HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Bùi Quang Tuyên

ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY BẰNG CÔNG NGHỆ AI HIỆN ĐẠI

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

TP. HCM – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Bùi Quang Tuyên

ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY BẰNG CÔNG NGHỆ AI HIỆN ĐẠI

CHUYÊN NGÀNH:HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SĨ 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 bằng công nghệ AI hiện đại"* 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 1 năm 2022

Học viên thực hiện luận văn

Bùi Quang Tuyên

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 1 năm 2022

Học viên thực hiện luận văn

Bùi Quang Tuyên

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh
CC	Cloud Computing
ML	Machine Learning
LB	Load Balancing
Cloud	Cloud computing environment
AI	Artificial Intelligence

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Ngày nay, với sự bùng nổ thông tin cũng như đòi hỏi nhu cầu về xử lý thông tin ngày càng cao thì nhu cầu về khả năng lưu trữ một lượng dữ liệu lớn là vô cùng cấp thiết. Sự phát triển không ngừng của nền kinh tế thế giới đã đẩy các doanh nghiệp, các tập đoàn lớn vào tình thế phải có được một giải pháp giúp họ lưu trữ được một khối lượng khổng lồ các dữ liệu liên quan đến công việc kinh doanh của họ…

Vì vậy để đáp ứng tất cả các nhu cầu nói trên thì đã có Điện toán đám mây (Cloud computing).

Đứng về khía cạnh người dùng thì họ muốn mình được phục vụ một cách tiện lợi nhất, dễ dàng nhất và không phải lúc nào cũng túc trực quản lý hay tiêu tốn vào đó một lượng tài chính nhất định.

Đứng về khía cạnh các doanh nghiệp thì họ luôn muốn làm hài lòng khách hàng của mình một cách tốt nhất để từ đó mới giữ chân được khách hàng.

Đứng về khía cạnh công nghệ thì hiện tại các công nghệ phát triển đã kịp thời đáp ứng các bài toán về tải, lưu trữ, băng thông…

Điện toán đám mây là việc cung cấp tài nguyên máy tính cho người dùng tùy theo mục đích sử dụng thông qua kết nối Internet. Nguồn tài nguyên đó có thể là bất kỳ thứ gì liên quan đến điện toán và máy tính, ví dụ như phần mềm, phần cứng, hạ tầng mạng cho đến các máy chủ và mạng lưới máy chủ cỡ lớn.

Người dùng không phải quan tâm đến kỹ năng cài đặt, triển khai và ứng dụng phần mềm hay các yêu cầu về phần cứng như máy chủ, cơ sở hạ tầng truyền thông để truy cập các dịch vụ. Người dùng chỉ cần trả tiền cho chất lượng tương ứng mà họ đã sử dụng.

Để đảm bảo chất lượng dịch vụ trên điện toán đám mây, việc quản lý tài nguyên đã 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.

Tuy nhiên, trong một số trường hợp xấu mà bộ cân bằng tải chưa xử lý kịp hoặc chưa được tính toán đến thì có thể có tài nguyên nhỏ hơn (số lượng máy ảo VM ít hơn) so nhu cầu cần xử lý công việc (các tiến trình có nhu cầu tài nguyên lớn, đặc biệt tài nguyên ở xa) mà người dùng yêu cầu. Trong tình huống như vậy, các loại (tiến trình) xử lí sẽ xung đột cạnh tranh để có được xử lý trên tài nguyên giới hạn (cùng một máy ảo VM) cùng một lúc, dẫn đến tắc nghẽn và đứng máy… hiểu một cách nôm nay là quá tải. Từ đó, gây gián đoạn dịch vụ cho khách hàng, dẫn đến việc thất thoát kinh tế và tài chính. Để giải quyết việc này một cách tốt nhất thì phải có các thuật toán cân bằng tải trên điện toán đám mây. Trong đó, xu hướng áp dụng AI vào tất cả các lĩnh vực đang được triển khai rất mạnh trên thế giới.

Chính vì vậy, thuật toán cân bằng tải trên điện toán đám mây bằng công nghệ AI hiện đại được đề xuất trong luận văn này, đề tài như sau: “Đề xuất thuật toán cân bằng tải trên điện toán đám mây bằng công nghệ AI hiện đại”.

Để tránh được việc gián đoạn dịch vụ, bộ cân bằng tải sẽ làm việc hiệu quả hơn, đặc biệt sẽ càng hiệu quả với việc áp dụng công nghệ trí tuệ nhân tạo (AI), hiệu quả kinh doanh của nhà cung cấp dịch vụ đám mây được cải thiện một cách đáng kể.

Luận văn bao gồm: Phần mở đầu, nội dung gồm bốn chương và 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 hay 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 nhưng 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 của hệ thống. Những yếu tố được đo lường trước khi gán công việc cho một nút tính toán

như thời gian đen, quy 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.

2.1 Lợi ích của điện toán đám mây

Giúp tiết kiệm chi phí: Vì không cần trung tâm dữ liệu tại chỗ nên không cần phải lắp đặt máy chủ, phần cứng, phần mềm...

Truy cập tức thì mọi lúc mọi nơi: Người dùng có thể truy cập vào tài khoản ngay khi đang di chuyển, thông qua bất cứ thiết bị nào, bất kỳ nơi nào trên thế giới miễn là thiết bị đó đang được kết nối với mạng Internet.

Khả năng biến đổi vô tận: Người dùng có thể tùy chọn tạo mô hình đám mây riêng, công cộng, kết hợp (hybrid) hoặc tùy chọn để quyết định vị trí của trung tâm dữ liệu ảo.

