

Báo Cáo Tuần 3 Thuật Toán Tách Công Việc Chồng Chéo

Ngày 16 tháng 3 năm 2025

1 Mô tả bài toán

Một nhân viên có thể thực hiện nhiều công việc trong cùng khoảng thời gian. Nếu các công việc của cùng một nhân viên chồng chéo, ta cần tách chúng thành các phân đoạn không chồng chéo để dễ quản lý. Dữ liệu đầu vào gồm danh sách công việc với các thuộc tính:

- Tên công việc (T_i)
- Thời gian bắt đầu (T_i^s)
- Thời gian kết thúc (T_i^e)
- Người thực hiện ($T_i^{assignee}$)

2 Mô hình toán học

Tại một thời điểm t , một nhân viên thực hiện các công việc T_i có thời gian:

$$T_i^d = T_i^e - T_i^s \quad (1)$$

Nếu có nhiều công việc trùng thời gian của cùng một nhân viên, chúng được chia thành các đoạn không chồng chéo. Tập công việc hoạt động tại thời điểm t được định nghĩa như sau:

$$PS^t = T_i \mid T_i^s \leq t < T_i^e, T_i^{assignee} = E \quad (2)$$

Với E là tập hợp các nhân viên.

3 Thuật toán tách công việc

3.1 Mô tả thuật toán

Thuật toán thực hiện các bước sau:

1. Tạo danh sách các sự kiện gồm thời điểm bắt đầu và kết thúc của từng công việc, kèm theo thông tin nhân viên phụ trách.
2. Sắp xếp các sự kiện theo thứ tự thời gian tăng dần. Nếu hai sự kiện xảy ra cùng thời điểm, ưu tiên xử lý sự kiện kết thúc trước sự kiện bắt đầu.
3. Duyệt qua các sự kiện theo từng nhân viên:
 - Theo dõi tập hợp các công việc đang hoạt động tại mỗi khoảng thời gian.
 - Khi có sự thay đổi trong tập hợp các công việc (có công việc bắt đầu hoặc kết thúc), ghi nhận một phân đoạn mới.
 - Nếu tại một khoảng thời gian có nhiều công việc cùng hoạt động, thuật toán sẽ chia đều khoảng thời gian đó cho các công việc này.
4. Trả về danh sách các phân đoạn công việc đã được chia nhỏ và không còn chồng lấn.

4 Mô tả dữ liệu

4.1 Danh sách công việc

- Tên công việc (T_i): Tên công việc.
- Thời gian bắt đầu (T_i^s): Thời gian bắt đầu công việc.
- Thời gian kết thúc (T_i^e): Thời gian kết thúc công việc.
- Người thực hiện ($T_i^{assignee}$): Người thực hiện công việc.
- Thời gian hoàn thành (T_i^d): Thời gian thực hiện công việc.
- Danh sách phân đoạn ($T_i^{segments}$): Danh sách các phân đoạn công việc.
- Thời gian bắt đầu phân đoạn ($T_i^{segments.s}$): Thời gian bắt đầu phân đoạn.
- Thời gian kết thúc phân đoạn ($T_i^{segments.e}$): Thời gian kết thúc phân đoạn.
- Người thực hiện phân đoạn ($T_i^{segments.assignee}$): Người thực hiện phân đoạn.

4.2 Mã giả thuật toán

Data: Danh sách công việc $\{T_i\}$

Result: Danh sách phân đoạn công việc không chồng chéo

1 **Function** *splitTask*tasks:

2 *events* $\leftarrow \emptyset$;

3 **foreach** *task* \in *tasks* **do**

4 *events.append*((*task.startTime*, 'start', *task.name*, *task.assignee*));

5 *events.append*((*task.endTime*, 'end', *task.name*, *task.assignee*));

6 **end**

7 *sort(events by time ascending, ưu tiên 'end' trước 'start');*

8 *activeTasks* $\leftarrow \{\}$;

// Map từ assignee \rightarrow tập công việc đang thực hiện

9 *prevTime* $\leftarrow \{\}$;

// Map từ assignee \rightarrow thời gian trước đó

10 *segmentCounters* $\leftarrow \{\}$;

// Map từ assignee \rightarrow bộ đếm cho từng task

11 *result* $\leftarrow \emptyset$;

12 **foreach** *event* \in *events* **do**

13 *assignee* \leftarrow *event.assignee*;

14 **if** *assignee* \notin *activeTasks* **then**

15 *activeTasks[assignee]* $\leftarrow \emptyset$;

16 *prevTime[assignee]* \leftarrow null;

17 *segmentCounters[assignee]* $\leftarrow \emptyset$;

18 **end**

19 *tasksSet* \leftarrow *activeTasks[assignee]*;

20 *counters* \leftarrow *segmentCounters[assignee]*;

