HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Huỳnh Phi Long

ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY THÔNG QUA HÀNH VI NGƯỜI DÙNG CLOUD

LUẬN VĂN THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Huỳnh Phi Long

ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY THÔNG QUA HÀNH VI NGƯỜI DÙNG CLOUD

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. HỒ CHÍ MINH – 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

Huỳnh Phi Long

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện luận văn, ngoài nỗ lực riêng có 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 thuận lợi 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

Huỳnh Phi Long

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
CNTT		Công nghệ thông tin
CC	Cloud Computing	Điện toán đám mây
Cloud	Cloud computing environment	Môi trường điện toán đám mây
ML	Machine Learning	Máy học
LB	Load Balancing	Cân bằng tải
AI	Artificial Intelligence	Trí tuệ nhân tạo
QoS	Quality of Service	Chất lượng dịch vụ
VM	Virtual Machine	Máy ảo
IaaS	Infrastructure as a Service	Cơ sở hạ tầng như dịch vụ
PaaS	Platform as a Service	Nền tảng như là dịch vụ
SaaS	Software as a Service	Phần mềm như là dịch vụ
DLB	Dynamic Load Balancing	Cân bằng tải động

MỤC LỤC

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Với sự phát triển của khoa học công nghệ, đặc biệt về lĩnh vực công nghệ thông tin – viễn thông, thông qua việc ứng dụng sự phát triển của công nghệ máy tính và dựa vào mạng Internet cho phép người dùng có thể lưu trữ, trích xuất dữ liệu từ nhiều nơi khác nhau và có thể truy cập qua mạng các dịch vụ công nghệ từ một nhà cung cấp nào đó, kết hợp sử dụng các công nghệ điện toán (song song, phân tán, ảo hóa...) gọi là "Điện toán đám mây" (ĐTĐM – Cloud Computing).

Trong những năm gần đây, sự tăng trưởng nhanh chóng của số lượng thiết bị đầu cuối di động làm phát sinh xu hướng không thể đảo ngược của việc ứng dụng rộng rãi điện toán đám mây. Sự chuyển dịch của điện toán đám mây từ thị trường máy tính để bàn sang thị trường di động trở thành hướng phát triển chính. Tuy nhiên, dường như có nhiều vấn đề phức tạp. Trong số đó, ba khía cạnh của "người dùng - môi trường - dịch vụ" là đặc biệt nổi bật. Việc phân loại rõ ràng hành vi của người dùng khi sử dụng các dịch vụ đám mây đã trở thành một vấn đề quan trọng cần được giải quyết khẩn cấp.

Hiện nay, việc thực hiện cân bằng tải trong điện toán đám mây thông qua hành vi người dùng Cloud là một thách thức lớn đối với các nhà nghiên cứu và nhà cung cấp dịch vụ đám mây. Việc thiết lập một thuật toán cân bằng tải hiệu quả đáp ứng được hiệu năng hệ thống và làm thế nào sử dụng nguồn tài nguyên điện toán đám mây một cách có hiệu quả nhất là mục đích cuối cùng của điện toán đám mây muốn đạt đến. Ở nước ta hiện nay, các công trình nghiên cứu về cân bằng tải trong điện toán đám mây thông qua hành vi người dùng Cloud Computing cũng còn ít, hạn chế. Vì vậy, luận văn **"Đề xuất thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi người dùng Cloud"** sẽ đi sâu nghiên cứu các kỹ thuật cân bằng tải đang được áp dụng hiện nay; đồng thời đề xuất cải tiến một kỹ thuật cân bằng tải và mô phỏng điện bằng toán đám mây CloudSim. Luận văn bao gồm 03 phần:

1. Phần mở đầu
2. Phần nội dung gồm:

- Chương 1: Giới thiệu tổng quan về hệ thống cân bằng tải của điện toán đám mây.

- Chương 2: Các Công trình nghiên cứu liên quan.

- Chương 3: Đề xuất Thuật toán Cân bằng tải trên điện toán đám mây thông qua hành vi người dùng Cloud.

- Chương 4: Mô phỏng, Thực nghiệm.

3. Phần Kết luận

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, để có thể 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; Đồng thời 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 thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi người dùng Cloud”.

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 các thuật toán cân bằng tải trên điện toán đám mây.

- Tìm hiểu khả năng xảy ra quá tải, tài nguyên phân bổ không đồng đều, máy chủ quá tải và ngưng hoạt động.

- Nghiên cứu về hành vi người dùng Cloud.

- Đề xuất thuật toán có thể sử dụng tài nguyên hiệu quả hơn, tiết kiệm năng lượng.

- Trên cơ sở lý thuyết đã nghiên cứu, Luận văn đề xuất thuật toán nâng cao hiệu quả cân bằng tải của điện toán đám mây. Đánh giá hiệu quả của đề xuất cải tiến này trong môi trường mô hình và mô phỏng điện toán đám mây CloudSim; đồng thời nghiên cứu hướng tiếp cận mới về điện toán đám mây thông qua môi trường CloudSim...

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

- Đối tượng nghiên cứu chính là thuật toán nâng cao hiệu quả cân bằng tải trên điện toán đám mây thông qua hành vi của người dùng Cloud (người dùng cá nhân – User, doanh nghiệp – Enterprise, tổ chức – Organization)

- Nghiên cứu các thuật toán cân bằng tải hiện đang sử dụng.

Phạm vi nghiên cứu

Phạm vi nghiên cứu trong Cloud:

- Xây dựng mô hình mô phỏng đám mây ở mức độ nhỏ: khoảng 10 – 30 máy ảo.

- Độ phức tạp trên mỗi máy ảo chỉ ở mức độ thấp: dưới 10 ứng dụng chạy trên trên các máy ảo.

- Yêu cầu (Request) gửi về máy chủ cũng đơn giản, đánh giá chính xác phải là hành động của người dùng Cloud.

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 bị 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 nó là một mô hình điện toán kinh doanh mới. Hiện tại, vẫn chưa có định nghĩa thống nhất về điện toán đám mây. Theo 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, 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 là một ý tưởng đang phát triển trong thế giới CNTT, được sinh ra từ nhu cầu sử dụng máy tính khi đang di chuyển. Nó mang lại cho người dùng quyền truy cập vào dữ liệu, ứng dụng và bộ nhớ không được lưu trữ trên máy tính của họ. Để có một cái nhìn tổng quan về điện toán đám mây rất đơn giản, nó có thể được hiểu là một hệ thống phân phối cung cấp điện toán giống như cách một lưới điện cung cấp điện. Đối với người dùng máy tính bình thường, nó mang lại lợi thế là cung cấp CNTT mà người dùng không cần phải có kiến thức chuyên sâu về công

nghệ. Tương tự như cách một người tiêu dùng có thể tiếp cận điện năng mà không cần phải là một thợ điện. Cụ thể hơn, trong mô hình điện toán đám mây, tất cả các tài nguyên, thông tin và phần mềm 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 người dùng sử dụng dịch vụ như cơ sở dữ liệu, website, lưu trữ, … trong mô hình điện toán đám mây không cần quan tâm đến vị trí địa lý cũng như các thông tin khác của hệ thống mạng đám. 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 (di động) 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 trung tâm dữ liệu – hay còn gọi là "Data Center". Đây là một thuật ngữ chỉ khu vực chứa server và các thiết bị lưu trữ, bao gồm nguồn điện và các thiết có khả năng sẵn sàng và độ ổn định cao. Bên cạnh đó 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 bao gồm: public (công cộng), private (riêng tư), và hybrid ("lai" giữa đám mây công cộng và riêng tư). Trong một đám mây công cộng, nhà cung cấp bên thứ ba cung cấp một loạt các dịch vụ cho công chúng qua internet. Dữ liệu từ một số khách hàng công ty hoặc cá nhân có thể dùng chung một máy chủ. Về nguyên tắc, đám mây riêng cũng tương tự như vậy, nhưng được thiết lập sau tường lửa và chỉ cung cấp các dịch vụ được lưu trữ cho một số lượng hạn chế người dùng được phê duyệt. Đám mây "lai" là môi trường đám mây mà kết hợp cung cấp các dịch vụ công cộng và riêng tư. Ngoài ra còn có "Community Cloud" là đám mây giữa các nhà cung cấp dịch vụ đám mây.

Điện toán đám mây bao gồm ba loại dịch vụ điện toán riêng biệt được phân phối từ xa tới khách hàng thông qua internet [1]. Khách hàng thường trả phí dịch vụ hàng tháng hoặc hàng năm cho nhà cung cấp để có quyền truy cập vào các hệ thống cung cấp phần mềm dưới dạng dịch vụ (SaaS), nền tảng dưới dạng dịch vụ (PaaS) và cơ sở hạ tầng như dịch vụ (IaaS) cho người đăng ký. Khách hàng đăng ký dịch vụ điện toán đám mây có thể gặt hái nhiều lợi ích, tùy thuộc vào nhu cầu kinh doanh cụ thể của họ tại một thời điểm nhất định. Những ngày đầu tư vốn lớn vào phần mềm và

cơ sở hạ tầng CNTT giờ đã trở thành dĩ vãng đối với bất kỳ doanh nghiệp nào chọn áp dụng mô hình điện toán đám mây để mua sắm các dịch vụ CNTT. Khả năng tiếp cận các nguồn lực CNTT mạnh mẽ trên cơ sở gia tăng đang san bằng sân chơi cho các tổ chức vừa và nhỏ, cung cấp cho họ các công cụ và công nghệ cần thiết để cạnh tranh trên thị trường toàn cầu mà không cần đầu tư trước đây vào các nguồn lực CNTT tiền đề. Những khách hàng đăng ký dịch vụ điện toán được cung cấp qua "cloud" có thể giảm đáng kể chi phí dịch vụ CNTT cho tổ chức của họ; và có quyền truy cập vào các dịch vụ tính toán cấp doanh nghiệp linh hoạt và nhanh nhẹn hơn, trong quá trình này.

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

Saas (Software as a Service) cung cấp cho khách hàng khả năng sử dụng các ứng dụng phần mềm từ xa thông qua trình duyệt web internet. Phần mềm như một dịch vụ còn được gọi là "phần mềm theo yêu cầu". Khách hàng có thể truy cập các ứng dụng SaaS từ mọi nơi thông qua web vì các nhà cung cấp dịch vụ lưu trữ các ứng dụng và dữ liệu liên quan của họ tại vị trí của họ. Lợi ích chính của SaaS là chi phí sử dụng thấp hơn, vì phí thuê bao đòi hỏi một khoản đầu tư nhỏ hơn nhiều so với những gì thường gặp trong mô hình phân phối phần mềm truyền thống. Hầu như có thể loại bỏ phí cấp phép, chi phí cài đặt, phí bảo trì và phí hỗ trợ liên quan đến mô hình cung cấp phần mềm truyền thống bằng cách đăng ký mô hình phân phối phần mềm SaaS. Ví dụ về SaaS bao gồm: Ứng dụng Google và các ứng dụng email dựa trên internet như Yahoo! Mail, Hotmail và Gmail.

PaaS (Platform as a Service) cung cấp cho khách hàng khả năng phát triển và xuất bản các ứng dụng tùy chỉnh trong môi trường được lưu trữ thông qua web. Nó đại diện cho một mô hình mới để phát triển phần mềm đang tăng nhanh chóng về mức độ phổ biến của nó. Một ví dụ về PaaS là Salesforce.com. PaaS cung cấp một khuôn khổ để phát triển, thử nghiệm, triển khai và bảo trì phần mềm nhanh trong một môi trường tích hợp. Giống như SaaS, lợi ích chính của PaaS là chi phí sử dụng thấp hơn, vì phí thuê bao yêu cầu đầu tư nhỏ hơn nhiều so với những gì thường gặp phải khi triển khai các công cụ truyền thống để phát triển, thử nghiệm và triển khai phần mềm. Các nhà cung cấp PaaS xử lý việc bảo trì nền tảng và nâng cấp hệ thống, dẫn

đến giải pháp hiệu quả hơn và tiết kiệm chi phí cho việc phát triển phần mềm doanh nghiệp.

IaaS (Infrastructure as a Service) 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.

Bằng cách cung cấp điện toán, lưu trữ và các ứng dụng này dưới dạng dịch vụ mà không phải là sản phẩm, đám mây mang lại lợi thế kinh doanh và chi phí. Đám mây di chuyển tất cả các dịch vụ này ra bên ngoài trang web đến một nhà thầu hoặc một cơ sở tập trung. Tập trung dữ liệu cho phép chia sẻ chi phí giữa tất cả người dùng. Đám mây hoàn thành những gì CNTT luôn tìm kiếm; tăng khả năng tính toán mà không cần phải cung cấp cơ sở hạ tầng mới. Các ứng dụng có thể có của điện toán đám mây là theo cấp số nhân. Người dùng giao diện với đám mây thông qua trình duyệt web của họ, loại bỏ nhu cầu cài đặt nhiều ứng dụng phần mềm.