Khả năng thích ứng [21]: Có thể chuyển đổi từ mạng riêng sang mạng kết hợp hoặc tạm thời mở rộng dung lượng lưu trữ thì điện toán đám mây có thể làm tất cả một cách suôn sẻ, đáp ứng mọi nhu cầu người dùng.

Hợp tác bền vững, không xáo trộn: Các file được tập trung lưu trữ cố định và nhất quán, tránh được tình trạng bị mất phương hướng khi đang theo dõi dự án.

Bảo mật dữ liệu: Các nhà cung cấp dịch vụ phải luôn đảm bảo rằng hệ thống bảo vệ được cập nhật liên tục và cùng lúc với tất cả các tính năng mới thông qua việc kiểm định chặt chẽ. Tất cả các hoạt động trên đám mây sẽ được bên thứ ba giám sát và kiểm tra thường xuyên để đảm bảo chuẩn an toàn được đáp ứng.

2.2 Các mô hình dịch vụ [3]

Mô hình dịch vụ của điện toán đám mây [19] được các nhà cung cấp dịch vụ chia thành 3 loại lớn:

2.2.1 Cơ sở hạ tầng như một dịch vụ (Infrastructure as a Service - IaaS)

IaaS là một dạng dịch vụ trả tiền theo định mức (pay-per-use) hay chỉ trả tiền cho những gì sử dụng. Dịch vụ này cho phép người sử dụng truy cập vào cơ sở hạ

tầng máy tính từ xa. IaaS bao gồm các máy chủ server, storage lưu trữ và các bảo vệ an ninh nâng cao. Tất cả những yếu tố này giúp cho IaaS trở thành nguồn lực vô giá cho cả doanh nghiệp lẫn cá nhân.

2.2.2 Nền tảng như một dịch vụ (Platform as a Service - PaaS)

Mô hình hệ thống của PaaS cũng tương tự như IaaS nhưng có thêm những công cụ phát triển doanh nghiệp thông minh (BI), middleware, các tool quản lý dữ liệu cũng như các hỗ trợ khác giúp phát triển và triển khai ứng dụng.

2.2.3 Phần mềm như một dịch vụ (Software as a Service - SaaS)

SaaS là một mô hình nổi trội trong điện toán đám mây, cho phép người dùng tận dụng các ứng dụng nền tảng đám mây thông qua Internet. Mô hình dịch vụ này mang đến khả năng truy cập tiện lợi hơn ở mọi góc độ thời gian và vị trí. Chẳng những vậy mà còn giúp doanh nghiệp giảm thiểu phần lớn chi phí ban đầu nhờ loại bỏ được các nhu cầu về server hay các giải pháp backup đắt tiền.

3. Mục đích nghiên cứu

Mục tiêu chính: Đề xuất thuật toán cân bằng tải trên điện toán đám mây bằng công nghệ AI hiện đại.

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 tổng quan các nguyên nhân dẫn đến deadlock.
- Tìm hiểu về các thuật toán trên điện toán đám mây.
- Đề xuất thuật toán.
- Trên cơ sở lý thuyết đã nghiên cứu, luận văn đề xuất thuật toán cân bằng tải trên điện toán đám mây bằng công nghệ AI hiện đại.

4. Đối tượng và phạm vi nghiên cứu

4.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu chính là thuật toán nâng cao hiệu quả cân bằng tải trên điện toán đám mây.

Nghiên cứu áp dụng thuật toán cân bằng tải bằng công nghệ AI hiện đại.

4.2 Phạm vi nghiên cứu

Xây dựng mô hình mô phỏng đám mây ở mức độ nhỏ: khoảng 10 – 20 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, dung lượng dữ liệu gửi về nhỏ: khoảng dưới 1 Mb.

5. Phương pháp nghiên cứu

Phương pháp luận: Dựa trên cơ sở 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 deadlock trên đám mây. Đề xuất ra thuật toán cân bằng tải để dự báo tránh khả năng xảy ra deadlock cao trên đám mây dựa trên thuật toán dự báo AI. 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 - ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY BẰNG CÔNG NGHỆ AI HIỆN ĐẠI

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 là điện toán phân tán và điện toán lưới. Như trong Hình 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 cũng như 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 với tài nguyên điện toán, lưu trữ và mạng cũng như 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 cũng như

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… luôn sẵn sàng và có độ ổn định cao. Ngoài ra còn có 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, mỗi đám mây vẫn là một thực thể độc lập. Nhưng cần kết hợp chúng với công nghệ tiêu chuẩn hoặc độc quyền để làm thành 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. Minh chứng là một loạt các nhà cung cấp lớn, 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, giải phóng người sử dụng khỏi việc phải đầu tư nhân lực, công nghệ và hạ tầng để triển khai hệ thống. Từ đó, 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, kế thừa các mạng lưới trước đây cũng như các khái niệm máy tính phân tán. Mục đích chính là 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 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 (ondemand service). Ngoài ra còn cung cấp khả năng truy cập dịch vụ qua mạng rộng rãi từ máy tính để bàn và 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, đáp ứng nhanh theo nhu cầu từ thấp lên 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 [17] hoàn chỉnh bằng cách tập hợp các phần cứng, phần mềm và các thiết bị có liên quan lại với nhau nhằm 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 tái sử dụng 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 nhanh, gọn và nhẹ hơn. 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 [5] 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… Trên thực tế, người dùng chỉ truy cập vào thiết bị đầu cuối để truy xuất vào các tài nguyên trên điện toán. Còn 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 cho đến thời gian hoàn thành nhiệm vụ.

