

Đại học Bách khoa Hà Nội
Viện Công nghệ Thông tin và Truyền thông

Sử dụng mạng Bayesian ước lượng tài nguyên khả dụng cho bài toán lập lịch trong môi trường Cloud Computing

Quang Khanh
khanh.tq170083@gmail.com

Ngày 30 tháng 6 năm 2021



Giới thiệu chung

Các vấn đề của bài toán lập lịch thời gian thực

Giải pháp

Đánh giá hiệu năng

Kết luận



Giới thiệu chung

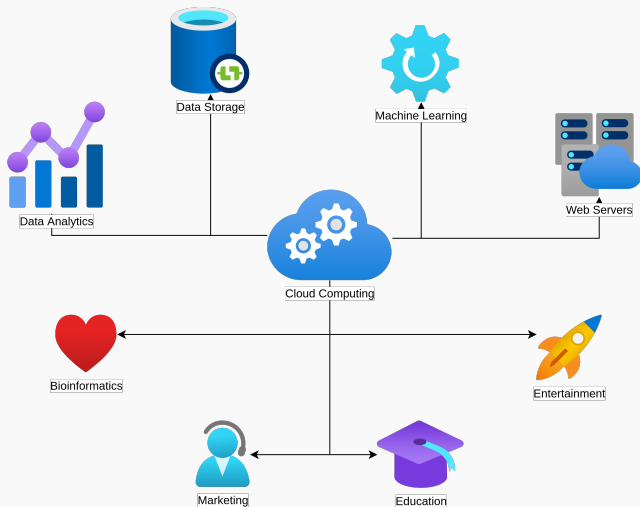
Các vấn đề của bài toán lập lịch thời gian thực

Giải pháp

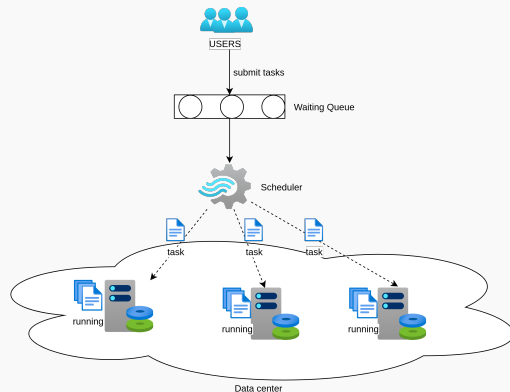
Đánh giá hiệu năng

Kết luận

Sự xuất hiện của Cloud Computing



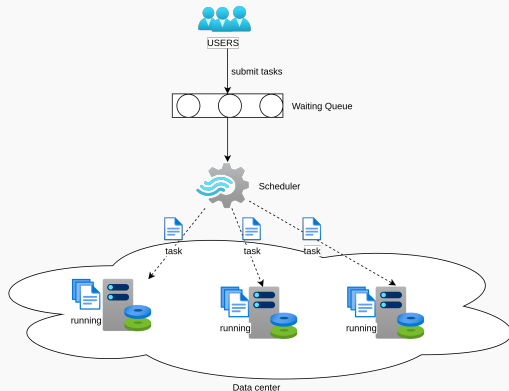
Luồng hoạt động



Hình: Người dùng gửi tasks đến hệ thống

Luồng hoạt động

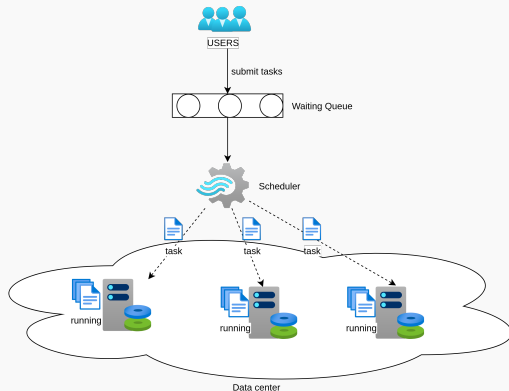
1. Người dùng gửi các tasks đến hệ thống



