HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Hoàng Tấn

ĐỀ XUẤT THUẬT TOÁN DỰ BÁO THỜI GIAN DI CHUYỂN TÁC VỤ NHẰM NÂNG CAO HIỆU NĂNG CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY

LUẬN VĂN THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

TP. HCM – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Hoàng Tấn

ĐỀ XUẤT THUẬT TOÁN DỰ BÁO THỜI GIAN DI CHUYỂN TÁC VỤ NHẰM NÂNG CAO HIỆU NĂNG CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

PGS.TS. TRẦN CÔNG HÙNG

TP. HCM – NĂM 2022

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: *"Đề xuất thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi người dùng cloud"* là công trình nghiên cứu của chính tôi.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Nguyễn Hoàng Tấn

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện luận văn, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới: Ban Giám Đốc, Phòng đào tạo sau đại học và quý Thầy Cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành luận văn.

Tôi xin chân thành cảm ơn Thầy PGS.TS Trần Công Hùng, người thầy kính yêu đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành luận văn.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn. Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Nguyễn Hoàng Tấn

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

DANH MỤC CHỮ VIẾT TẮT

CC	Cloud Computing
ML	Machine Learning
LB	Load Balancing
Cloud	Cloud computing environment
AI	Artificial Intelligence
ACO	Ant Colony Optimization
GA	Genetic Algorithm
FCFS	First Come First Serve

MỤC LỤC

PHẦN MỞ ĐẦU

1. Tính cấp thiết của đề tài

Cuộc cách mạng công nghiệp lần thứ tư [1] được cho là đã bắt đầu từ vài năm gần đây, tập trung chủ yếu vào sản xuất thông minh dựa trên các thành tựu đột phá trong công nghệ thông tin, công nghệ sinh học và công nghệ nano. Đây là một cơ hội và cũng là một thách thức đối với Việt Nam chúng ta. Cơ hội để chúng ta có thể đi tắt đón đầu, rút ngắn khoảng cách với các nước phát triển. Song là một thách thức lớn vì tiềm lực ta có nhưng chưa có kinh nghiệm khai thác và phát huy hiệu quả tối đa các nguồn lực này.

Có thể thấy thời gian gần đây việc ứng dụng Công nghệ thông tin phục vụ phát triển Chính quyền điện tử [2] hướng đến Chính quyền số đang được Chính phủ và nhiều địa phương quan tâm và ưu tiên phát triển. Từ đó, nhu cầu về triển khai ứng dụng, lưu trữ dữ liệu lớn và xử lý, khai thác thông tin ngày càng cao. Vì vậy, để đáp ứng được các nhu cầu nói trên thì có một công nghệ đã và đang được triển trong nhiều năm qua và vẫn sẽ là xu thế phát triển trong tương lai, đó là Điện toán đám mây (Cloud computing). Điện toán đám mây là một công nghệ đầy hứa hẹn [3] vì:

- Tính sẵn sàng cao: Hạ tầng ảo hoá từ nền tảng công nghệ hàng đầu thế giới của VMware, Cisco, Netapp, IBM… sẽ cho bạn một Cloud Server mạnh mẽ, ổn định, uptime lên đến 99.99%.

- Tính linh hoạt: Cloud Server cho phép bạn chủ động lựa chọn cấu hình và tăng giảm tài nguyên theo nhu cầu sử dụng thực tế. Việc này được thực hiện nhanh chóng trong vài phút.

- Tính an toàn dữ liệu: Hệ thống lưu trữ phân tán và cơ chế sao lưu hàng ngày đảm bảo dữ liệu luôn luôn sẵn sàng và liên tục.

- Tính tiết kiệm: việc hao hụt và dung lượng lưu trữ dự phòng được tổ chức tập trung nên không cần phải tốn nhiều, sẽ tiết kiệm được chi phí.

- Tiết kiệm thời gian: Với đám mây, bạn có thể mở rộng sang các khu vực địa lý mới và triển khai trên toàn cầu trong vài phút. Ví dụ: AWS có cơ sở hạ tầng trên toàn thế giới. Vì vậy, bạn có thể triển khai ứng dụng của mình ở nhiều địa điểm thực tế chỉ bằng vài cú nhấp chuột. Đặt các ứng dụng gần hơn với người dùng cuối giúp giảm độ trễ và cải thiện trải nghiệm của họ.

- Quản lý dễ dàng: Giao diện quản lý Cloud Server rất thân thiện, dễ sử dụng. Có thể quản lý thông qua cổng website, các giao thức API hay ngay cả các ứng dụng di động mọi lúc, mọi nơi.

- Hệ điều hành mẫu đa dạng: hệ thống có khả năng tương thích với nhiều hệ điều hành từ Linux như CentOS, Redhat, Fedora, Ubuntu, Debian, Opensuse đến Windows, Free BSD...

Trên quan điểm chất lượng dịch vụ [4] trên điện toán đám mây, việc quản lý tài nguyên trở thành một công việc phức tạp từ góc nhìn kinh doanh của nhà cung cấp dịch vụ đám mây. Do đó, ta phải khắc phục vấn đề thiếu thốn tài nguyên, giảm độ trễ trên đám mây và khả năng cải thiện hiệu suất mạng. Điều này được bộ cân bằng tải xử lý và điều phối. Vì vậy, cần phải có thuật toán dự báo thời gian di chuyển tác vụ nhằm nâng cao hiệu quả cân bằng tải trên điện toán đám mây. Cụ thể, đề tài như sau:

Tên tiếng Việt là: “Đề xuất thuật toán dự báo thời gian di chuyển tác vụ nhằm nâng cao hiệu năng cân bằng tải trên điện toán đám mây”.

Tên tiếng Anh là: “Proposed Task Migration Time Algorithm to Improve Load Balancing in Cloud Computing” .

2. Tổng quan về vấn đề nghiên cứu

Cân bằng tải là kỹ thuật phân phối khối lượng công việc đồng đều giữa hai hoặc nhiều máy tính, kết nối mạng, CPU, ổ cứng, hoặc các nguồn lực phân tán to lớn trên mạng. Với mục đích chính là tận dụng có hiệu quả các nguồn lực, tối đa hóa thông lượng, cải thiện thời gian đáp ứng và thời gian xử lý dữ liệu. Ngoài ra, tránh tình trạng quá tải một số nút tính toán trong khi những nút khác được nạp tải nhẹ khi

có nhiều yêu cầu xử lý cần được đáp ứng. Kỹ thuật cân bằng tải hiện nay chủ yếu tập trung vào hai kỹ thuật là cân bằng tải tĩnh và cân bằng tải động.

Kỹ thuật cân bằng tải tĩnh không thu thập thông tin trạng thái hiện tại hệ thống. Những yếu tố được đo lường trước khi gán công việc cho một nút tính toán như thời gian đến, qui mô nguồn tài nguyên, thời gian thực thi và giao tiếp các tiến trình.

Kỹ thuật cân bằng tải động trong tự nhiên không xem xét trạng thái trước đó hoặc hành vi của hệ thống, nó chỉ phụ thuộc vào hành vi hiện tại của hệ thống.

3. Mục đích nghiên cứu

Mục tiêu chính: Đề xuất ra một thuật toán dự báo thời gian di chuyển tác vụ (Task Migration Time) nhằm nâng cao hiệu quả cân bằng tải trên điện toán đám mây.

Từ mục tiêu chính trên, luận văn sẽ dự kiến các kết quả đạt được như sau:

- Tìm hiểu tổng quan về điện toán đám mây.

- Tìm hiểu về các thuật toán trên điện toán đám mây.

- Tìm hiểu về thời gian di chuyển tác vụ (Migration Time).

- Tìm hiểu thuật toán dự báo thời gian di chuyển tác vụ (Migration Time) trong việc cân bằng tải trên điện toán đám mây.

- Đề xuất thuật toán nhằm dự báo thời gian di chuyển một task bất kỳ trên cloud. Có thể là chuyển từ VM này sang VM khác hoặc từ data-center này sang data-center khác, ứng với các loại task khác nhau thì thời gian di chuyển tương ứng dự báo là bao nhiêu. Từ đó, phân bổ task từ resource đang full sang resource ít full hơn… Nghiên cứu sâu về mô hình cloud, những chỗ có khả năng xảy ra overload,… overload trên cloud là gì? Thường xảy ra ở đâu, mức nào nhiều nhất? Khi xảy ra thì chuyển task như thế nào?

- Trên cơ sở lý thuyết đã nghiên cứu, luận văn đề xuất thuật toán dự báo thời gian di chuyển tác vụ (Migration Time) nhằm nâng cao hiệu quả cân bằng tải trên điện toán đám mây. Mô phỏng và thực nghiệm thuật toán đã đề xuất.

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu

- Đối tượng nghiên cứu chính là thời gian di chuyển tác vụ (Migration Time) trong cân bằng tải trên điện toán đám mây.

- Nghiên cứu các thuật toán dự báo thời gian di chuyển tác vụ (Migration Time) trong cân bằng tải trên điện toán đám mây.

- Phạm vi nghiên cứu

Phạm vi nghiên cứu trong Cloud:

- Xây dựng mô hình mô phỏng đám mây ở mức độ nhỏ: khoảng từ 10~15 máy ảo.

- Độ phức tạp trên mỗi máy ảo chỉ ở mức độ thấp: khoảng 1 – 4 ứng dụng trên các máy ảo đó.

5. Phương pháp nghiên cứu

Phương pháp luận: Dựa trên cơ sở là các lý thuyết về điện toán đám mây, các thuật toán cân bằng tải trên cloud.

Phương pháp đánh giá dựa trên cơ sở toán học: Trên cơ sở các lý thuyết về điện toán đám mây, khả năng xảy ra tắc nghẽn trên đám mây. Đề xuất ra thuật toán để nâng cao hiệu quả cân bằng tải trên đám mây dựa trên các thuật toán đã nghiên cứu. Chứng minh thuật toán và đánh giá hiệu quả của thuật toán.

Phương pháp đánh giá bằng mô phỏng thực nghiệm: Xây dựng mô hình mô phỏng và thực nghiệm thuật toán đã đề xuất.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ HỆ THỐNG CÂN BẰNG TẢI CỦA ĐIỆN TOÁN ĐÁM MÂY

1.1. Tổng quan về điện toán đám mây

Lịch sử của điện toán đám mây bắt đầu từ năm 1983, khi Sun Microsystems đề xuất rằng "web là máy tính". Trong tháng 3 năm 2006, Amazon giới thiệu dịch vụ đám mây điện toán đàn hồi. Vào tháng 8 năm 2006, Eric Schmidt, Giám đốc điều hành của Google, lần đầu tiên đề xuất khái niệm "Điện toán đám mây" tại hội nghị công cụ tìm kiếm. Năm 2009, Nair M K. và Gopalakrishnan V. đã phát triển một khung hệ thống, sử dụng các dịch vụ web như SaaS và môi trường web để hiện thực hóa PaaS, thúc đẩy hiệu quả sự phát triển của điện toán đám mây. Takahiro Miyamoto và nhóm của ông đã nhận ra chức năng mạng của điện toán đám mây vào năm 2009, đặt nền tảng vững chắc cho sự phát triển của điện toán đám mây. Kể từ đó, điện toán đám mây đã bước vào thời kỳ phát triển nhanh chóng. Điện toán đám mây được phát triển từ điện toán song song: điện toán phân tán và điện toán lưới, như trong hình 1.1, nó là một mô hình điện toán kinh doanh mới. Hiện tại, vẫn chưa có định nghĩa thống nhất về điện toán đám mây. Wikipedia định nghĩa điện toán đám mây là một phương thức tính toán mới dựa trên Internet, cung cấp tính toán theo yêu cầu cho người dùng cá nhân và doanh nghiệp thông qua các dịch vụ không đồng nhất và tự trị trên Internet. Eric Schmidt, Giám đốc điều hành của Google, cho rằng điện toán đám mây về cơ bản là một mô hình cung cấp dịch vụ, ảo hóa tài nguyên máy tính, tài nguyên lưu trữ và tài nguyên mạng bao gồm một số lượng lớn máy chủ. Chúng tạo thành một nhóm tài nguyên ảo bao gồm tài nguyên điện toán, lưu trữ và mạng, quản lý và lên lịch thông qua một nền tảng điện toán đám mây thống nhất.

Điện toán đám mây (cloud computing) hay còn gọi là điện toán máy chủ ảo, nơi các tính toán được “định hướng dịch vụ” và phát triển dựa vào Internet. Cụ thể hơn, trong mô hình điện toán đám mây, tất cả các tài nguyên, thông tin cùng với software đều được chia sẻ và cung cấp cho các máy tính, thiết bị, người dùng dưới dạng dịch vụ trên nền tảng một hạ tầng mạng công cộng (thường là mạng Internet).

Các user sử dụng dịch vụ như cơ sở dữ liệu, website, lưu trữ,… trong mô hình cloud computing không cần quan tâm đến vị trí địa lý cũng như các thông tin khác của hệ thống mạng đám mây - "điện toán đám mây trong suốt đối với người dùng". Người dùng cuối truy cập và sử dụng các ứng dụng đám mây thông qua các ứng dụng như trình duyệt web, các ứng dụng mobile hoặc máy tính cá nhân thông thường. Hiệu năng sử dụng phía người dùng cuối được cải thiện khi các phần mềm chuyên dụng, các cơ sở dữ liệu được lưu trữ và cài đặt trên hệ thống máy chủ ảo trong môi trường điện toán đám mây trên nền của "data center". "Data center" là thuật ngữ chỉ khu vực chứa server và các thiết bị lưu trữ, bao gồm nguồn điện và các thiết bị khác như rack, cables… có khả năng sẵn sàng và độ ổn định cao. Ngoài ra còn bao gồm các tiêu chí khác như: tính module hóa cao, khả năng mở rộng dễ dàng, nguồn và làm mát, hỗ trợ hợp nhất server và lưu trữ mật độ cao.