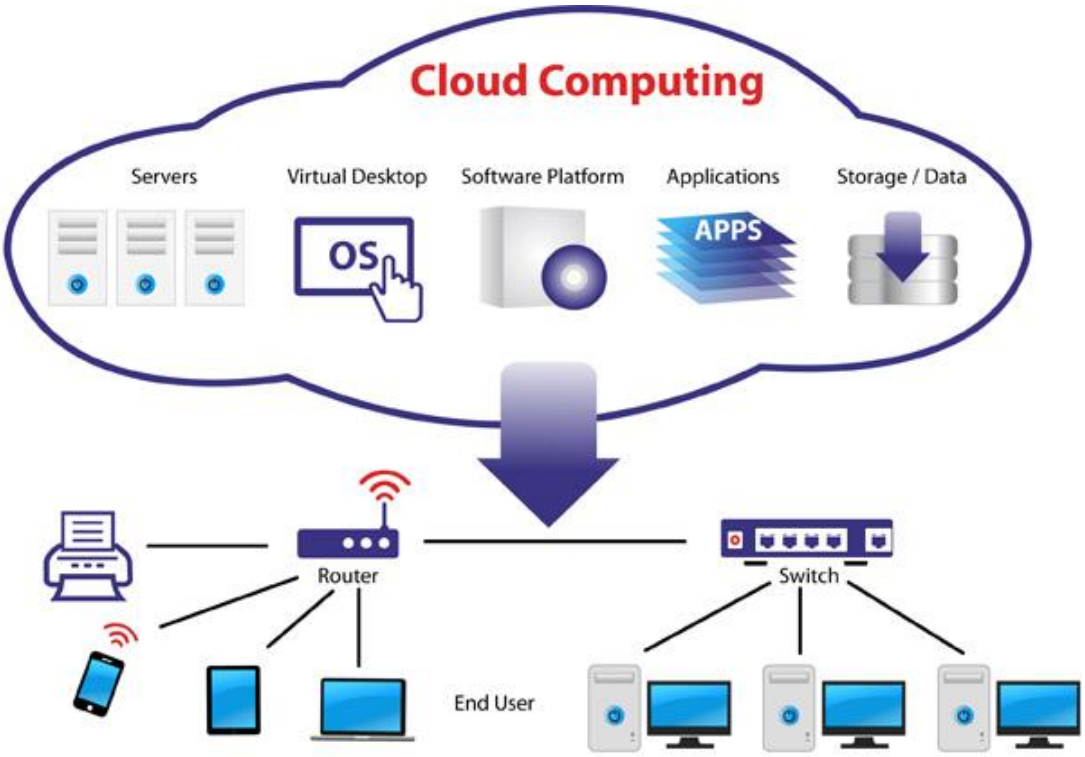
Điện toán đám mây [6] đang chuyển đổi ngành công nghệ thông tin, thay đổi cách thức sử dụng và cung cấp phần mềm cũng như phần cứng. Làm cho việc sử dụng các tài nguyên máy tính theo yêu cầu như băng thông, lưu trữ hoặc các ứng dụng phần mềm và điện toán có sẵn trở nên dễ dàng hơn bao giờ hết. Nó che giấu được 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 các máy bị quá tải đến các máy không được sử dụng. 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 ra đời. 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 lại rất nhẹ khi lưu lượng mạng truy cập và dữ liệu tăng lên nhanh chóng. 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 của dữ liệu lớn và phát triển của điện toán đám mây, khi giải quyết bài toán công việc bằng các máy ảo giao dịch với dữ liệu, sẽ mang lại nhiều chi phí truyền thông giữa các máy chủ trong quá trình di chuyển và tính toán. Qua đó, 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

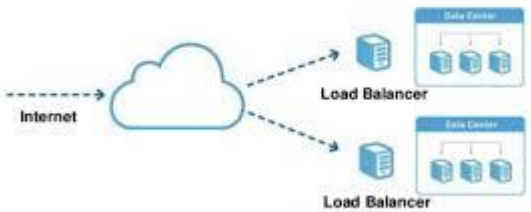
lực cùng với nhiệm vụ lập kế hoạch giữa các hệ thống phân phối. Trong cân bằng tải truyền thống, ta phải đối mặt với một số vấn đề khác nhau của các giai đoạn cung cấp tài nguyên trong môi trường đám mây. Nó cũng có tác động to lớn trong các hệ thống đám mây về hiệu suất và vấn đề đo lường do sự tham 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 rồi sử dụng nó từ bất kỳ nơi nào trên thế giới. Nó đã làm cho phần cứng được yêu cầu trở nên linh hoạt hơn nhiều. Do đó, mọi người có cơ hội sử dụng nhiều tài nguyên khi cần và chỉ phải trả số tiền cho khoảng thời gian họ đã sử dụng nguồn dung lượng cụ thể. Dịch vụ đó được gọi là dịch vụ trả tiền cho mỗi lần sử dụng, đã góp phần làm cho ngành công nghiệp công nghệ thông tin hướng đến gần hơn việc kinh doanh điện toán đám mây. Giống như một CPU có 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.