Hình 1.1: Mô hình điện toán đám mây [2]

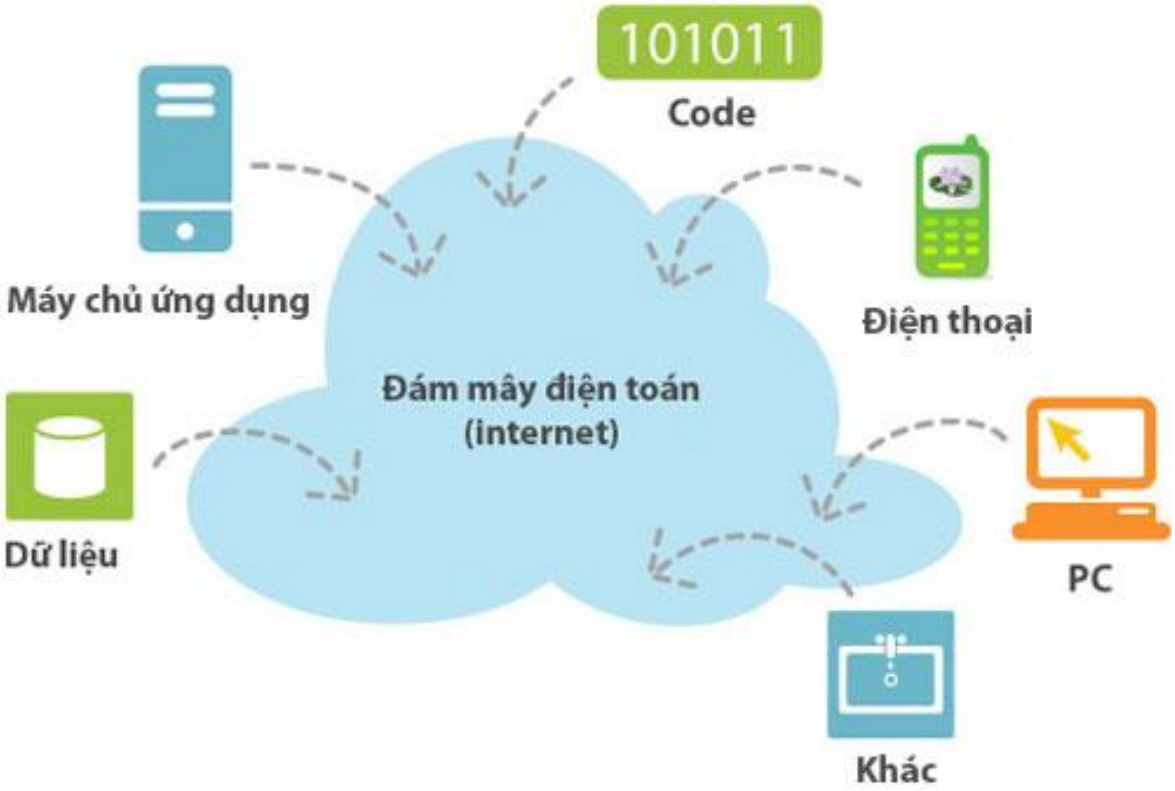
Điện toán đám mây là một mô hình điện toán nơi mà mọi giải pháp liên quan đến CNTT đều được cung cấp dưới dạng các dịch vụ qua mạng Internet, giúp người dùng thoát khỏi việc phải đầu tư không ít nhân lực, công nghệ và hạ tầng để triển khai hệ thống. Qua đó tối giản chi phí cũng như thời gian triển khai, 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 của họ. Chính vì vậy, lợi ích mà 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. Bên cạnh đó, điện toán đám mây [2], [3] cũng đang là xu hướng được phát triển mạnh mẽ hiện nay. Kế thừa các mạng lưới trước đây cùng vớicác 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 cho phép kết thúc sử dụng dịch vụ, giải phóng tài nguyên dễ dàng, giảm thiểu các giao tiếp với nhà cung cấp. Theo đó, mô hình chính là cho phép sử dụng dịch vụ theo yêu cầu; 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; với tài nguyên tính toán động, phục vụ nhiều người năng lực tính toán phần mềm dẻo, đáp ứng nhanh với nhu cầu từ thấp đến cao.

Điện toán đám mây được dựa trên công nghệ ảo hóa [4], 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. Ngoài ra, điện toán đám mây [5] còn 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 nhanh, gọn, nhẹ. Cung cấp các dịch vụ về cơ sở hạ tầng, nền tảng phần mềm, các dịch vụ theo yêu cầu người dùng thông qua Internet.

Điện toán đám mây [6] là một mô hình dịch vụ công nghệ thông tin, kế thừa các mạng lưới trước đây trên thế giới giúp người dùng truy cập tài nguyên dữ liệu, lưu trữ đến hệ thống quản lý, xử lý dữ liệu phức tạp của các hệ thống như Google, Facebook… Người dùng chỉ truy cập vào thiết bị đầu cuối để truy xuất vào các tài nguyên trên điện toán và bên trong hệ thống điện toán sẽ lập lịch xử lý các yêu cầu trên bao gồm xử lý thời gian chờ, thời gian xử lý tín hiệu đến thời gian hoàn thành nhiệm vụ. Chính vì vậy mà điện toán đám mây [7] đang chuyển đổi ngành CNTT, thay đổi cách thức sử dụng và cung cấp phần mềm và phần cứng. Làm cho việc sử dụng các tài nguyên máy tính theo yêu cầu như băng thông, lưu trữ hoặc các ứng dụng phần mềm và điện toán có sẵn. 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.

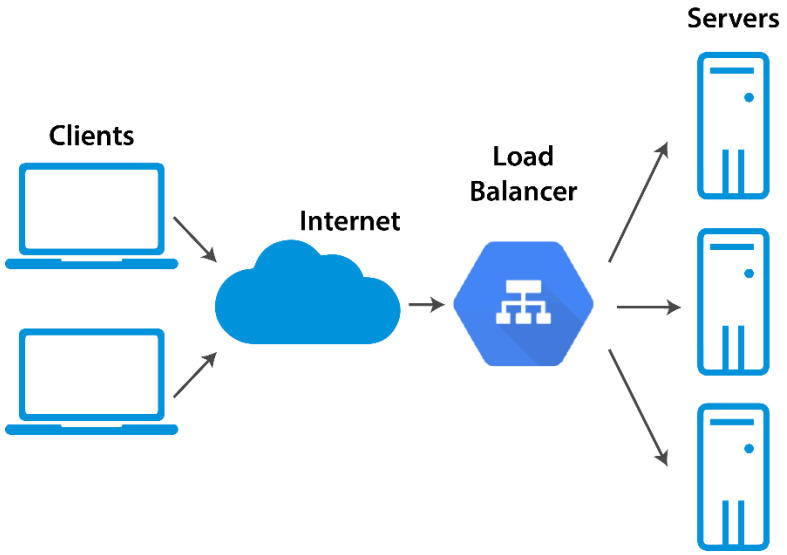
Điện toán đám mây là một kiểu mẫu mới [8], [9] và đang không ngừng tiến hóa trong tính toán. Cơ chế cân bằng tải được chia thành các nguồn lực và cung cấp các nguồn lực cùng với nhiệm vụ lập kế hoạch giữa các hệ thống phân phối. Trong cân bằng tải truyền thống phải đối mặt với một số vấn đề khác nhau 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 và bản chất của môi trường đám mây. Ngày nay [10], 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 trong thế giới này. 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, được gọi là

dịch vụ trả tiền cho mỗi lần sử dụng, làm cho ngành công nghiệp công nghệ thông tin hướng tới việc kinh doanh điện toán đám mây. Giống như một CPU với nhiều lỗi, những doanh nghiệp sở hữu một cụm của các CPU/Máy vật lý đó được gọi là đám mây. Các cụm có một số lượng hữu hạn không gian và bộ nhớ.



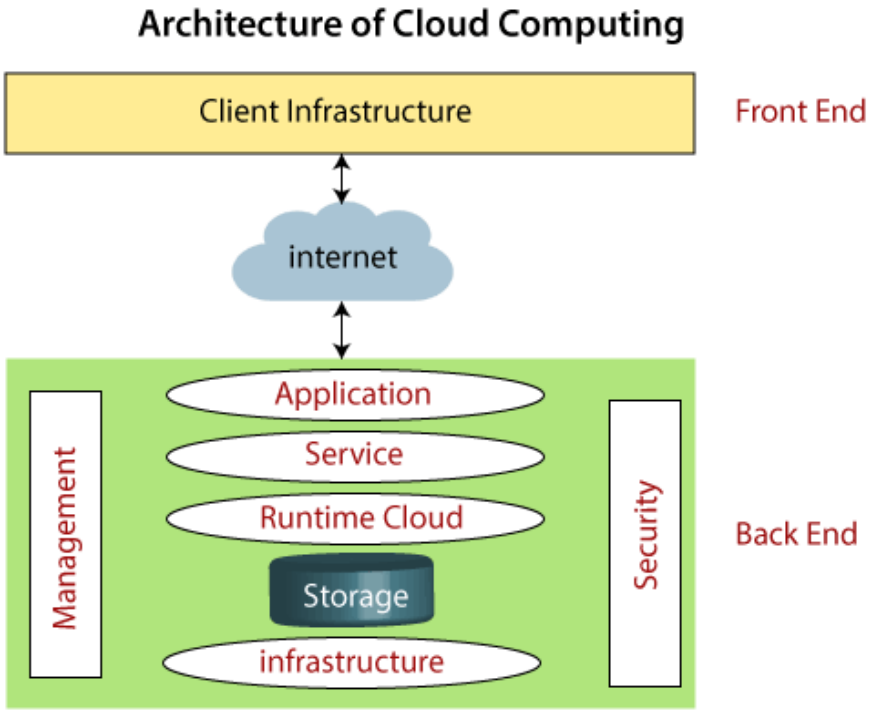
Hình 1.2: Cung cấp tài nguyên đám mây [5]

Người dùng sẽ nhận được không gian và bộ nhớ trong một khoảng thời gian từ cụm được phân bổ khi trả tiền cho dịch vụ này. Khi người dùng đòi hỏi các nguồn lực bao gồm bộ nhớ, không gian và băng thông 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) hay còn được gọi là máy ảo (VM). Lúc này, bộ cân bằng tải sẽ quản lý ghép kênh các tài nguyên theo đúng với yêu cầu từ người dùng.



Hình 1.3: Cân bằng tải trong điện toán đám mây [6]

Những 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 tuy nhiên nó lại không cung cấp đúng chất lượng dịch vụ. Các QoS có thể 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, 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.



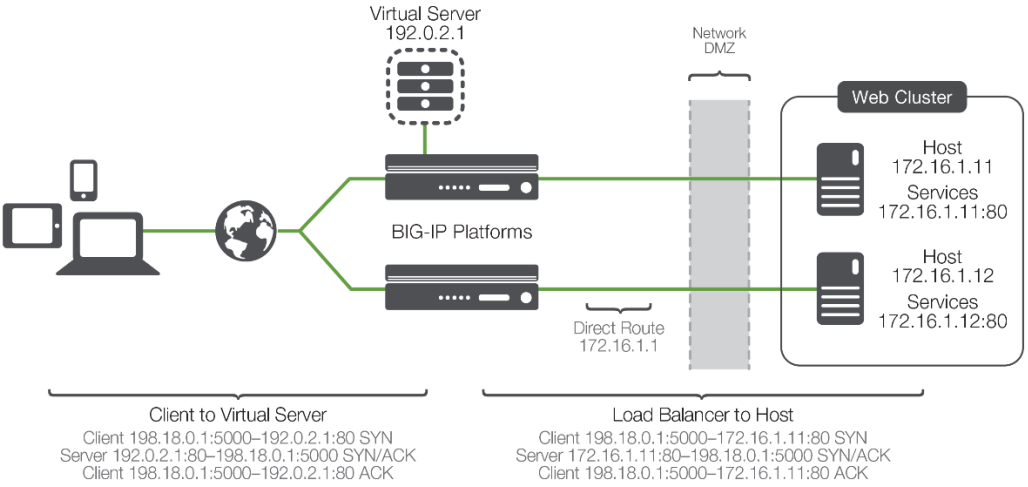
Hình 1.4: Kiến trúc của điện toán đám mây [8]

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

1.2.1. Giới thiệu về cân bằng tải

Hiện nay, lĩnh vực CNTT đang phát triển ngày càng mạnh mẽ và nhu cầu về tài nguyên lưu trữ và tính toán tăng trưởng không kém. Với lượng lớn dữ liệu được tạo và trao đổi qua mạng ngày một tăng lên thì sự đòi hỏi thêm về nhu cầu tài nguyên máy tính ngày càng nhiều. Đám mây đã giúp các doanh nghiệp tận dụng lợi ích của tài nguyên điện toán được chia sẻ trên môi trường ảo hóa, cùng với đó rất nhiều doanh nghiệp đã sử dụng các dịch vụ dựa trên đám mây ở dạng này hoặc dạng khác. Điều này đưa chúng ta đến khái niệm cân bằng tải trong điện toán đám mây.

Kéo theo sự phát triển rộng khắp của Internet, các website hay các ứng dụng trực tuyến đang được rất nhiều người truy cập và sử dụng. Lượng truy cập này quá lớn thường dẫn đến các vấn đề về hạ tầng mạng, ngoài ra khả năng xử lý của Server sẽ bị tắc nghẽn cục bộ. Chính vì vậy, cân bằng tải luôn 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 các tài nguyên. Giải pháp của cân bằng tải chính 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. Với phương pháp này, hệ thống sẽ 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, đẩy thời gian uptime (thời gian hoạt động) 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 [9]

Trong môi trường phân tán, cân bằng tải là một trong những chủ đề hết sức quan trọng. Vì điện toán đám mây được coi 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 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 thuật toán để tăng cường nhằm tìm ra những phương án tốt nhất cho cân bằng tải.