Có 3 mô hình triển khai điện toán đám mây chính là public (công cộng), private (riêng) và hybrid ("lai" giữa đám mây công cộng và riêng). Đám mây công cộng là mô hình đám mây mà trên đó, các nhà cung cấp đám mây cung cấp các dịch vụ như tài nguyên, platform hay các ứng dụng lưu trữ trên đám mây và public ra bên ngoài. Các dịch vụ trên public cloud có thể miễn phí hoặc có phí. Đám mây riêng thì các dịch vụ được cung cấp nội bộ và thường là các dịch vụ kinh doanh. Mục đích nhắm đến cung cấp dịch vụ cho một nhóm người và đứng đằng sau firewall. Đám mây "lai" là môi trường đám mây mà kết hợp cung cấp các dịch vụ công cộng và riêng. Ngoài ra còn có "community cloud" là đám mây giữa các nhà cung cấp dịch vụ đám mây. Về mô hình cung cấp dịch vụ có 3 loại chính là IaaS – cung cấp hạ tầng như một service, PaaS – cung cấp Platform như một service và SaaS – cung cấp software như một service.

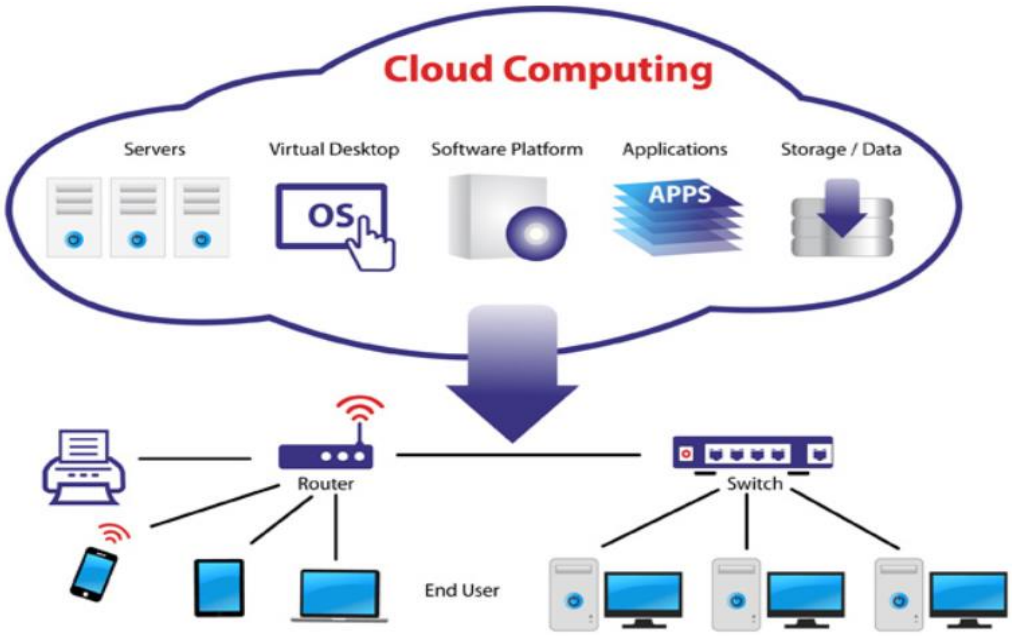
Theo các loại hình dịch vụ, điện toán đám mây có thể được chia thành ba loại sau:

- IaaS, hoặc cơ sở hạ tầng như một dịch vụ, cho phép người dùng truy cập trực tiếp vào tài nguyên lưu trữ, tài nguyên mạng và tài nguyên máy tính bên

dưới. IaaS sử dụng công nghệ ảo hóa để ảo hóa và đóng gói tài nguyên máy tính, tài nguyên lưu trữ và tài nguyên mạng của máy chủ, đồng thời cung cấp các tài nguyên này dưới dạng API. Khi cần sử dụng các tài nguyên này, người dùng không cần mua các thiết bị phần cứng như máy chủ mà chỉ cần mua các tài nguyên này từ các nhà sản xuất cung cấp dịch vụ IaaS. Nền tảng điện toán đám mây IaaS cung cấp quản lý và lập kế hoạch của các tài nguyên này. Ví dụ điển hình bao gồm Đám mây tính toán đàn hồi (EC2) và Dịch vụ lưu trữ đơn giản (S3) của Amazon.

- PaaS, hoặc nền tảng làm nền tảng dịch vụ, cung cấp nền tảng và môi trường cho hoạt động kinh doanh phần mềm. PaaS cung cấp giải pháp cho các công ty không thể hoặc không muốn xây dựng môi trường vận hành phần mềm. PaaS cung cấp môi trường hoạt động và hệ điều hành cho các doanh nghiệp khác nhau. "Máy chủ ảo" thuộc danh mục dịch vụ PaaS. Chỉ có mã nguồn cần được tải lên địa chỉ của "máy chủ ảo". "Máy chủ ảo" sẽ chạy mã và tạo một trang web theo mã. Ví dụ điển hình bao gồm GoogleAppEngine của Google và MicrosoftWindowsAzure của Microsoft.

Theo các phương pháp triển khai khác nhau, điện toán đám mây có thể được chia thành đám mây riêng, đám mây công cộng và đám mây lai. Đám mây riêng là cơ sở hạ tầng đám mây do một tổ chức sở hữu hoặc thuê, có thể được đặt tại địa phương hoặc ở một nơi khác. Đám mây công cộng là cơ sở hạ tầng đám mây thuộc sở hữu của một tổ chức điều hành cung cấp dịch vụ điện toán đám mây, tổ chức này bán các dịch vụ điện toán đám mây cho công chúng hoặc một số lượng lớn các nhóm doanh nghiệp vừa và nhỏ. Đám mây kết hợp bao gồm đám mây riêng và đám mây công cộng và mỗi đám mây vẫn là một thực thể độc lập. Song, kết hợp chúng với công nghệ tiêu chuẩn hoặc độc quyền để làm cho dữ liệu và ứng dụng di động.



Hình 1.1: Mô hình điện toán đám mây [1]

Điện toán đám mây là một xu hướng công nghệ nổi bật trên thế giới trong những năm gần đây và đã có những bước phát triển nhảy vọt cả về chất lượng, quy mô cung cấp và loại hình dịch vụ. Tiêu biểu là một loạt các nhà cung cấp lớn và nổi tiếng như Google, Amazon, Microsoft,...

Điện toán đám mây là mô hình điện toán mà mọi giải pháp liên quan đến công nghệ thông tin đều được cung cấp dưới dạng các dịch vụ qua mạng Internet. Từ đó, giải phóng người sử dụng khỏi việc phải đầu tư nhân lực, công nghệ và hạ tầng để triển khai hệ thống. Hơn nữa, điện toán đám mây giúp tối giản chi phí và thời gian triển khai, tạo điều kiện cho người sử dụng nền tảng điện toán đám mây tập trung được tối đa nguồn lực vào công việc chuyên môn. Lợi ích của điện toán đám mây mang lại không chỉ gói gọn trong phạm vi người sử dụng nền tảng điện toán đám mây mà còn từ phía các nhà cung cấp dịch vụ điện toán.

Điện toán đám mây (Cloud Computing) [1], [2] là xu hướng phát triển mạnh nhất hiện nay. Nó kế thừa các mạng lưới trước đây và các khái niệm máy tính phân tán để tích hợp các tài nguyên máy tính, lưu trữ, nền tảng và các dịch vụ khác theo nhu cầu một cách thuận tiện và nhanh chóng. Đồng thời, điện toán đám mây còn cho

phép kết thúc sử dụng dịch vụ, giải phóng tài nguyên dễ dàng và giảm thiểu các giao tiếp với nhà cung cấp. Theo đó, mô hình chính là cho phép sử dụng dịch vụ theo yêu cầu (on-demand service); cung cấp khả năng truy cập dịch vụ qua mạng rộng rãi từ máy tính để bàn, máy tính xách tay tới thiết bị di động (broad network access); với tài nguyên tính toán động, phục vụ nhiều người (resource pooling for multi-tenanci), năng lực tính toán phần mềm dẻo và đáp ứng nhanh với nhu cầu từ thấp đến cao (rapidelasticity).

Điện toán đám mây được dựa trên công nghệ ảo hóa [3], thông qua các dịch vụ mạng để cung cấp cho người dùng với các nguồn lực cơ bản, nền tảng ứng dụng, phần mềm và các dịch vụ khác. Trong trường hợp IaaS (cơ sở hạ tầng như một dịch vụ), các nhà phát triển cung cấp một môi trường ứng dụng phần mềm hoàn chỉnh bằng cách tập hợp các phần cứng, phần mềm và các thiết bị có liên quan lại với nhau để đáp ứng thỏa thuận chất lượng dịch vụ với người dùng. Công nghệ máy ảo (Virtual Machine) thường được sử dụng trong các trung tâm dữ liệu, máy tính cụm và các ứng dụng khác. Công nghệ này cho phép nhiều hệ điều hành có thể chạy trên cùng một máy tính và cung cấp các dịch vụ độc lập đáng tin cậy, cải tiến rất nhiều khả năng sử dụng lại các tài nguyên vật lý.

Điện toán đám mây [4] là một hướng nghiên cứu rộng, sẽ đem lại giá trị lớn về các chi phí cho các doanh nghiệp trên toàn thế giới. Điện toán đám mây sẽ giúp giải quyết được việc lưu trữ dữ liệu trên hệ thống một cách nhanh, gọn, nhẹ. Cung cấp các dịch vụ về cơ sở hạ tầng, nền tảng phần mềm và các dịch vụ theo yêu cầu người dùng thông qua Internet.

Điện toán đám mây (cloud computing) hay còn gọi là điện toán máy chủ ảo nơi các tính toán được "định hướng dịch vụ" và phát triển dựa vào Internet. Cụ thể hơn, trong mô hình điện toán đám mây, tất cả các tài nguyên, thông tin, và software đều được chia sẻ và cung cấp cho các máy tính, thiết bị, người dùng dưới dạng dịch vụ trên nền tảng một hạ tầng mạng công cộng (thường là mạng Internet). Các user sử dụng dịch vụ như cơ sở dữ liệu, website, lưu trữ,… trong mô hình cloud computing

không cần quan tâm đến vị trí địa lý cũng như các thông tin khác của hệ thống mạng đám mây - "điện toán đám mây trong suốt đối với người dùng". Người dùng cuối truy cập và sử dụng các ứng dụng đám mây thông qua các ứng dụng như trình duyệt web, các ứng dụng mobile hoặc máy tính cá nhân thông thường. Hiệu năng sử dụng phía người dùng cuối được cải thiện khi các phần mềm chuyên dụng, các cơ sở dữ liệu được lưu trữ và cài đặt trên hệ thống máy chủ ảo trong môi trường điện toán đám mây trên nền của "data center". "Data center" là thuật ngữ chỉ khu vực chứa server và các thiết bị lưu trữ, bao gồm nguồn điện và các thiết bị khác như rack, cables… có khả năng sẵn sàng và độ ổn định cao. Ngoài ra còn bao gồm các tiêu chí khác như: tính module hóa cao, khả năng mở rộng dễ dàng, nguồn và làm mát, hỗ trợ hợp nhất server và lưu trữ mật độ cao. Có 3 mô hình triển khai điện toán đám mây chính là public (công cộng), private (riêng), và hybrid ("lai" giữa đám mây công cộng và riêng). Đám mây công cộng là mô hình đám mây mà trên đó, các nhà cung cấp đám mây cung cấp các dịch vụ như tài nguyên, platform hay các ứng dụng lưu trữ trên đám mây và public ra bên ngoài. Các dịch vụ trên public cloud có thể miễn phí hoặc tính phí. Đám mây riêng thì các dịch vụ được cung cấp nội bộ và thường là các dịch vụ kinh doanh. Mục đích của đám mây riêng nhắm đến là cung cấp dịch vụ cho một nhóm người và đứng đằng sau firewall. Đám mây "lai" là môi trường đám mây mà kết hợp cung cấp các dịch vụ công cộng và riêng. Ngoài ra còn có "community cloud" là đám mây giữa các nhà cung cấp dịch vụ đám mây. Về mô hình cung cấp dịch vụ có 3 loại chính là IaaS – cung cấp hạ tầng như một service, PaaS – cung cấp Platform như một service, và SaaS – cung cấp software như một service.

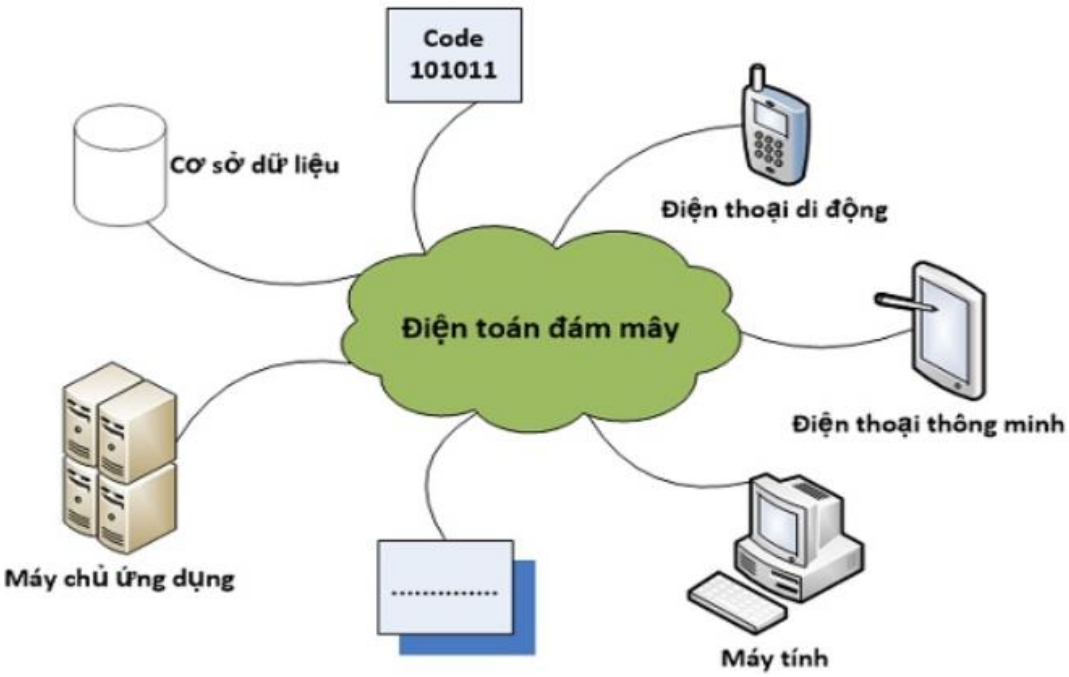
Điện toán đám mây [5] là một mô hình dịch vụ công nghệ thông tin kế thừa các mạng lưới trước đây trên thế giới. Mục tiêu hướng đến là giúp người dùng truy cập tài nguyên dữ liệu, lưu trữ đến hệ thống quản lý và xử lý dữ liệu phức tạp của các hệ thống như Google, Facebook… Trên thực tế, người dùng chỉ truy cập vào thiết bị đầu cuối để truy xuất vào các tài nguyên trên điện toán. Còn ở bên trong hệ thống điện toán sẽ lập lịch xử lý các yêu cầu trên bao gồm xử lý thời gian chờ và thời gian xử lý tín hiệu đến thời gian hoàn thành nhiệm vụ.

