HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN XUÂN PHI

NÂNG CAO HIỆU NĂNG CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY

Chuyên ngành : Hệ thống thông tin
Mã số : 9.48.01.04

TÓM TẮT LUẬN ÁN TIẾN SĨ KỸ THUẬT

Hà Nội – Năm 2022

Công trình được hoàn thành tại:

Học viện Công nghệ Bưu chính Viễn thông

Người hướng dẫn khoa học: **PGS.TS Trần Công Hùng**

Phản biện 1: ……..……

Phản biện 2: ……..……

Phản biện 3: …….……..……

Luận án được bảo vệ trước Hội đồng chấm luận cấp Học viện họp tại: **Học viện Công nghệ Bưu chính Viễn thông**
Vào hồi …... giờ…..... ngày…….. tháng…….. năm…..…..…

Có thể tìm hiểu luận án tại thư viện:

- **Thư viện Quốc gia Việt Nam**
- **Thư viện Học viện Công nghệ Bưu chính Viễn thông**

MỞ ĐẦU

1. Lý do chọn đề tài

Điện toán đám mây hiện đã trở thành một nguồn tài nguyên tính toán và lưu trữ phổ biến và quan trọng với người dùng. Sự phát triển của các hệ thống điện toán đám mây cũng kéo theo những bài toán tối ưu hóa hạ tầng kỹ thuật của điện toán đám mây nhằm tận dụng tối đa năng lực của hạ tầng cũng như đem lại chất lượng dịch vụ tốt nhất[3], [11], [21], [28], [44], [72], [93], [104], [109].

Trong điện toán đám mây, ngoài các vấn đề về trao đổi, xử lí, an toàn dữ liệu, vấn đề cân bằng tải (load balancing) truy nhập có vai trò quan trọng nhằm nâng cao chất lượng dịch vụ. Cho nên việc nghiên cứu cải tiến các thuật toán nâng cao hiệu năng cân bằng tải trên điện toán đám mây là rất cần thiết, thu hút được sự quan tâm nghiên cứu của nhiều nhà khoa học cũng như các nhà khai thác, cung cấp dịch vụ thông tin truyền thông ở trong và ngoài nước [5], [18], [23], [31], [34],[35], [42], [83], [84], [108].

Vì vậy, việc nghiên cứu các giải pháp mới nâng cao hiệu năng cân bằng tải là vấn đề cấp thiết. Đề tài ***"Nâng cao hiệu năng cân bằng tải trên điện toán đám mây***" được thực hiện trong khuôn khổ luận án tiến sĩ chuyên ngành Hệ thống thông tin, góp phần giải quyết một số hạn chế trong các thuật toán nhằm nâng cao hiệu năng cân bằng tải trên điện toán đám mây.

2. Mục tiêu của luận án

- Mục tiêu thứ nhất: Nghiên cứu, phát triển một số thuật toán cân bằng tải nhằm cải thiện thời gian đáp ứng trên điện toán đám mây.
- Mục tiêu thứ hai: Nghiên cứu, phát triển một số thuật toán cân bằng tải nhằm cải thiện thời gian xử lý trên điện toán đám mây.

3. Phạm vi, đối tượng và phương pháp nghiên cứu

- Phạm vi: Các thuật toán cân bằng tải nhằm cải thiện các tham số thời gian đáp ứng, thời gian xử lý trên môi trường điện toán đám.
- Đối tượng nghiên cứu: Các thuật toán cân bằng tải, các tham số ảnh hưởng đến cân bằng tải, các phương pháp đánh giá thuật toán cân bằng tải trên điện toán đám mây.
- Phương pháp nghiên cứu: Xây dựng mô hình; Cài đặt/thử nghiệm mô hình trên các phần mềm mô phỏng; So sánh, đánh giá kết quả so với các thuật toán khác.

4. Các đóng góp của luận án

- Đóng góp thứ nhất: Đề xuất 02 thuật toán cân bằng tải nhằm giảm thời gian đáp ứng trên điện toán đám mây: Thuật toán LBAIRT (CT4) cải tiến tham số thời gian đáp ứng trong giải thuật Throttled [89]. Thuật toán RRTA (CT7) cải tiến tham số thời gian đáp ứng của tác giả Sharma [2] bằng kỹ thuật dự báo thời gian đáp ứng ARIMA.
- Đóng góp thứ hai: Đề xuất 02 thuật toán cân bằng tải nhằm giảm thời gian xử lý trên điện toán đám mây: Thuật toán TMA (CT5) cải tiến bảng chỉ mục trạng thái của thuật toán Throttled [89]. Thuật toán MMSIA (CT6) cải tiến tham số thời gian hoàn thành dự kiến của nhóm yêu cầu và máy ảo trong thuật toán lập lịch Min-Max [69].

5. Bố cục luận án

Nội dung của luận án chia thành 03 chương như sau:

Chương 1 - Giới thiệu điện toán đám mây, bài toán cân bằng tải trên điện toán đám mây; phân tích ảnh hưởng của các tham số đến cân bằng tải. Giới thiệu các nghiên cứu liên quan và xác định rõ hướng nghiên cứu của đề tài là.

Chương 2 - Đề xuất 02 thuật toán nhằm cải thiện thời gian đáp ứng trên điện toán đám mây: Cải tiến thời gian đáp ứng trong giải thuật

Throttled để đưa ra thuật toán LBAIRT (CT4). Cải tiến thời gian đáp ứng trong thuật toán của tác giả Sharma [2] để đưa ra thuật toán RRTA (CT7). Tiến hành thử nghiệm và đánh giá kết quả của các phương pháp đề xuất.

Chương 3 – Đề xuất 02 thuật toán nhằm cải thiện thời gian xử lý trên điện toán đám mây: Cải tiến bảng chỉ mục trạng thái của thuật toán Throttled để đưa ra thuật toán TMA (CT5). Cải tiến thời gian hoàn thành dự kiến của nhóm yêu cầu và máy ảo trong thuật toán lập lịch Min-Max để đưa ra thuật toán MMSIA (CT6). Tiến hành thử nghiệm để đánh giá kết quả của các phương pháp đề xuất.

Kết luận: Tóm tắt các kết quả đã đạt được, các đóng góp mới và đề xuất hướng phát triển của luận án.

CHƯƠNG 1.

TỔNG QUAN VỀ CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY

1.1. Cân bằng tải trên điện toán đám mây

1.1.1. Giới thiệu chung về điện toán đám mây

Điện toán đám mây, còn gọi là điện toán máy chủ ảo, là mô hình điện toán sử dụng các công nghệ máy tính và phát triển dựa vào mạng Internet. Thuật ngữ "đám mây" ở đây chỉ độ phức tạp của các cơ sở hạ tầng mạng Internet [66].

1.1.2. Cân bằng tải và hiệu năng cân bằng tải trên điện toán đám mây

Khái niệm cân bằng tải chủ yếu đề cập đến ý tưởng phân phối tải đồng đều trên các tài nguyên công nghệ thông tin có sẵn, đảm bảo rằng không có máy chủ nào bị quá tải [110], [111], [112]. Nâng cao hiệu năng cân bằng tải là nhiệm vụ của các thuật toán cân bằng tải thông qua việc tối ưu các tham số ảnh hưởng đến cân bằng tải.