Trong cân bằng tải thời gian gần đây đã nổi lên chủ đề phân tán dự đoán quá tải [11] 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, vì sẽ không có đủ dòng chảy đồng thời xảy ra đi cùng 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 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 sử dụng thông tin 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 làm giảm đáng kể sự phức tạp mà 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, 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, 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 [12]. Thuật toán lập lịch cân bằng tải tĩnh thường bao gồm round robin, weighted rounded robin [13]. 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 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 dễ sử dụng. Nhưng trong các trung tâm dữ liệu đám mây quy mô lớn có tính không đồng nhất của tài nguyên và nhu cầu người sử dụng là không nhất quán, hiệu quả cân bằng tải tĩnh không được lý tưởng. Cân bằng tải động (DLB - Dynamic Load Balancing), nó chủ yếu được sử dụng trong lĩnh vực phân phối máy tính song song, và 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 mà một số các nút máy tính bị quá tải và một số nút có tải nhẹ và do đó 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 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. Vì

vậy, những cách phân phối tiếp cận thường có được sự tối ưu cục bộ của các giải pháp. Và hiệu quả của việc giải quyết vấn đề phân phối tải trong một số trường hợp đặc biệt không phải là lý tưởng. Vì vậy, nó có thể đảm bảo cân bằng tải và sử dụng hiệu quả tài nguyên vật lý của toàn bộ cụm. Tuy nhiên, cân bằng tải 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 và nâng cao hiệu quả hoạt động của các trung tâm dữ liệu điện toán đám mây.

Theo tài liệu [14], Cân bằng tải có thể được chia thành 2 loại là: Cân bằng tải cục bộ và cân bằng tải toàn cục. 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 đáp ứng nhu cầu. Về 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. Nó 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 là 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 tiếp tục hoạt động như mong đợi.

1.2.2. Mục đích cân bằng tải

Cân bằng tải có một số mục đích chính sau đây:

- Tăng khả năng đáp ứng, tránh tình trạng quá tải trên máy chủ, 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 khả dụng cao (HA - High Availability) cho hệ thống, đồng thời 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, 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, 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 là một lĩnh vực liên quan đến chuyên ngành khoa học máy tính và công nghệ thông tin, bản chất của trí tuệ nhân tạo vẫn do con người làm ra, họ xây dựng các thuật toán, lập trình bằng các công cụ phần mềm công nghệ thông tin, giúp các máy tính có thể tự động xử lý các hành vi thông minh như con người. Trí tuệ nhân tạo có khả năng tự thích nghi, tự học và tự phát triển, tự đưa ra các lập luận để giải quyết vấn đề, có thể giao tiếp như người…tất cả là do AI được cài một cơ sở dữ liệu lớn, được lập trình trên cơ sở dữ liệu đó và tái lập trình trên cơ sở dữ liệu mới sinh ra. Cứ như vậy cấu trúc của AI luôn luôn thay đổi và thích nghi trong điều kiện và hoàn cảnh mới.

1.4. Tổng quan về học máy (ML)

Học máy (Machine Learning/ML) là một phương pháp để tạo ra AI. ML 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. 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 sau đó 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. Người dùng cloud và hành vi người dùng cloud

Việc xác định hành vi người dùng có vai trò rất quan trọng trong nghiên cứu này. Việc xác định hành vi người dùng thông qua đoạn request họ gửi lên cloud, với từng loại người dùng với những mục đích sử dụng khác nhau thì cấu trúc của những đoạn request này cũng sẽ khác nhau.

Vậy nên dựa vào sự khác nhau trong cấu trúc, thông tin của các request được gửi trên Cloud ta có thể dự đoán những tác vụ mà người dùng sẽ yêu cầu thực hiện khi kết nối với Cloud. Từ đó có sự phân bổ tài nguyên cho các người dùng hợp lý, tránh việc lãng phí tài nguyên.

1.6. Kết luận chương 1

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, gói tin mất mát khi truyền dữ liệu qua môi trường điện toán, mục đích tăng hiệu quả cân bằng của hệ thống.

CHƯƠNG 2. CÁC CÔNG TRÌNH LIÊN QUAN

2.1. Giới thiệu chung

Chương này sẽ giới thiệu các công trình nghiên cứu của Việt Nam và trên thế giới có liên quan, ứng dụng vào đề tài. Những công trình này đã đề xuất nhiều kỹ thuật phân bổ tài nguyên, phát hiện hành vi người dùng, cân bằng tải,... trong cloud. Qua việc nghiên cứu các công trình này cũng góp phần củng cố thêm cơ sở lý thuyết và định hình hướng nghiên cứu cho đề tài.

2.2. Các công trình nghiên cứu tại Việt Nam

Trong bài báo [15] của Trần Công Hùng và các cộng sự đăng trên tạp chí Khoa học công nghệ Thông tin và truyền thông số 04 (CS.01) 2018 của Học viện Công nghệ Bưu chính viễn thông, đã đề xuất một thuật toán cân bằng tải nhằm giảm thời gian đáp ứng trên điện toán đám mây, ý tưởng chính là sử dụng thuật toán dự báo ARIMA để dự báo thời gian đáp ứng, từ đó đưa ra cách giải quyết phân phối tài nguyên hiệu quả dựa vào giá trị ngưỡng thời gian. Bài báo đã đưa ra thuật toán, thử nghiệm mô phỏng với mô hình nhỏ và đã đạt được một số kết quả mô phỏng khá tích cực, và tiềm năng trong dự báo tương lai gần.

Trong bài báo [16] của tác giả Nguyễn Thanh Thủy và các cộng sự đăng trên tạp chí "International Journal of Computer Science and Network, Volume 4, Issue 2, April 2015", đã trình bày một cách tiếp cận để cải thiện thuật toán ngăn chặn bế tắc, để lên lịch cho các chính sách cung cấp tài nguyên để phân bổ tài nguyên không đồng nhất. Thuật toán ngăn chặn bế tắc có độ phức tạp thời gian chạy là O (min (m, n)), trong đó m là số lượng tài nguyên và n là số lượng quy trình. Họ đề xuất thuật toán phân bổ nhiều tài nguyên cho các dịch vụ cạnh tranh đang chạy trong các máy ảo trên nền tảng phân tán không đồng nhất. Các thí nghiệm cũng so sánh hiệu suất của phương pháp đề xuất với các công việc liên quan khác.

Trên tạp chí Innovative Technology and Interdisciplinary Sciences năm 2018 [17] nhóm tác giả Ha Huy Cuong Nguyen, cũng đã công bố quốc tế "Avoid Deadlock Resource Allocation (ADRA) Model V VM-out-of-N PM". Trong bài báo này, các tác giả đã chú trọng hơn về cloud, và mô hình hoạt động của cloud, từ đó đề xuất ra thuật toán ADRA, trong đó xây dựng mô hình V, có nghĩa là VM-out-of-N, tức là các

máy ảo đã hết khả năng xử lý. Ở bài nghiên cứu này, các tác giả đã giải thích rõ sự phát triển của cloud trên nền tảng grid, và đặc trưng nhất là sự đa dạng người dùng rất vô tận nhưng tài nguyên thì có giới hạn. Và Deadlock xảy ra trên cloud là ở mức lớn, tức là deadlock trên cloud lớn hơn tất cả deadlock trước giờ vốn có. Trong bài viết này, một ý tưởng mới được phát triển dựa trên vùng không gian trống trên cloud, nhằm tránh deadlock, thông qua việc biết các tài nguyên trống từ các người dùng đã được phục vụ. Thuật toán mới đề xuất là phân bổ các tài nguyên tới các dịch vụ đang được phục vụ trong các máy ảo trên môi trường cloud không đồng nhất. Trong nghiên cứu này, được thực nghiệm mô phỏng trên cloudSim và kết quả thu được tương đối tốt.

2.3. Một số công trình nghiên cứu trên thế giới

Trên tạp chí Journal of Internet Technology Volume năm 2019 [18], nhóm tác giả Ruoshui Liu, Xin Wang, Juan Du, Ping Xie, cũng đã công bố quốc tế "A Cloud User Behavior Authentication Model Based on Multi-label Hyper-network". Với sự ra đời của kỷ nguyên dữ liệu lớn, việc bảo mật thông tin người dùng là đặc biệt quan trọng. Làm thế nào để xây dựng lòng tin giữa người dùng và đám mây là một vấn đề quan trọng. Để giải quyết vấn đề này, bài báo này đề xuất một mô hình xác thực hành vi người dùng đám mây dựa trên mạng đa cấp nhãn, thực hiện phân chia chi tiết hành vi của người dùng và cải thiện độ chính xác của phát hiện bất thường. Phương pháp này huấn luyện cơ sở dữ liệu hành vi bình thường của người dùng thành một siêu mạng, thêm hành vi của người dùng hiện tại như một thể hiện vào siêu mạng để phân loại. Nếu một nhãn được tìm thấy thành công trong phân loại này, nhãn đó được xác định là người dùng bình thường. Nếu không, mô hình cập nhật trọng số của siêu mạng, thay thế siêu cạnh và tìm kiếm lại nhãn. Nếu nhãn được tìm thấy, nó được xác định là người dùng rủi ro, nếu không nó được xác định là người dùng độc hại. Kết quả mô phỏng cho thấy có sự cải thiện đáng kể về độ chính xác của phân loại. Áp dụng phương pháp trong bài báo này để phát hiện hành vi của người dùng có thể cải thiện hiệu quả tỷ lệ phát hiện hành vi của người dùng, thực hiện phân tích chi tiết về hành vi của người dùng và cải thiện khả năng xử lý hành vi của người dùng. Trong những năm gần đây, với sự phát triển của công nghệ thông tin và sự hội nhập của nền kinh tế thế giới, nhân loại đã bước vào kỷ nguyên dữ liệu lớn. Sự phát triển nhanh chóng của số lượng thiết bị đầu cuối di động làm phát sinh xu hướng không thể đảo

ngược của việc ứng dụng rộng rãi điện toán đám mây di động. Chuyển dịch điện toán đám mây từ thị trường máy tính để bàn sang thị trường di động trở thành hướng phát triển chính. Tuy nhiên, dường như có nhiều vấn đề phức tạp. Trong số đó, ba khía cạnh của "người dùng-môi trường-dịch vụ" là đặc biệt nổi bật. Thiết bị đầu cuối thông minh di động truy cập thông tin từ các dịch vụ đám mây di động thông qua Internet di động hoặc Internet of Things để có thể cung cấp nhiều loại dịch vụ tích hợp khác nhau. Việc cung cấp dịch vụ đầu cuối xanh, đáng tin cậy và kịp thời là chức năng cốt lõi của các dịch vụ đám mây di động. Theo quan điểm của người dùng, việc thiết lập mối quan hệ tin cậy giữa người dùng và đám mây trở thành điều tối quan trọng. Việc phân loại tốt các hành vi của người dùng trước khi dịch vụ đám mây đi vào quá trình cung cấp dịch vụ thực chất đã trở thành một vấn đề quan trọng cần được giải quyết khẩn cấp. Bài toán phân loại hành vi người dùng thực sự là tập con của bài toán phân loại. Thuật toán phân loại cũng là một hướng nghiên cứu quan trọng trong các lĩnh vực nhận dạng mẫu, khai thác dữ liệu và phát hiện dị thường, và nó đã thu hút được nhiều sự quan tâm. Các đặc tính của tập dữ liệu được sử dụng để xây dựng một bộ phân loại trong bài toán phân loại, và sau đó bộ phân loại này được sử dụng để gán các danh mục cho các danh mục đối tượng chưa biết. Hiện tại, bài toán phân loại nhãn đơn đã được nghiên cứu sâu và có nhiều thuật toán liên quan đã hoàn thiện như Naive Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN). Thuật toán phân loại nhiều nhãn sử dụng sự phụ thuộc giữa các nhãn hành vi để cải thiện hiệu suất của bộ phân loại. Đối với một số dữ liệu quan hệ ít phức tạp hơn và nhỏ hơn, thuật toán cho phép phân loại chính xác hơn. Tuy nhiên, rất khó để tìm hiểu các đặc tính của các nút thông qua việc phân loại các mô hình liên quan. Với việc nghiên cứu ngày càng nhiều về thuật toán phân loại, ngày càng có nhiều ứng dụng thực tế xuất hiện trong các lĩnh vực khai thác dữ liệu, khuyến nghị quan tâm, phát hiện dị thường,... Trong bài báo này, những đóng góp chính của công trình này được tóm tắt như sau: Chúng tôi đề xuất mô hình xác thực hành vi người dùng đám mây dựa trên siêu mạng đa nhãn, mô hình này thực hiện phân chia chi tiết hành vi của người dùng và cải thiện độ chính xác của việc phát hiện bất thường. Kết quả mô phỏng cho thấy nó có thể cải thiện hiệu quả độ chính xác của việc phân loại.