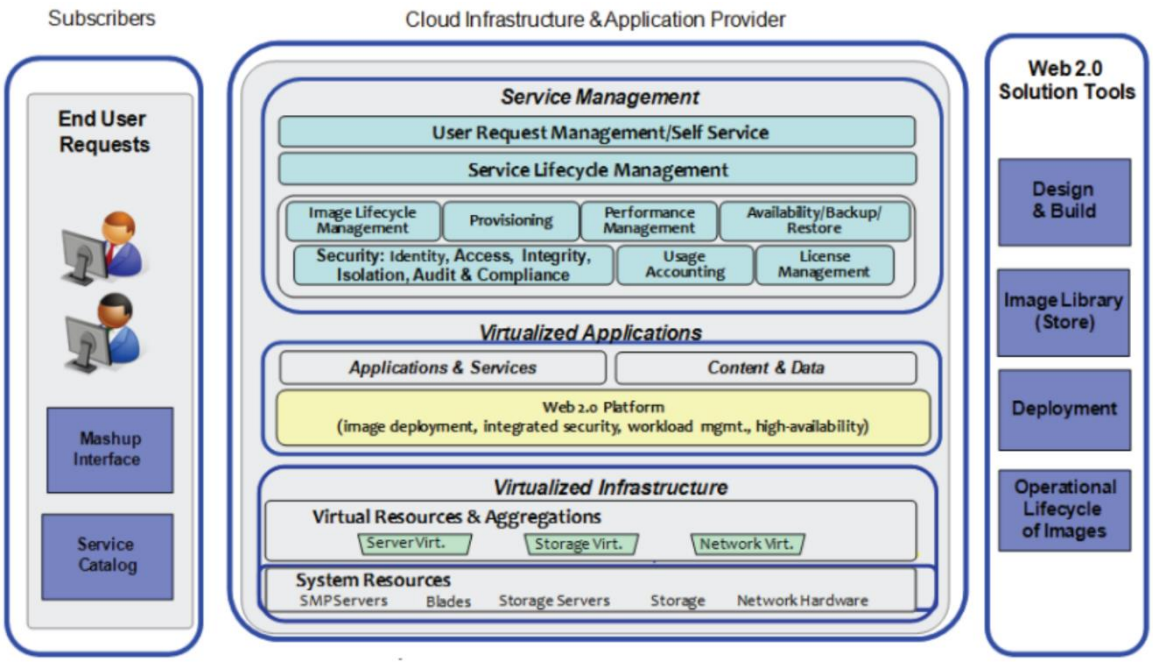
Hình 1.2: Cung cấp tài nguyên đám mây [4]

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. Khi người sử dụng 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) 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, ta phải đối mặt với các vấn đề suy giảm và những vấn đề khác mà 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 đúng với 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

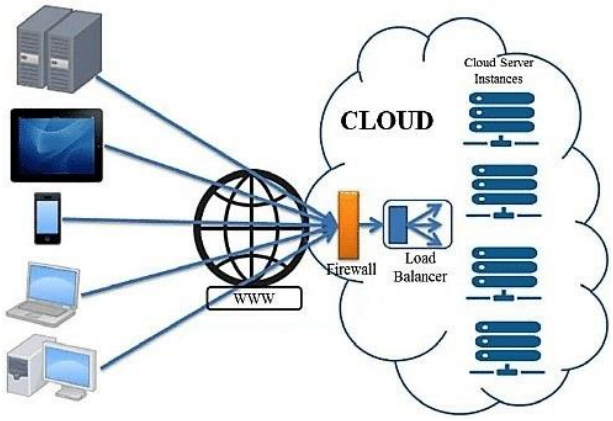
1.2.1 Giới thiệu về cân bằng tải

Ngày nay, ngành công nghiệp CNTT đang phát triển mỗi ngày và nhu cầu về tài nguyên lưu trữ và tính toán cũng ngày càng tăng. Một lượng lớn dữ liệu được tạo ra 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 đang ngày càng được rất nhiều người truy cập và sử dụng. Khi lượng truy cập này quá lớn thường xảy ra 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[15].

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, 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. Bởi, Cloud Computing 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 mọi nơi qua thông qua Internet. Cân bằng tải cho điện toán đám mây đã trở thành một lĩnh vực nghiên cứu vừa thú vị lại quan trọng. Cân bằng tải nhằm mục đích thỏa mãn người dùng 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 nhằm tăng cường 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, giải pháp 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. 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 một lúc với tất cả chúng. 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 ra 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 đã sử dụng một phần thông tin của đường dẫn để tìm được 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, 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 giải pháp: 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, 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 [14]. 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 tính không đồng nhất của tài nguyên và nhu cầu không nhất quán của người sử dụng 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 mà một số các nút máy tính bị quá tải hay một số nút có tải nhẹ. Từ đó mà 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 có tầm ảnh hưởng đến hiệu suất của DLB. Tuy nhiên, một số thuật toán ở trên không thể đáp ứng được sự lựa chọn cũng như 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 cũng không phải là lý tưởng. Thế nên, nó có thể đảm bảo cân bằng tải và sử dụng hiệu quả tài nguyên vật lý của toàn bộ cụm. Mặt khác, 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 và 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 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 phía 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. 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 cũng 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 sẽ 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, đảm bảo tính linh hoạt và mở rộng cho hệ thống [18].

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, đồng thời đảm bảo cho người dùng không bị gián đoạn dịch vụ khi lỗi sự cố xảy ra 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 (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 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 vì công nghệ phần cứng và phần mềm được cải tiến. Các cá nhân và tổ chức trên một số các ngành công nghiệp đang bắt đầu nhận ra tiềm năng của AI để cải thiện các hoạt động hiện tại và nghiên cứu AI đã được 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 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 (Hình 1). Trong học tập có giám sát, máy được cung cấp tập dữ liệu, với mỗi ví dụ trong tập dữ liệu đã được gắn nhãn kèm theo câu trả lời. Sau đó, các máy học thông qua việc 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 [20]. 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 và mất mát gói tin khi truyền dữ liệu qua môi trường điện toán. Mục đích cân bằng tải là để làm tăng hiệu năng của hệ thống.

