai tập tin phụ dung để sắp xếp

Phương pháp trộn Run

Bước 1: **M=1**

F0	24	12	67	33	58	42	11	34	29	31
F1	24	67	58	11	29					
F2	12	33	42	34	31					
F0 mới	12	24	33	67	42	58	11	34	29	31

Phương pháp trộn Run

Bước 2: **M=M*2=2**

F0	12	24	33	67	42	58	11	34	29	31
F1	12	24	42	58	29	31				
F2	33	67	11	34						
F0 mới	12	24	33	67	11	34	42	58	29	31

Phương pháp trộn Run

Bước 3: **M=M*2=4**

F0	12	24	33	67	11	34	42	58	29	31
F1	12	24	33	67	29	31				
F2	11	34	42	58						
F0 mới	11	12	24	33	34	42	58	67	29	31

Phương pháp trộn Run

Bước 4: **M=M*2=8**

F0	11	12	24	33	34	42	58	67	29	31
F1	11	12	24	33	34	42	58	67		
F2	29	31								
F0 mới	11	12	24	29	31	33	34	42	58	67

Phương pháp trộn Run

Bước 5:

Lặp lại tương tự các bước trên cho tới khi chiều dài m của run cần phân bố lớn hơn chiều dài n của F0 thì dừng

Phương pháp trộn Run

Thuật toán tổng quát:

m = 1 while (m < số phần tử của f0)
{
Chia[Distribute] m phần tử của f0 lần lượt cho f1, f2
Trộn[Merge] f1, f2 lần lượt vào f0
M = M * 2
}

Phương pháp trộn Run

❖ **Đánh giá:**

- Cần ít nhất N không gian trống trên đĩa để hoạt động.
- Số bước log2N (vì mỗi lần xử lý 1 dãy tăng gấp 2)
- Mỗi bước:
  - Distribute: Copy N lần
  - Merge: Copy N lần, so sánh N/2 lần

❖ **Tổng cộng:**

- Copy: 2N * log2N
- So sánh: N/2 * log2N

❖ **Hạn chế:**

- Không tận dụng được dữ liệu đã được sắp bộ phận
- Độ dài dãy con xử lý ở bước k <= 2k

2

PHƯƠNG PHÁP TRỘN TỰ NHIÊN

Phương pháp trộn tự nhiên

- Trong phương pháp trộn ở mục 1, giải thuật chưa tận dụng được chiều dài cực đại của các Run trước khi phân bố → chưa tối ưu.
- Đặc điểm của PP trộn tự nhiên là tận dụng chiều dài "tự nhiên" của các Run ban đầu; nghĩa là thực hiện việc trộn các Run có độ dài cực đại với nhau cho tới khi dãy chỉ còn 1 Run duy nhất → dãy đã được sắp.

Phương pháp trộn tự nhiên

F0 | 1 | 2 | 9 | 8 | 7 | 6 | 5 |

Bước 1:

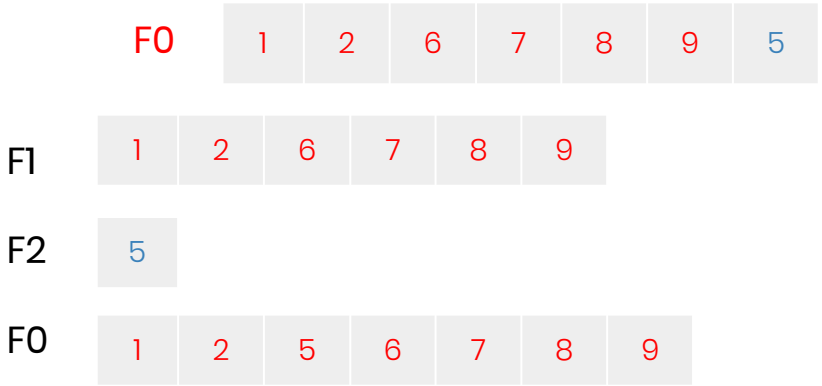
F1	1	2	9	7	5		
F2	8	6					
F0	1	2	8	9	6	7	5

Bước 2:

F1	1	2	8	9	5		
F2	6	7					
F0	1	2	6	7	8	9	5

Phương pháp trộn tự nhiên

Bước 3:



Bước 4: Dừng vì F0 chỉ có một run

Phương pháp trộn tự nhiên

Thuật toán tổng quát:

While (số Run của F0 > 1)

{

Phân bố F0 vào F1, F2 theo các Run tự nhiên.

Trộn các Run của F1, F2 vào F0.

}

- [Distribute] Chia xoay vòng dữ liệu của file F0 cho F1 và F2, mỗi lần 1 run cho đến khi file F0 hết.
- - [Merge] Trộn từng cặp run của F1 và F2 tạo thành run mới trên F0

Phương pháp trộn tự nhiên

Thuật toán tổng quát:

m = 1 while (m < số phần tử của f0)

{

Chia[Distribute] m phần tử của f0 lần lượt cho f1, f2

Trộn[Merge] f1, f2 lần lượt vào f0

M = M * 2

}

3

PHƯƠNG PHÁP TRỘN ĐA LỐI CÂN BẰNG

Phương pháp trộn đa lối cân bằng

ĐẶC ĐIỂM

- Thay vì thực hiện 2 giai đoạn, ta chỉ cần thực hiện 01 giai đoạn trộn.
  - Tiết kiệm ½ chi phí Copy.
  - Cần số lượng file trung gian gấp đôi.