Điện toán đám mây [6] đang chuyển đổi ngành công nghệ thông tin, thay đổi cách thức sử dụng và cung cấp phần cứng cũng như phần mềm. Làm cho việc sử dụng các tài nguyên máy tính theo yêu cầu như băng thông, lưu trữ hoặc các ứng dụng phần mềm, điện toán có sẵn trở nên dễ dàng và nhanh chóng hơn. Nó che giấu sự phức tạp của cơ sở hạ tầng cơ bản, cho phép người dùng cuối tập trung vào sản phẩm của chính họ mà không cần nhiều khoản đầu tư vào phần cứng. Theo hợp đồng dịch vụ đã được thiết lập giữa nhà cung cấp điện toán và khách hàng, các ràng buộc về chất lượng dịch vụ (QoS) nhất định được xác định thông qua các thỏa thuận theo mức dịch vụ (SLA). Tuân thủ với các SLA này, nhà cung cấp đảm bảo cung cấp một chất lượng nhất định cho dịch vụ đã thỏa thuận. Việc sử dụng các máy ảo cho phép sử dụng tốt hơn các tài nguyên phần cứng hiện tại trong khi vẫn duy trì QoS yêu cầu. Để tránh sự xuống cấp của hiệu suất, máy ảo được di chuyển từ quá tải đến các máy không sử dụng được. Vì vậy, các thuật toán phát hiện là cần thiết để chủ động phân loại quá tải và không quá tải. Các thuật toán chủ động xác định một kế hoạch tối ưu cho việc di chuyển và phân bổ các máy ảo trong thời gian chạy.

Là một mô hình tính toán mới, [7] được phát triển sau khi công nghệ phân phối máy tính, điện toán lưới, lưu trữ mạng, công nghệ cụm và tính toán song song. Do tính đa dạng ứng dụng trong nền điện toán đám mây và sự không đồng nhất của các nút nguồn máy chủ, một số máy tính bị quá tải và một số máy tính rất nhẹ khi sự tăng trưởng nhanh chóng của lưu lượng mạng truy cập và dữ liệu. Do đó, chúng ta cần chiến lược cân bằng tải để điều chỉnh tải máy chủ, giảm chi phí truyền thông và cải thiện việc sử dụng tài nguyên. Tuy nhiên, với sự xuất hiện dữ liệu lớn và phát triển của điện toán đám mây đã làm thay đổi một số góc độ ở một số vấn đề. Điển hình như, giải quyết bài toán công việc dữ liệu lớn bằng các máy ảo trong điện toán đám mây, sự liên quan của dữ liệu cũng như sự di chuyển của một số máy ảo giao dịch với dữ liệu sẽ gây ra một vài ảnh hưởng. Cụ thể sẽ mang lại nhiều chi phí truyền thông giữa các máy chủ trong quá trình di chuyển và tính toán. Qua đó, làm giảm tỷ lệ sử dụng tài nguyên hệ thống.

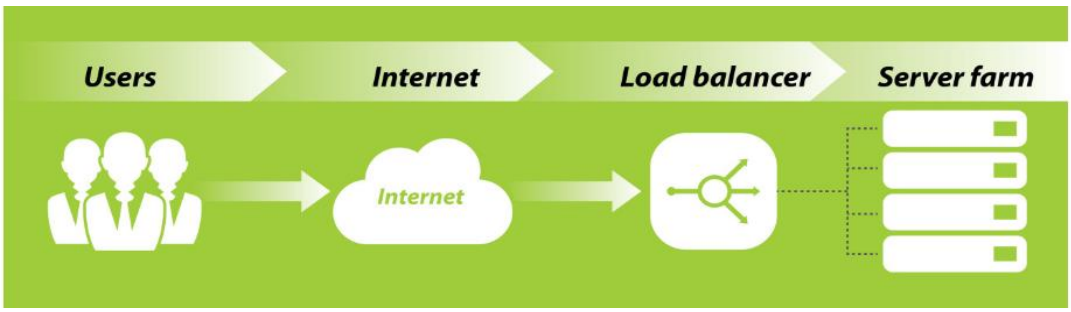
Điện toán đám mây là một kiểu [8] mẫu mới và tiến hóa đáng chú ý nhất trong tính toán. Cơ chế cân bằng tải được chia thành các nguồn lực và cung cấp các nguồn lực cùng với nhiệm vụ lập kế hoạch giữa các hệ thống phân phối. Cân bằng tải truyền thống phải đối mặt với một số vấn đề khác nhau của các giai đoạn cung cấp tài nguyên trong môi trường đám mây. Nó cũng có tác động to lớn trong các hệ thống đám mây về hiệu suất và về vấn đề đo lường do sự tham gia của các thông số cân bằng tải khác nhau cũng như bản chất của môi trường đám mây.

Trong thế giới ngày nay [9], điện toán đám mây là một cách để giữ phần cứng cũng như phần mềm ở một nơi và sử dụng nó từ mọi nơi trên thế giới. Nó đã làm cho phần cứng yêu cầu linh hoạt hơn nhiều. Do đó, mọi người có cơ hội sử dụng nhiều tài nguyên khi cần và phải trả số tiền chỉ cho khoảng thời gian họ đã sử dụng nguồn dung lượng cụ thể. Cái đó được gọi là dịch vụ trả tiền cho mỗi lần sử dụng. Nó sẽ dẫn dắt ngành công nghiệp công nghệ thông tin hướng đến việc kinh doanh điện toán đám mây. Giống như một CPU nhiều lõi, những doanh nghiệp sở hữu một cụm các CPU/Máy vật lý đó được gọi là đám mây. Các cụm có một số lượng hữu hạn không gian và bộ nhớ.



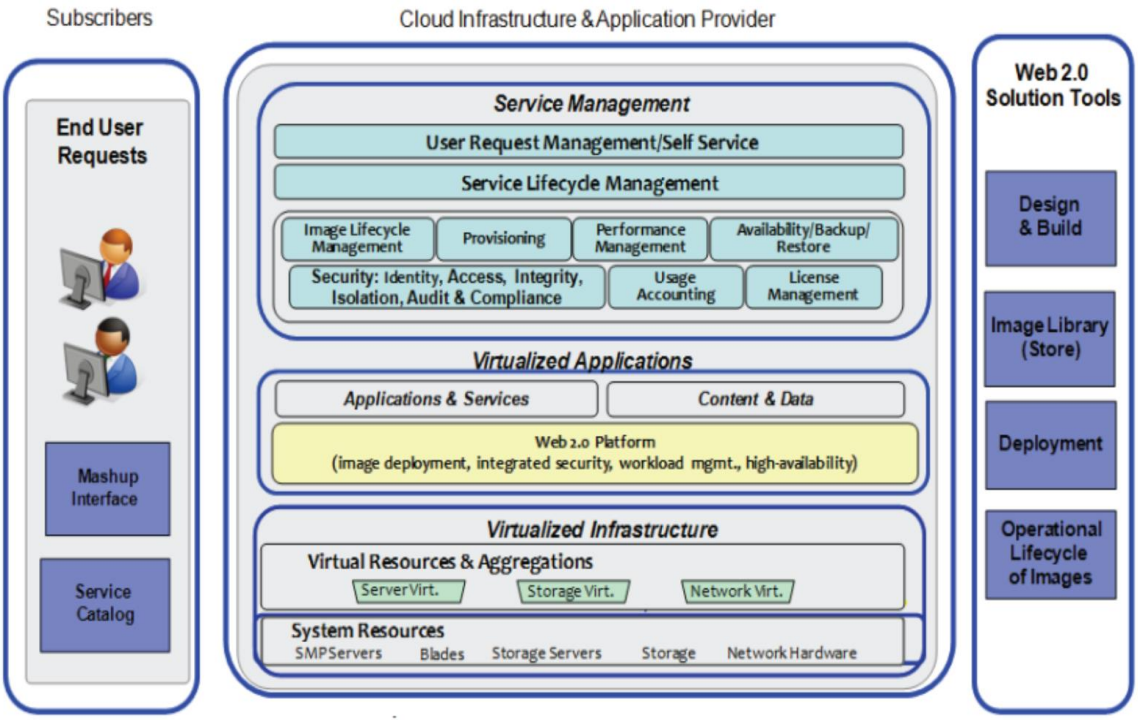
Hình 1.2: Cung cấp tài nguyên đám mây [4]

Vì vậy, khách hàng phải trả tiền để có không gian và bộ nhớ trong một khoảng thời gian từ cụm được phân bổ cho người dùng. Người sử dụng thường đòi hỏi các nguồn lực bao gồm: bộ nhớ, không gian và băng thông. Khi đó, nguồn lực sẽ được thực hiện bởi các công ty thông qua phân bổ các máy chủ đến nền tảng nhu cầu khách hàng. Cung cấp tài nguyên trên đám mây là quá trình cung cấp không gian bộ nhớ ảo từ các nguồn lực bằng cách tổng hợp máy vật lý (PM) được gọi là máy ảo (VM). Bộ cân bằng tải quản lý ghép kênh các tài nguyên theo yêu cầu.



Hình 1.3: Cân bằng tải trong điện toán đám mây [5]

Các biện pháp cân bằng trước đây có hiệu quả trong việc cải thiện thời gian phản hồi và thời gian phục vụ của đám mây, nhưng không cung cấp đúng chất lượng dịch vụ. Các QoS có thể được cung cấp hiệu quả bằng cách thêm tham số của nó vào tham số cân bằng tải. Xem xét băng thông như tham số, mà phải đối mặt với các vấn đề suy giảm và những vấn đề khác sẽ làm cho ngưỡng giá trị chính xác hơn. Do đó, QoS sẽ được coi là có hiệu quả. Vì vậy, cần giảm thiểu yêu cầu được cấp phát cho các máy vật lý với đúng khả năng cung cấp của các máy ảo và duy trì trạng thái ổn định trong suốt thời gian cung cấp dịch vụ.



Hình 1.4: Kiến trúc của điện toán đám mây [7]

Trong khi sử dụng tính toán tự động, tránh chi phí chung là một vấn đề lớn và giải quyết bằng cách đặt ra các nguồn lực thông qua thuật toán quy mô. Sau đó, vấn đề cuối cùng là giữ tải cân bằng ngay cả trong thời gian của giai đoạn phát triển. Điều này được thực hiện bằng cách sử dụng các thuật toán khác nhau.

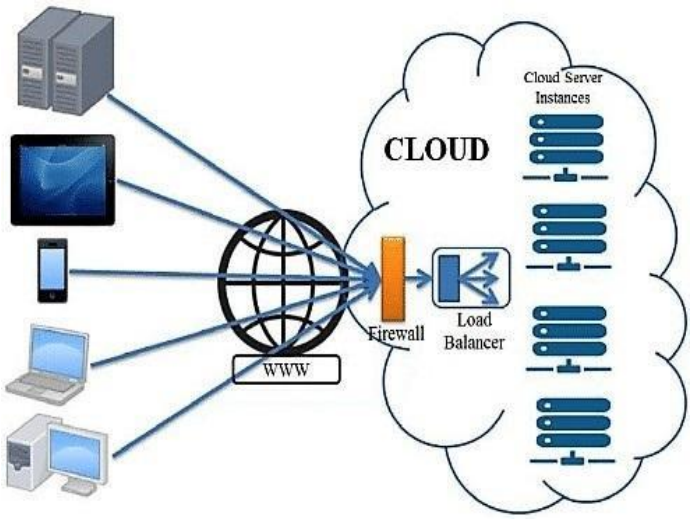
1.2. Tổng quan về cân bằng tải trong điện toán đám mây

1.2.1. Giới thiệu về cân bằng tải

Ngày nay Ngành công nghiệp CNTT đang phát triển mỗi ngày và nhu cầu về tài nguyên lưu trữ và tính toán cũng vậy. Một lượng lớn dữ liệu được tạo và trao đổi qua mạng, điều này đòi hỏi nhu cầu về tài nguyên máy tính ngày càng nhiều. Cloud đã giúp các doanh nghiệp tận dụng lợi ích của tài nguyên điện toán được chia sẻ trên môi trường ảo hóa. Rất nhiều doanh nghiệp đã sử dụng các dịch vụ dựa trên đám mây ở dạng này hay dạng khác. Điều này đưa chúng ta đến khái niệm cân bằng tải trong điện toán đám mây.

Cùng với việc phát triển rộng rãi của Internet, các website hay các ứng dụng trực tuyến cũng đang được rất nhiều người truy cập và sử dụng. Khi lượng truy cập quá lớn thường xảy ra các vấn đề là hạ tầng mạng và khả năng xử lý của Server sẽ bị tắc nghẽn cục bộ. Vì vậy, Cân Bằng Tải luôn là một trong những tính năng công nghệ rất quan trọng giúp các máy chủ ảo hoạt động đồng bộ và hiệu quả hơn thông qua việc phân phối đồng đều tài nguyên.

Giải pháp cân bằng tải là việc phân bố đồng đều lưu lượng truy cập giữa hai hay nhiều các máy chủ có cùng chức năng trong cùng một hệ thống. Bằng cách đó sẽ giúp cho hệ thống giảm thiểu tối đa tình trạng một máy chủ bị quá tải và ngưng hoạt động. Hoặc khi một máy chủ gặp sự cố, Cân Bằng Tải sẽ chỉ đạo phân phối công việc của máy chủ đó cho các máy chủ còn lại đồng thời đẩy thời gian uptime của hệ thống lên cao nhất và cải thiện năng suất hoạt động tổng thể.