Năm 2020, Sudipta Sahana và các cộng sự [19] đã công bố nghiên cứu *"A Conceptual Framework Towards Implementing a Cloud-Based Dynamic Load*

Balancer Using a Weighted Round-Robin Algorithm", sự ra đời của điện toán đám mây đã mang đến cho thế giới công nghệ một sự bùng nổ lớn khi hầu hết các tổ chức, đơn vị kinh doanh và công ty công nghệ thông tin đã đón nhận công nghệ này trong những năm gần đây. Những lợi ích do công nghệ này mang lại là vô cùng to lớn khiến nó trở nên phổ biến trên thị trường công nghệ. Trên thực tế, cả nhà cung cấp dịch vụ và khách hàng đám mây đều được hưởng những lợi ích của công nghệ này. Phương pháp tiếp cận "trả tiền khi bạn sử dụng" một cách hạn chế, có nghĩa là khách hàng chỉ phải trả tiền cho tất cả những dịch vụ mà họ muốn sử dụng. Tuy nhiên, với sự gia tăng nhu cầu của công nghệ này và sự gia tăng về quy mô của đám mây, đòi hỏi các nhà cung cấp dịch vụ đám mây phải xử lý một số lượng lớn các yêu cầu. Do đó, vấn đề nằm ở nhà cung cấp dịch vụ để giải quyết vấn đề này một cách hợp lý trong khi vẫn duy trì hoặc nâng cấp hiệu suất của đám mây. Bất chấp một tương lai tuyệt vời mà công nghệ này đã có, một số vấn đề lớn liên quan đến nó cần được giải quyết. Một trong những vấn đề đó là cân bằng tải. Cân bằng tải trên đám mây là quá trình phân phối khối lượng công việc trên nhiều tài nguyên máy tính. Nói cách khác, đó là quá trình phân bổ nhiều yêu cầu của khách hàng giữa một số máy chủ trong môi trường đám mây. Việc phân bổ các yêu cầu giữa các máy chủ phụ thuộc vào một số thuật toán. Các thuật toán được sử dụng phổ biến là Vòng quay vòng, Vòng quay có trọng số, Kết nối ít nhất, Kết nối ít nhất có trọng số và Ngẫu nhiên. Ngoài những lợi ích như tính sẵn sàng cao, khả năng mở rộng, tính liên tục trong kinh doanh với tính linh hoạt, hiệu suất kinh tế và mức độ cao, việc sử dụng công nghệ có thể giúp ngăn máy chủ bị quá tải và hỏng hóc. Ngoài ra, trong trường hợp máy chủ bị lỗi, khối lượng công việc có thể được chuyển sang các máy chủ khác đang hoạt động tích cực. Do đó, kỹ thuật này giúp duy trì sự cân bằng giữa các máy chủ trong khi đồng thời nâng cao hiệu suất của đám mây và giúp sử dụng tài nguyên tối ưu. Trong bài báo của chúng tôi, chúng tôi đã thảo luận về một kỹ thuật cân bằng tải hiệu quả có thể xử lý các yêu cầu của máy khách giữa nhiều máy chủ với thời gian phản hồi tối thiểu. Chúng tôi đã sử dụng bộ cân bằng tải động dựa trên đám mây có thể được hỗ trợ tốt trong môi trường đám mây và có thể đạt được mức hiệu suất cao. Chúng tôi cũng đã sử dụng một thuật toán vòng tròn có trọng số vì chúng tôi nghĩ rằng thuật toán này có thể khắc phục những lỗ hổng tồn tại trong kịch bản hiện tại của cân bằng tải. Do đó,

chúng tôi bắt buộc phải thiết kế một kỹ thuật thích hợp sẽ giúp giảm thiểu vấn đề với cân bằng tải trong môi trường đám mây càng nhiều càng tốt…

Năm 2020, Aparna Kumaria, Rajesh Guptaa, Sudeep Tanwara, Neeraj Kumarb [20] đã công bố nghiên cứu "Blockchain and AI Amalgamation for Energy Cloud Management: Challenges, Solutions, and Future Directions" trong những năm gần đây, hệ thống Smart Grid (SG) phải đối mặt với nhiều thách thức khác nhau như nhu cầu năng lượng ngày càng tăng, sự tăng trưởng của các nguồn năng lượng tái tạo (RES) với sản xuất năng lượng phân tán (EG), các thiết bị Internet of Things (IoT) rộng lớn thích ứng, các mối đe dọa an ninh đang nổi lên và mục tiêu hàng đầu là duy trì sự ổn định, hiệu quả và độ tin cậy của SG. Để đối phó những vấn đề này đang tồn tại, hệ thống quản lý đám mây năng lượng (ECM), kết hợp cơ sở hạ tầng cho năng lượng, với sử dụng năng lượng và các dịch vụ giá trị gia tăng theo nhu cầu của người tiêu dùng. Để đạt được những điều này, dự báo từ phía nhu cầu hiệu quả và an toàn truyền dữ liệu là yếu tố quan trọng. Các vấn đề quản lý năng lượng đặt ra vấn đề hết sức nghiêm trọng trong việc tìm kiếm các giải pháp bền vững bằng cách sử dụng chuỗi khối (BC) và trí tuệ nhân tạo (AI). Các kỹ thuật dựa trên AI hỗ trợ các dịch vụ khác nhau như dự đoán tải lượng năng lượng, phân loại người tiêu dùng, quản lý tải và phân tích trong đó BC cung cấp tính bất biến của dữ liệu và cơ chế tin cậy cho quản lý năng lượng an toàn. Do đó, bài báo này xem xét một số phương pháp tiếp cận dựa trên AI hiện có cùng với những ưu điểm và những thách thức của việc tích hợp công nghệ BC và AI trong hệ thống ECM. Chúng tôi đã trình bày một khung ECM dựa trên AI phi tập trung để quản lý năng lượng bằng cách sử dụng BC và xác nhận nó bằng cách sử dụng một nghiên cứu điển hình. Nó chỉ ra rằng cách BC và AI có thể được sử dụng để giảm thiểu ECM với các vấn đề về bảo mật và quyền riêng tư. Cuối cùng, chúng tôi nhấn mạnh các vấn đề nghiên cứu mở và những thách thức của hệ thống ECM dựa trên BC-AI.

Năm 2020, Ahmed và các cộng sự [21] đã công bố nghiên cứu "An Empirical Analysis on Load Balancing and Service Broker Techniques using Cloud Analyst Simulator", Công nghệ điện toán đám mây cung cấp các dịch vụ công nghệ thông tin phức tạp được trình bày cho người dùng theo những cách khác nhau dựa trên yêu cầu kinh doanh của họ. Cả dịch vụ và nội dung đều được cung cấp tự động tại một thời

điểm cụ thể. Cân bằng tải là một trong những cách tiếp cận quan trọng được sử dụng trong môi trường đám mây để xác nhận mức hiệu suất của tài nguyên đám mây. Tải trọng được phân bổ trong tất cả các tài nguyên góp phần vào kiến trúc của trung tâm dữ liệu ảo hóa để giảm thiểu thời gian yêu cầu của người dùng trong trung tâm dữ liệu và tối đa hóa thông lượng hệ thống. Nhiều thuật toán cân bằng tải đã được các nhà nghiên cứu phát triển trong những năm gần đây. Các thuật toán đó được phân thành hai loại chính (thuật toán tĩnh và thuật toán động). Mục tiêu chính là tối ưu hóa hiệu suất môi trường đám mây bằng cách giảm chi phí tổng thể bằng cách truyền các yêu cầu của người dùng đến các nút chính. Trong bài báo này, một phân tích thực nghiệm được trình bày trên cả hai kỹ thuật (Cân bằng tải và Nhà môi giới dịch vụ) bằng cách sử dụng trình mô phỏng phân tích đám mây. Mục tiêu phân tích là nghiên cứu hành vi của ba thuật toán cân bằng tải khác nhau (Round Robin, Throttled và Active Monitoring). Các thuật toán đó chứa một số kỹ thuật môi giới dịch vụ trong các trung tâm dữ liệu đám mây ảo hóa. Công nghệ điện toán đám mây đang mở rộng nhanh chóng trên toàn thế giới, điều này đặt ra nhu cầu phát triển các kỹ thuật cân bằng tải hiệu quả; nhằm cung cấp dịch vụ đám mây nâng cao cho người dùng và thích ứng với tốc độ phát triển công nghệ nhanh chóng của môi trường đám mây. Trong bài báo này, đã phân tích hiệu suất của ba kỹ thuật cân bằng tải (Round Robin, Throttled và Active Monitoring) cùng với các chính sách môi giới dịch vụ khác nhau. Kết quả là kỹ thuật cân bằng tải điều chỉnh đã cung cấp thời gian phản hồi trung bình lý tưởng (mili giây) với chính sách trung tâm dữ liệu gần nhất. Trong tương lai, một kỹ thuật cân bằng tải được tối ưu hóa động có thể được phát triển để mở rộng chức năng giám sát tích cực của kỹ thuật cân bằng tải và để giảm thiểu cả thời gian phản hồi và thực thi của DC trong môi trường đám mây ảo hóa.

Năm 2021, Xun Xu; Shuo Zeng; Yuanjie He [22], đã công bố nghiên cứu về "The impact of information disclosure on consumer purchase behavior on sharing economy platform Airbnb", thông qua các bằng chứng thực nghiệm từ Airbnb từ tám thành phố lớn ở Hoa Kỳ, nhóm tác giả xem xét vai trò của thông tin tiết lộ trong việc tác động đến hành vi mua hàng của người tiêu dùng trên nền tảng kinh tế chia sẻ này. Chúng tôi phân tích việc công bố thông tin từ bốn khía cạnh - đó là thông tin gì (tức là nội dung thông tin), từ đâu (tức là, nguồn thông tin), ở định dạng nào (hình thức trình bày thông tin), và bao nhiêu (số lượng thông tin). Chúng tôi tìm thấy cả ba nguồn

thông tin - nhà cung cấp, nền tảng và người tiêu dùng ngang hàng - ảnh hưởng đến người tiêu dùng hành vi mua hàng. Liên quan đến thông tin do các nhà cung cấp đăng tải, chúng tôi nhận thấy mối quan hệ lõm giữa thông tin (tức là số lượng ảnh và độ dài của mô tả) và hành vi mua hàng của người tiêu dùng. Tuy nhiên, không có mối quan hệ đáng kể nào giữa phần tự mô tả của nhà cung cấp (văn bản và ảnh) và hành vi mua hàng của người tiêu dùng hành vi được tìm thấy. Về thông tin được đăng bởi nền tảng, cả khuyến nghị từ nền tảng và thông tin xác minh nhà cung cấp ảnh hưởng tích cực đến hành vi mua hàng của người tiêu dùng. Đối với thông tin về tương tác giữa nhà cung cấp và người tiêu dùng, tỷ lệ phản hồi cao và tốc độ phản hồi nhanh của nhà cung cấp sẽ nâng cao khả năng mua hàng của người tiêu dùng hành vi. Tuy nhiên, việc cung cấp kết nối với hồ sơ mạng xã hội của nhà cung cấp ảnh hưởng tiêu cực đến người tiêu dùng hành vi mua hàng. Về thông tin từ người tiêu dùng ngang hàng, chúng tôi nhận thấy mặc dù về tổng thể, người tiêu dùng xếp hạng ảnh hưởng tích cực đến hành vi mua hàng của người tiêu dùng, ảnh hưởng đó giảm đi khi xếp hạng vượt quá mức nhất định các ngưỡng. Nghiên cứu của chúng tôi cung cấp các gợi ý cho chủ sở hữu nền tảng để tối ưu hóa bố cục trình bày thông tin trực tiếp thông qua thiết kế nền tảng hoặc gián tiếp thông qua hướng dẫn tiết lộ thông tin của nhà cung cấp để tạo điều kiện thuận lợi cho việc tìm kiếm và thu thập thông tin của người tiêu dùng nhằm giảm rủi ro được nhận thức, nâng cao lòng tin đối với các nhà cung cấp và nền tảng, đồng thời nâng cao ý định và hành vi mua hàng của họ.

2.4. Kết luận Chương 2

Chương này đã trình bày một số công trình nghiên cứu liên quan trong nước và quốc tế. Các nghiên cứu này giúp ta hiểu rõ hơn về cân bằng tải trên môi trường điện toán đám mây, nắm bắt được ưu nhược điểm của các thuật toán cân bằng tải cũng như hành vi của người dùng hiện nay, từ đó đưa ra các thuật toán cải tiến với mục tiêu nâng cao hiệu suất. Chương tiếp theo sẽ trình bày đề xuất thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi người dùng cloud.

CHƯƠNG 3. ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY THÔNG QUA HÀNH VI NGƯỜI DÙNG CLOUD

3.1. Giới thiệu chung

Trong vài năm trở lại đây, nhiều bài báo đã trình bày các thuật toán trong cân bằng tải [23], [24], [25] và đề xuất cải tiến chúng với mục đích nâng cao hiệu suất cân bằng tải trong môi trường điện toán đám mây. Sau quá trình nghiên cứu và tham khảo những công trình nghiên cứu trên, đề tài này quyết định sẽ sử dụng một số thuật toán phân lớp (classification) trong AI, cụ thể là sử dụng thuật toán Cây quyết định J48 [26] kết hợp với việc nghiên cứu hành vi người dùng cloud nhằm đề xuất một ý tưởng cân bằng tải mới, từ đó có thể giúp cho các nhà cung cấp dịch vụ cũng như người dùng cloud hoạt động hiệu quả hơn. Hướng đến mục tiêu này, chương này sẽ trình bày tổng quát về ý tưởng bên cạnh đó cũng đề xuất việc xây dựng thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi của người dùng cloud nhằm nâng cao hiệu suất cân bằng tải trên môi trường điện toán đám mây bao gồm hệ thống host/datacenter và các máy ảo.