1.1.3. Sự cần thiết của cân bằng tải trên điện toán đám mây

Do sự phát triển không ngừng của điện toán đám mây nên vấn đề trao đổi, xử lý, an toàn dữ liệu và đặc biệt vấn đề cân bằng tải truy nhập là những vấn đề cấp thiết và có ý nghĩa thực tiễn cao, vì vấn đề cơ bản nhất trong truyền thông là những vấn đề liên quan đến dữ liệu [14].

1.1.4. Ảo hóa và quản lý máy ảo trên đám mây

Ảo hóa là việc tách hệ điều hành khỏi phần cứng, tạo ra sự di chuyển hệ điều hành và các ứng dụng từ phần cứng này sang phần cứng khác mà mọi thứ vẫn nguyên vẹn [3], [12], [13], [40], [91], [92],

1.1.5. Quản lý và phân bổ tài nguyên trên điện toán đám mây

Trong điện toán đám mây, phân bổ tài nguyên là quá trình chỉ định động các tài nguyên có sẵn cho các ứng dụng đám mây được yêu

cầu [1], [26, [53], [60], [95]. Việc phân bổ tài nguyên xảy ra ở lớp IaaS và tài nguyên sử dụng có thể bao gồm hệ điều hành và ứng dụng cho người dùng.

1.2. Bài toán cân bằng tải

1.2.1. Phát biểu bài toán và mô hình nghiên cứu

- Đặt vấn đề [43]:
  - Có *m* máy M = {M_1, M_2, M_3,…..,M_m}.
  - Có *n* công việc J = {j_1, j_2, j_3,….., j_n}, với mỗi công việc có thời gian xử lý là $t_j > 0$.

- Yêu cầu đặt ra: Gán tập công việc J cho tập máy M sao cho tải trên tất cả các máy M là đồng đều nhất có thể. Hạn chế tình trạng quá tải trên một máy bất kỳ, trong khi máy khác còn lại thì không phục vụ công việc nào cả.

- Đặt A(i) là tập công việc gán cho máy ảo Mi, nên máy Mi cần làmviệc trong tổng thời gian [43]:

$$T_i = \sum_{j \in A(i)} t_j \qquad (1.1)$$

Đại lượng T_i là thời gian cần thiết để hoàn thành việc thực hiện tất cả các yêu cầu đầu vào và đó cũng là tải (load) trên các máy M_i [43]. Mục tiêu của bài toán là tối thiểu hóa đại lượng $T = max_i T_i$, với T là tải lớn nhất trên bất kỳ máy nào.

1.2.2. Các yếu tố ảnh hưởng đến cân bằng tải

Có nhiều yếu tố ảnh hưởng đến khả năng cân bằng tải trên điện toán đám mây [10], [15], [67], [70], [72], [98], [106]. Các kết công bố

trong công trình (CT1), (CT2) và (CT3) đã nghiên cứu về ảnh hưởng các tham số ảnh hưởng trực tiếp đến cân bằng tải: thời gian đáp ứng, thời gian xử lý, thời gian chờ, mức độ sử dụng tài nguyên, mức độ ưu tiên của các yêu cầu đầu vào, trạng thái của các nút mạng, băng thông cho máy ảo và các cơ chế phân phối tải.

1.2.3. Phân loại các thuật toán cân bằng tải

- Thuật toán cân bằng tĩnh [98]: không xét đến trạng thái hoặc hành vi trước đó của nút trong khi phân phối tải.
- Thuật toán cân bằng động [98]: kiểm tra trạng thái trước đó của một nút trong khi phân phối tải.

1.2.4. Đo lường cân bằng tải

Có rất nhiều các công cụ mô phỏng để đánh giá hiệu năng một thuật toán các thuật toán trên điện toán đám mây như: CloudSim, Cloud Analyst, GridSim, CSIM for Java, Matlab [9], [22], [65], [74],...tất cả các công cụ này đã và đang hỗ trợ đắc lực cho việc đánh giá hiệu năng các thuật toán cân bằng tải.

1.3. Các hướng giải quyết bài toán cân bằng tải

1.3.1. Phương pháp xấp xỉ

Bài toán cân bằng tải là một bài toán NP-Complete [79], hiện chưa có lời giải tối ưu, tuy nhiên có thể giải quyết thông qua các thuật toán xấp xỉ. Tuy không thể tính toán chính xác giá trị của T* (là giá trị tối ưu), nhưng rõ ràng đã có thuật toán cân bằng tải để tính toán giá trị T tiệm cận đến giá trị tối ưu T*. Điều đó có nghĩa là có thể dùng thuật toán xấp xỉ để giải quyết bài toán cân bằng tải

1.3.2. Chiến lược lập lịch phân bổ tài nguyên

Hướng nghiên cứu tiếp theo là cải tiến các chiến lược lập lịch phân bổ tài nguyên nhằm cân bằng nguồn lực trên điện toán đám mây [7], [16], [19], [25], [63], [87], [97], [107]. Công trình [16] tác giả đề xuất một thuật toán quản lý tải động để phân phối có hiệu quả toàn bộ các yêu cầu đến các máy ảo. Giải thuật được mô phỏng bằng công cụ

CloudAnalyst dựa trên các thông số khác nhau như thời gian xử lý dữ liệu và thời gian đáp ứng. Công trình [51], [56], [57], [95] đưa ra phương pháp tối đa hóa việc sử dụng tài nguyên bằng mô hình cân bằng tải, các phương pháp này hỗ trợ trong việc dự báo xu hướng của các tài nguyên, việc cấp phát tài nguyên động và việc giải phóng bộ nhớ của các máy chủ một cách hiệu quả.

Các công trình gần đây nhất [24], [36], [71], [88] cũng đã tập trung nghiên cứu cải tiến các chiến lực phân bổ tài nguyên. Tài liệu [88] đề xuất cơ chế tránh lãng phí tài nguyên, công trình [36] đề xuất cơ chế di trú trực tiếp các yêu cầu hiệu quả, giảm chi phí về thời gian thực hiện. Công trình [24], [71] đề xuất các chiến lược phân bổ tải cho các máy ảo một cách hiệu quả.