Hình 1.5: Mô hình Cân bằng tải trong điện toán đám mây [8]

Cân bằng tải là một trong những chủ đề quan trọng nhất trong môi trường phân tán. Vì Cloud Computing được xem là một trong những nền tảng tốt nhất giúp lưu trữ dữ liệu với chi phí tối thiểu và có thể truy cập mọi lúc qua Internet. Cân bằng tải cho điện toán đám mây nay đã trở thành một lĩnh vực nghiên cứu rất thú vị và quan trọng. Cân bằng tải nhằm mục đích thỏa mãn người dùng và sử dụng tỷ lệ tài nguyên cao bằng cách đảm bảo phân bổ hợp lý. Có rất nhiều khó khăn trong các kỹ thuật cân

bằng tải như bảo mật, khả năng chịu lỗi, v.v... vốn phổ biến trong môi trường điện toán đám mây hiện đại. Nhiều nhà nghiên cứu đã đề xuất một số kỹ thuật và thuật toán để không ngừng tìm ra những phương án tốt nhất cho Cân bằng tải.

Phân tán dự đoán quá tải trong cân bằng tải [10] thời gian gần đây đã nổi lên như một giải pháp đầy hứa hẹn. Trong đó, chuyển sang cấp độ giám sát tình trạng tắc nghẽn của mỗi con đường và phân tán dòng chảy trực tiếp đến con đường ít tắc nghẽn. Cách tiếp cận này có nhiều lợi thế thực tiễn. Là một lược đồ phân phối, nó có thể mở rộng hơn và có thể đối phó với lưu lượng truy cập nhanh hơn cách lịch trình tập trung. Là một phương pháp tiếp cận dữ liệu, nó không phụ thuộc vào ngăn xếp mạng của máy chủ lưu trữ và ngay lập tức mang lại lợi ích cho tất cả lưu lượng truy cập khi triển khai. Khả năng hiển thị tắc nghẽn cuối cùng của nó cũng làm cho nó trở nên mạnh mẽ hơn mà không cần cấu hình lại máy điều khiển. Mấu chốt của việc thiết kế một giao thức cân bằng tải tắc nghẽn là chúng ta cần phải biết thông tin về tắc nghẽn thời gian thực từ tất cả các đường đi giữa nguồn dòng chảy và điểm đến. Một cách tiếp cận đơn giản là sử dụng thông tin định hướng đường đi cuối: Một switch ToR duy trì các chỉ số tắc nghẽn đầu cuối cho tất cả các đường dẫn từ chính nó đến các thiết bị chuyển mạch ToR khác trong mạng. Các chỉ số tắc nghẽn có thể được thu thập bằng các gói dữ liệu. Thông thường, có hàng trăm đường dẫn tồn tại giữa hai ToR thiết bị chuyển mạch và công tắc ToR có thể giao tiếp với hàng trăm các thiết bị chuyển mạch ToR khác. Quan trọng hơn, không thể để thu thập thông tin tắc nghẽn thời gian thực cho tất cả các đường dẫn này. Bởi, sẽ không có đủ dòng chảy đồng thời xảy ra đi với tất cả chúng cùng một lúc. Trong giai đoạn đầu, chỉ có nguồn và thiết bị chuyển mạch ToR đích tham gia để lựa chọn tốt nhất đường dẫn từ ToR đến tầng tổng hợp. Chuyển đổi nguồn ToR sẽ gửi số liệu tắc nghẽn của nó đến đích ToR, chúng sẽ kết hợp với các chỉ số tắc nghẽn để chọn con đường tốt nhất cho lớp tổng hợp. Trong giai đoạn thứ hai, tập hợp đã chọn sau đó sẽ chọn công tắc lõi tốt nhất theo một cách tương tự về tình trạng tắc nghẽn của bước nhảy thứ hai và thứ ba. Con đường quyết định lựa chọn sau đó được duy trì tại ToR và tập hợp thiết bị chuyển mạch. Về cơ bản, hai giai đoạn lựa chọn đường dẫn chỉ sử dụng thông tin của một phần đường dẫn

để tìm đường tốt nhất cho dòng chảy. Bằng cách khai thác các tính chất cấu trúc của 3 tầng, lựa chọn đường dẫn hai giai đoạn đã làm giảm đáng kể các vấn đề phức tạp và không có nhiều hiệu suất. Trên thực tế, đánh giá cho thấy rằng thực hiện lựa chọn đường dẫn trên mỗi cơ sở lưu lượng trong TCP là tốt nhất và không gây ra việc sắp xếp lại gói tin cũng như không gây bất kỳ độ trễ nào.

Cân bằng tải luôn là chủ đề nghiên cứu nóng của các trung tâm dữ liệu đám mây và mục tiêu của nó là đảm bảo rằng mọi tài nguyên máy tính có thể xử lý các nhiệm vụ một cách hiệu quả và nhanh chóng. Cuối cùng, việc sử dụng nguồn lực được cải thiện. Các nhà nghiên cứu đã đề xuất một loạt cân bằng tĩnh, cân bằng động và chiến lược lập kế hoạch cân bằng tải. Ngoài ra, cũng có một số nghiên cứu sử dụng công nghệ di chuyển trực tiếp của máy ảo để đáp ứng các yêu cầu đám mây cũng như nhiệm vụ của trung tâm dữ liệu là yêu cầu hiệu suất và giới hạn tải. Các chiến lược cân bằng tải hiện được chia thành hai loại: cân bằng tải tĩnh và cân bằng tải năng động. Thuật toán lập lịch cân bằng tải tĩnh thường bao gồm Round Robin, Rounded Robin Weighted. Các thuật toán tĩnh chỉ sử dụng một số thông tin tĩnh mà không thể phản ánh tải động. Hiện nay, hầu hết các nền tảng mã nguồn mở, kể cả IaaS, đã sử dụng các thuật toán tĩnh để tiến hành lập kế hoạch tài nguyên. Lợi thế của thuật toán lập kế hoạch cân bằng tải tĩnh là nó rất đơn giản và dễ sử dụng. Nhưng trong các trung tâm dữ liệu đám mây quy mô lớn, với tài nguyên có tính không đồng nhất và nhu cầu người sử dụng không nhất quán thì hiệu quả cân bằng tải tĩnh không được lý tưởng. Cân bằng tải động (DLB), nó chủ yếu được sử dụng trong lĩnh vực phân phối máy tính song song. Mục tiêu chính của nó là làm thế nào để phân phối tải hợp lý hơn giữa nhiều máy chủ để tránh một số hiện tượng như một số các nút máy tính bị quá tải và một số nút có tải nhẹ. Do đó, cần tìm ra giải pháp để cải thiện toàn bộ hiệu suất của hệ thống. Chi phí truyền thông bổ sung được tạo ra trong quá trình DLB sẽ làm suy giảm hiệu năng hệ thống của cân bằng tải động. Vì vậy, làm thế nào để giảm truyền gói tin trên cao nhất giữa các nút trong quá trình DLB đã trở thành một vấn đề quan trọng, sẽ ảnh hưởng đến hiệu suất của DLB. Tuy nhiên, một số thuật toán ở trên không thể đáp ứng được sự lựa chọn và bản chất của cơ cấu cân bằng tải tối ưu cùng một

lúc. Thế nên, những cách phân phối tiếp cận thường có được sự tối ưu cục bộ của các giải pháp. Hiệu quả của việc giải quyết vấn đề phân phối tải trong một số trường hợp đặc biệt không phải là lý tưởng. Chính vì điều đó, nó có thể đảm bảo cân bằng tải và sử dụng hiệu quả tài nguyên vật lý của toàn bộ cụm. Dẫu vậy, cân bằng tải lại là vấn đề và chi phí chung của đám mây trong các trung tâm dữ liệu không được xem xét. Nó chỉ tập trung vào quản lý máy ảo để tăng cường quản lý các trung tâm dữ liệu đám mây đồng thời nâng cao hiệu quả hoạt động của các trung tâm dữ liệu điện toán đám mây.

Cân bằng tải [11] có thể được chia thành 2 thể loại:

- Cân bằng tải cục bộ
- Tải toàn cầu

Cân bằng tải cục bộ được sử dụng để cân bằng dự báo tải trong một trung tâm. Nó phân phối yêu cầu từ phía máy khách sang máy chủ để đáp ứng nhu cầu. Loại cân bằng tải thứ hai là cân bằng tải toàn cục. Nó quản lý và kiểm soát yêu cầu từ phía khách hàng tự động đến máy chủ qua nhiều trung tâm dữ liệu. Ngoài ra, nó còn xử lý lưu lượng trên cả hai mặt gói truyền tải. Xử lý cân bằng tải toàn cầu cho sự phức tạp, nhưng đồng thời điều này c rất hữu ích cho truyền tải gói tin trên trung tâm dữ liệu mạng. Tính khả dụng đảm bảo rằng, trong trường hợp thất bại, hệ thống vẫn tiếp tục hoạt động như mong đợi.

1.2.2. Mục đích cân bằng tải

Tăng khả năng đáp ứng, tránh tình trạng quá tải trên máy chủ đồng thời đảm bảo tính linh hoạt và mở rộng cho hệ thống.

Tăng độ tin cậy và khả năng dự phòng cho hệ thống: Sử dụng Cân bằng tải giúp tăng tính HA (High Availability) cho hệ thống. Mặt khác, đảm bảo cho người dùng không bị gián đoạn dịch vụ khi xảy ra lỗi sự cố lỗi tại một điểm cung cấp dịch vụ.

Tăng tính bảo mật cho hệ thống: Thông thường khi người dùng gửi yêu cầu dịch vụ đến hệ thống, yêu cầu đó sẽ được xử lý trên bộ Cân bằng tải. Sau đó, thành phần Cân bằng tải mới chuyển tiếp các yêu cầu cho các máy chủ bên trong. Quá trình trả lời cho khách hàng cũng thông qua thành phần Cân bằng tải. Chính vì vậy mà người dùng không thể biết được chính xác các máy chủ bên trong cũng như phương pháp phân tải được sử dụng. Bằng cách này có thể ngăn chặn người dùng giao tiếp trực tiếp với các máy chủ, ẩn các thông tin và cấu trúc mạng nội bộ. Đồng thời, ngăn ngừa các cuộc tấn công trên mạng hoặc các dịch vụ không liên quan đang hoạt động trên các cổng khác.

1.3. Tổng quan về trí tuệ nhân tạo (AI)

Trí tuệ nhân tạo (AI) [1] là một ngành khoa học máy tính liên quan đến việc tạo ra các chương trình nhằm mục đích tái tạo nhận thức con người và các quá trình liên quan đến việc phân tích sự phức tạp của dữ liệu. Sự ra đời của khái niệm này được liên kết phổ biến với hội nghị Dartmouth năm 1956 [2]. Tuy nhiên, công nghệ tại thời điểm này đã giới hạn việc ứng dụng AI. Gần đây, những tiến bộ đáng kể đã được thực hiện trong lĩnh vực sức mạnh máy tính do công nghệ phần cứng và phần mềm được cải tiến. Các cá nhân và tổ chức ở một số các ngành công nghiệp đang bắt đầu nhận ra tiềm năng của AI để cải thiện các hoạt động hiện tại. Vậy nên, nghiên cứu AI đã được tiến hành trong nhiều lĩnh vực: y tế, điện toán đám mây, xử lý ảnh,…

1.4. Tổng quan về machine learning

Học máy (Machine Learning / ML) [3] là một ứng dụng của trí tuệ nhân tạo (AI) cung cấp cho máy móc khả năng tự động học hỏi và cải thiện mà không cần được lập trình rõ ràng cho từng tác vụ. Machine Learning liên quan đến các chương trình máy tính viết lập trình của riêng chúng để hoàn thành một nhiệm vụ định trước. Quá trình này có thể được giám sát, bán giám sát hoặc không giám sát (Hình 1.1). Trong học tập có giám sát, máy được cung cấp dữ liệu trong đó mỗi ví dụ trong tập dữ liệu được gắn nhãn với câu trả lời. Các câu trả lời sau đó sẽ được máy học thông qua thử và sai để dự đoán câu trả lời từ tập dữ liệu đã nhập. Học tập không giám sát

liên quan đến việc phân tích dữ liệu đầu vào mà không có câu trả lời xác định. Điều này thường được sử dụng để mô hình hóa cấu trúc và phân phối dữ liệu. Cuối cùng, học tập bán giám sát là một phương pháp kết hợp liên quan đến việc kết hợp dữ liệu được gắn nhãn và không được gắn nhãn. Điều này có thể giúp giảm bớt gánh nặng của nhiệm vụ ghi nhãn. Sử dụng các thuật toán phân lớp của ML để tiến hành phân lớp người dùng dựa trên các đặc trưng của họ để thực hiện việc cân bằng tải.

1.5. Kết luận chương

Hiểu biết được những khái niệm tổng quan về điện toán đám mây. Hiểu biết thuật toán điện toán đám mây giải quyết những vấn đề tắc nghẽn cũng như gói tin mất mát khi truyền dữ liệu qua môi trường điện toán và mục đích cân bằng tải để làm tăng hiệu năng của hệ thống.

CHƯƠNG 2: CÁC CÔNG TRÌNH LIÊN QUAN

2.1. Giới thiệu chương

Trong chương này xin giới thiệu các công trình liên quan đến điện toán đám mây, cân bằng tải và các thuật toán AI.