3.2. Mô hình nghiên cứu

Thuật toán phân lớp cây quyết định J48 (Decision Tree J48 Algorithm) sẽ được áp dụng vào mô hình nghiên cứu với mục tiêu phân bổ tài nguyên tương ứng với các Request thông qua hành vi của người dùng cloud mà ta đã dự đoán thông qua việc sử dụng thuật toán J48. Hành vi người dùng ở đây được phân chia trên request mà họ mang tới dựa theo các tiêu chí ứng với mỗi cloudlet [27] như kích thước của đầu vào (cloudletFileSize), kích thước đầu ra (cloudletOutputSize), độ dài (cloudletLength), thời gian bắt đầu thực hiện (execStartTime), thời gian kết thúc (finishTime),... Sau quá trình phân loại các tác vụ theo hành vi người dùng, bộ cân bằng tải sẽ tiến hành phân bổ request chứa tác vụ với hành vi cần nhiều tài nguyên hơn vào những máy ảo có khả năng xử lý tốt hoặc ngược lại, vào những máy vừa và thấp. Thông qua cách tiếp cận này, thuật toán mà luận văn đề xuất sẽ cải thiện được thời gian xử lý cân bằng tải trên đám mây, và có thể ứng dụng trên môi trường đám mây theo thời gian

thực. Luận văn cũng xin tạm đặt tên cho thuật toán là J-TUBA (J48 classifier of Task through User Behavior Algorithm).

Mục tiêu:

- Hạn chế và giảm thiểu các rủi ro cho datacenter.
- Làm giảm thời gian sống của các request trong đám mây.
- Ngăn chặn mất cân bằng tải và hạn chế tối đa sự mất cân bằng tải giữa các máy ảo.
- Giải quyết các request nhanh hơn, phân loại các tác vụ với các hành vi người dùng khác nhau tương ứng với các request, sử dụng có hiệu quả nguồn tài nguyên đám mây, đáp ứng một cách tốt nhất cho người dùng.
- Có thể phân lớp được các request tiếp theo tương ứng với hành vi người dùng đã được phân lớp ở trên, từ đó có kế hoạch đưa các request này sang những máy ảo/host có khả năng xử lý tải tương ứng.
- Sắp xếp các máy ảo/host/tài nguyên sao cho mức độ sử dụng từ cao đến thấp để phân bổ các tác vụ hợp lý.

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.
- Đề xuất tập trung vào mô phỏng dịch vụ Web (Web Service), các web server (máy ảo) sẽ biết trước thời gian xử lý của từng dịch vụ chạy trên web và trên từng máy ảo.
- 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 là khác nhau.

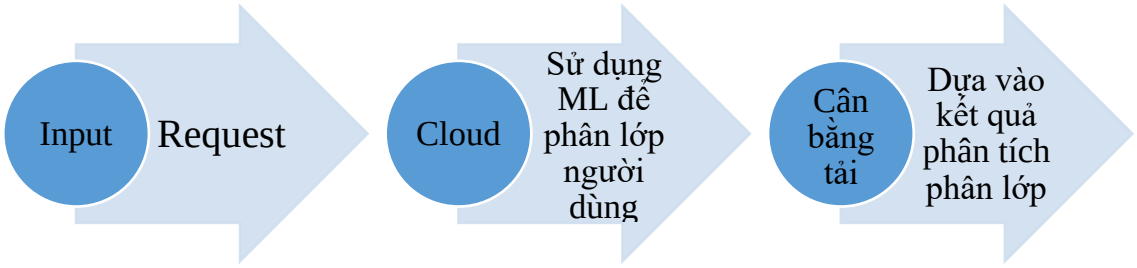
Mô hình nghiên cứu:

- 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 phân lớp của máy học để tiến hành phân lớp các request dựa trên các đặc trưng của các request. Các đặc trưng này dựa trên các hành vi của người dùng trên internet.

+ Bước 3: Dựa vào kết quả phân lớp các request tiến hành cân bằng tải.



Hình 3.1: Mô hình cân bằng tải

- Với đầu vào là các yêu cầu từ người dùng (request) thuật toán đề xuất sẽ xử lý và đưa vào các máy ảo phù hợp để tiến hành quá trình cân bằng tải.

- Mô hình này áp dụng K-means (dựa vào tính chất của các request) mà phân loại các đầu vào này, sau đó từ các tác vụ mà request mang tới (CPU Usage, RAM Usage, Power) ta dự báo thông số cloud sao cho phù hợp. Để có thể tiến hành phân lớp với kỹ thuật này, thuật toán sẽ sử dụng bộ dữ liệu trong lịch sử cloud đã được lưu lại (dữ liệu gần đây nhất).

- Tiếp theo, với các số liệu (CPU Usage, RAM Usage, Power) mà cloud cần để xử lý tác vụ tương ứng đã tính toán phía trên, ta tiếp tục sử dụng thuật toán J48 để phân lớp các tác vụ với bộ dữ liệu là dữ liệu dự đoán tính toán ở phía trên kết hợp với dữ liệu thực đã được lưu lại, phân lớp các request dựa trên đặc trưng (hành vi của người dùng), cuối cùng phân bổ vào các máy ảo tương ứng.

- Mô hình mô phỏng thuật toán một cách tự nhiên, đồng thời lên kế hoạch cho các yêu cầu kế tiếp nhằm đảm bảo cân bằng tải. Thuật toán này sẽ giúp giảm các tải liên lạc giữa các nguồn tài nguyên và máy ảo hiện có, chính vì thế làm giảm được băng thông cùng với thông lượng không cần thiết, phục vụ tốt hơn yêu cầu của người dùng.

3.3. Thuật toán Cây quyết định J48, K-Means và ứng dụng vào phân lớp request dựa trên hành vi người dùng

- Thuật toán Cây quyết định J48

J48 là một mã nguồn mở của Java triển khai thuật toán cây quyết định C4.5. J48 là một phần mở rộng của ID3, các tính năng bổ sung của J48 là tính toán các giá trị bị thiếu, cắt tỉa cây quyết định, dẫn xuất các quy tắc, v.v.. Là một bộ phân loại cây quyết định (decision tree classifier), J48 sử dụng mô hình học máy dự đoán để tính

toán giá trị kết quả của một mẫu mới dựa trên các giá trị thuộc tính khác nhau của dữ liệu có sẵn. Các nút bên trong của cây quyết định biểu thị các thuộc tính khác nhau; các nhánh giữa các nút cho chúng ta biết các giá trị có thể có mà các thuộc tính này có thể có trong các mẫu quan sát, trong khi các nút đầu cuối cho chúng ta biết giá trị cuối cùng (phân loại) của biến phụ thuộc.

- Thuật toán K-Means

Thuật toán Kmeans là một thuật toán lặp lại để cố gắng phân vùng tập dữ liệu thành các nhóm (cụm) con không trùng lặp được xác định bởi Kpre, trong đó mỗi điểm dữ liệu chỉ thuộc về một nhóm. Nó cố gắng làm cho các điểm dữ liệu trong cụm càng giống nhau càng tốt đồng thời giữ cho các cụm càng khác biệt (càng xa) càng tốt. Nó chỉ định các điểm dữ liệu cho một cụm sao cho tổng khoảng cách bình phương giữa các điểm dữ liệu và trung tâm của cụm (trung bình cộng của tất cả các điểm dữ liệu thuộc cụm đó) là nhỏ nhất. Khi càng có ít biến thể trong các cụm, thì các điểm dữ liệu trong cùng một cụm càng đồng nhất (tương tự).

- Phân lớp tác vụ dựa theo hành vi người dùng

Dựa vào bộ dữ liệu trước đây khi xử lý những tác vụ đầu tiên, ta sử dụng J48 để phân lớp dựa trên hành vi người dùng cho các nhiệm vụ (hay task) tiếp theo. Tương ứng với mỗi yêu cầu (Request) sẽ có một tác vụ mà máy tính cần phải thực hiện để phục vụ người dùng. Chính vì thế, bất kỳ một request được gửi đến cloud đều có thể được phân lớp dựa trên tác vụ tương ứng của nó.

3.4. Thuật toán đề xuất J-TUBA

Thuật toán đề xuất J-TUBA (J48 classifier of Task through User Behavior Algorithm), dựa vào hành vi người dùng (User Behavior được mô tả ở trên) tương ứng với các request. Để có thể phân bố tài nguyên cho các request một cách tối ưu và hiệu quả nhất ta sử dụng thuật toán K-Means với mục tiêu tính toán nhanh tức thời các thông số cho việc phân cụm máy ảo, và phù hợp với yêu cầu real-time; sau đó áp dụng J48 để tiến hành phân lớp các request để tính toansra máy ảo phù hợp, từ đó tìm ra máy ảo phù hợp với request để đưa vào xử lý. Bên cạnh đó, các tài nguyên (máy ảo/host) sẽ được sắp xếp theo mức độ sử dụng tăng dần. Kết hợp với đánh giá số lần

sai cùng với 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.

Dựa vào tham khảo từ tài liệu [20], luận văn này xin đề xuất thuật toán gồm 3 nhóm module chính:

(1) Module tính toán ra các thông số của request bằng thuật toán K-means: Trong module này, thuật toán K-means sẽ dựa vào các thuộc tính của request và các yếu tố mà tính toán ra các thông số sử dụng tài nguyên của Task/job tương ứng với request đó. Các thuộc tính bao gồm: Size, Response Length, Max Length, …

Po_{New} = K-means(Request, Power)

CPU_{New} = K-means(Request, CPU)

RAM_{New} = K-means(Request, RAM)

Trong đó X_i = là các thuộc tính của Request khi gửi lên cloud.

Request = { X_1, X_2,…, X_n }, với X_i là các thuộc tính của Request

Po_{New} : Power dự đoán Power new: Power ghi nhận được trong quá khứ

CPU_{New} : CPU dự đoán CPU new: CPU ghi nhận được trong quá khứ

RAM_{New} : RAM dự đoán RAM new: RAM ghi nhận được trong quá khứ

Ở đây có thể sử dụng nhóm 3 yếu tố {Po, CPU, RAM} để tổng hợp tính toán, hoặc tính toán riêng biệt từng đại lượng.

(2) Module phân lớp tác vụ theo hành vi người dùng:

Trong module này sẽ sử dụng thuật toán phân lớp J48 để phân lớp request đang xét, dựa vào hành vi người dùng của các tác vụ.

VM_{select} = J48(Po, CPU, RAM);

Trong đó:

VM_{select} là máy ảo được chọn ra

J48 là hàm phân lớp từ mô hình Cây quyết định đã được xây dựng dựa trên bộ dữ liệu trước đây của các request

Po: là Power dự đoán tính toán từ Module 1

CPU: là mức sử dụng CPU dự đoán tính toán từ Module 1

RAM: là mức sử dụng RAM dự đoán tính toán từ Module 1

(3) Module phân bổ các dịch vụ (chọn máy ảo)

Module này có nhiệm vụ phân bổ các yêu cầu đến các máy ảo thông qua loại request và máy ảo phù hợp. Nếu một yêu cầu được gửi tới 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. Ở đây, Module 3 có nhiệm vụ phân bổ Request đang xét vào máy ảo đã tìm thấy từ Module 2, và từ đó xử lý cho request đó, và trả về kết quả của request, lưu vào lịch sử bộ nhớ các request gần nhất đã xử lý, làm dữ liệu đầu vào cho quá trình xây dựng mô hình Cây quyết định J48 ở Module 2.

Thuật toán J-TUBA

```
1.  For each Request in CloudRequests
2.      isLocated = false;
3.      UserBehavior = {Po, CPU, RAM}new = K-means(T1,T2,…,Tn); // Module 1
4.      Request.UserBehaviorClass = J48(UserBehavior); //J48 là mô hình phân lớp
        tác vụ
5.      For each VM in VMList
6.          If isFitSituation(Request. UserBehaviorClass, VM)
7.          AllocateRequestToVM(VM, Request); // Module 3
8.          isLocated = true;
9.          End If
10.     End For
11.     If(!isLocated)
12.         VM = VMList.getSelectedVM(); // Module 2
13.         AllocateRequestToVM(VM, Request);
14.     End If
15. End For
```

Trong đoạn mã giả trên của thuật toán J-TUBA, thuật toán sẽ sử dụng một vòng lặp để lắng nghe tất cả các Request có trong danh sách hàng đợi các Request được gửi lên bộ cân bằng tải (ở đây là CloudRequests). Khi nào hết danh sách này thì sẽ không phân bổ nữa. Trong đó, thuật toán sử dụng biến isLocated (kiểu luận lý) để làm cờ đánh dấu rằng Request đang xét đã được phân bổ hay chưa. Mới vào vòng lặp, biến isLocated được tạo giá trị mặc định là false. Sau đó, thuật toán tính toán ra vector UserBehavior với 3 chiều là PowerConsume, CPU Usage và RAM Usage (UserBehavior ={Po, CPU, RAM}) cần dùng để thực hiện Request đang xét. Việc tính toán này dựa trên lịch sử số liệu của các Request trước đó T1, T2,… Tn , trong đó n là số request đã được lưu, mỗi T bao gồm các đại lượng đầu vào: MaxLength,

FileSize, OutputSize…; và các đại lượng xử lý do Cloud đã thực hiện để xử lý Request thứ i bao gồm PowerConsume, CPU Usage và RAM Usage. Dữ liệu n Request trong lịch sử này sẽ xây dựng nên hàm K-means để dự báo và tính toán ra các đại lượng UserBehavior cho Request đang xét. Sau đó, dùng dữ liệu UserBehavior này để chạy J48 phân lớp cho Request đang xét. Và lớp này được gán vào thuộc tính UserBehaviorClass của Request. Sau khi chạy ra thông số UserBehavior cho Request, thì thuật toán duyệt vòng lặp để duyệt qua các máy ảo đang có trên cloud. Tương ứng với từng máy, thuật toán xem xét máy ảo đó có phù hợp với độ ưu tiên của Request đang xét hay không, thông qua hàm isFitSituation (Request. UserBehaviorClass, VM). Nếu thỏa thì sẽ phân bổ request đang xét vào máy ảo đó AllocateRequestToVM(VM, Request), và đồng thời gán giá trị biến isLoacated=true. Nếu trường hợp không tìm ra máy ảo nào phù hợp, thì sẽ kết thúc vòng lặp. Lúc này, chạy hết vòng lặp và biến isLocated vẫn mang giá trị false, và lúc này Request chưa được phân bổ. Vì vậy, thuật toán phân bổ Request này vào máy ảo đầu tiên của danh sách máy ảo thông qua đoạn lệnh VM = VMList.getSelectedVM(). Việc phân bổ này đảm bảo nếu có request nào được dự báo không nằm trong dữ liệu của thuật toán, vẫn được phân bổ và xử lý phục vụ người dùng.