Hình: Người dùng gửi tasks đến hệ thống

Luồng hoạt động

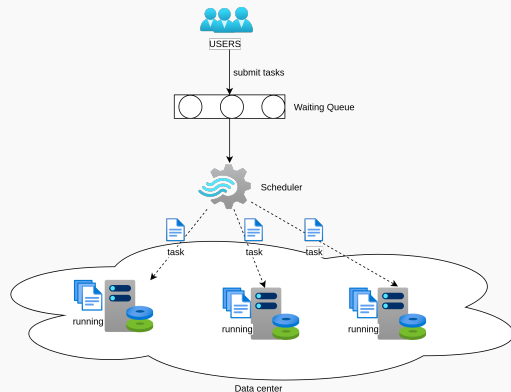
1. Người dùng gửi các tasks đến hệ thống
2. Các tasks được đưa đến hàng đợi cho đến khi được lập lịch



Hình: Người dùng gửi tasks đến hệ thống

Luồng hoạt động

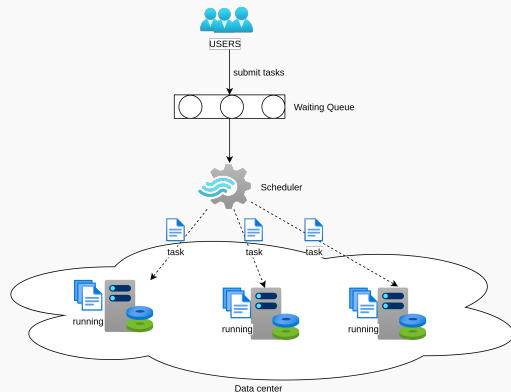
1. Người dùng gửi các tasks đến hệ thống
2. Các tasks được đưa đến hàng đợi cho đến khi được lập lịch
3. Bộ lập lịch tìm các máy tính phù hợp cho các tasks



Hình: Người dùng gửi tasks đến hệ thống

Luồng hoạt động

1. Người dùng gửi các tasks đến hệ thống
2. Các tasks được đưa đến hàng đợi cho đến khi được lập lịch
3. Bộ lập lịch tìm các máy tính phù hợp cho các tasks
4. Chuyển các tasks đến các máy tính và thực thi



Hình: Người dùng gửi tasks đến hệ thống



Giới thiệu chung

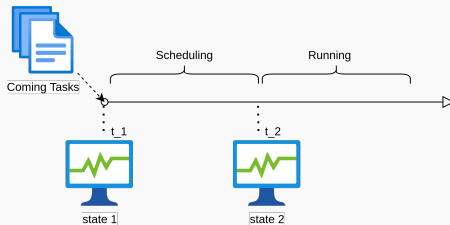
Các vấn đề của bài toán lập lịch thời gian thực

Giải pháp

Đánh giá hiệu năng

Kết luận

Trạng thái tài nguyên

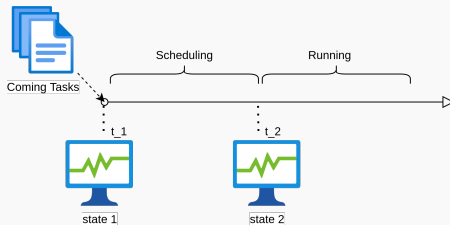


Hình: Trạng thái máy tính



Trạng thái tài nguyên

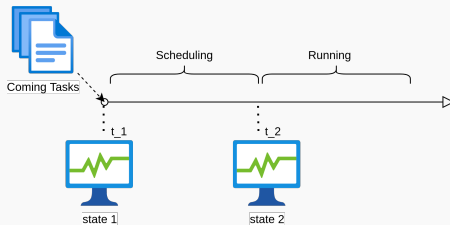
- ▶ Tại thời điểm t_1
 - ▶ $\text{available_cpu} = c_1$
 - ▶ $\text{available_memory} = m_1$