21 **if** *prevTime[assignee]* \neq null **and** *tasksSet* $\neq \emptyset$ **and** *prevTime[assignee]* \neq *event.time* **then**

22 *duration* \leftarrow *event.time* - *prevTime[assignee]*;

23 *splitDuration* \leftarrow *duration* / *tasksSet.size*;

24 *offset* $\leftarrow 0$;

25 **foreach** *taskName* \in *tasksSet* **do**

26 *segmentStart* \leftarrow *prevTime[assignee]* + *offset*;

27 *segmentEnd* \leftarrow *segmentStart* + *splitDuration*;

28 **if** *counters[taskName]* chưa tồn tại **then**

29 *counters[taskName]* $\leftarrow 1$;

30 **end**

31 *result.append*(*Segment*(*taskName* + ' ' +

counters[taskName], *segmentStart*, *segmentEnd*, *assignee*));

32 *counters[taskName]* ++;

33 *offset* \leftarrow *offset* + *splitDuration*;

34 **end**

35 **end**

36 **if** *event.type* == 'start' **then**

37 *tasksSet.add*(*event.task*);

38 **end**

39 **else**

40 *tasksSet.remove*(*event.task*);

41 **end**

42 *prevTime[assignee]* \leftarrow *event.time*;

43 **end**

44 **return** *result*;

5 Ví dụ minh họa

5.1 Dữ liệu đầu vào

Tên công việc	Thời gian bắt đầu	Thời gian kết thúc	Người thực hiện
A	2025-02-12 00:00	2025-02-20 00:00	E1
B	2025-02-14 00:00	2025-02-21 00:00	E1
C	2025-02-18 08:00	2025-02-20 00:00	E1
D	2025-02-16 16:00	2025-02-18 08:00	E2

5.2 Kết quả

Tên công việc	Thời gian bắt đầu	Thời gian kết thúc	Người thực hiện
A.1	2025-02-12 00:00	2025-02-14 00:00	E1
A.2	2025-02-14 00:00	2025-02-16 00:00	E1
B.1	2025-02-16 00:00	2025-02-18 00:00	E1
D.1	2025-02-16 16:00	2025-02-18 08:00	E2
A.3	2025-02-18 00:00	2025-02-18 16:00	E1
B.2	2025-02-18 16:00	2025-02-19 08:00	E1
C.1	2025-02-19 08:00	2025-02-20 00:00	E1
B.3	2025-02-20 00:00	2025-02-21 00:00	E1

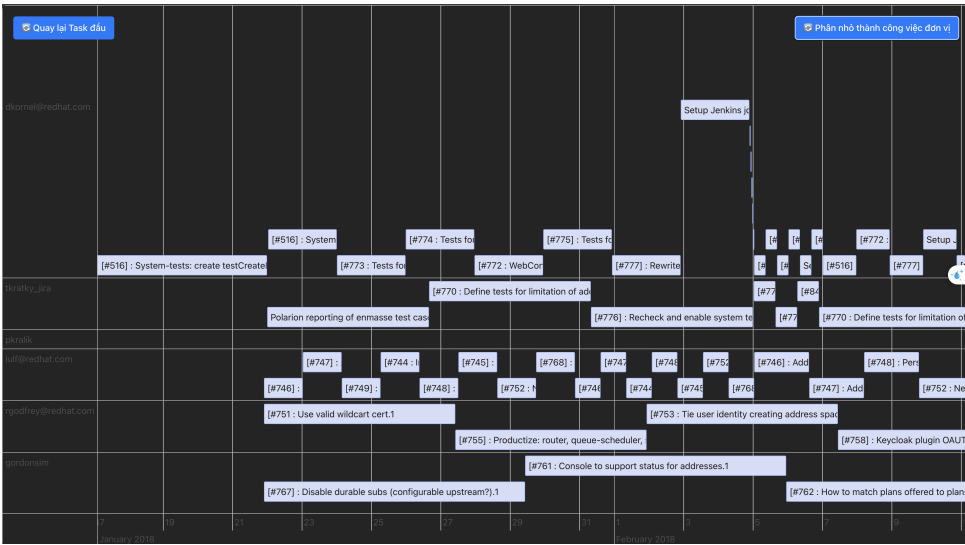
Chú thích: A.1, A.2, A.3 là phân đoạn của công việc A, B.1, B.2, B.3 là phân đoạn của công việc B, C.1 là phân đoạn của công việc C.

5.3 Demo

Trước khi phân đoạn:



Sau khi phân đoạn:



6 Kết luận

Thuật toán giúp chia nhỏ công việc chồng chéo theo nhân viên, đảm bảo không có thời gian nào bị trùng lặp giữa các công việc của cùng một người. Điều này giúp quản lý công việc dễ dàng và trực quan hơn.