CHƯƠNG 2 - CÁC CÔNG TRÌNH LIÊN QUAN

2.1 Tình hình nghiên cứu trong nước

Trong bài báo [5] của tác giả Trần Công Hùng & các cộng sự đã nghiên cứu và đề xuất các giải pháp nhằm nâng cao hiệu suất trong điện toán đám mây, đặc biệt là về cân bằng tải dựa vào thời gian đáp ứng. Các tác giả đã đưa ra công trình nghiên cứu về các tham số của tính hiệu quả nhằm cân bằng tải trong đám mây (Study the effect of Parameters to load balancing in cloud computing), trong đó chỉ rõ rằng các kỹ thuật cân bằng tải có rất nhiều cách giải quyết: (i) Cân bằng tải sau khi máy chủ bị quá tải; (ii) Cân bằng tải và dự đoán tải tiếp theo nhằm phân bổ tài nguyên; (iii) Cải thiện các tham số ảnh hưởng đến cân bằng tải trên đám mây. Trong nghiên cứu này cũng đề xuất một số phương pháp nhằm nâng cao hiệu quả cân bằng tải và tăng hiệu suất hoạt động của đám mây.

Trong bài báo [6] 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, tiềm năng dự báo trong tương lai gần.

Trong bài báo [7] 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. Đồng thời, lên lịch cho các chính sách cung cấp tài nguyên để phân bổ tài nguyên không đồng nhất[13]. 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.

Bên cạnh đó, có rất nhiều nghiên cứu và bài báo từ Việt Nam được công bố rộng rãi về cân bằng tải trên đám mây. Tuy nhiên, đa số đều ở mức thực nghiệm mô phỏng và chưa áp dụng vào thực tế công nghệ cloud hiện tại do tính chất quy mô của đề tài nghiên cứu.

2.2 Tình hình nghiên cứu trên thế giới

Trong bài báo [8] *"Deadlock Avoidance through Efficient Load Balancing to Control Disaster in Cloud Environment"* của nhóm tác giả Mahitha. O và Suma. V, Ấn Độ năm 2013, đã trình bày một kỹ thuật cân bằng tải hiệu quả để kiểm soát thảm họa trên môi trường điện toán đám mây. Thuật toán đề xuất được áp dụng để cân bằng tải, đề cập đến thời gian đáp ứng tổng thể, thời gian xử lý và thông lượng. Thuật toán cải tiến được triển khai bằng cách sử dụng công cụ Cloud Analyst. Các kết quả mô phỏng thu được tốt hơn với thời gian đáp ứng tổng thể và thời gian xử lý cũng được cải thiện.

Trong bài báo năm 2012 [9] *"Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud"* của nhóm các tác giả Rashmi. K. S, Suma. V, Vaidehi. M, Ấn Độ đã đề cập đến vấn đề là cần thiết phải có một kỹ thuật cân bằng tải hiệu quả để tránh các bế tắc. Đó là phương pháp cân bằng tải nâng cao sử dụng hiệu quả hệ thống quản lí đám mây. Để đạt được mục tiêu đã đề cập ở trên, trong bài báo này các tác giả đã đề xuất một thuật toán cân bằng tải. Trong môi trường điện toán đám mây, tải liên quan đến số lượng yêu cầu phải được phục vụ bởi các máy ảo có sẵn trong đám mây. Thuật toán đề xuất tránh bế tắc bằng cách cung cấp các nguồn lực theo yêu cầu dẫn đến tăng số lần thực hiện công việc. Bài báo đã thảo luận các vấn đề hiện còn tồn tại cũng như đề xuất thuật toán và mô hình hóa thuật toán theo công thức toán học. Đồng thời, bài báo cũng đưa ra các thiết lập để mô phỏng thuật toán đã đề xuất, so sánh kết quả của thuật toán đề xuất với các thuật toán hiện có và trình bày các kết quả đạt được.

Năm 2013, trong bài báo [10] *"Intelligent Computing Relating to Cloud Computing"* của tác giả B S Panda Asst và nhóm sinh viên nghiên cứu, trình bày các khái niệm về trí tuệ nhân tạo trong điện toán đám mây.

Năm 2018, Usman Ahmen và các cộng sự [11] đã công bố nghiên cứu về *"RALB-HC: A resource-aware load balancer for heterogeneous cluster"*, trong nghiên cứu này, một bộ cân bằng tải nhận biết tài nguyên mới cho cụm không đồng nhất (RALB-HC) được đề xuất để phân phối khối lượng công việc dựa trên tính toán khả năng của tài nguyên và nhu cầu tính toán của ứng dụng. RALB-HC sử dụng học máy có giám sát để phân loại các ứng dụng bằng cách sử dụng các tính năng mã tĩnh. Khung RALB-HC bao gồm hai giai đoạn: (1) Lập bản đồ công việc dựa trên sự sẵn có của các nguồn lực; (2) Cân bằng tải nhận biết tài nguyên để đạt được tỷ lệ sử dụng tài nguyên cao hơn. Kết quả thử nghiệm trên một tập hợp lớn khối lượng công việc tổng hợp và trong thế giới thực cho thấy RALB-HC giảm thời gian thực hiện 31,61%, tăng tỷ lệ sử dụng tài nguyên lên 67,8% đồng thời cải thiện 147,35% minh bạch so với bộ lập lịch cơ sở.