1.3.3. Phương pháp cải tiến các tham số

Một hướng nghiên cứu nữa là cải tiến các tham số ảnh hưởng đến khả năng cân bằng tải thu hút rất nhiều tác giả trên thế giới [2], [5],], [49], [52], [58], [70], [77], [80], [90], [94]. Các tham số cơ bản đó là: thời gian đáp ứng, thời gian xử lý, thời gian hoàn thành, tỉ lệ sử dụng tài nguyên, độ ưu tiên của các nhiệm vụ... Nhóm tác giả Agraj Sharma trong công trình [2] đã cho rằng yếu tố thời gian đáp ứng ảnh hưởng lớn đến hiệu năng cân bằng tải trên điện toán đám mây. Tác giả [2] nêu ra 2 vấn đề còn tồn tại của các giải thuật trước đây là: 1) cân bằng tải chỉ xảy ra sau khi các máy chủ bị quá tải; 2) liên tục truy vấn thông tin tài nguyên sẵn có dẫn đến tăng chi phí tính toán và tiêu thụ băng thông. Vì vậy, tác giả đã đề xuất thuật toán cải tiến thời gian đáp ứng của các yêu cầu để quyết định gán các yêu cầu cho các máy chủ một cách thích hợp, cách tiếp cận của giải thuật này đã giảm được sự truy vấn thông tin về các nguồn lực sẵn có, giảm sự giao tiếp và tính toán trên mỗi máy chủ.

Các công trình gần đây [27], [64], [85], [86], [103] cũng đã công bố những kết quả cải tiến các tham số ảnh hưởng trực tiếp đến cân bằng tải. Bằng các phương pháp khác nhau, các tác giả đã thực hiện cải tiến các tham số chính và đã cho kết quả tốt hơn các phương pháp đã công bố trước đó. Thời gian đáp ứng là tham số rất quan trọng, nó

ảnh hưởng đến hiệu năng của đám mây cũng như là QoS của các nhà cung cấp dịch vụ điện toán. Chính vì vậy, đã có nhiều thuật toán cân bằng tải đề xuất với mục tiêu tối ưu hóa thời gian đáp ứng trên điện toán đám mây [18], [29], [33], [56], [60], [108].

Có nhiều tham số ảnh hưởng đến khả năng cân bằng tải trên điện toán đám mây, tuy nhiên trong khuôn khổ luận án chỉ nghiên cứu 2 tham số chính phục vụ mục tiên nghiên cứu đó là: thời gian đáp ứng, thời gian xử lý.

1.4. Các vấn đề mà luận án cần giải quyết

Thông qua khảo sát, đánh giá các công trình nghiên cứu liên quan ở mục 1.2 và 1.3, nghiên cứu sinh rút ra một số nhận xét:

- Thuật toán [89] đã không xem xét tới lượng tải hiện tại của máy ảo, vì vậy khi một máy ảo không đủ năng lực thực hiện yêu cầu nó sẽ phải quay trở lại tìm một máy ảo tiếp theo, hệ thống sẽ tiêu tốn thời gian hơn. Vấn đề này được giải quyết trong Chương 2 và kết quả nghiên cứu được công bố ở công trình (CT4).
- Phương pháp giảm thời gian xử lý của thuật toán trong công trình [89] tồn tại một vấn đề là: bộ cân bằng tải phải tìm kiếm toàn bộ danh sách các máy ảo để tìm ra máy ảo nào đang sẵn sàng cho việc phân bổ tải, điều này làm tăng thời gian xử lý các công việc. Phương pháp giải quyết được thực hiện trong Chương 3 và được công bố trong công trình (CT5).
- Cũng nhằm mục đích cải thiện thời gian xử lý, thuật toán Max – Min [69] có điểm hạn chế là bộ cân bằng tải duyệt trên toàn bộ danh sách các máy ảo để thực hiện việc gán yêu cầu cho máy ảo nào thỏa mãn yêu cầu, vấn đề này sẽ làm tăng thời gian xử lý, do các thao tác này lặp đi lặp lại. Do đó, vấn đề đặt ra là phải giảm thiểu các các thao tác lặp lại này để cải thiện thời gian xử lý. Và vấn đề này cũng được giải quyết trong Chương 3, các kết quả nghiên cứu cũng được công bố trên công trình (CT6).

- Thuật toán RRTA (CT7): Sử dụng thuật toán dự báo ARIMA để cải thiện thời gian đáp ứng trong [2], để đưa ra cách phân phối tài nguyên hợp lý. Vấn đề này được trình bày trong Chương 2 của luận án và được công bố trên công trình (CT7).

1.5. Kết luận Chương 1

Trong chương này, đã trình bày tổng quan về cân bằng tải trên điện toán đám mây. Trong Chương 2, việc phát triển các thuật toán cân bằng tải nhằm cải thiện thời gian đáp ứng được nghiên cứu và đã chứng minh được hiệu quả của nó trên điện toán đám mây.

CHƯƠNG 2.

PHÁT TRIỂN MỘT SỐ THUẬT TOÁN CÂN BẰNG TẢI NHẰM CẢI THIỆN THỜI GIAN ĐÁP ỨNG TRÊN ĐIỆN TOÁN ĐÁM MÂY

2.1. Đặt vấn đề

Việc lập kế hoạch cho sử dụng tài nguyên dựa trên thời gian đáp ứng của dịch vụ là rất quan trọng.

2.2. Thuật toán LBAIRT

2.2.1. Cơ sở lý thuyết

Hạn chế của thuật toán Throttled [89]: Giải thuật phải dò tìm VM đang sẵn sàng '0' với toàn bộ kích thước bảng danh sách VM ban đầu do đó làm tăng chi phí thời gian.

2.2.2. Đề xuất thuật toán

Thuật toán LBAIRT được công bố trong công trình (CT4). trên hạ tầng ảo hóa đám mây là phức tạp, tùy thuộc vào cơ chế lập lịch tài nguyên tính toán trong hệ thống.

Điểm mới của thuật toán LBAIRT là xét thêm số tham số thời gian hoàn thành công việc dự kiến của mỗi VM khi có các danh sách yêu cầu đến. Thời gian đáp ứng dự kiến được tính theo công thức sau [74]:

$$TR_{dk} = F_t - A_t + T_{delay} \quad (2.1)$$

Trong đó:

- TR_{dk}: Thời gian đáp ứng dự kiến.
- F_t: là thời điểm hoàn thành dự kiến xử lý Cloudlet.
- A_t: là thời điểm đến của Cloudlet.
- T_{delay}: là thời gian truyền tải các yêu cầu. Vì thuật toán thực hiện công việc điều phối tải của DatacenterBroker nên mức độ của thuật toán chỉ ảnh hưởng đến thời gian xử lý trong một môi trường mạng nội bộ của một Datacenter. Do đó tham số về độ trễ truyền có thể bỏ qua, nên $T_{delay} = 0$.