2.2. Các công trình liên quan

Năm 2015, Anita Rani và Pankajdeep Kaur [12] công bố các nghiên cứu về di chuyển các tác vụ trên cloud thông qua "Migration Jobs in Cloud Computing". Điện toán đám mây là việc cung cấp các dịch vụ điện toán qua Internet. Dịch vụ đám mây cho phép các cá nhân và tổ chức doanh nghiệp khác sử dụng dữ liệu được quản lý bởi bên thứ ba hoặc một người khác tại các địa điểm từ xa. Trong điện toán đám mây, có hai kỹ thuật di chuyển là quá trình di chuyển và di chuyển máy ảo. Quá trình di chuyển là một kỹ thuật mà trong đó, một quá trình đang hoạt động được di chuyển từ máy này sang máy khác với kiến trúc có thể khác nhau. Máy ảo (VM) là một mô phỏng của một hệ thống máy tính cụ thể. Di chuyển máy ảo là một công cụ hữu ích để di chuyển các phiên bản HĐH trên nhiều máy vật lý. Cả hai kỹ thuật đều được sử dụng với mục đích cân bằng tải, quản lý lỗi, bảo trì hệ thống mức thấp, giao tiếp tốt hơn và giảm tiêu thụ năng lượng. Bài báo này trình bày các kỹ thuật di chuyển máy ảo và quy trình khác nhau. Đồng thời mô tả cách tiếp cận không thăm dò, phù hợp với các ứng dụng mong muốn độ trễ tối thiểu và có thể đủ khả năng để kích hoạt hiệu quả tất cả các điểm tương đương. Để giảm thiểu tiêu thụ tài nguyên, một số giải pháp được đưa ra và sử dụng để giải quyết vấn đề. Do đó, những cách tiếp cận này đạt được chi phí hiệu suất tối thiểu trong quá trình thực hiện quy trình. Các kỹ thuật di chuyển trực tiếp của máy ảo, bao gồm việc chuyển một máy ảo, đang chạy qua các máy chủ vật lý. Có nhiều kỹ thuật được sử dụng để giảm thiểu thời gian ngừng hoạt động và tổng thời gian di chuyển để mang lại hiệu suất tốt hơn trong băng thông thấp. Hiện có rất ít kỹ thuật di chuyển nhận biết mạng và giúp ích nhiều cho người dùng. Với sự gia tăng phổ biến của các hệ thống điện toán đám mây, việc di chuyển máy ảo qua các trung tâm dữ liệu và vùng tài nguyên sẽ có lợi rất nhiều cho các quản trị viên trung

tâm dữ liệu. Có ít hơn các kỹ thuật di chuyển nhận biết mạng khả dụng. Với sự gia tăng phổ biến của các hệ thống điện toán đám mây, việc di chuyển máy ảo qua các trung tâm dữ liệu và vùng tài nguyên sẽ có lợi rất nhiều cho các quản trị viên trung tâm dữ liệu. Do đó, thời gian ngừng hoạt động và thời gian di chuyển sẽ được giảm thiểu bớt.

Năm 2015, một nghiên cứu của Weishan Zhang [13] và cộng sự đã công bố bài báo "A Genetic-Algorithm-Based Approach for Task Migration in Pervasive Clouds". Trong bài báo này, điện toán lan tỏa đang hội tụ với điện toán đám mây, trở thành điện toán đám mây phổ biến như một mô hình điện toán mới nổi. Người dùng có thể chạy các ứng dụng hoặc tác vụ của họ trong môi trường đám mây phổ biến để đạt được hiệu quả thực thi cùng hiệu suất tốt hơn nhờ khả năng lưu trữ và tính toán mạnh mẽ của các đám mây phổ biến thông qua việc di chuyển tác vụ. Trong quá trình di chuyển nhiệm vụ, có thể có một số mục tiêu mâu thuẫn nhau cần được xem xét khi đưa ra quyết định di chuyển, chẳng hạn như tiêu thụ ít năng lượng hơn và phản ứng nhanh, để tìm ra con đường di chuyển tối ưu. Trong bài báo này, nhóm tác giả đã đề xuất một cách tiếp cận dựa trên thuật toán di truyền (GAs-) với hiệu quả trong việc giải quyết các vấn đề tối ưu hóa đa mục tiêu. Các tác giả đã thực hiện một số đánh giá sơ bộ về cách tiếp cận được đề xuất cho thấy kết quả khá hứa hẹn, sử dụng một trong những thuật toán di truyền cổ điển. Kết luận là GA có thể được sử dụng để ra quyết định trong việc di chuyển nhiệm vụ trên các đám mây phổ biến.

Trong nghiên cứu của Geetha Megharaj [14] và cộng sự vào năm 2018, bài báo "Run Time Virtual Machine Task Migration Technique for Load Balancing in Cloud" đã nghiên cứu sâu hơn về di chuyển tác vụ. Cân bằng tải là một khía cạnh quan trọng của các trung tâm dịch vụ đám mây để tối ưu hóa việc sử dụng tài nguyên. Việc tiêu thụ điện năng dư thừa trong các trung tâm đám mây có thể dẫn đến lãng phí tiền tệ. Điều quan trọng là, các tài nguyên trong trung tâm đám mây được sử dụng một cách tối ưu sao cho vừa tiết kiệm được tiền vừa đạt được sự hài lòng của khách hàng. Một trong những kỹ thuật phổ biến nhất để đạt được cân bằng tải là kỹ thuật di chuyển Máy ảo (VM). Trong đó, một số máy ảo từ Máy vật lý (PM) quá tải được

chuyển sang PM được tải nhẹ. Tuy nhiên, kỹ thuật này đòi hỏi quá nhiều thời gian và chi phí tiền tệ. Gần đây, một kỹ thuật cân bằng tải di chuyển các tác vụ VM thay vì VM thực tế đã được đề xuất trong tài liệu. Kỹ thuật này đã có thể khắc phục một số hạn chế của kỹ thuật di chuyển VM. Ở đây, VM quá tải không chấp nhận bất kỳ tác vụ mới nào. Song, các tác vụ mới được chuyển sang các máy ảo được tải nhẹ. Mặc dù kỹ thuật này di chuyển các tác vụ bổ sung để đạt được cân bằng tải VM, nhưng các máy ảo đã quá tải vẫn không được giải tỏa khỏi gánh nặng tác vụ hiện có của chúng. Nếu một số tác vụ hiện có và phù hợp trong máy ảo quá tải được di chuyển, nó có thể cải thiện hiệu quả của việc cân bằng tải. Trong công việc này, một kỹ thuật di chuyển tác vụ VM thời gian chạy mới được đề xuất. Kỹ thuật này sẽ di chuyển các tác vụ từ các máy ảo quá tải. Các nhiệm vụ phù hợp cho quá trình di chuyển được lựa chọn thông qua một hàm phân biệt, xác định các nhiệm vụ tiêu tốn nhiều tài nguyên và các nhiệm vụ được thực hiện hạn chế đối với việc di chuyển. Kể từ đó, nó đã được chỉ ra trong tài liệu rằng, lập bản đồ tài nguyên nhiệm vụ tối ưu là NPhard, kỹ thuật tìm kiếm giải pháp dựa trên Particle Swarm Optimization (PSO) được đề xuất. Kỹ thuật được đề xuất này làm giảm đáng kể tải tính toán. Đồng thời đạt được mức bảo toàn năng lượng / điện năng tốt trong các máy ảo quá tải khi so sánh với các kỹ thuật di chuyển tác vụ VM hiện đại và kỹ thuật di chuyển máy ảo.

Gần đây, năm 2020, các tác giả Chen Ling, Hui He [15] và cộng sự đã công bố “Network perception task migration in cloud-edge fusion computing”. Với sự phát triển của điện toán đám mây, điện toán biên đã được đề xuất để cung cấp các dịch vụ thời gian thực và độ trễ thấp cho người dùng. Nghiên cứu hiện tại thường tích hợp điện toán đám mây và điện toán biên dưới dạng điện toán tổng hợp cạnh đám mây cho các dịch vụ được cá nhân hóa hơn. Tuy nhiên, cả điện toán đám mây và điện toán biên đều phải chịu mức tiêu thụ mạng cao. Đây vẫn là một vấn đề quan trọng chưa được giải quyết trong môi trường điện toán tổng hợp cạnh đám mây. Chi phí tiêu thụ mạng có thể được chia thành hai phần: chi phí di chuyển và chi phí truyền thông. Để giải quyết vấn đề tiêu thụ mạng cao, một số máy ảo có thể được di chuyển từ máy vật lý quá tải sang máy khác với sự trợ giúp của công nghệ ảo hóa. Các chiến lược di

chuyển nhận thức mạng hiện tại tập trung nhiều hơn vào chi phí truyền thông bằng cách tối ưu hóa cấu trúc liên kết truyền thông. Xem xét cả chi phí truyền thông và di chuyển, bài báo này giải quyết vấn đề tiêu thụ mạng cao về mối tương quan truyền thông của các máy ảo cũng như lưu lượng mạng của quá trình di chuyển. Nó đề xuất ba thuật toán di chuyển máy ảo heuristic: LM, mCaM và mCaM2 để cân bằng giữa chi phí truyền thông và chi phí di chuyển. Hiệu suất của các thuật toán này được so sánh với hiệu suất của các thuật toán di chuyển máy ảo hiện có thông qua các thử nghiệm. Kết quả thử nghiệm cho thấy các thuật toán di chuyển máy ảo của chúng tôi rõ ràng tối ưu hóa chi phí truyền thông và chi phí di chuyển. Ba thuật toán này có chi phí mạng thấp hơn AppAware, một thuật toán hiện có, trung bình 20%. Điều này có nghĩa là ba thuật toán này cải thiện hiệu suất mạng và giảm mức tiêu thụ mạng trong các môi trường điện toán tổng hợp cạnh đám mây. Chúng cũng vượt trội so với các thuật toán hiện tại về thời gian hoạt động trung bình 70%..

2.3. Tổng kết chương

Trong chương này thông qua việc nghiên cứu tìm hiểu được một số thuật toán và những công trình liên quan tới cân bằng tải trong điện toán đám mây. Qua đó giúp luận văn hiểu rõ hơn về cân bằng tải và tải trên điện toán đám mây. Từ đó, hiểu được những ưu nhược điểm của các thuật toán cũng như các cách xử lý cân bằng tải, tạo tiền đề và cơ sở vững chắc cho nghiên cứu của đề tài luận văn này.

CHƯƠNG 3 : ĐỀ XUẤT THUẬT TOÁN DỰ BÁO THỜI GIAN DI CHUYỂN TÁC VỤ NHẰM NÂNG CAO HIỆU NĂNG CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY

3.1. Giới thiệu chung

Công nghệ cân bằng tải trên cloud dần trở thành xu hướng nghiên cứu của các nhà khoa học, các tổ chức cũng như các doanh nghiệp lớn nhỏ nói chung. Bên cạnh đó, các thuật toán cân bằng tải [16] trên đám mây cũng được cải thiện và phát triển để tối ưu hóa thời gian xử lí cũng như phân bổ các request vào các máy ảo một cách phù hợp. Một trong các yếu tố có khả năng gây ảnh hưởng đến hiệu suất cân bằng tải trên cloud đó chính là thời gian di chuyển tác vụ đến các máy ảo. Vì thế, trong chương này, luận văn sẽ đề xuất thuật toán dự báo thời gian di chuyển tác vụ nhằm giảm thiểu sự mất cân bằng tải trên cloud.

3.2. Mô hình nghiên cứu

Mô hình nghiên cứu sử dụng thuật toán Linear Regression nhằm mục đích dự đoán thời gian thực hiện và phân loại các task tương ứng với các Request dựa trên độ lịch sử di chuyển task giữa các máy ảo. Độ ưu tiên ở đây được tính toán dựa trên mức độ tiêu thụ năng lượng của task (Power consumed), mức độ sử dụng CPU (CPU Usages), mức độ sử dụng RAM (RAM Usages) và chi phí (Costing) để thực hiện task đó trong cloud. Sau khi phân loại các job/task theo độ ưu tiên, bộ cân bằng tải sẽ phân bổ các request có task đó với độ ưu tiên cao hơn vào những máy ảo/host có năng lực xử lý tốt hơn, tức là mức độ rảnh task cao. Từ đó, phân bổ request có nhu cầu xử lý nhiều vào máy ảo/host có mức độ hoạt động thấp nhất. Với cách tiếp cận này, thuật toán đề xuất sẽ cải thiện thời gian xử lý cân bằng tải trên cloud và ứng dụng trên môi trường cloud theo thời gian thực. Trong luận văn này tạm đặt tên thuật toán là TLRegA.

Mô hình nghiên cứu sử dụng thuật toán

Quá trình cân bằng tải được thực hiện gồm các bước như sau:

Bước 1: Nhận thông tin input (các request nhận được)

Bước 2: Sử dụng các thuật toán dự báo Linear Regression để tiến hành dự báo thời gian di chuyển tác vụ giữa các máy ảo dựa trên các đặc trưng và trạng thái hoạt động của máy ảo.

Về mục tiêu:

- Giảm thiểu rủi ro cho hệ thống máy chủ.

- Giảm thiểu thời gian sống cho các yêu cầu trong điện toán đám mây.

- Hạn chế tối đa hoặc ngăn chặn sự mất cân bằng tải giữa các máy ảo.

Giả định:

- Bộ cân bằng tải sẽ biết trước các dịch vụ nào đang chạy trên các máy ảo vào bất cứ thời điểm nào.

- Ở đây tập trung vào dịch vụ Web (Web Service), các máy chủ web sẽ biết trước thời gian xử lý của từng dịch vụ chạy trên web và trên từng máy ảo.

- Nếu hai máy ảo có cấu hình tương đương nhau về RAM, vi xử lý và I/O thì thời gian thực thi của các dịch vụ sẽ không mấy khác nhau.

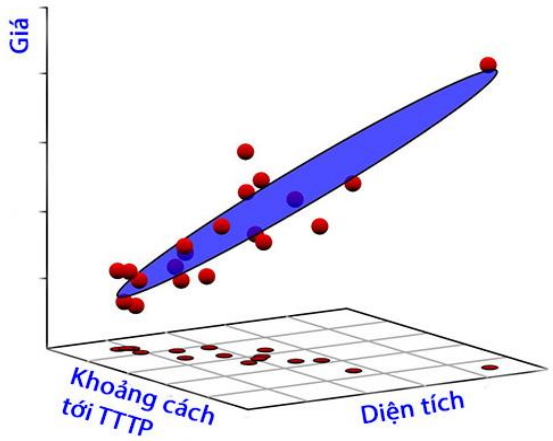
3.3. Thuật toán Linear Regression (LR)

Linear Regression (LR) là một thuật toán học giám sát (supervised learning) được sử dụng rộng rãi trong các bài toán liên quan đến việc dự đoán đầu ra của dữ liệu dựa trên các đặc trưng từ tập dữ liệu. Nói một cách khác, thuật toán tìm ra phương trình tuyến tính dựa trên tập dữ liệu quan hệ giữa X (dữ liệu đầu vào) và Y (dữ liệu đầu ra). X là biến giải thích và Y là biến phụ thuộc. Ví dụ, tạo ra mô hình quan hệ giữa chiều cao và cân nặng bằng mô hình hồi quy tuyến tính. Trước khi thử tạo ra mô hình quan hệ cần xác định sự liên quan giữa các mối quan hệ này. Điều này chỉ ra rằng, không nhất thiết phải có sự tương tác giữa các biến nhưng cần phải có sự liên quan. Có thể dùng biểu đồ phân tán như một công cụ trong việc xác định mức độ liên

quan để chỉ ra mối quan hệ giữa các biến. Nếu không có mối quan hệ nào giữa các biến được đưa vào mô hình, thì mô hình hồi quy tuyến tính sẽ không giúp ích được trong trường hợp này.