Năm 2018, Priyanshu Srivastava và Rizwan Khan [1] đã công bố nghiên cứu về *"A Review Paper on Cloud Computing"*, nghiên cứu này nhận định về những lợi ích mà điện toán đám mây mang lại. Đồng thời cũng so sánh sự khác nhau trước và sau khi xuất hiện công nghệ điện toán đám mây.

Năm 2019, Hogarty Daniel và các cộng sự [4] đã công bố nghiên cứu *"Artificial Intelligence in Dermatology—Where We Are and the Way to the Future: A Review"*, đã nêu lên các định nghĩa cơ bản về trí tuệ nhân tạo. Đây là cơ sở lý thuyết về trí tuệ nhân tạo được sử dụng cho đề cương này.

Năm 2019, Michael Haenlein và Andreas Kaplan [12] đã công bố nghiên cứu *"A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence"*, nghiên cứu này khái quát lịch sử phát triển của trí tuệ nhân tạo cũng như dự đoán các hướng phát triển của trí tuệ nhân tạo trong thời gian tới. Nghiên cứu này cũng đóng góp các lý thuyết về trí tuệ nhân tạo cho đề cương này.

2.3 Tổng kết chương

Trong chương này thông qua việc nghiên cứu tìm hiểu được một số thuật toán cũng như những công trình liên quan đến cân bằng tải trong điện toán đám mây, giúp luận văn này 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 việc nghiên cứu đề tài của luận văn này.

CHƯƠNG 3 - ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY BẰNG CÔNG NGHỆ AI

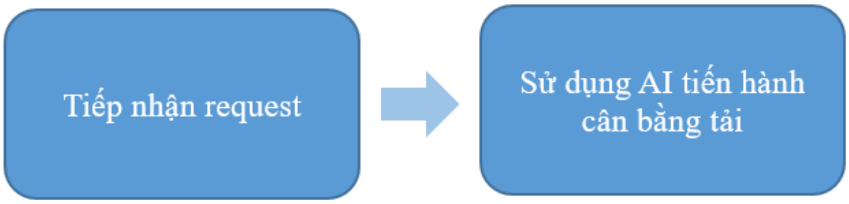
3.1 Giới thiệu chung

Ngày nay, các thuật toán trong cân bằng tải đã được nhiều bài báo nêu lên và cải tiến nhằm nâng cao hiệu năng cân bằng tải cũng như cải tiến thời gian xử lý/thực hiện nhiệm vụ. Ngoài ra, thời gian hoàn thành và tài nguyên của máy ảo cũng đã giảm thiểu được sự mất cân bằng tải trong môi trường điện toán đám mây, tránh được tình trạng quá tải trên máy chủ. Các thuật toán cùng với công nghệ AI hiện đại đã góp phần làm tăng tính hiệu quả một cách đáng kể trong việc phân chia tác vụ trên cloud, tăng tính bảo mật, đảm bảo tính linh hoạt và mở rộng của hệ thống. Với những mục đích đã nêu trên, chương này sẽ trình bày ý tưởng thuật toán đề xuất nhằm nâng cao khả năng cân bằng tải dựa trên công nghệ AI hiện đại để ứng dụng vào mô hình cloud với hệ thống host và các máy ảo.

3.2 Mô hình nghiên cứu

Để phân loại các request, mô hình cân bằng tải trên điện toán đám mây sử dụng thuật toán phân lớp Decision Tree – một trong những thuật toán được đánh giá cao trong việc phân lớp. Ngoài ra, mô hình cũng sử dụng thêm thuật toán K-means để tiến hành phân cụm các máy ảo (các host) có tính sẵn sàng cao, vừa và thấp. Các request có yêu cầu phải xử lí nhiều sẽ được phân bổ vào các máy ảo có tính sẵn sàng cao (máy ảo có mức độ hoạt động thấp) và tương tự như vậy, các request có yêu cầu xử lí vừa hay ít hơn sẽ được phân bổ vào các máy ảo tương ứng. Theo hướng tiếp cận này, thuật toán đề xuất sẽ tối ưu được thời gian xử lí các request trên cloud và có thể ứng dụng trên môi trường cloud theo thời gian thực. Mô hình cân bằng tải gồm 2 bước thực hiện chính:

- Bước 1: Nhận thông tin từ các request.
- Bước 2: Sử dụng các thuật toán trong trí tuệ nhân tạo để tiến hành cân bằng tải.



Hình 3.1: Mô hình cân bằng tải

Về mục tiêu:

- Hệ thống máy chủ giảm thiểu được các rủi ro nhất định.

- Thời gian hoạt động cho các yêu cầu trong điện toán đám mây được giảm thiểu.

- Ngăn chặn mất cân bằng tải đồng thời hạn chế tối đa sự mất cân bằng tải giữa các máy ảo.

- Các yêu cầu được giải quyết nhanh chóng [22], đồng thời dự đoán thời gian xử lí của các máy chủ cũng như tổng thời gian hoàn thành cho từng hay toàn bộ yêu cầu.

Giả định:

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

- Quá trình nghiên cứu sẽ tập trung vào dịch vụ Web (Web Service).