Xác định Ft [74]: Do sử dụng chính sách Timeshared nên F_t của yêu cầu *p* được quản lý bởi VM_i được tính như sau:

$$F_t = ct + \frac{rl(p)}{capacity \times cores(p)} \quad (2.2)$$

Với:

$$capacity = \frac{\sum_{i=1}^{np} cap(i)}{max\left(\sum_{j=1}^{cloudlets} cores(j), np\right)} \quad (2.3)$$

Thời gian thực thi của một Cloudlet được xác định theo công thức sau:

$$Thời\ gian\ thực\ thi\ một\ cloudlet = \frac{rl}{capacity \times cores(p)} \quad (2.4)$$

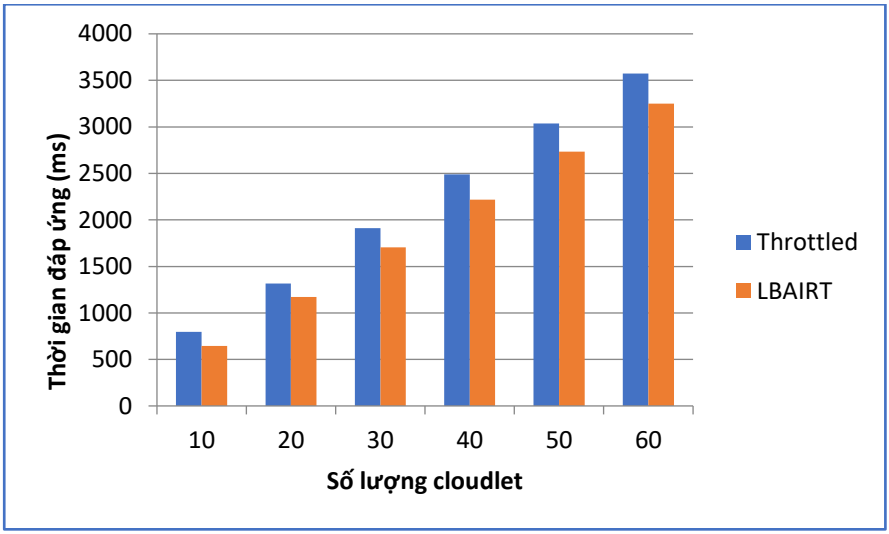
Trong đó:

- *rl*: là tổng số lệnh mà Cloudlet *p* cần được thực thi trên một bộ xử lý.
- *capacity*: là năng lực xử lý trung bình (tính theo MIPS) của một core dành cho Cloudlet *p*.
- *ct*: là thời gian mô phỏng hiện tại.
- *cores(p)*: là số lượng core mà Cloudlet p cần.
- *np*: là số lượng core thực mà host đang xét có.
- *cap:* là năng lực xử lý của core

Độ phức tạp của thuật toán LBAIRT: O((n+p) mq).

2.2.3. Kết quả mô phỏng

Mục tiêu của mô phỏng này là so sánh, phân tích, đánh giá thời gian đáp ứng của thuật toán Throttled [89] và thuật toán đề xuất LBAIRT.



Hình 2.4. Thực nghiệm mô phỏng so sánh thời gian đáp ứng của Throttled và LBAIRT khi thay đổi số lượng cloudlet.

Thuật toán LBAIRT được đề xuất dựa trên thuật toán Throttled. Trong thuật toán Throttled, không xét đến lượng tải trên VM. Trong thuật toán đề xuất, ngoài việc xét thời gian hoàn thành dự kiến còn xét đến năng lực xử lý nhiệm vụ/yêu cầu của VM. Kết quả mô phỏng cho thấy thuật toán LBAIRT đã giảm được thời gian đáp ứng.

2.3. Thuật toán RRTA

2.3.1. Đề xuất thuật toán

Thuật toán RRTA dựa vào thuật toán dự báo ARIMA [75] để dự báo thời gian đáp ứng, giúp phân bổ hiệu quả các yêu cầu đầu vào.

Các bước của thuật toán:

1. For each Request in CloudRequests
2. T_{new} = ARIMA(RT_i); *// Module 1*
3. isLocated = false;
4. For each VM in VMList

```
5.          If VM.getPredictedRT()  < Tnew
6.              AllocateRequestToVM(VM, Request); // Module 3
7.              isLocated = true;
8.          End If
9.      End For
10.     If (!isLocated)
11.         VM = VMList.getMinDistance(Tnew);  // Module 2
12.         AllocateRequestToVM(VM, Request);
13.     End If
14.  End For
```

Theo tài liệu [2], ngưỡng được tính toán chính là thời gian đáp ứng lớn nhất xét trong tập các VM. Vì thế, thuật toán RRTA này sử dụng lại phương pháp chọn ngưỡng này, tuy nhiên sẽ hiệu chỉnh một số thay đổi, hoặc đưa vào các hệ số và tham số, tùy thuộc vào kết quả thực nghiệm.

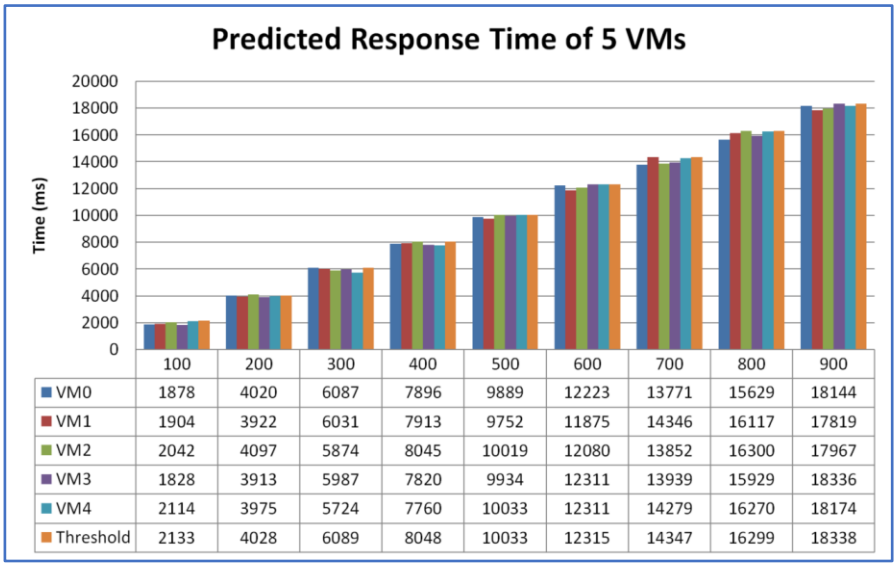
Độ phức tạp tính toán: O(nm).

2.3.2. Thực nghiệm mô phỏng:

Hình 2.8, so sánh thời gian đáp ứng dự báo của các VM với ngưỡng tính toán ứng với trường hợp 5 VM, có thể thấy sự phân bổ khá ổn định và hợp lý của thuật toán, thời gian đáp ứng dự báo của các VM không quá khác biệt so với thời gian dự báo của đám mây (tức là ngưỡng).

Đánh giá thuật toán:

Mô phỏng này chưa tính tới việc mở rộng tập các VM (VM pool) để giảm tải trong trường hợp cần thiết, do giả định nhóm các VM này xử lý tối đa bao nhiêu yêu cầu, nếu vượt quá mới mở rộng pool. Tuy nhiên, việc mô phỏng với lượng request lớn trên 1000 yêu cầu đòi hỏi cấu hình phần cứng máy tính mạnh hơn và bộ xử lý tốt hơn, đây là hạn chế của mô phỏng này. Việc mô phỏng với 5 VM, chịu tải từ 100 tới 900 yêu cầu đã cho thấy kết quả tương đối tốt, việc phân bổ các request tới các VM xử lý khá đồng đều và dự đoán với sai số nhỏ.