Tùy vào mục đích tính toán và sử dụng mà mỗi bài toán sẽ có cách ứng dụng thuật toán LR khác nhau. Song, để có một kết quả dự đoán thuyết phục thì đầu vào của thuật toán cần phải phụ thuộc vào nhiều đặc trưng khác nhau. Bên cạnh đó, độ chính xác của thuật toán còn phụ thuộc vào hàm mất mát (loss function) vì để tìm được một đường thẳng đi qua tất cả các điểm input cho trước là điều không thể. Một ví dụ đơn giản để mô phỏng ý tưởng của thuật toán:

Để dự đoán giá của một căn nhà, cần phải phụ thuộc vào nhiều yếu tố như diện tích, vị trí, thời điểm,… Tuy nhiên, giả sử giá nhà phụ thuộc vào hai yếu tố là diện tích và khoảng cách tới trung tâm thành phố, khi đó thuật toán Linear Regression với hai biến có nhiệm vụ tìm ra mặt phẳng biểu diễn sự phụ thuộc giữa giá nhà với hai yếu tố trên một cách chính xác nhất.



Hình 3.1: Không gian ba chiều biểu diễn các mối quan hệ đầu vào

Phương trình mặt phẳng trên có thể viết dưới dạng tổng quát sau:

$$\hat{P}(s,d) = w_0 + w_1 s + w_2 d$$

với $\hat{P}(s,d)$ là giá căn nhà diện tích s và cách trung tâm thành phố

d là khoảng cách

Trong trường hợp tổng quát, input bao gồm biến x_1, x_2,…, x_n và output y phụ thuộc vào n biến đó theo phương trình tuyến tính thì thuật toán tìm ra phương trình gọi là Linear Regression n biến:

$$\hat{y} = w_0 + w_1x_1 + \ldots + w_{n-1}x_{n-1} + w_nx_n$$

Để đơn giản hóa phương trình ở trên, ta có thể đặt w = $[w_0\ w_1\ \ldots\ w_n\ ]$ là vector các hệ số của phương trình, x = $[1\ x_1\ \ldots\ x_n\ ]$ là vector các biến của input (phần tử 1 trong vector đóng vai trò như x_0 chỉ nhằm mục đích thuận tiện cho tính toán) thì có thể viết lại phương trình trên như sau:

$$\hat{y} = xTw$$

3.4. Thuật toán đề xuất cân bằng tải

Task Migration Time **P**rediction using **L**inear **Reg**ression Analysis **A**lgorithm TLRegA

Dựa vào các mô hình cân bằng tải đã được công bố hiện nay [17], [18], chúng ta tiến hành đánh giá mô hình của mình. Từ đó, ta biết cách phân bố tài nguyên cho các request này.

Dựa vào tham khảo từ tài liệu [19], luận văn này xin đề xuất thuật toán gồm 3 nhóm module chính tuần tự như sau:

(1) Module dự đoán thời gian di chuyển tác vụ dùng thuật toán Linear Regression:

Trong module này, thuật toán Linear Regression sẽ dựa vào lịch sử di chuyển tác vụ giữa các máy ảo. Từ đó có thể dự đoán và đưa ra các kế hoạch cũng như chiến lược phân bổ tài nguyên một cách hợp lý đến các máy ảo, nâng cao hiệu suất cân bằng tải trên điện toán đám mây.

Nhóm Thời Gian xử lý = LR (X_1, X_2,…, X_n)

Trong đó X_i là thời gian di chuyển các tác vụ.

(2) Module phân lớp tác vụ:

Trong module này sẽ sử dụng thuật toán phân cụm K-Means (với k = 3) để phân cụm các máy ảo dựa vào mức động hoạt động cũng như sử dụng tài nguyên của máy ảo bao gồm cụm cao, trung bình và thấp. Việc phân cụm máy ảo này dựa vào thông số tức thời của các máy ảo:

$Cluster_i$ = K-Means (CPU usage, RAM, …);

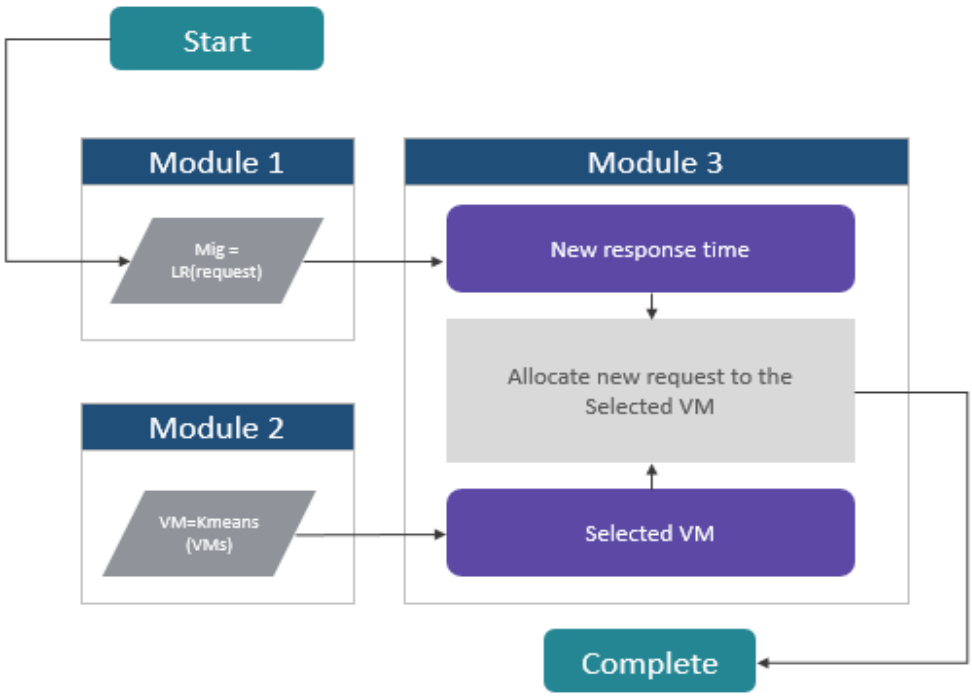
Trong đó: i = 1 là nhóm thấp

i = 2 là nhóm trung bình

i = 3 là nhóm cao

(3) Module phân bổ tác vụ vào máy ảo dựa trên thời gian dự báo

Module này có nhiệm vụ phân bổ các tác vụ đến các máy ảo theo thời gian dự báo các tác vụ. Nếu một yêu cầu được gửi đến thì yêu cầu này được phân loại bởi Module 1 và các VM đang xét, kể cả VM không tải cũng được phân cụm theo Module 2.



Hình 3.2: Sơ đồ hoạt động của thuật toán TLRegA

Thuật toán TLRegA

```
(1)  For each Task in VMTasks
(2)      isLocated = true;
(3)      Mig_Time = LR(X1, X2…..);  // Module 1
(4)      VM_Cluster = kMeans(state);  // state: Trạng thái của các VM Module 2
(5)      For each VM in VMList
(6)          If  isFitSituation(Task.Mig_Time , VM.VM_Cluster)
(7)              AllocateRequestToVM(VM, Request); // Module 3
(8)              isLocated = true;
(9)              break;
(10)         End If
(11)     End For
(12)     If (!isLocated)
(13)         VM = VMList.getMinFromMean();  // Module 2
(14)         AllocateRequestToVM(VM, Request);
(15)     End If
(16) End For
```

3.5. Kết luận chương 3

Chương này đưa ra mô hình nhằm giải quyết vấn đề cân bằng tải thông qua các kỹ thuật AI hiện đại. Với những mục tiêu ban đầu là duy trì tính ổn định và hoạt động liên tục của cloud, thuật toán đề xuất TLRegA đã chứng minh được tính hiệu quả của nó trong quá trình cân bằng tải. Cụ thể, thuật toán giảm thiểu được thời gian hoạt động cho các yêu cầu và các rủi ro nhất định đồng thời ngăn chặn mất cân bằng tải và hạn chế tối đa sự mất cân bằng tải giữa các máy ảo.

CHƯƠNG 4: MÔ PHỎNG CHƯƠNG TRÌNH VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Giới thiệu chương 4

Trong chương này trình bày về cài đặt mô phỏng thuật toán TLRegA, cụ thể như sau: sử dụng thuật toán Linear Regression nhằm mục đích loại các task tương ứng với các Request dựa trên độ ưu tiên xử lý task đó. Độ ưu tiên ở đây được tính toán dựa trên mức độ tiêu thụ năng lượng của task (Power consumed), mức độ sử dụng CPU (CPU Usages), mức độ sử dụng RAM (RAM Usages) và chi phí (Costing) để thực hiện task đó trong cloud. Sau khi phân loại các tasks theo độ ưu tiên, bộ cân bằng tải sẽ phân bổ các request có task đó với độ ưu tiên cao hơn vào những máy ảo/host có năng lực xử lý tốt hơn, tức là mức độ rảnh task cao. Từ đó, phân bổ request có nhu cầu xử lý nhiều vào máy ảo/host có mức độ hoạt động thấp nhất. Với cách tiếp cận này, thuật toán đề xuất TLRegA sẽ cải thiện thời gian xử lý cân bằng tải trên cloud và ứng dụng trên môi trường cloud theo thời gian thực. Sau khi tiến hành các bước như trên ta thu được các kết quả, từ đó phân tích tính hiệu quả của thuật toán đề ra.

4.2. Mô tả môi trường mô phỏng thực nghiệm

Dựa vào dữ liệu của các request mà ta có thể biết, ta sử dụng thuật toán Regression để phân loại request bằng cách tính toán ra bộ Priority = {Power, CPU, RAM}. Qua đó, ta biết cách phân bố tài nguyên cho các request vào các máy ảo đã phân cụm. Kết hợp với đánh giá số lần sai và sai số, ta cải thiện thuật toán bằng cách áp dụng máy học vào. Tuy nhiên, việc áp dụng này sẽ ít diễn ra vì có sai số cho phép. Giả lập môi trường cloud sử dụng bộ thư viện CloudSim và lập trình trên ngôn ngữ JAVA; Môi trường giả lập cloud là từ 25 đến 75 máy ảo, và tạo môi trường request ngẫu nhiên tới các dịch vụ trên cloud này. Bao gồm dịch vụ cung cấp máy ảo, dịch vụ cung cấp và đáp ứng người dùng của CloudSim để thử nghiệm. Cài đặt thuật toán Linear Regression trên môi trường mô phỏng Cloud Analyst.

- Dựa vào mô hình được đề xuất tiến hành xây dựng mô hình cân bằng tải có kết hợp với các thuật toán dự báo thời gian di chuyển tác vụ, được cài đặt với ngôn ngữ Java và Cloud Analyst [19].

Bước 1: Tiếp nhận giá trị input

Bước 2: Phân tích các input để rút trích các đặc trưng của các input

Bước 3: Dựa vào các đặc trưng trên chúng ta sử dụng machine learning để phân lớp các đầu vào.

Bước 4: Dựa vào kết quả phân lớp ta tiến hành cân bằng tải cho hệ thống.

Các tham số của mô hình mạng mô phỏng:

Thực nghiệm mô phỏng thuật toán đề xuất được cài đặt trên ngôn ngữ JAVA và sử dụng ECLIPSE IDE để chạy thử và hiển thị kết quả dưới dạng console. Môi trường giả lập với bộ thư viện mã nguồn mở CloudSim 4.0 (được cung cấp bởi http://www.cloudbus.org/) và Cloud Analyst.

Môi trường mô phỏng giả lập gồm các thông số sau:

- 01 Datacenter với thông số như sau:

Bảng 4.1: Thông số cấu hình Datacenter

Thông tin Datacenter	***Thông tin Host trong Datacenter***
- Không sử dụng Storage (các ổ SAN) - Kiến trúc(arch): x86 - Hệ điều hành (OS): Linux - Xử lý (VMM): Xen - TimeZone: +7 GMT - Cost: 3.0 - Cost per Memory: 0.05 - Cost per Storage: 0.1 - Cost per Bandwidth: 0.1	Mỗi host trong Datacenter có cấu hình như sau: - CPU có 4 nhân, mỗi nhân có tốc độ xử lý là 1000 (mips) - Ram: 16384 (MB) - Storage: 1000000 - Bandwidth: 10000

- Các máy ảo có cấu hình giống nhau khi được khởi tạo:

Bảng 4.2: Cấu hình máy ảo

Kích thước (size)	Ram	Mips	Bandwidth	Số lượng cpu (pes no.)	VMM
10000 MB	512 MB	250	1000	1	Xen

- Các Request (các request chạy trên web, WebRequest) được đại diện bởi Cloudlet trong CloudSim và kích thước của các Cloudlet được khởi tạo một cách ngẫu nhiên bằng hàm random của JAVA. Số lượng Cloudlet lần lượt là 20 □ 1000.