Hình: Trạng thái máy tính

Trạng thái tài nguyên

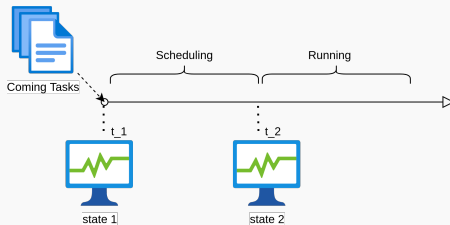
- ▶ Tại thời điểm t_1
 - ▶ available_cpu = c_1
 - ▶ available_memory = m_1
- ▶ Tại thời điểm t_2
 - ▶ available_cpu = c_2
 - ▶ available_memory = m_2



Hình: Trạng thái máy tính

Trạng thái tài nguyên

- ▶ Tại thời điểm t_1
 - ▶ available_cpu = c_1
 - ▶ available_memory = m_1
- ▶ Tại thời điểm t_2
 - ▶ available_cpu = c_2
 - ▶ available_memory = m_2

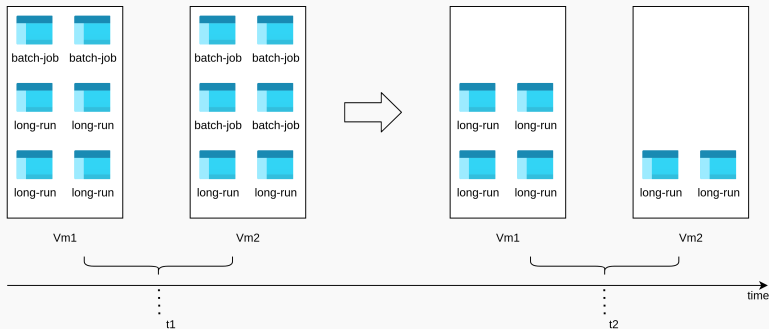


Hình: Trạng thái máy tính

Vấn đề

$$c_1 \neq c_2 \text{ hoặc } m_1 \neq m_2$$

Mất cân bằng khối lượng công việc khi hoạt động



Hình: Mất cân bằng giữa các task do sự kết thúc của batch-job tasks



Giới thiệu chung

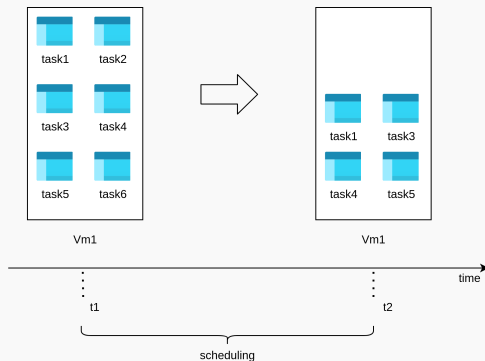
Các vấn đề của bài toán lập lịch thời gian thực

Giải pháp

Đánh giá hiệu năng

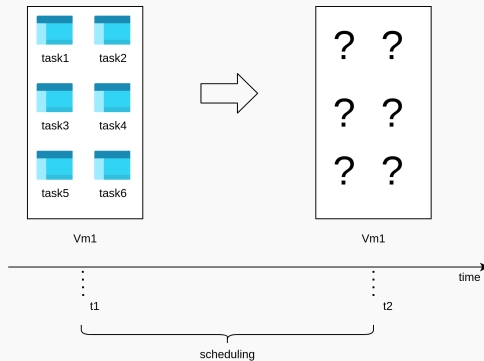
Kết luận

Ước lượng trạng thái các tasks



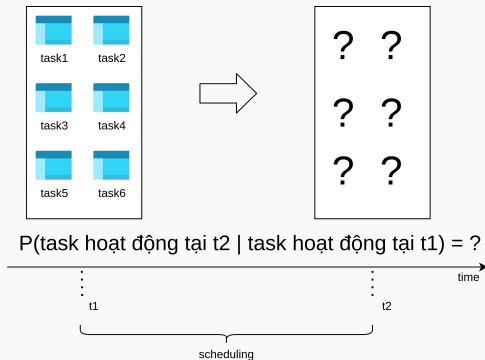
Hình: Trạng thái tài nguyên tại thời điểm lập lịch và thực thi