	100	200	300	400	500	600	700	800	900
VM0	1878	4020	6087	7896	9889	12223	13771	15629	18144
VM1	1904	3922	6031	7913	9752	11875	14346	16117	17819
VM2	2042	4097	5874	8045	10019	12080	13852	16300	17967
VM3	1828	3913	5987	7820	9934	12311	13939	15929	18336
VM4	2114	3975	5724	7760	10033	12311	14279	16270	18174
Threshold	2133	4028	6089	8048	10033	12315	14347	16299	18338

Hình 2.8. So sánh thời gian đáp ứng dự báo của 5 VM và ngưỡng

2.4. Kết luận Chương 2

Chương 2 trình bày 02 thuật toán nhằm mục đích cải tiến thời gian đáp ứng trên điện toán đám mây cùng các kết quả mô phỏng thực nghiệm để chứng minh hiệu quả của thuật toán đề xuất. Các kết quả nghiên cứu được công bố ở các công trình (CT4) và (CT7).

Có nhiều phương pháp để nâng cao hiệu năng cân bằng tải trên điện toán đám mây, trong Chương 3 tiếp tục nghiên cứu nâng cao khả năng này dựa trên một trong những tham số quan trọng là thời gian xử lý.

CHƯƠNG 3.

PHÁT TRIỂN MỘT SỐ THUẬT TOÁN CÂN BẰNG TẢI NHẰM CẢI THIỆN THỜI GIAN XỬ LÝ TRÊN ĐIỆN TOÁN ĐÁM MÂY

3.1. Đặt vấn đề

Các thuật toán cân bằng tải hiện nay đang cố gắng cải thiện thời gian xử lý các yêu cầu trên điện toán đám mây một cách tối ưu nhằm tăng hiệu năng phục vụ của trung tâm dữ liệu đám mây.

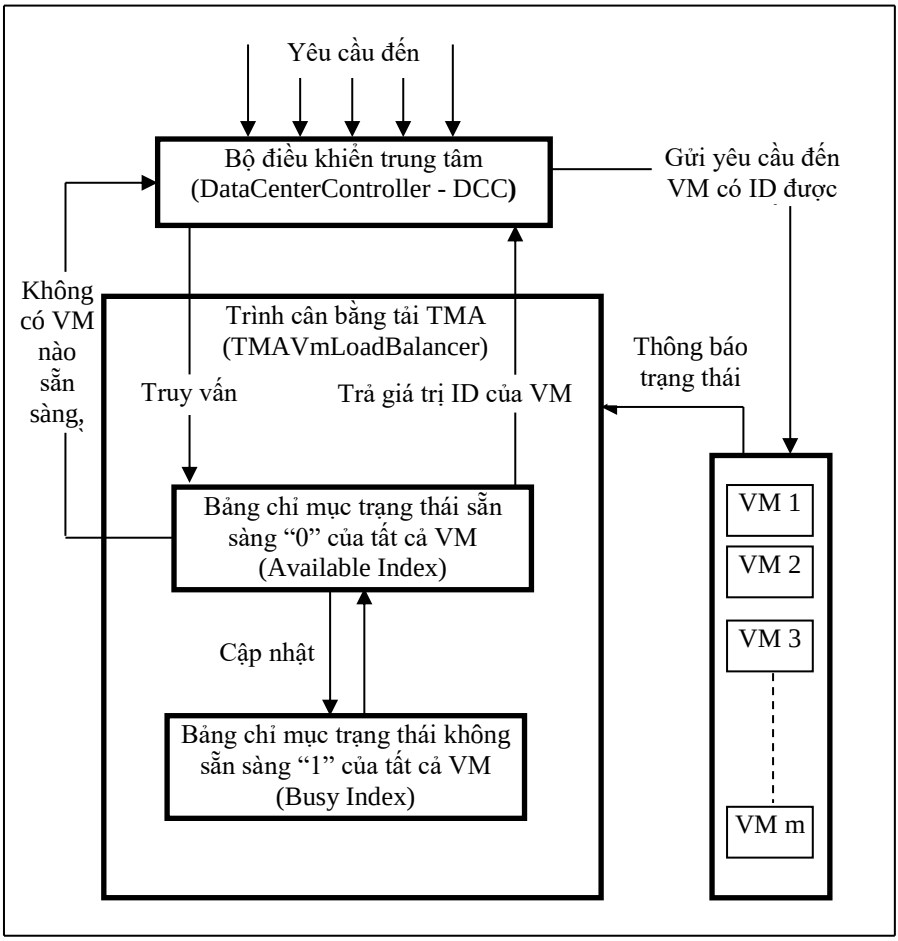
3.2. Thuật toán TMA

3.2.1. Đề xuất thuật toán

Thuật toán TMA (Throttled Modified Algorithm) cải thiện thời gian xử lý của trung tâm dữ liệu (Data Center) dựa trên thuật toán gốc Throttled [89].

Điểm mới của thuật toán TMA: Việc dò tìm VM đang sẵn sàng '0' với kích thước bảng "Available Index" thay đổi linh động hơn so với thuật toán Throttled. Bộ cân bằng tải tốt ít chi phí thời gian do duy trì 2 bảng danh sách các VM "sẵn sàng" và "bận", bộ cân bằng tải chỉ việc gán VM cho các yêu cầu mới đến. Điều này giúp tăng hiệu suất xử lý cho hệ thống đồng nghĩa với giảm thời gian xử lý các yêu cầu đầu vào.

Độ phức tạp tính toán của thuật toán: $O(n^2)$.



Hình 3.1. Sơ đồ hoạt động của thuật toán TMA

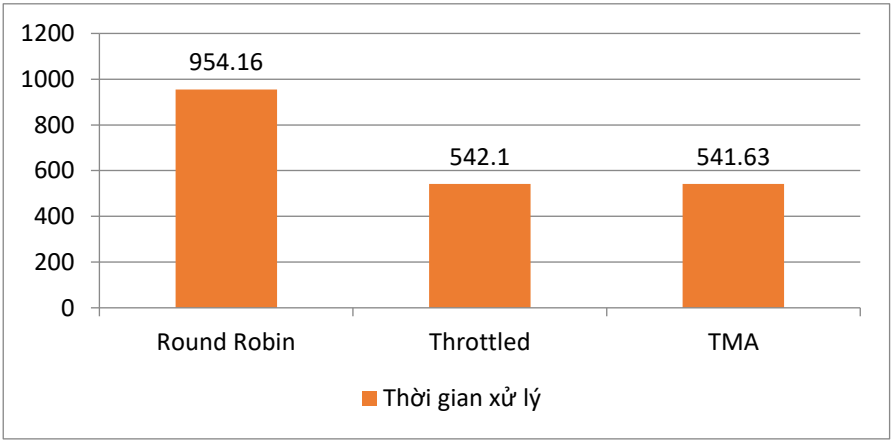
3.2.2 Kết quả mô phỏng

Sử dụng bộ công cụ mô phỏng Cloud Analyst để thực hiện mô phỏng và đánh giá thuật toán đề xuất TMA so với 2 thuật toán Round-Robin và Throttled.