Phương pháp đánh giá thuật toán J-TUBA

Kết quả đạt được từ thuật toán mà luận văn đề xuất đã đáp ứng các mục tiêu, chẳng hạn như giới hạn số lượng yêu cầu xếp hàng để phân phối, cải thiện thời gian xử lý và phản hồi của đám mây trung tâm so với bốn thuật toán cũ. Điều này cũng có nghĩa là với thuật toán được đề xuất, hiệu năng của điện toán đám mây được cải thiện so với bốn thuật toán **FCFS, MaxMin, MinMin** và **Round Robin**.

3.5. Kết luận chương 3

Để phục vụ cho thuật toán cân bằng tải, chương này đã trình bày lý do tác giả chọn phương pháp phân lớp tác vụ [17], [21], [22] theo hành vi của người dùng cloud. Để cloud có thể giữ được trạng thái an toàn và hoạt động liên tục, thuật toán đề tài thực hiện đã cải tiến các thuật toán cân bằng tải trong điện toán đám mây – J-TUBA (J48 classifier of Task through User Behavior Algorithm) sẽ giải quyết được sự cân bằng tải dựa trên việc cải thiện thời gian thực thi tác vụ từ đó lượng số lượng thất bại trong việc triển khai sẽ giảm đi, số điểm chết nút sẽ ít hơn các thuật toán hiện tại.

CHƯƠNG 4. MÔ PHỎNG, THỰC NGHIỆM

4.1. Giới thiệu chung

Chương này sẽ trình bày việc cài đặt và mô phỏng thuật toán J-TUBA (J48 classifier of Task through User Behavior Algorithm), cụ thể là sử dụng thuật toán phân lớp Cây quyết định J48 (Decision Tree J48 Algorithm) với mục tiêu là phân loại các tác vụ tương ứng với các Request dựa trên đặc trưng của tác vụ đó. Đặc trưng này tính toán dựa trên hành vi của người dùng cloud (loại người dùng và mục đích sử dụng). Sau khi phân loại các tác vụ theo đặc trưng, bộ cân bằng tải sẽ tiến hành phân phối các request có tác vụ đó vào những máy ảo/host phù hợp. 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 J-TUBA 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. Xây dựng mô hình mô phỏng – thực nghiệm

Dựa vào bộ dữ liệu chứa các request, luận văn này sử dụng thuật toán K-Means để phân loại request dựa trên các đặc trưng của request (hành vi người dùng cloud)

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.

Dựa vào dữ liệu của các request mà ta có thể biết, ta sử dụng thuật toán K-means để phân loại request bằng cách tính toán ra bộ Priority = {Power, CPU, RAM}, từ đó ta biết cách phân bố tài nguyên cho cái 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.

Ta tiến hành giả lập môi trường cloud bằng cách lập trình trên ngôn ngữ JAVA và sử dụng bộ thư viện CloudSim. Môi trường này sẽ bao gồm 5 đến 15 máy ảo, và đồng thời 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 tiến hành thử nghiệm.

Cài đặt thuật toán K-means, J48 trên môi trường mô phỏng được phát triển bởi bộ thư viện Weka và kiểm nghiệm kết quả.

Các tham số của mô hình mạng mô phỏng:

Quá trình thực nghiệm mô phỏng thuật toán được cài đặt trên ngôn ngữ JAVA và sử dụng Eclipse IDE hoặc NETBEAN IDE để chạy thử và hiển thị kết quả dưới dạng console. Môi trường giả lập với bộ thư viện mã nguồn mở CloudSim (được cung cấp bởi http://www.cloudbus.org/), kết hợp với bộ thư viện về Data Mining là WEKA.

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

- 1 Datacenter với thông số sau:

Bảng 4.1: Thông số cấu hình Datacenter

Thông tin Datacenter	***Thông tin Host trong Datacenter***
- Số lượng máy (host) trong datacenter: 5 - 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 các ngẫu nhiên bằng hàm random của JAVA. Số lượng Cloudlet lần lượt là 500 → 1500.

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 **JTUBASchedulingAlgorithm**, kế thừa từ đối tượng **BaseSchedulingAlgorithm**, cập nhật thêm một số phương thức và thuộc tính liên quan tới **predictRequestKmeans**, và điều chỉnh các hàm dựng sẵn để phù hợp với thuật toán đề xuất:

@Override

*public void **run()** // Module 3*

*public CondorVM **getFittingVm** (double label)*

// Module 2

*public String **predictRequestPowerConsume**(Cloudlet req)*

*public String **predictRequestCpuUsage**(Cloudlet req)*

*public String **predictRequestRamUsage** (Cloudlet req)*

// Module 1

Tiêu chí đánh giá

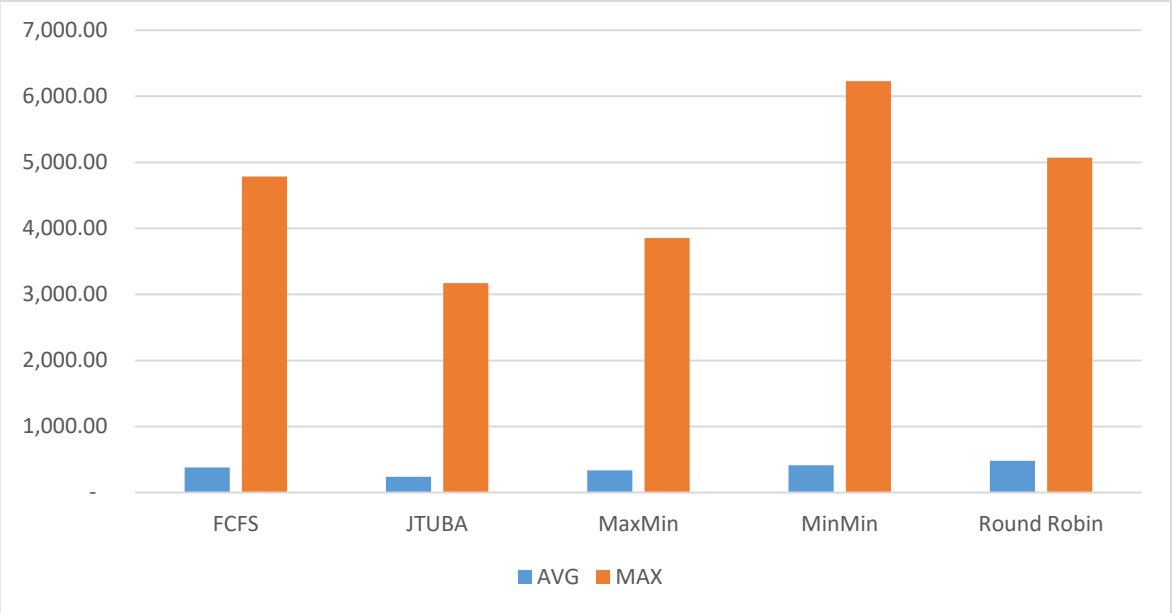
Thực nghiệm mô phỏng cloud với các tham số trên sau đó chạy thuật toán cân bằng tải của CloudSim có sẵn cuối cùng là chạy thuật toán đề xuất mới cài đặt kết hợp với dữ liệu đầu vào và so sánh kết quả đầu ra, đặc biệt là thông số thời gian thực hiện (Makespan) [28]. Thời gian đáp ứng dự đoán của các máy ảo và thời gian đáp ứng dự đoán của cloud với sai số càng thấp thì hiệu quả của thuật toán càng tốt.

4.3. Kết quả thực nghiệm của mô hình

Tiến hành chạy thực nghiệm mô phỏng trên CloudSim với 5 máy ảo đã được dựng sẵn để đáp ứng các request, với các request được khởi tạo có chiều dài và kích thước ngẫu nhiên và số lượng Request lần lượt là 30, 60, 100 và 1000. Tiếp theo ta so sánh kết quả này với các thuật toán FCFS, MaxMin, MinMin và Round-Robin. Đầu tiên, với trường hợp với 30 Requets, ta có bảng kết quả như bảng 4.4.

Bảng 4.4: Kết quả thực nghiệm mô phỏng với 30 request

Thời gian thực hiện (ms)	FCFS	JTUBA	MaxMin	MinMin	Round Robin
AVG	378.67	239.51	333.53	412.75	483.14
MAX	4,782.24	3,172.69	3,852.00	6,228.22	5,071.06
MIN	0.84	0.13	0.17	0.12	1

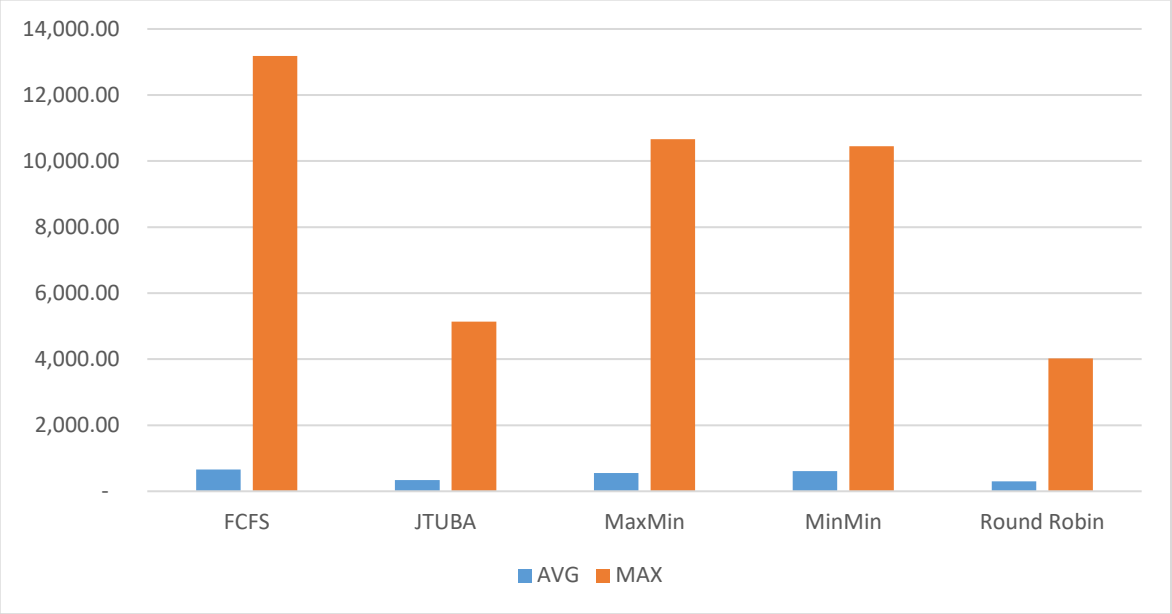


Hình 4.1: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 30 Request

Kết quả thực nghiệm với số request là 30 Request, có thể thấy rằng thuật toán J-TUBA đang chiếm ưu thế và xử lý tốt hơn các thuật toán khác, bên cạnh đó thuật toán MaxMin cũng khá ổn định. Thuật toán MinMin thì chưa có ưu thế mạnh. Tuy nhiên để chứng tỏ thuật toán đề xuất tốt hơn ta sẽ tiến hành quan sát kết quả khi xử lý nhiều request hơn.