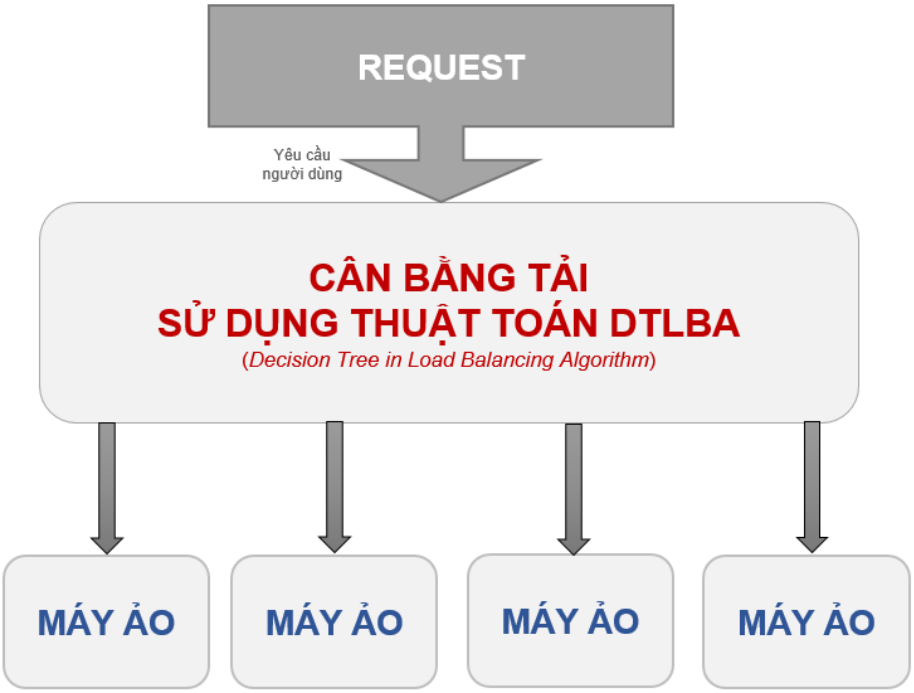
- Thời gian xử lý của từng dịch vụ chạy trên web và trên từng máy ảo sẽ được thông tin trước cho các máy chủ web.

- RAM, vi xử lý và I/O nếu có cấu hình tương tự nhau thì thời gian thực thi của các dịch vụ sẽ có sự chênh lệch không đáng kể.

Mô hình nghiên cứu:

- Mô hình dự báo tác vụ trên điện toán đám mây được thực hiện như sau:

  - Bước 1: Nhận thông tin từ các request.
  - Bước 2: Sử dụng các thuật toán trong trí tuệ nhân tạo để tiến hành quá trình cân bằng tải.



Hình 3.2: Cân bằng tải sử dụng thuật toán DTLBA

Thuật toán đề xuất là nơi xử lý các yêu cầu và đưa vào các máy ảo phù hợp để cân bằng tải.

Trong mô hình này sử dụng Regression (dựa vào công nghệ AI hiện đại) để phân loại các request đầu vào và dự báo các thông số cloud cần để xử lý task mà request này đem đến (Power, CPU Usage, RAM Usage). Để phân lớp với kỹ thuật Regression này, thuật toán sử dụng bộ data trong lịch sử cloud được lưu lại (sử dụng dữ liệu gần nhất).

Sau đó với số liệu (Power, CPU Usage, RAM Usage) mà cloud cần có để xử lý Task/Job tương ứng đã được tính toán ở trên, thuật toán được sử dụng tiếp theo là DT, dùng để phân lớp Task/Job. Trong đó, bộ dữ liệu là dữ liệu thực đã được lưu lại kết hợp với dữ liệu dự đoán mới tính toán ra ở trên và phân lớp các tác vụ dựa vào độ ưu tiên. Từ đó, phân bổ vào các máy ảo tương ứng.

Mô hình này là mô phỏng thuật toán một cách tự nhiên và lên kế hoạch cho các yêu cầu tiếp theo để không bị mất cân bằng tải. Theo thuật toán này sẽ giảm được

các tải liên lạc giữa máy ảo và các nguồn tài nguyên hiện có. Vì vậy, giảm được băng thông và thông lượng không cần thiết, tăng phục vụ cho yêu cầu người dùng.

3.3 Thuật toán K-mean

Thuật toán phân cụm K- mean là một trong những thuật toán khai thác dữ liệu mạnh mẽ và mạnh mẽ nhất trong cộng đồng nghiên cứu. Tuy nhiên, mặc dù mức độ phổ biến của nó, thuật toán có một số hạn chế nhất định, bao gồm các vấn đề liên quan đến khởi tạo ngẫu nhiên của centroid dẫn đến sự hội tụ bất ngờ. Ngoài ra, một thuật toán phân cụm như vậy yêu cầu số lượng cụm được xác định trước, chịu trách nhiệm cho các hình dạng cụm khác nhau và hiệu ứng ngoại lệ. Một vấn đề cơ bản của thuật toán K-mean là không có khả năng xử lý các loại dữ liệu khác nhau.

Thuật toán k-mean phụ thuộc vào giá trị của k, luôn cần được chỉ định để thực hiện bất kỳ phân tích phân cụm nào. Việc phân cụm với các giá trị k khác nhau cuối cùng sẽ tạo ra các kết quả khác nhau.

Kmeans 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).

3.4 Thuật toán Decision Trees

Decision Trees (DTs) là một thuật toán học giám sát phi tham số (non-parametric supervised learning) sử dụng cho các bài toán phân lớp (classification) và hồi quy [16] (regression). Ý tưởng của thuật toán là tạo ra mô hình để dự đoán giá trị của một biến mục tiêu bằng cách học các quy luật được suy ra từ các đặc trưng của dữ liệu.