Mô phỏng với số lượng 50 VM

Bảng 3.5: Kết quả mô phỏng trường hợp 2 (50VM)

Thuật toán	Thời gian xử trung bình (ms)
Round Robin	954.16
Throttled	542.10
TMA	541.63



Hình 3.3. Kết quả mô phỏng trường hợp 50 VM.

Hình 3.3 cho thấy, thời gian xử lý của Data Center của thuật toán TMA đã giảm so với thuật toán Throttled khi số lượng VM tăng lên.

3.2.3 Đánh giá:

Thông qua kết quả thực nghiệm mô phỏng thì thuật toán TMA có khả năng cân bằng tải tốt hơn thuật toán Throttled và Round Robin. Các kết quả thu được từ thuật toán TMA đã đáp ứng các mục tiêu này, chẳng hạn như giới hạn số lượng yêu cầu được xếp hàng để phân phối, cải thiện thời gian xử lý và thời gian đáp ứng của đám mây so với hai

thuật toán cũ. Với thuật toán TMA, hiệu suất của điện toán đám mây được cải thiện so với hai thuật toán Round Robin và Throttled.

3.3. Thuật toán MMSIA:

Thuật toán MMSIA cải tiến thuật toán lập lịch Max-Min [69].

3.3.1 Giới thiệu thuật toán Max - Min

Thuật toán Max-Min [69] chọn yêu cầu với thời gian hoàn thành dự kiến tối đa và gán yêu cầu đó cho máy ảo với thời gian thực hiện tổng thể tối thiểu.

3.3.2. Đề xuất thuật toán MMSIA

Mục tiêu thuật toán:

- Giảm thời gian xử lý tất cả các yêu cầu vào.
- Tăng độ xử lý yêu cầu của các máy ảo mà không làm mất cân bằng tải.

Ưu điểm thuật toán MMSIA:

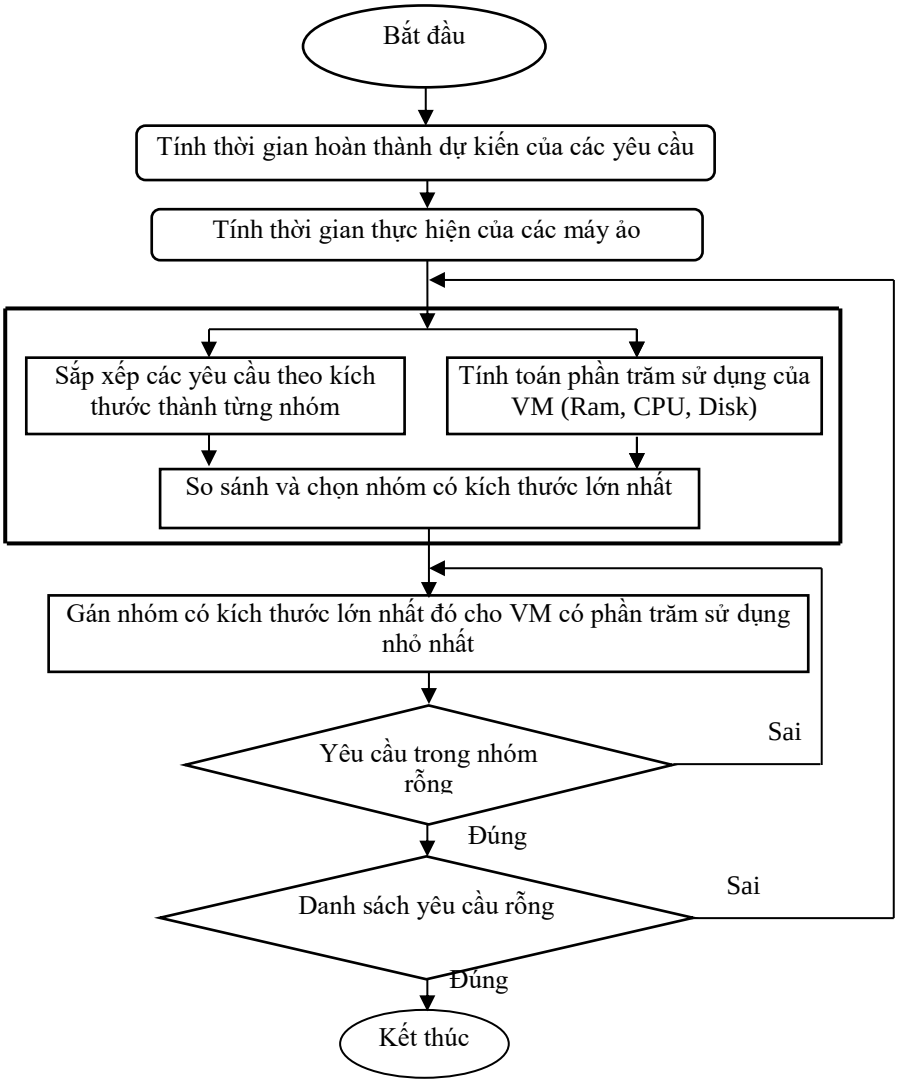
So với thuật toán lập lịch Max-Min [69] thì MMSIA tập trung vào tính kích thước của các yêu cầu đầu vào. Đồng thời cũng tính phần trăm sử dụng của máy ảo tại thời điểm đó dựa trên các thông số Ram, CPU và ổ đĩa. MMSIA đã bỏ qua giai đoạn tính thời gian hoàn thành xử lý một yêu cầu của máy ảo, nên đã giảm thời gian xử lý không cần thiết. Mặt khác, khi tính kích thước các yêu cầu đầu vào đồng thời sẽ phân nhóm thành các kích thước khác nhau, đưa những nhóm có kích thước lớn nhất cho các máy ảo có phần trăm sử dụng thấp nhất để làm giảm thời gian xử lý.

Mô tả:

MMSIA tính kích thước các yêu cầu và tính phần trăm sử dụng của VM theo những nhóm khác nhau từ đó thực hiện 2 hàm sau:

- Hàm so sánh và gán nhóm yêu cầu có kích thước file lớn nhất cho VM có phần trăm dung lượng sử dụng ít nhất.
- Sau khi gán xong sẽ thực hiện xử lý yêu cầu để đưa ra kết quả và tính toán lại phần trăm sử dụng hiện tại của VM.

Thuật toán MMSIA lặp lại đến khi các bảng yêu cầu trống. Các yêu cầu sẽ được xử lý nhanh hơn, giảm mất cân bằng tải cho điện toán đám mây.