Bảng 4.3: Cấu hình thông số các Request

Chiều dài (Length)	Kích thước file (File Size)	Kích thước file xuất ra (Output Size)	Số CPU xử lý (PEs)
3000 ~ 1700	5000 ~ 45000	450 ~ 750	1

- Thuật toán đề xuất được xây dựng bằng cách tạo ra lớp TLRegALoadBalancer kế thừa từ đối tượng VmLoadBalancer, đồng thời thực thi giao diện CloudSimEventListener và điều chỉnh các hàm dựng sẵn để phù hợp với thuật toán đề xuất:

@Override

public int getNextAvailableVm()

// Module 2

public void cloudSimEventFired(CloudSimEvent e)

// Module 1

Tiêu chí đánh giá:

Thực nghiệm mô phỏng cloud với các tham số như trên và chạy thuật toán cân bằng tải của CloudSim có sẵn. Song song đó, chạy thuật toán đề xuất mới cài đặt với cùng đầu vào và so sánh kết quả đầu ra, đặc biệt là thông số thời gian xử lý.

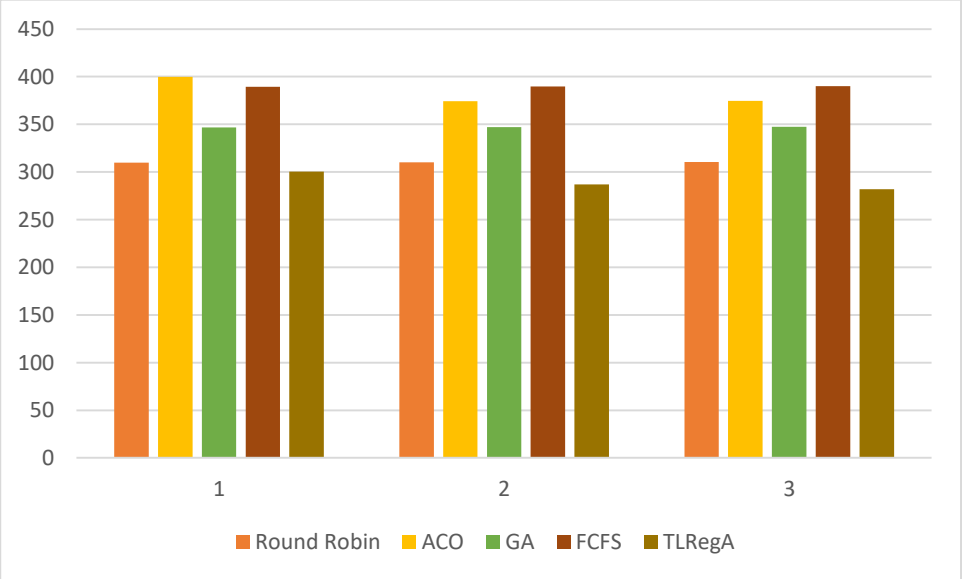
Thời gian di chuyển tác vụ của các máy ảo cũng như thời gian xử lý của cloud với sai số càng thấp thì hiệu quả của thuật toán càng tốt.

4.3. Thực nghiệm và kết quả mô phỏng

Kết quả chạy thực nghiệm mô phỏng trên CloudSim với 1 Datacenter và số lượng các máy ảo lần lượt là 25, 50, 75 được dựng sẵn để đáp ứng các yêu cầu. Các yêu cầu được khởi tạo với chiều dài và kích thước ngẫu nhiên. Thực hiện với các thuật toán Round Robin, ACO, GA và FCFS thì có thời gian thực hiện là:

Bảng 4.4: Kết quả thực nghiệm mô phỏng với 1 DC

Cấu hình Cloud	Số máy ảo trong Datacenter (DC)	Round Robin	ACO	GA	FCFS	TLRegA
CC1	25	309.92	399.8	346.85	389.45	300.35
CC2	50	310.12	374.27	347.16	389.87	286.82
CC3	75	310.38	374.59	347.45	390.09	282.09

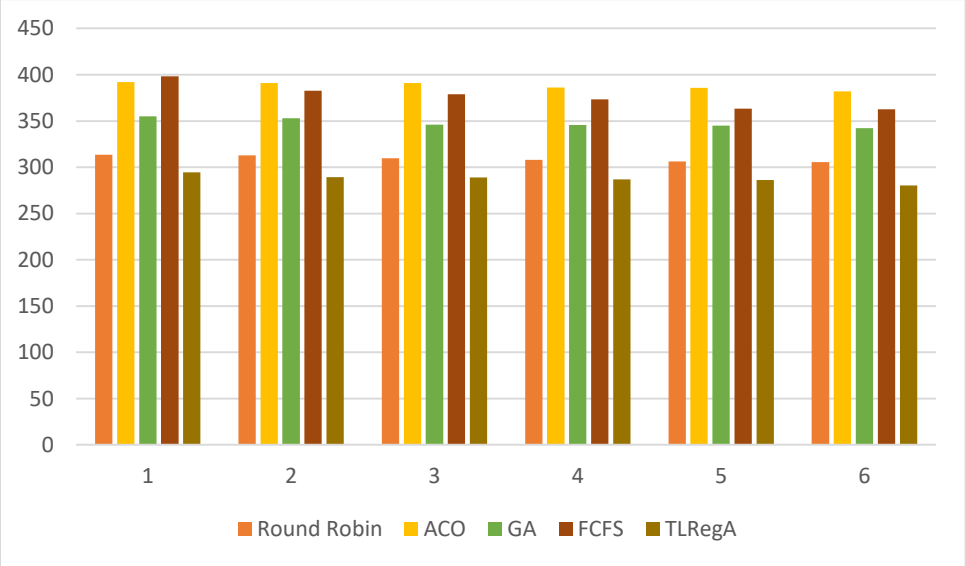


Hình 4.1: Biểu đồ thể hiện hiệu quả của thuật toán đề xuất so với các thuật toán Round Robin, ACO, GA, FCFS sử dụng 1 Datacenter

Kết quả thực nghiệm sử dụng 1 Datacenter với số lượng máy ảo lần lượt là 25, 50, và 75 cho thấy thuật toán đề xuất có thời gian phản hồi tốt hơn so với các thuật toán khác ngay từ bước thực nghiệm đầu tiên.

Bảng 4.5: Kết quả thực nghiệm mô phỏng với 2 DC

Cấu hình Cloud	Số máy ảo trong Datacenter (DC)	Round Robin	ACO	GA	FCFS	TLRegA
CC1	25	313.37	392.02	354.99	398.12	294.66
CC2	50	312.69	390.93	352.87	382.64	289.16
CC3	75	309.54	390.86	345.93	378.73	289.08
CC4	25, 50	308.12	386.32	345.53	373.47	287
CC5	25, 75	306.18	385.67	345.13	363.28	286.29
CC6	75, 50	305.42	382.12	342.27	362.68	280.32



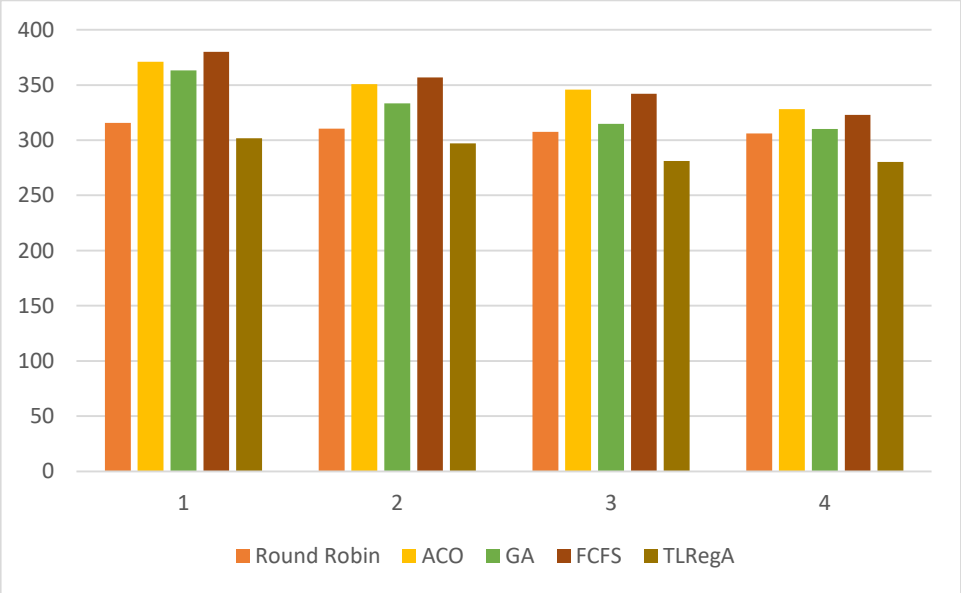
Hình 4.2: Biểu đồ thể hiện hiệu quả của thuật toán đề xuất so với các thuật toán Round Robin, ACO, GA, FCFS sử dụng 2 Datacenter

Từ 2 Datacenter trở đi, thuật toán TLRegA trở nên ổn định và có xu hướng giảm khi thay đổi (tăng) số lượng các máy trong Datacenter. Tương tự như thuật toán TLRegA, các thuật toán khác cũng có thời gian phản hồi tỉ lệ nghịch với số lượng máy ảo (số máy ảo càng tăng thời gian xử lí càng giảm). Đối với môi trường thực nghiệm này, điểm chung giữa các thuật toán là đều có xu hướng giảm khi số lượng

máy ảo tăng. Dù vậy, thuật toán đề xuất vẫn dẫn đầu với thời gian xử lí thấp hơn các thuật toán còn lại.

Bảng 4.6: Kết quả thực nghiệm mô phỏng với 3 DC

Cấu hình Cloud	Số máy ảo trong Datacenter (DC)	Round Robin	ACO	GA	FCFS	TLRegA
CC1	25	315.72	370.93	363.12	379.88	301.87
CC2	50	310.55	350.86	333.46	356.96	297.11
CC3	75	307.48	345.67	314.65	342.04	281.12
CC4	25, 50, 75	306.19	328.12	310.05	322.97	280.36

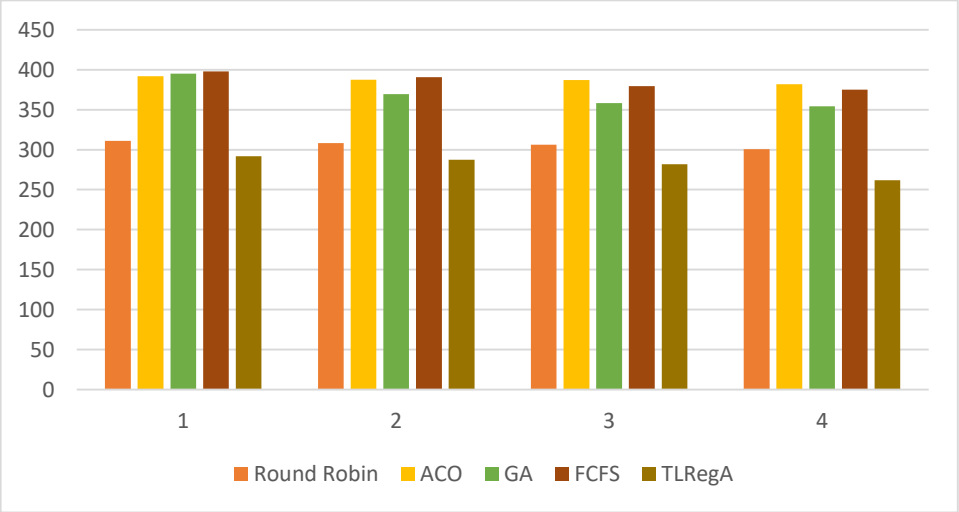


Hình 4.3: Biểu đồ thể hiện hiệu quả của thuật toán đề xuất so với các thuật toán Round Robin, ACO, GA, FCFS sử dụng 3 Datacenter

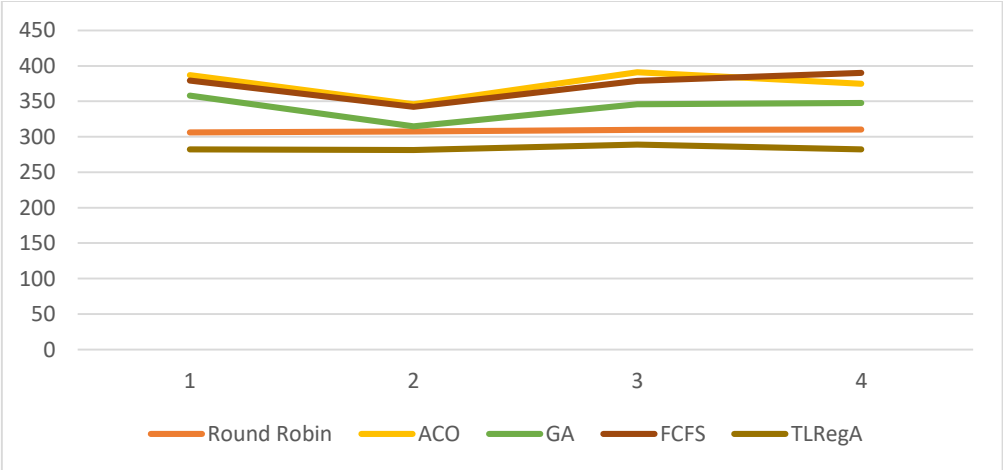
Kết quả thực nghiệm với 3 Datacenter càng làm rõ được mối quan hệ trái ngược nhau giữa số lượng Datacenter và thời gian thực thi tác vụ. Đồng thời, nó cũng thể hiện được tính hiệu quả và ổn định của thuật toán đề xuất.

Bảng 4.7: Kết quả thực nghiệm mô phỏng với 4 DC

Cấu hình Cloud	Số máy ảo trong Datacenter (DC)	Round Robin	ACO	GA	FCFS	TLRegA
CC1	25	310.86	392.06	395.17	397.97	291.81
CC2	50	308.39	387.3	369.3	390.79	287.25
CC3	75	306.03	386.95	358.12	379.37	281.99
CC4	25, 50, 75	300.48	381.75	354.25	375.25	261.72



Hình 4.4: Biểu đồ thể hiện hiệu quả của thuật toán đề xuất so với các thuật toán Round Robin, ACO, GA, FCFS sử dụng 4 Datacenter



Hình 4.5: Biểu đồ thể hiện so sánh thuật toán đề xuất với các thuật toán Round Robin, ACO, GA, FCFS sử dụng 75 máy ảo và các giá trị thay đổi của Datacenter

Lần thực nghiệm cuối cùng với 4 Datacenter, có thể thấy rằng thuật toán đề xuất đã chứng tỏ được năng lực của mình so với các thuật toán còn lại. Qua tất cả các lần thực nghiệm với số lượng Datacenter và máy ảo tương ứng, ta càng thấy rõ hiệu quả và ưu thế của thuật toán TLRegA. Thông qua biểu đồ đường so sánh kết quả thực nghiệm các thuật toán với số lượng máy ảo là 75, có thể nhận thấy thuật toán FCFS và ACO có thời gian tải trễ hơn so với 3 thuật toán còn lại. Trong đó, FCFS chiếm ưu thế hơn ở các cấu hình đầu tiên. Tuy vậy, khi số lượng các máy ảo thay đổi, ACO có xu hướng giảm về mặt thời gian hay nói cách khác là tính hiệu quả cao hơn FCFS.