Ước lượng trạng thái các tasks



Hình: Trạng thái tài nguyên tại thời điểm lập lịch và thực thi

Ước lượng trạng thái các tasks



Hình: Trạng thái tài nguyên tại thời điểm lập lịch và thực thi

Ước lượng tài nguyên khả dụng





Tập các tasks đang chạy trên máy tính ảo tại thời điểm lập lịch

$$\mathcal{T} = \{task_1, task_2, ..., task_K\}$$

Tập các tasks đang chạy trên máy tính ảo tại thời điểm lập lịch

$$\mathcal{T} = \{task_1, task_2, \dots, task_K\}$$

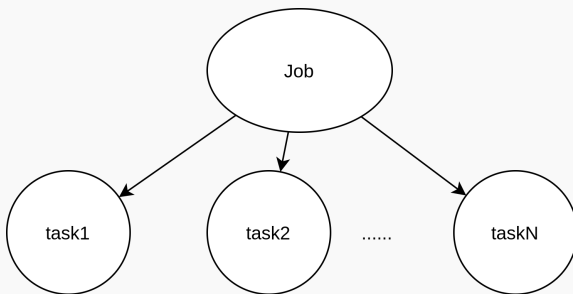
Ước lượng kì vọng tài nguyên khả dụng tại thời điểm thực thi

$$available_resources = resources_capacity - \sum_{i=1}^K resources_usage_i \times p_i$$

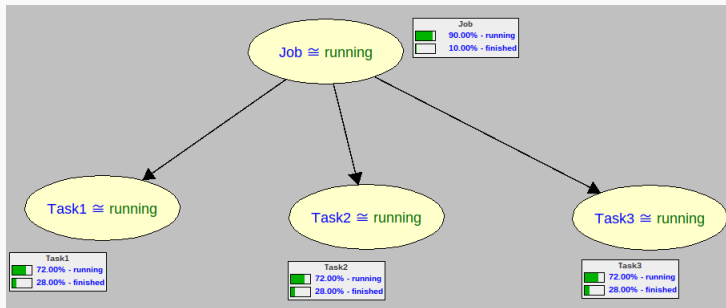
► p_i là xác suất $task_i$ còn hoạt động tại thời điểm thực thi