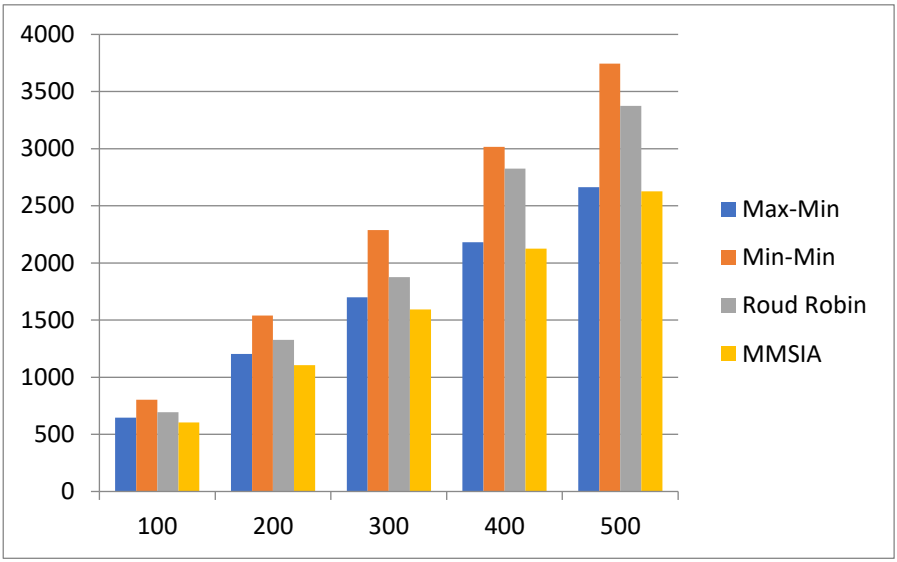
Hình 3.7. Sơ đồ thuật toán MMSIA

Thuật toán MMSIA hoạt động trên cơ sở sắp xếp các yêu cầu đầu vào thành nhiều nhóm khác nhau, sau đó sẽ gán cho các VM có phần trăm sử dụng nhỏ nhất (phần trăm sử dụng tính trên CPU, Ram, và Disk) theo cơ chế (Max-Min). Thuật toán này đã cải thiện thời gian xử lý, hạn chế mất cân bằng tải giữa các tài nguyên.

Độ phức tạp tính toán: O(n(n+m)) = O(n^2+nm).

3.3.3. Kết quả mô phỏng

Phần này trình bày về cài đặt, mô phỏng thuật toán đề xuất và từ kết quả thực nghiệm mô phỏng đánh giá phương pháp đề xuất này.



Hình 3.11. Biểu đồ so sánh thời gian xử lý lần 4

Kết quả Hình 3.11 cho thấy thời gian xử lý của các VM trên thuật toán MMSIA đã được cải thiện so với các thuật toán Max-Min [69].

Thực nghiệm này mô phỏng nhóm các VM, chưa tính tới việc mở rộng tập các máy ảo để giảm tải trong trường hợp cần thiết, vì giả định là nhóm các máy ảo này xử lý tối đa bao nhiêu yêu cầu (request), nếu

vượt quá mới mở rộng và việc mô phỏng này thực hiện ở những mô hình nhỏ và số lượng yêu cầu ít. Với việc gom nhóm yêu cầu theo kích thước file giúp VM xử lý nhanh hơn, đồng thời làm cho hệ thống phân loại những yêu cầu, từ đó đưa vào những VM có phần trăm xử lý thấp nhất để xử lý. Thực nghiệm cho thấy thuật toán đề xuất đã giảm thiểu thời gian xử lý các yêu cầu. Các thông số cũng như kịch bản đưa ra dựa vào quá trình request của các browser trên môi trường đám mây. Từ đó, ghi nhận các thông số về kích thước file của các yêu cầu và trung bình phần trăm của VM. Thực nghiệm với 5 máy ảo, số lượng từ 25 đến 500 request đã cho thấy kết quả tương đối tốt, việc phân bổ các request tới các máy ảo xử lý khá đồng đều và kết quả xử lý có sự sai lệch không quá lớn.

3.3. Kết luận Chương 3

Chương 3 đã đề xuất 02 thuật toán cân bằng tải nhằm cải tiến thời gian xử lý trên điện toán đám mây, bao gồm: TMA, MMSIA. Thuật toán TMA (được công bố trong công trình CT5). Thuật toán MMSIA (được công bố trong công trình CT6). Các thuật toán này được so sánh, đánh giá với các thuật toán liên quan Round Robin, Throttled, Max-Min, Min-Min. Thử nghiệm được tiến hành trên các kịch bản khác nhau với số lượng các yêu cầu và số lượng VM khác nhau. Kết quả mô phỏng đã chứng tỏ các thuật toán đề xuất đã cải thiện được thời gian xử lý các yêu cầu.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

I. Những kết quả chính của luận án:

1. Nghiên cứu phát triển một số thuật toán cân bằng tải nhằm cải thiện thời gian đáp ứng trên điện toán đám mây:

- **Đề xuất thuật toán LBAIRT (CT4):** cải tiến thuật toán Throttled với đóng góp chính là việc phân bổ yêu cầu đầu vào đến các máy ảo dựa trên thời gian đáp ứng nhỏ nhất và bằng cách xem xét tham số thời gian hoàn thành các yêu cầu công việc dự kiến của mỗi tài nguyên. Thuật toán đưa vào thời gian hoàn thành dự kiến của mỗi VM cho các yêu cầu trong hàng đợi. Dựa trên tham số này, thuật toán sẽ chọn VM với thời gian hoàn thành dự kiến nhỏ nhất và tỷ lệ sử dụng thấp nhất để phân bổ yêu cầu.
- **Đề xuất thuật toán RRTA (CT7):** ứng dụng thuật toán ARIMA để dự đoán ngưỡng thời gian đáp ứng chung của hệ thống và dự đoán thời gian đáp ứng của các máy ảo dựa trên tập yêu cầu tương tự trước đó nhằm đưa ra cách phân phối tài nguyên hợp lý. Thuật toán RRTA tiếp cận một cách khái quát và phát huy ý tưởng của dự báo và xử lý chuỗi thời gian, điển hình là thuật toán ARIMA. Thuật toán đề xuất có hướng tiếp cận mới trong cân bằng tải ở môi trường đám mây, đồng thời đạt được một số kết quả thực nghiệm mô phỏng khá tích cực, cho thấy hướng phát triển tốt của thuật toán

Các kết quả nghiên cứu thực nghiệm dựa trên bộ dữ liệu mô phỏng đã chứng minh hiệu quả và tính đúng đắn của 02 thuật toán đề xuất. Qua đó, giúp cho các nhà cung cấp dịch vụ điện toán đám mây nâng cao chất lượng dịch vụ cho người dùng trong thực tế