4.4. Kết luận chương 4

Chương 4 của luận văn trình bày mô hình thực nghiệm mô phỏng, các thông số cũng như kịch bản đưa ra là dựa vào quá trình request của các browser trên môi trường Cloud Analyst. Qua quá trình thực nghiệm nhiều lần với số lượng Datacenter khác nhau, luận văn đã ghi nhận các kết quả, sau đó phân tích và so sánh với các thuật toán cân bằng tải khác và nhận thấy thuật toán đề xuất có tính hiệu quả hơn một số thuật toán còn lại trong nhiều môi trường khác nhau. Từ đó, ghi nhận các thông số về thời gian dự báo di chuyển các tác vụ giữa các máy ảo của thuật toán đề xuất. Việc chạy thực nghiệm mô phỏng với các số lượng Datacenter từ 1 - 4 và số lượng các máy ảo linh hoạt lần lượt là 25, 50 và 75 đã cho thấy kết quả tương đối tốt. Qua đó thấy được việc phân bổ các request đến các máy ảo xử lý khá đồng đều và có tính khả thi cao.

KẾT LUẬN

Luận văn "Đề xuất thuật toán Dự báo thời gian di chuyển tác vụ nhằm nâng cao hiệu năng cân bằng tải trên điện toán đám mây" về cơ bản đã đáp ứng được những mục tiêu ban đầu đề ra. Dựa vào các thuật toán đã có sẵn như FCFS, Round Robin, ACO, GA, luận văn đã phân tích, đánh giá cách thức xây dựng các thuật toán. Từ đó, ta có thể tìm ra được các ưu, nhược điểm của từng thuật toán và đưa ra giải pháp phù hợp. Nhận thấy được những điều còn chưa tốt ở các thuật toán thế hệ trước, luận văn đã đề xuất ra một thuật toán có khả năng cải tiến và nâng cao cân bằng tải một cách tối ưu hơn.

Luận văn sử dụng ba mô hình chính để nghiên cứu tổng quan về đám mây và các đám mây. Trong đó, các kỹ thuật cân bằng tải được áp dụng trong môi trường điện toán đám mây.

Nghiên cứu cách tiếp cận điện toán đám mây thông qua môi trường mô phỏng CloudSim và Cloud Analyst với công cụ giao diện dễ sử dụng và thân thiện với người dùng. Từ đó, cài đặt và mô phỏng các kỹ thuật cân bằng tải cũng như các thuật toán đã được đưa vào so sánh. Các giá trị thu được sẽ dùng để phân tích nhằm đúc kết những mặt lợi thế và hạn chế của mỗi thuật toán. Qua đó, ta sẽ kịp thời định hướng đề xuất một thuật toán với mục đích cao nhất là khắc phục những thiếu sót còn hiện hữu.

Kết quả đạt được từ thuật toán đề xuất đáp ứng được các mục tiêu như việc đáp ứng thời gian được cải thiện, hạn chế của các tài nguyên bị đói, máy ảo có năng lực xử lý mạnh sẽ được xử lý nhiều yêu cầu hơn. Kết quả từ các thực nghiệm mô phỏng cho thấy, thuật toán đề xuất có khả năng đáp ứng được hầu hết các mục tiêu mà chúng ta mong đợi. Cụ thể, TLRegA cải thiện được thời gian đáp ứng, giảm thiểu thực trạng các tài nguyên bị đói cũng như tăng cường năng lực xử lý cho các máy ảo để giải quyết được nhiều hơn yêu cầu của người dùng. Hơn nữa, thuật toán đề xuất

TLRegA giúp việc cân bằng tải được thực hiện hiệu quả hơn so với các thuật toán còn lại: Round Robin, ACO, GA và FCFS.

Thuật toán đề xuất có thể xem xét đưa vào áp dụng trong thực tế.

- Hạn chế luận văn:

+ Vẫn chỉ là nghiên cứu được đề xuất và chưa được ứng dụng vào môi trường thực tế.

+ Thời gian xử lý và đáp ứng được cải thiện hơn so với các thuật toán cũ nhưng chưa nhiều.

- Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu:

+ Tiếp tục nâng cấp thuật toán đề xuất và đưa vào ứng dụng thực tế trong tương lai.

+ Xây dựng biểu đồ phân bổ tải cho cloud dựa trên việc áp dụng mô hình năng lượng của Datacenter hoặc cloud tương ứng.

TÀI LIỆU THAM KHẢO

[1] K. Schwab, "The Fourth Industrial Revolution: what it means, how to respond," the World Economic Forum, [Online]. Available: https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/. [Accessed 15 5 2021].

[2] V. ICT, "Tây Ninh dự kiến chi gần 600 tỷ đồng cho phát triển Chính quyền số," Vietnam ICT News, 1 10 2020. [Online]. Available: https://ictnews.vietnamnet.vn/cuoc-song-so/tay-ninh-du-kien-chi-gan-600-ty-dong-cho-phat-trien-chinh-quyen-so-265531.html.

[3] Admin Globaldots, "13 Key Cloud Computing Benefits for Your Business," Global Dots, [Online]. Available: https://www.globaldots.com/resources/blog/cloud-computing-benefits-7-key-advantages-for-your-business/. [Accessed 1 5 2021].

[4] Danilo Ardagna, Giuliano Casale, Michele Ciavotta, Juan F Pérez and Weikun Wang, "Quality-of-service in cloud computing: modeling techniques and their applications," *Journal of Internet Services and Applications*, vol. 5, no. 11, pp. 2-17, 2014.

[5] Nguyen Xuan Phi, Tran Cong Hung, "Study the effect of Parameters to load balancing in cloud computing," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 8, no. 3, 2016.

[6] Bui Khiet Thanh, Pham Tran Vu, Tran Cong Hung, "A Load Balancing Game Approach for VM Provision Cloud Computing Based on Ant Colony Optimization," *Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 52-63, 2017.

[7] Nguyen Xuan Phi, Le Ngoc Hieu, Tran Cong Hung, "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 công nghệ Thông tin và truyền thông* , vol. 4, no. 1, pp. 43-48, 2018.

[8] Rashmi. K. S, Suma. V, Vaidehi. M, "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud," *Special Issue of International Journal of Computer Applications* , pp. 31-33, 2012.

[9] Agraj Sharma & Sateesh K. Peddoju, "Response time based load balancing in cloud computing," *International Conference on Control, Instrumentation, Communication and Computational Technologies* , 2014.

[10] Rajwinder Kaur, Pawan Luthra, "Load Balancing in Cloud Computing," *Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC*, 2014.

[11] Lazaros Gkatzikis, Iordanis Koutsopoulos, "Migrate or Not? Exploiting Dynamic Task Migration in Mobile Cloud Computing Systems," *World Communication Magazine,* 2013.

[12] Anita Rani and Pankajdeep Kaur, "Migration Jobs in Cloud Computing," *International Journal of Grid Distribution Computing*, vol. 8, no. 6, pp. 151-160, 2015.

[13] Weishan Zhang, Shouchao Tan, Qinghua Lu, Xin Liu, and Wenjuan Gong, "A Genetic-Algorithm-Based Approach for Task Migration in Pervasive Clouds," *International Journal of Distributed Sensor Networks*, pp. 2-11, 2015.

[14] Geetha Megharaj, Mohan Kabadi, "Run Time Virtual Machine Task Migration Technique for Load Balancing in Cloud," *International Journal of Intelligent Engineering & System*, vol. 11, no. 8, pp. 265-275, 2018.

[15] Chen Ling, Weizhe Zhang, Hui He and Yu-chu Tian, "Network perception task migration in cloud-edge fusion computing," *Journal of Cloud Computing: Advances, Systems and Applications*, pp. 9-43, 2020.

[16] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, "CloudSim: A Framework For Modeling And Simulation Of Cloud

Computing Infrastructures And Services," School of Computing and Information Systems, The University of Melbourne, Australia, [Online]. Available: http://www.cloudbus.org/cloudsim/. [Accessed 1 5 2021].

[17] Kumaria, Aparna; Gupta, Rajesh; Tanwar, Sudeep; Kumar, Neeraj;, "Blockchain and AI Amalgamation for Energy Cloud Management: Challenges, Solutions, and Future Directions," *Journal Pre-proof,* 2020.

[18] Al-Mashhadi, Saif; Anbar, Mohammed; Jalal, Rana A.; Al-Ani, Ayman;, "Design of Cloud Computing Load Balance System Based on SDN Technology," *Computational Science and Technology,* pp. 123-135, 2020.

[19] G. Mandal, S. Dam, K. Dasgupta nad P. Dutta, “A Linear Regression-Based Resource Utilization Prediction Policy for Live Migration in Cloud Computing,” trong *Studies in Computational Intelligence*, Springer, 2020, pp. 109-128.

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn/luận án qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng 15 % toàn bộ nội dung luận văn. Bản luận văn kiểm tra qua phần mềm là bản cứng luận văn đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Tây Ninh, ngày 25 tháng 01 năm 2022

HỌC VIÊN CAO HỌC

Nguyễn Hoàng Tấn

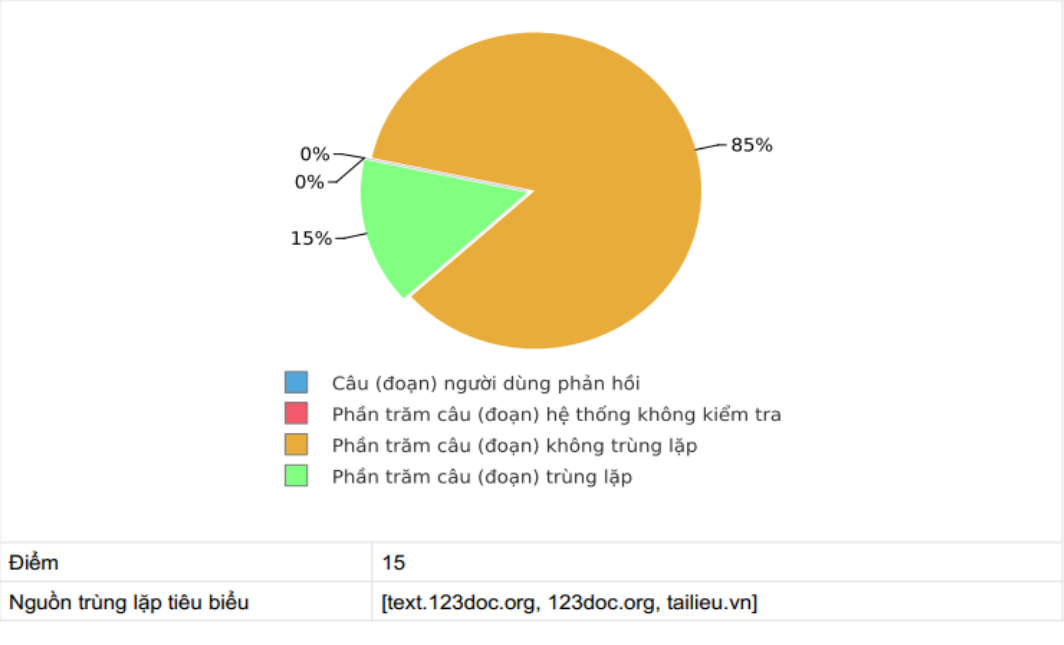
Hệ thống hỗ trợ nâng cao chất lượng tài liệu

KẾT QUẢ KIỂM TRA TRÙNG LẶP TÀI LIỆU

THÔNG TIN TÀI LIỆU

Tác giả	Nguyễn Hoàng Tấn
Tên tài liệu	1. LuanVan_NguyenHoangTan_v5 25012022 - DOIT
Thời gian kiểm tra	09-02-2022, 09:12:54
Thời gian tạo báo cáo	09-02-2022, 09:14:52

KẾT QUẢ KIỂM TRA TRÙNG LẶP



Điểm	15
Nguồn trùng lặp tiêu biểu	[text.123doc.org, 123doc.org, tailieu.vn]

(*) Kết quả trùng lặp phụ thuộc vào dữ liệu hệ thống tại thời điểm kiểm tra

Học viên thực hiện luận văn **Người hướng dẫn khoa học**

Nguyễn Hoàng Tấn **PGS.TS Trần Công Hùng**

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN LUẬN VĂN THẠC SĨ

Họ và tên học viên: **Nguyễn Hoàng Tấn**

Chuyên ngành: **Hệ Thống Thông Tin**

Khóa : **2020- 2022**

Tên đề tài: **Đề xuất thuật toán dự báo thời gian di chuyển tác vụ nhằm nâng cao hiệu năng cân bằng tải trên điện toán đám mây**

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

Ngày bảo vệ: **15/01/2022**

Các nội dung học viên đã sửa chữa, bổ sung trong luận văn theo ý kiến đóng góp của Hội đồng chấm luận văn:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Bổ sung cơ sở lý luận để đề xuất thuật toán	Học viên đã bổ sung và giải thích tại phần mở đầu, mục 1: Tính cấp thiết của đề tài.
2	Bổ sung bình luận về kết quả thực nghiệm, giải thích vì sao thuật toán đề xuất hiệu quả hơn các thuật toán khác.	Học viên đã bổ sung và giải thích ở phần kết luận chương 4 trang 39.

Tp.HCM, ngày 25 tháng 01 năm 2022

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM LUẬN VĂN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
TS. Tân Hạnh	TS. Huỳnh Trọng Thưa	PGS.TS. Trần Công Hùng	Nguyễn Hoàng Tấn