Mạng Bayesian ước lượng xác suất task còn hoạt động

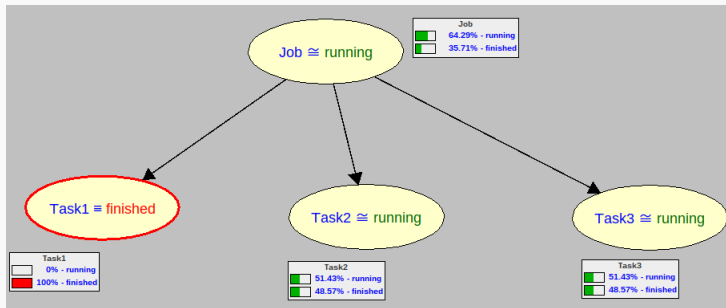




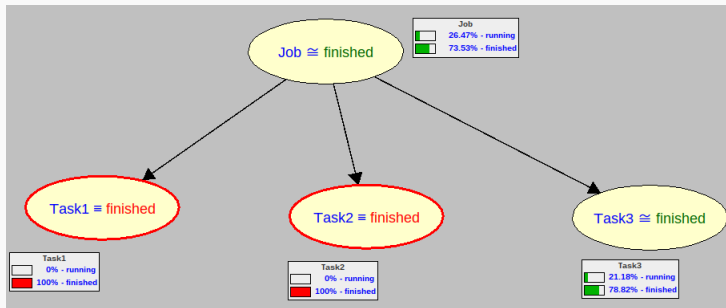
Hình: Task trong cùng job



Hình: Ví dụ về mạng Bayesian cho 3 tasks



Hình: Ví dụ về mạng Bayesian cho 3 tasks



Hình: Ví dụ về mạng Bayesian cho 3 tasks

Cân bằng khối lượng công việc giữa các máy ảo



Cân bằng khối lượng công việc giữa các máy ảo



Khối lượng công việc của M máy ảo

$$\mathcal{L} = \{l_1, l_2, \dots, l_M\}$$



Khối lượng công việc của M máy ảo

$$\mathcal{L} = \{l_1, l_2, \dots, l_M\}$$

Độ mất cân bằng

$$\sigma(\mathcal{L}) = \frac{1}{M} \times \sum_{i=1}^M (l_i - \bar{l})^2$$

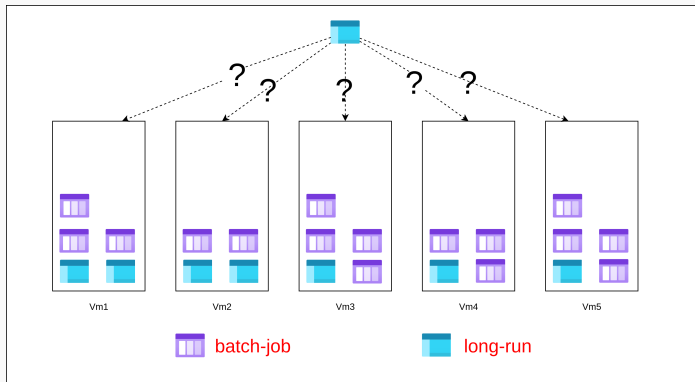
với

$$\bar{l} = \frac{1}{M} \sum_{i=1}^M l_i$$

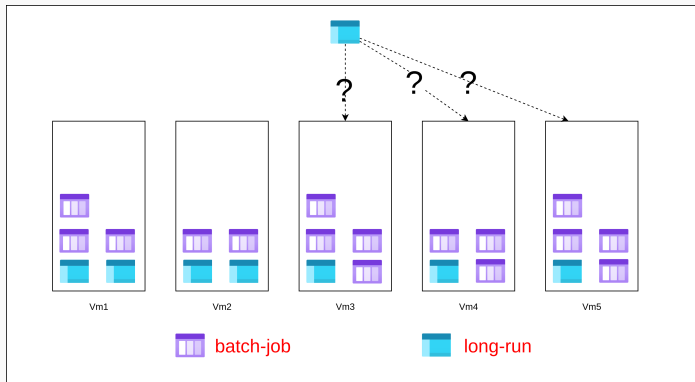
Lựa chọn máy tính ảo cho tasks