2. Nghiên cứu phát triển một số thuật toán cân bằng tải nhằm cải thiện thời gian xử lý trên điện toán đám mây:

- **Đề xuất thuật toán TMA (CT5):** Thuật toán TMA cải tiến thuật toán Throttled bằng cách chia bảng chứa thông tin máy ảo chung thành hai bảng máy ảo ở trạng thái sẵn sàng và trạng thái không sẵn sàng nhằm giảm thời gian tìm kiếm máy ảo sẵn sàng cho mỗi yêu cầu đầu vào. Điểm mới của thuật toán TMA: Việc dò tìm VM đang sẵn sàng '0' với kích thước bảng "Available Index" thay đổi linh động hơn so với thuật toán Throttled. Bộ cân bằng tải tốt ít chi phí thời gian do duy trì 2 bảng danh sách các VM "sẵn sàng" và "bận", bộ cân bằng tải chỉ việc lấy gán VM cho các yêu cầu mới đến. Điều này giúp tăng hiệu suất xử lý cho hệ thống đồng nghĩa với giảm thời gian xử lý các yêu cầu đầu vào
- **Đề xuất thuật toán MMSIA (CT6):** Thuật toán MMSIA cải tiến thuật toán lập lịch Min – Max bằng cách nhóm các yêu cầu và máy áo theo thời gian hoàn thành dự kiến và thời gian thực hiện hoàn thành tổng thể. Thuật toán MMSIA hoạt động trên cơ sở sắp xếp các yêu cầu đầu vào thành nhiều nhóm khác nhau, sau đó sẽ gán cho các VM có phần trăm sử dụng nhỏ nhất (phần trăm sử dụng tính trên CPU, Ram, và Disk) theo cơ chế (Max-Min). Thuật toán này đã cải thiện thời gian xử lý, hạn chế mất cân bằng tải giữa các tài nguyên

Thông qua thực nghiệm với nhiều kịch bản mô phỏng đã chứng minh tính hiệu quả và tính đúng đắn của 02 thuật toán đề xuất. Đây cũng là cơ sở lý luận cho các nhà phát triển dịch vụ điện toán đám mây nâng cao chất lượng dịch vụ và hiệu năng phục vụ của trung tâm dữ liệu đám mây.

Về mặt thực tiễn: kết quả của luận án đã được thực nghiệm trên các bộ dữ liệu mô phỏng trong các kịch bản khác nhau, kết quả thực nghiệm của phương pháp đề xuất được đánh giá là có hiệu quả hơn các phương pháp đã công bố trong đa số trường hợp, đồng thời là cơ sở khoa học để chế tạo ra các bộ cân bằng tải ứng dụng vào các trung tâm dữ thực tế. Đây là cơ sở cho thấy, có thể áp dụng kết quả nghiên cứu của đề tài trong việc triển khai các hệ thống cân bằng tải nhằm đối phó với sự bùng

nổ trao đổi dữ liệu đám mây hiện nay ở đa dạng các lĩnh vực. Các thuật toán đề xuất được mô phỏng để đánh giá tính hiệu quả so với các thuật toán gốc đã được công bố trước đó.

Phạm vi ứng dụng của các thuật toán đề xuất: Các thuật toán đề xuất được định hướng cho các bộ cân bằng tải (Load Balaner) trong các trung tâm dữ liệu của các nhà cung cấp dịch vụ đám mây, do tính hiệu quả của nó đã được chứng minh thông qua cơ sở lý luận cũng như mô hình thực nghiệm trong luận án. Áp dụng các thuật toán để cải thiện thời gian đáp ứng, thời gian xử lý các yêu cầu từ phía người dùng truy cập đến trung tâm điện toán đám mây.

II. Hướng phát triển của luận án:

1. Luận án có thể được phát triển theo hướng xây dựng mô hình cơ sở dựa vào công nghệ trí tuệ nhân tạo (AI) để nhận diện theo đặc tính riêng lẻ của các yêu cầu đầu vào nhằm đánh giá hiệu năng của hệ thống điện toán đám mây. Từ đó có được mô hình lý thuyết đầy đủ hỗ trợ hoạt động nghiên cứu và triển khai hệ thống điện toán đám mây trong thực tế.
2. Ngoài ra, luận án có thể được phát triển theo hướng cải thiện đồng thời hai tham số: thời gian đáp ứng và thời gian xử lý trên môi trường điện toán đám mây. Đây cũng là một cách tiếp cận rất thiết thực trong bối cảnh bùng nổ trao đổi dữ liệu trên môi trường điện toán đám mây hiện nay.
3. Nghiên cứu cân bằng tải trên mạng lưới vạn vật kết nối (IoT) cũng có thể là một hướng phát triển của luận án khi mà cuộc cách mạng công nghệ 4.0 đang làm thay đổi mọi lĩnh vực trong đời sống hàng ngày, hàng giờ.

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

[CT1] Tran Cong Hung, Nguyen Khoi, Nguyen Xuan Phi (2013), *"Survey traffic matrix for optimizing network performance"*, Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), October Edition, 2013 Volume 3, Issue 10, ISSN 1925-2676, pages 29-35, October 2013, Canada.

[CT2] Nguyễn Xuân Phi, Trần Công Hùng (2015), *"Giải Thuật Phòng Tránh Tình Trạng Quá Tải Trong Điện Toán Đám Mây"*, Proceedings of The 2015 National Conference on Electronics, Communications and Information Technology ECIT 2015, pages 66-70, ISBN: 978-604-67-0635-9, December, 10-11, 2015, Ho Chi Minh City, Viet Nam.

[CT3] Nguyen Xuan Phi, Tran Cong Hung (2016), *"Study the Effect of Parameters to Load Balancing in Cloud Computing",* International Journal of Computer Networks & Communications (IJCNC) Vol.8, No.3, May 2016.ISSN:0974-9322 [Online]; 0975-2293 [Print], DOI: 10.5121/ijcnc.2016.8303, pp.33-45, SCOPUS, the Australian Research Council (ARC) Journal Ranking,

[CT4] Nguyen Xuan Phi, Tran Cong Hung (2017), *"Load Balancing Algorithm to Improve Response time on Cloud Computing"*, International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 7, No. 6, December 2017, DOI: 10.5121/ijccsa.2017.7601, pp.1-12,

[CT5] Nguyen Xuan Phi, Cao Trung Tin, Luu Nguyen Ky Thu, Tran Cong Hung (2018), *"Proposed Load Balancing Algorithm to Reduce Response time and Processing time on Cloud Computing*", International Journal of Computer Networks &

Communications (IJCNC) Vol.10, No.3, May 2018, DOI: 10.5121/ijcnc.2018.10307, pp.87-98, ISSN 0974-9322 (Online), 0975- 2293 (Print), SCOPUS.

[CT6] Tran Cong Hung, Phan Thanh Hy, Le Ngoc Hieu, Nguyen Xuan Phi, *"MMSIA: Improved Max-Min Scheduling Algorithm for Load Balancing on Cloud Computing*", ICMLSC 2019 (Proceedings of The 3rd International Conference on Machine Learning and Soft Computing), pp.60-64 ACM New York, NY, USA @2019 (ISBN: 978-1-4503-6612-0), indexed by Ei Compendex, SCOPUS, Da Lat, Vietnam, January 25-28, 2019.

[CT7] Nguyễn Xuân Phi, Lê Ngọc Hiếu, Trần Công Hùng (2019), *"Thuật toán cân bằng tải nhằm giảm thời gian đáp ứng dựa vào ngưỡng thời gian trên điện toán đám mây"*, Tạp chí Khoa học và Công nghệ về Thông tin và Truyền thông (JSTIC-Journal of Science & Technology on Information and Communications, ISSN:2525-2224,pp.43-48, 04(CS.01)2018, PTIT, 01/2019