Phương pháp trộn đa lối cân bằng

- B1: Gọi tập nguồn S = {f1, f2, …, fn }
  Gọi tập đích D = {g1, g2, ... gn }

  Chia xoay vòng dữ liệu của file F0 cho các file thuộc tập nguồn, mỗi lần 1 Run cho tới khi F0 hết.

- B2: Trộn từng bộ Run của các file thuộc tập nguồn S, tạo thành Run mới, mỗi lần ghi lên các file thuộc tập đích D.

- B3: Nếu (số Run trên các file của D > 1) thì:
  - Hoán vị vai trò tập nguồn (S) và tập đích (D).
  - Quay lại B2

  Ngược lại kết thúc thuật toán.

VÍ DỤ

Cho dãy số sau :

3 5 2 7 12 8 4 15 20 1 2 8 23 7 21 27

Input :

F0 : 3 5 2 7 12 8 4 15 20 1 2 8 23 7 21 27

Output :

F0 : 1 2 2 3 4 5 7 7 8 8 12 15 20 21 23 27

Input :

F0 : 3 5 2 7 12 8 4 15 20 1 2 8 23 7 21 27

Bước 0 : đặt m = 3

Bước 1 : Phân phối các run luân phiên vào f1, f2, f3

F1 : 3 5 4 15 20

F2 : 2 7 12 1 2 8 23

F3 : 8 7 21 27

Bước 2 : Trộn các run của f1, f2, f3, và luân phiên phân phối vào các file g1, g2, g3

F1 : 3 5 4 15 20

F2 : 2 7 12 1 2 8 23

F3 : 8 7 21 27

G1 : 2 3 5 7 8 12

G2 : 1 2 4 7 8 15 20 21 23 27

G3 :

Do số run sau khi trộn >1 nên tiếp tục trộn run từ g1, g2, g3 vào ngược lại f1, f2, f3

G1 : 2 3 5 7 8 12

G2 : 1 2 4 7 8 15 20 21 23 27

G3 :

F1 : 1 2 2 3 4 5 7 7 8 8 12 15 20 21 23 27

F2 :

F3 :

Do số run trộn = 1 nên kết thúc thuật toán

4 PHƯƠNG PHÁP TRỘN ĐA PHA

Phương pháp trộn đa pha

Ta xét ví dụ sau với 3 tập tin f1, f2, f3

- Bước 1: Phân phối luân phiên các run ban đầu của f0 vào f1 và f2
- Bước 2: Trộn các run của f1, f2 vào f3 . Giải thuật kết thúc nếu f3 chỉ có một run
- Bước 3: Chép nữa run của f3 vào f1
- Bước 4: Trộn các run của f1 và f3 vào f2. Giải thuật kết thúc nếu f2 chỉ có một run.
- Bước 5: Chép nữa số run của f2 vào f1. Lặp lại bước 2. Phương pháp này còn có nhược điểm là mất thời gian sao chép nữa số run của tập tin này vào tập tin kia. Việc sao chép này có thể loại bỏ nếu ta bắt đầu với Fn run của tập tin f1 và fn-1 run của tập tin f2, với fn và fn-1 là các số liên tiếp trong dãy Fibonaci.

Phương pháp trộn đa pha

Ví dụ: Trường hợp n=7, tổng số rung 13+8=21 run

Pharse	F1	F2	F3	
0	1,1,1,1,1,1,1	1,1,1,1		Sort
1	1,1,1		2,2,2,2,2	Merge 1
2		3,3,3	2,2	Merge 2
3	5,5	3		Merge 3
4	5		8	Merge 4
5		13		Merge 5

Phương pháp trộn đa pha

- Phase 0: Phân phối các run ban đầu
- Phase 1: Trộn 8 run của f1 và f2 vào f3
- Phase 2: Trộn 5 run của f1 và f3 vào f2
- Phase 3: Trộn 3 run của f2 và f3 vào f1
- Phase 4: Trộn 2 run của f1 và f2 vào f3
- Phase 5: Trộn 1 run của f1 và f3 vào f2
- Phase 6: Trộn 1 run của f2 và f3 vào f1

5

Demo External merge sort

Phân công công việc

1. Demo: ALL
2. SLide: ALL
3. Thuyết trình: Thanh, Quang
4. Ghi biên bản: Truyền
5. Tài liệu: ALL
6. Kahoot: Truyền,Quang
7. Lên kế hoạch: Thanh

Thanks!

Any questions?

You can find me at

- 19522245@gm.uit.edu.vn
- 19522093@gm.uit.edu.vn
- 19522448@gm.uit.edu.vn