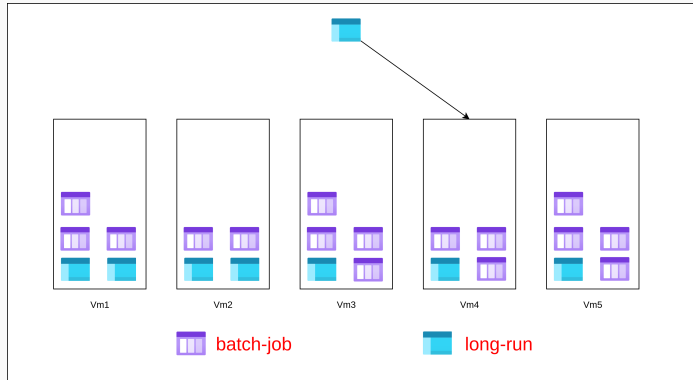
Lựa chọn máy tính ảo cho tasks



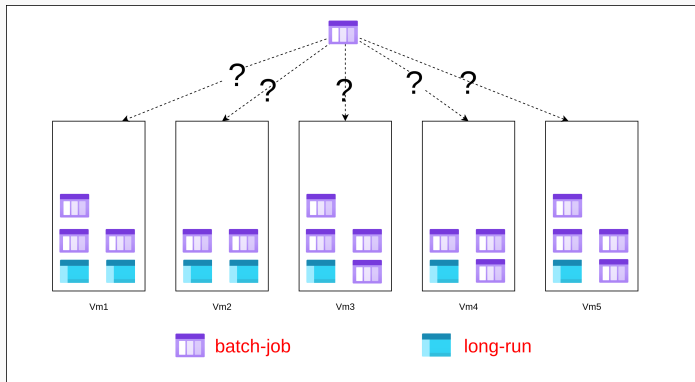
Lựa chọn máy tính ảo cho tasks



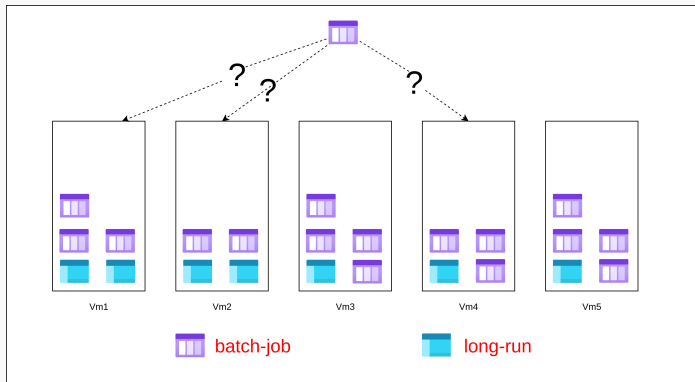
Lựa chọn máy tính ảo cho tasks



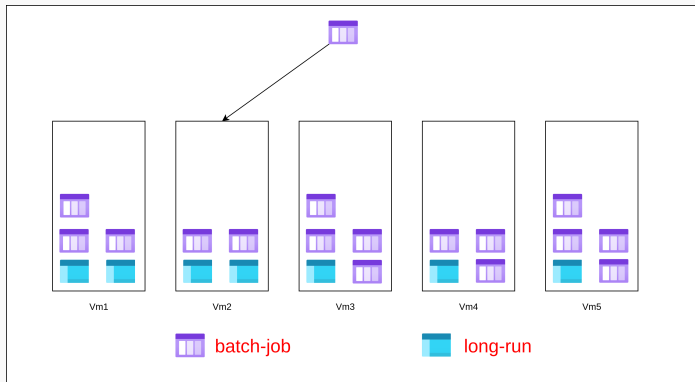
Lựa chọn máy tính ảo cho tasks



Lựa chọn máy tính ảo cho tasks



Lựa chọn máy tính ảo cho tasks



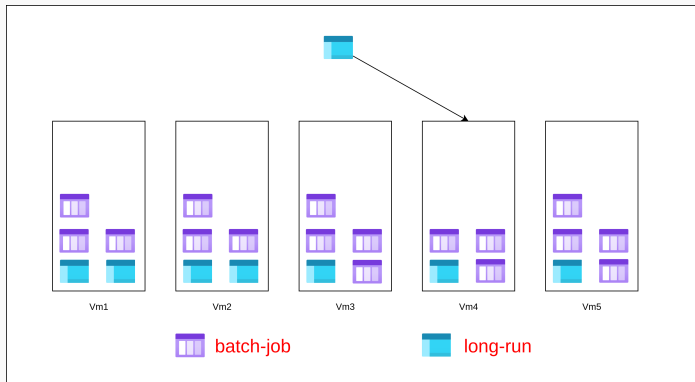
Lựa chọn máy tính ảo cho tasks



Lựa chọn máy tính ảo cho tasks



26



Lựa chọn máy tính ảo cho tasks





Giới thiệu chung

Các vấn đề của bài toán lập lịch thời gian thực

Giải pháp

Đánh giá hiệu năng

Kết luận



- ▶ $M = 10$ máy tính ảo
- ▶ $N = 15000$ tasks
- ▶ $K = 3$
- ▶ $T \sim \mathcal{P}(\lambda = 15)$
- ▶ $delay_time \sim \mathcal{N}(\mu = 5, \sigma = 0.5)$