Bảng 4.5: Kết quả thực nghiệm mô phỏng với 60 request

Thời gian thực hiện (ms)	FCFS	JTUBA	MaxMin	MinMin	Round Robin
AVG	661.45	335.71	550.44	610.61	303.08
MAX	13,187.07	5,139.11	10,657.80	10,451.42	4,024.57
MIN	0.48	0.17	0.13	0.14	0.1

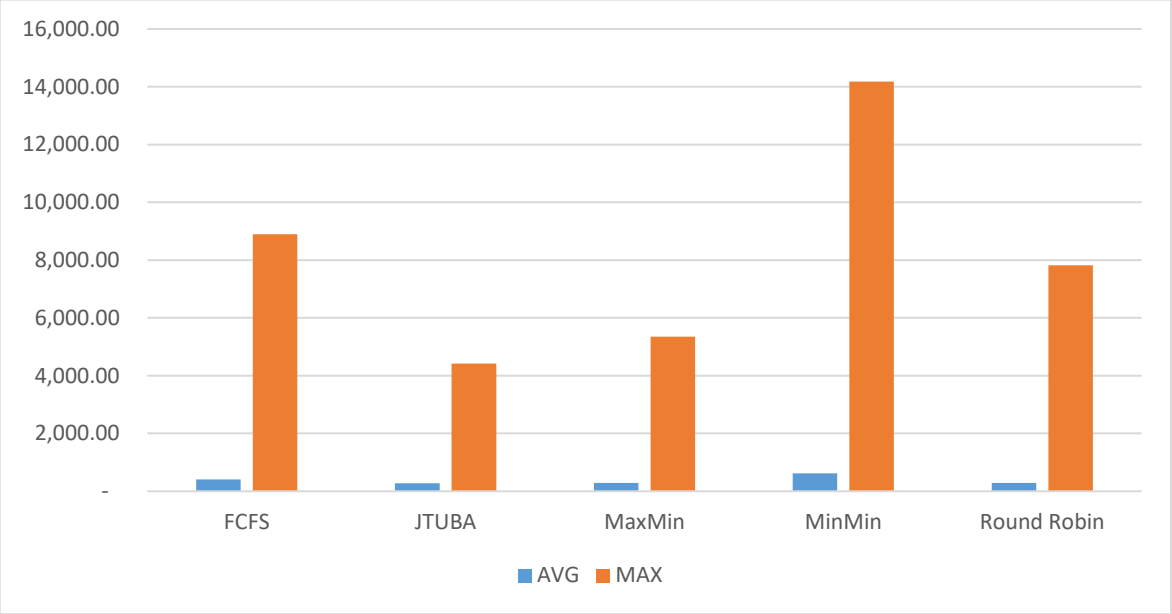


Hình 4.2: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 60 Request

Với request từ 60 trở đi, ta thấy thuật toán J-TUBA vượt trội hẳn so với MaxMin, MinMin và FCFS. Dù vậy vẫn chưa chiếm ưu thế so với RoundRobin. Ở trường hợp này, do 60 Request tương đối nhỏ, và các request không đủ làm cho cloud gồm các máy ảo bị overload, và thuật toán RR mạnh về xử lý các request nhỏ và tức thời, nên sẽ nhỉnh hơn so với các thuật toán còn lại. Tuy nhiên thuật toán J-TUBA cũng khá ổn định với số lượng request nhỏ và kích thước nhỏ. Ta cũng nhận thấy rằng FCFS thể hiện sự thiếu thông minh và tính tự nhiên của giải thuật.

Bảng 4.6: Kết quả thực nghiệm mô phỏng với 100 request

Thời gian thực hiện (ms)	**FCFS**	**JTUBA**	**MaxMin**	**MinMin**	**Round Robin**
AVG	411.00	280.15	285.03	624.96	284.91
MAX	8,898.95	4,424.93	5,351.33	14,181.70	7,821.67
MIN	0.18	0.16	0.12	0.12	0.13

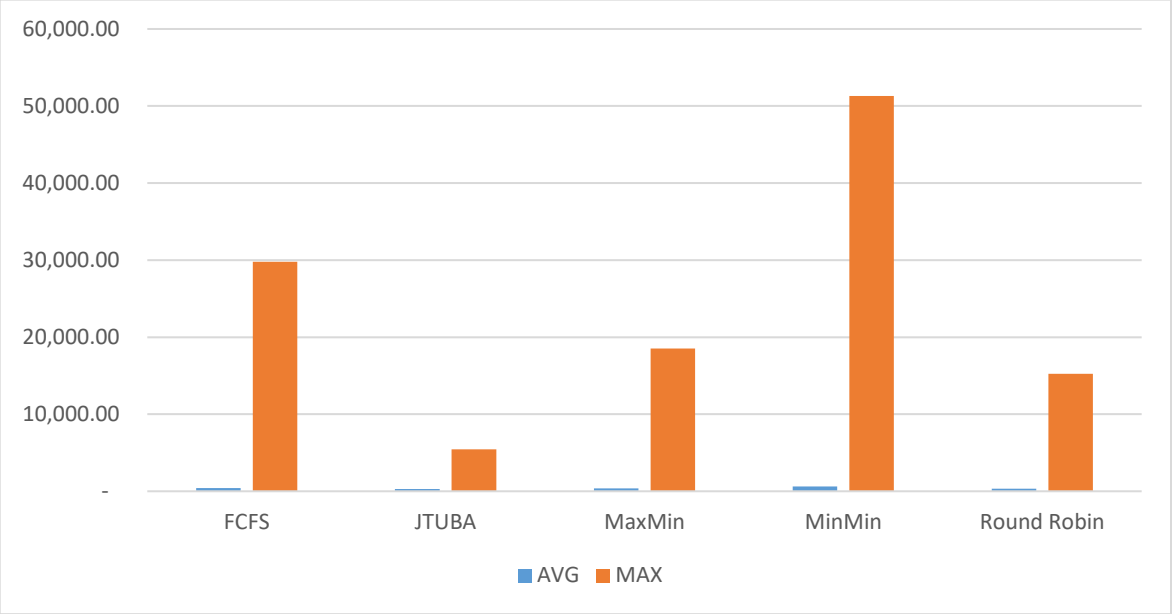


Hình 4.3: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 100 Request

Khi thử nghiệm với 100 request, ta thấy J-TUBA vượt trội hơn hẳn so với FCFS, MinMin, Round Robin. Với số lượng request càng lớn thì J-TUBA càng lợi thế hơn hẳn. Cuối cùng ta tăng lên 1000 request để quan sát liệu J-TUBA có bỏ xa các thuật toán khác hay không.

Bảng 4.7: Kết quả thực nghiệm mô phỏng với 1000 request

Thời gian thực hiện (ms)	FCFS	JTUBA	MaxMin	MinMin	Round Robin
AVG	424.48	278.72	377.15	640.01	342.62
MAX	29,804.76	5,427.77	18,542.97	51,311.11	15,266.81
MIN	0.1	0.15	0.1	0.11	0.11

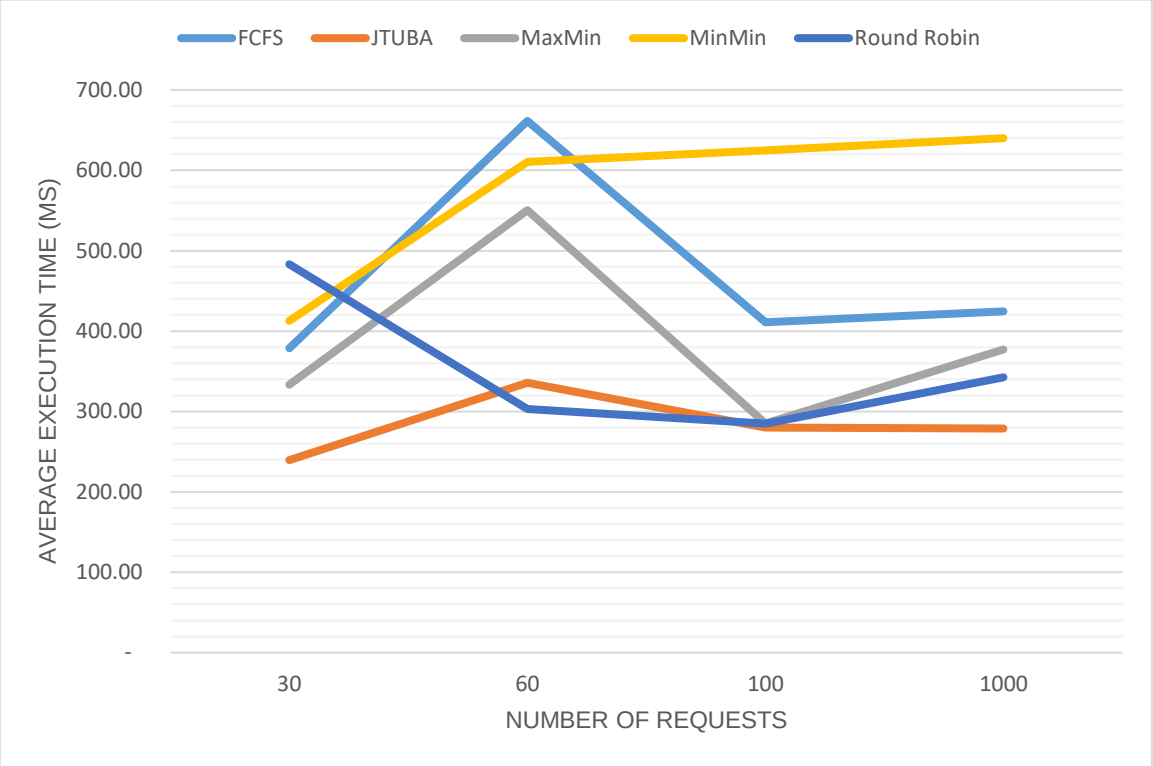


Hình 4.4: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 1000 Request

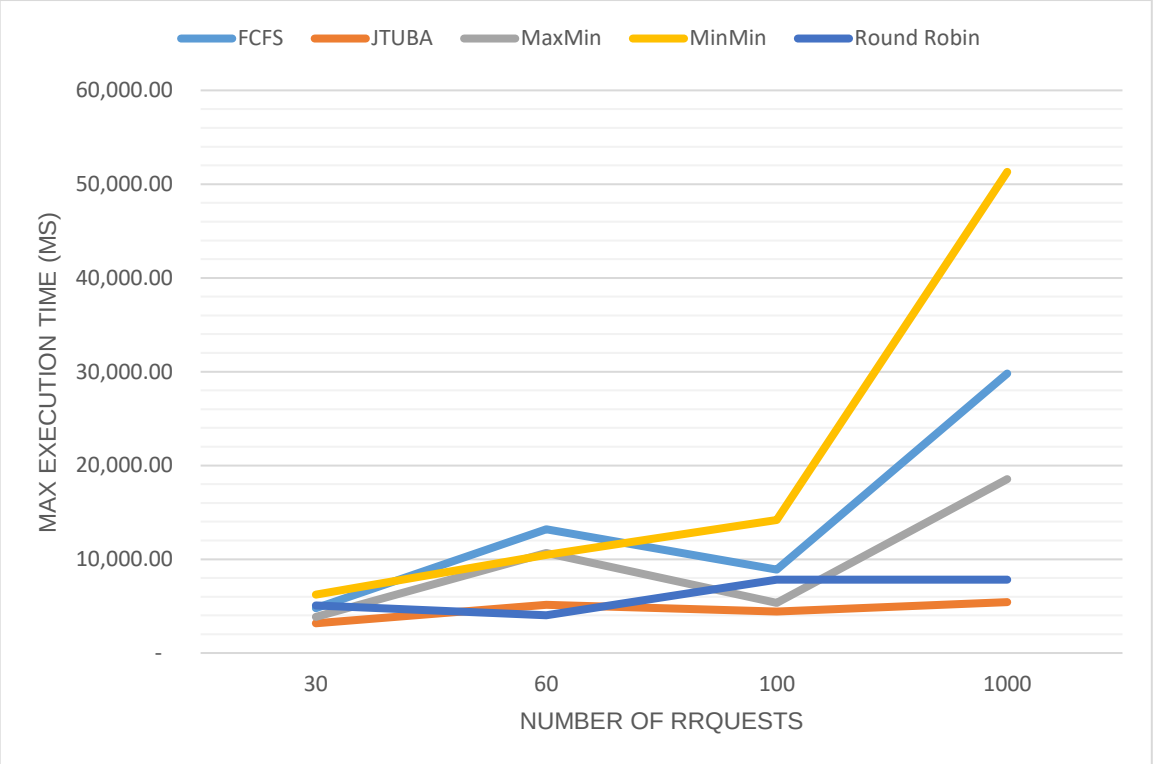
Trường hợp 1000 request này đã chứng minh J-TUBA vượt trội hơn hẳn so với 4 thuận toán còn lại. Nếu lượng request bùng nổ, việc xử lý của thuật toán vẫn đáp ứng được, phụ thuộc vào cấu hình thiết bị chạy bộ cân bằng tải, tuy nhiên, J-TUBA không lưu trữ lịch sử request và phân bổ request quá nhiều, vì vậy, không quá khó khăn trong việc xử lý cân bằng tải nếu lượng request nhiều, vì tất cả các bộ LB đều có cơ chế hàng đợi.

Thống kê so sánh thời gian thực hiện của các thuật toán

Sau khi chạy kiểm thử 4 thuật toán cơ bản với thuật toán đề xuất J-TUBA trên với số lượng request tăng dần từ 30 – 1000, ta tiến hành vẽ biểu đồ để thống kê thời gian thực hiện trung bình của các thuật toán để quan sát và đánh giá tính ổn định giữa chúng.



Hình 4.5: Thời gian thực hiện trung bình của 5 thuật toán từ 30-1000 Request



Hình 4.6: Thời gian thực hiện lớn nhất của 5 thuật toán từ 30-1000 Request

Với 4 trường hợp 30, 60, 100 và 1000 request, sau khi so sánh thời gian xử lý của các thuật toán với cùng điều kiện ta đã thấy được sự phân bổ khá ổn định và hợp lý của thuật toán đề xuất J-TUBA, thời gian xử lý của các máy ảo không quá khác

biệt so với thời gian xử lý của các thuật toán khác trên cloud (ở trường hợp ít và nhiều request). Hình 4.5 và 4.6 cho thấy J-TUBA luôn thấp nhất, kể cả giá trị trung bình lẫn giá trị max.