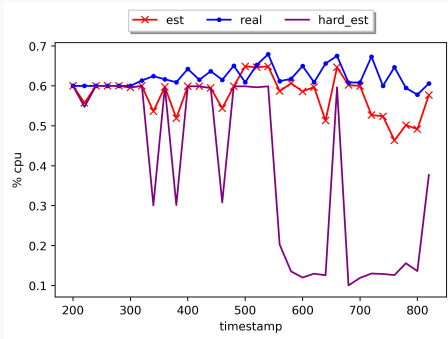
Bảng: Kết quả về thời gian chạy của các tasks

| Running statistics over 1000s | | | |
|-------------------------------|--------|----------|------------------------|
| stats | FCFS | Worstfit | Resources balancing |
| count | 13214 | 13925 | 14235 |
| mean | 10.62 | 6.34 | 5.42 |
| std | 54.24 | 33.37 | 27.31 |
| min | 0.31 | 0.12 | 0.21 |
| 50% | 5.78 | 3.52 | 3.41 |
| max | 829.92 | 616.35 | 640.39 |

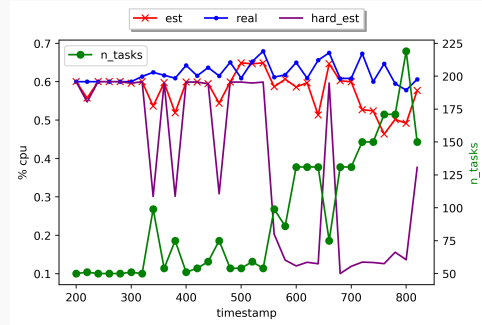
So sánh độ chính xác của mạng Bayesian



31



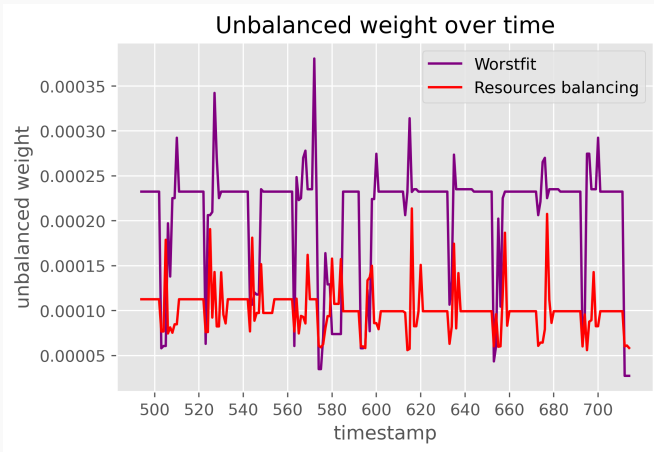
(a) Tài nguyên khả dụng tại thời điểm thực thi



(b) Sai số với số lượng tasks đang chạy

Hình: Thông số tại thời điểm kết thúc lập lịch

So sánh mức độ mất cân bằng giữa các máy tính



Hình: Mức độ mất cân bằng trong quá trình hoạt động



Giới thiệu chung

Các vấn đề của bài toán lập lịch thời gian thực

Giải pháp

Đánh giá hiệu năng

Kết luận





Kết luận

- ▶ Thuật toán đã cải thiện được sai số giữa thời điểm lập lịch và thực thi



Kết luận

- ▶ Thuật toán đã cải thiện được sai số giữa thời điểm lập lịch và thực thi
- ▶ Có thể cân bằng được khối lượng công việc trong quá trình chạy

Kết luận

- ▶ Thuật toán đã cải thiện được sai số giữa thời điểm lập lịch và thực thi
- ▶ Có thể cân bằng được khối lượng công việc trong quá trình chạy

Định hướng phát triển

Kết luận

- ▶ Thuật toán đã cải thiện được sai số giữa thời điểm lập lịch và thực thi
- ▶ Có thể cân bằng được khối lượng công việc trong quá trình chạy

Định hướng phát triển

- ▶ Phát triển mô hình đồ thị Bayesian phù hợp với học liên tục

Kết luận

- ▶ Thuật toán đã cải thiện được sai số giữa thời điểm lập lịch và thực thi
- ▶ Có thể cân bằng được khối lượng công việc trong quá trình chạy

Định hướng phát triển

- ▶ Phát triển mô hình đồ thị Bayesian phù hợp với học liên tục
- ▶ Mở rộng bài sang bài toán resource scaling



Thank You For Listening!