Thực nghiệm mô phỏng này chỉ là mô phỏng nhóm các máy ảo, chưa tính tới việc mở rộng tập các máy ảo (VM pool) để giảm tải trong trường hợp cần thiết, vì giả định là nhóm các máy ảo này xử lý tối đa bao nhiêu request, nếu vượt quá ta mới mở rộng pool. Tuy nhiên, việc thí nghiệm mô phỏng với lượng request lớn là trên 1000 request đòi hỏi máy tính mạnh hơn và bộ xử lý tốt hơn, chính vì vậy đây là hạn chế của thí nghiệm mô phỏng này.

Thuật toán đề xuất đã cho thấy hiệu quả khi máy ảo có số lượng cao lên thì J-TUBA đảm bảo thời gian phản hồi và thời gian xử lý tốt, giảm chi phí của các trung tâm dữ liệu đám mây. Tuy nhiên thuật toán vẫn còn 1 số nhược điểm như:

- Chưa sắp xếp các máy ảo theo danh sách tăng dần.

- Nếu số lượng máy ảo nhiều thì việc tìm ra máy có Usage nhỏ nhất là khó khăn hơn, nên có thể tìm một máy có Usage phù hợp là đạt.

4.4. Phân tích, đánh giá hiệu quả của mô hình

Chương này đã 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. Từ đó, ghi nhận các thông số về thời gian xử lý của các máy ảo, và của cloud. Việc chạy thực nghiệm mô phỏng với thông số 5 máy ảo, chịu tải từ 30 tới 1000 request đã cho thấy kết quả tương đối tốt, việc phân bổ các request tới các máy ảo xử lý khá đồng đều, và tính khả thi cao.

KẾT LUẬN

Luận văn "**Đề xuất thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi người dùng cloud**" nghiên cứu các thuật toán phân lớp từ các đặc trưng của request – hành vi người dùng. Từ đó phân bổ các tác vụ sao cho hợp lý và nâng cao cân bằng tải trong môi trường điện toán đám mây, sử dụng hiệu quả nguồn tài nguyên đám mây. Từ các thuật toán [23] đã có, tiến hành phân tích làm rõ chúng sau đó đánh giá đưa ra lợi thế cũng như nhược điểm của từng thuật toán, xem xét các nhược điểm đã phân tích để đề xuất thuật toán mới nhằm cải tiến và nâng cao khả năng cân bằng tải so với thuật toán cũ. Quá trình nghiên cứu của tác giả đã đạt được những mục tiêu như sau:

- Nghiên cứu tổng quan về đám mây và các kỹ thuật cân bằng tải được dùng trong môi trường điện toán đám mây.

- Nghiên cứu cách tiếp cận điện toán đám mây thông qua mô phỏng sử dụng bộ thư viện CloudSim. Cài đặt, mô phỏng các kỹ thuật cân bằng tải, các thuật toán MaxMin, MinMin, Round Robin và thuật toán tự nhiên FCFS. Các giá trị thu được khi mô phỏng đưa ra để phân tích so sánh với nhau để tổng hợp các ưu nhược điểm của các thuật toán từ đó có hướng đề xuất một thuật toán mới để khắc phục những mặt hạn chế đó.

- Kết quả đạt được từ thuật toán đề xuất đáp ứng được các mục tiêu như: cải thiện thời gian đáp ứng, hạn chế tài nguyên rảnh rỗi, máy ảo có năng lực xử lý mạnh sẽ được xử lý nhiều yêu cầu hơn. Giúp cân bằng tải hiệu quả hơn thuật toán được so sánh là MaxMin, MinMin, Round Robin và thuật toán tự nhiên FCFS.

- Thuật toán đề xuất J-TUBA có thể dùng để áp dụng trên thực tế.

Hạn chế của luận văn

- Chưa được ứng dụng vào môi trường thực tế.

- Thời gian đáp ứng và xử lý chưa cải thiện được nhiều.

Hướng phát triển đề tài

- Đưa thuật toán đề xuất ứng dụng vào môi trường đám mây cụ thể, hệ thống thực.

- Tiếp tục nghiên cứu, tối ưu các tham số đầu vào nhằm nâng cao hơn hiệu năng cân bằng tải.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Apprenda, "Introduction to Cloud Computing," THE APPRENDA LIBRARY, [Online]. Available: https://apprenda.com/library/cloud/introduction-to-cloud-computing/.

[2] Y. Wen and C. Chang, "Load balancing job assignment for cluster-based cloud computing," *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN),* pp. 199-204, 2014.

[3] Bui Thanh Khiet, Nguyen Thi Nguyet Que, Ho Dac Hung, Pham Tran Vu, Tran Cong Hung, "A Fair VM Allocation for Cloud Computing based on Game Theory," *Proceedings of the 10th National Conference on Fundamental and Applied Information Technology Research (FAIR'10),* 2017.

[4] J. Zhang, Q. Liu and J. Chen, "An Advanced Load Balancing Strategy for Cloud Environment," in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2016.

[5] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu and G. Xu, "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment," *IEEE Transactions on Parallel and Distributed Systems,* 2016.

[6] GIBET TANI, H. and C. EL AMRANI, "Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data," *Journal of Data Mining and Digital Humanities,* 2018.

[7] Matthias Sommer, Michael Klink, Sven Tomforde, Jörg Hähner, "Predictive Load Balancing in Cloud Computing Environments Based on Ensemble Forecasting," in *2016 IEEE International Conference on Autonomic Computing (ICAC2016),* 2016.

[8] G. Shao and J. Chen, "A Load Balancing Strategy Based on Data Correlation in Cloud Computing," in *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing* , 2016.

[9] Ashok U., C.K. Jha, Shikha P., "Suboptimal Mechanism For Load Balancing In CloudEnvironment," in *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, 2017.

[10] K. S. Umadevi and P. Chaturvedi, "Predictive load balancing algorithm for cloud computing, "Predictive load balancing algorithm for cloud computing," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, 2017.

[11] P. Wang, H. Xu, Z. Niu, D. Han and Y. Xiong, "Expeditus: Congestion-Aware Load Balancing in Clos Data Center Networks," *IEEE/ACM Transactions on Networking,* 2017.

[12] A. Garg, "A comparative study of static and dynamic Load Balancing Algorithms," *International Journal of Advance Research in Computer Science and Management IJARCSMS,* vol. 2, no. 12, pp. 386-392, 2014.

[13] Tychalas, Dimitrios & Karatza, Helen, "An Advanced Weighted Round Robin Scheduling Algorithm," in *24th Pan-Hellenic Conference on Informatics*, 2020.

[14] C. Jayashri, P. Abitha, S. Subburaj, S. Y. Devi, Suthir S and Janakiraman S, "Big data transfers through dynamic and load balanced flow on cloud networks," in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 2017.

[15] Tran Cong Hung, Nguyen Xuan Phi, "Thuật Toán Cân Bằng Tải Nhằm Giảm Thời Gian Đáp Ứng Dựa Vào Ngưỡng Thời Gian Trên Điện Toán Đám Mây," *Journal of Science and Technology on Information and Communications ,* 2018.

[16] Phi, Nguyen Xuan; Hung, Tran Cong, "Study the effect of Parameters to load balancing in cloud computing," *International Journal of Computer Networks & Communications (IJCNC),* 2016.

[17] N. H. H. Cuong, "Avoid Deadlock Resource Allocation (ADRA) Model V VM-out-of-N PM," *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 2, pp. 98-107, 2018.

[18] Liu, R., Wang, X., Du, J., & Xie, P, "A Cloud User Behavior Authentication Model Based on Multi-label Hyper-network," *Journal of Internet Technology*, 2019.

[19] S. Sahana, T. Mukherjee and D. Sarddar, "A Conceptual Framework Towards Implementing a Cloud-Based Dynamic Load Balancer Using a Weighted Round-Robin Algorithm," *International Journal of Cloud Applications and Computing (IJCAC),* 2020.

[20] Kumari, Aparna & Gupta, Rajesh & Tanwar, Sudeep & Kumar, Neeraj, "Blockchain and AI Amalgamation for Energy Cloud Management: Challenges, Solutions, and Future Directions," *Journal of Parallel and Distributed Computing.*

[21] A. I. El Karadawy, A. A. Mawgoud and H. M. Rady, "An Empirical Analysis on Load Balancing and Service Broker Techniques using Cloud Analyst Simulator," in *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, 2020.

[22] Xu, Xun & Zeng, Shuo & He, Yuanjie, "The impact of information disclosure on consumer purchase behavior on sharing economy platform Airbnb," *International Journal of Production Economics,* 2020.

[23] Tran Cong Hung, Phan Thanh Hy, Le Ngoc Hieu, Nguyen Xuan Phi, "MMSIA: Improved Max-Min Scheduling Algorithm for Load Balancing on Cloud Computing," *Proceedings of The 3rd International Conference on Machine Learning and Soft Computing (CMLSC 2019),* pp. 60-64, 2019.

[24] Mehmet Muzaffer Kösten, Murat Barut, Nurettin Acir, "Deep neural network training with iPSO algorithm," in *26th Signal Processing and Communications Applications Conference (SIU)*, 2018.

[25] Michael Richmond, Michael Hitchens, "A new process migration algorithm," *ISBN*, 1996.

[26] Wikipedia contributors, "C4.5 algorithm," 2018. [Online]. Available: https://en.wikipedia.org/wiki/C4.5_algorithm. [Accessed 20 April 2021].

[27] A. Singh, "Cloudlet in Cloudsim Simulation," Cloudsim Tutorials, 12 February 2021. [Online]. Available: https://www.cloudsimtutorials.online/cloudlet-in-cloudsim-simulation/.

[28] T. F. Encyclopedia, "Makespan," Wikipedia , 15 June 2021. [Online]. Available: https://en.wikipedia.org/wiki/Makespan.

[29] Rajwinder Kaur, Pawan Luthra, "Load Balancing in Cloud Computing," *Recent Trends in Information, Telecommunication and Computing, Association of Computer Electronics and Electrical Engineers*, pp. 374-381, 2014.

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 12% toàn bộ nội dung luận văn/luận án. Bản luận văn/luận án 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.

TP.Hồ Chí Minh, ngày 25 tháng 01 năm 2021

HỌC VIÊN CAO HỌC

Huỳnh Phi Long

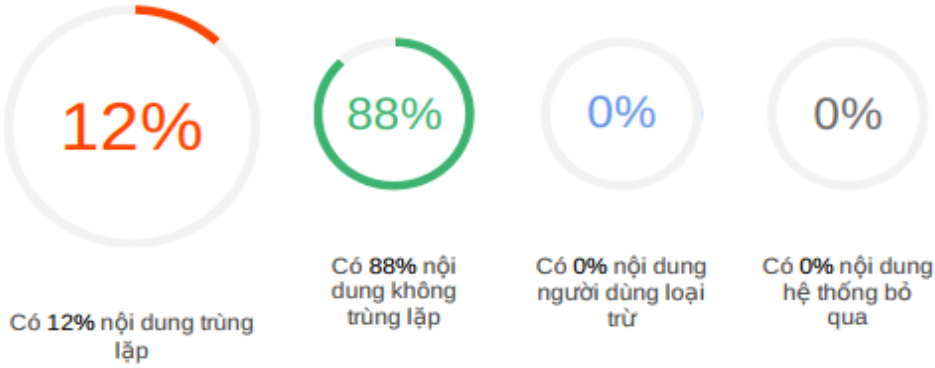
KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	1_LuanVan_HuynhPhiLong_Final
Tác giả:	n20chis023@student.ptithcm.edu.vn
Điểm trùng lặp:	12
Thời gian tải lên:	09:31 10/02/2022
Thời gian sinh báo cáo:	09:34 10/02/2022
Các trang kiểm tra:	56/56 trang

Kết quả kiểm tra trùng lặp



Nguồn trùng lặp tiêu biểu

123docz.net tailieu.vn

HỌC VIÊN **NGƯỜI HƯỚNG DẪN KHOA HỌC**

PGS.TS. Trần Công Hùng

Huỳnh Phi Long

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

CHƯƠNG 1.

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

Họ và tên học viên: **Huỳnh Phi Long**

Chuyên ngành: **Hệ Thống Thông Tin**

Khóa: **2020- 2022**

Tên đề tài: **Đề xuất thuật toán cân bằng tải trên điện toán đám mây thông qua hành vi người dùng cloud**

Người hướng dẫn khoa học: **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	Hiệu chỉnh luận văn phần nội dung cho khác với các nguồn khác	Học viên đã điều lại phần nội dung của luận văn.
2	Bổ sung giải thích các thuật toán	Học viên đã điều chỉnh bổ sung giải thích cụ thể thuật toán ở chương 3, mục 3.3 của luận văn.
3	Hiệu chỉnh lỗi trình bày, lỗi văn phong	Học viên đã tiến hành rà soát và điều chỉnh lại lỗi trình bày, văn phong trong luận văn.

Tp.Hồ Chí Minh, 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
PGS.TS. Đinh Đức Anh Vũ	**TS. Trần Trung Duy**	**PGS.TS. Trần Công Hùng**	**Huỳnh Phi Long**