Một vài lợi ích của thuật toán Decision Trees:

- Thuật toán tương đối dễ hiểu, dễ dàng diễn giải và hình dung.
- Decision Trees ngầm thực hiện và "sàng lọc biến" (variable screening) hoặc lựa chọn đặc trưng (feature selection).
- Decision Trees không yêu cầu quá nhiều các bước tiền xử lí dữ liệu.
- Mối quan hệ phi tuyến tính giữa các tham số không ảnh hưởng đến hiệu suất của cây.

Một điểm đáng lưu ý của Decision Trees là nó có thể làm việc với các đặc trưng (trong các tài liệu về Decision Trees, các đặc trưng thường được gọi là *thuộc tính – attribute*) dạng *categorical*, thường là rời rạc và không có thứ tự. Ví dụ, *mưa, nắng* hay *xanh, đỏ,* v.v. Decision Trees cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorical và liên tục (*numeric*). Một điểm đáng lưu ý nữa là Decision Trees ít yêu cầu việc chuẩn hoá dữ liệu.

3.5 Đề xuất thuật toán dự báo thời gian tải tối đa/tối thiểu trong ngày nhằm nâng cao hiệu quả cân bằng tải của điện toán đám mây

Dựa vào yếu tố thời gian xử lý (Makespans) của các request và một số thuộc tính khác, ta sử dụng thuật toán Decision Trees để phân lớp các request này. Từ đó, ta biết cách phân bổ tài nguyên cho các request này. Song song đó, các tài nguyên (máy ảo/host) được phân cụm theo mức độ sử dụng. 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.

Luận văn này xin đề xuất thuật toán DTLBA *(Decision Trees in load balancing algorithm)* gồm 3 nhóm Module chính:

(1) Module tính toán ra các thông số của request bằng thuật toán Decision Trees:

Trong Module này, thuật toán Decision Trees sẽ dựa vào các thuộc tính của request mà tính toán ra thời gian xử lý của request đó, từ đó phân lớp Request này. Các thuộc tính bao gồm: Size, Response Length, Max Length,…

Nhóm Thời Gian xử lý = $MK_{New} = DT(X_1, X_2,\ldots, X_n)$

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

Ở đây có thể chia thành nhiều nhóm (từ 4 ~10 nhóm) hoặc hơn nữa dựa vào độ biến thiên của Request.

(2) Module phân lớp tác vụ theo độ ưu tiên:

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, 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, …);

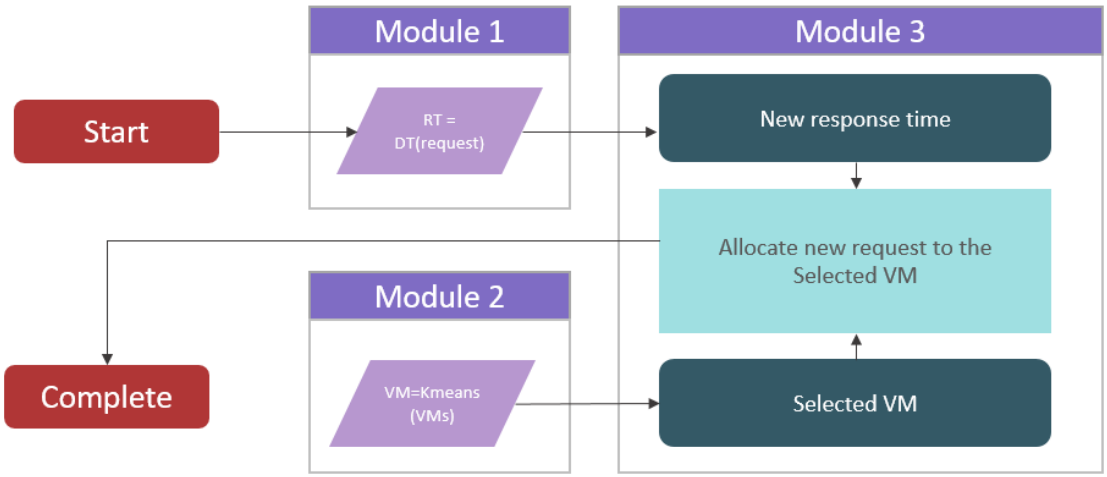
Trong đó: i = 1 là nhóm thấp

i = 2 là nhóm trung bình

i = 3 là nhóm cao

(3) Module phân bổ 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 cụm máy ảo phù hợp. Nếu một yêu cầu được gửi đến thì yêu cầu này được phân loại bởi Module 1. Còn các VM đang xét kể cả VM không tải cũng được phân cụm theo Module 2. Sau đó thuật toán sẽ tính toán ra request nào phù hợp nhất với máy ảo nào thông qua thông số trả về của 2 hàm DT và K-Means ở trên. Nếu thời gian xử lý tính toán của Request đang xét (được tính toán từ module 1) nhỏ nhất thì yêu cầu này sẽ được xử lý trên VM với mức độ xa means nhất (tức thuộc nhóm 1 và có mức độ sử dụng thấp nhất). Đối với các request không nhỏ không lớn ta có thể dùng các phương pháp tính toán như loại suy hay sai phân để tính toán việc phân bổ.



Hình 3.4: Sơ đồ của thuật toán DTLBA

Thuật toán DTLBA

```
1.    For each Request in CloudRequests
2.          isLocated = true;
3.          RT_new = DT(RT1, RT2…..);  // Module 1
4.          VM_Cluster = kMeans(situation);  // situation:
Trạng thái của các VM Module 2
5.          For each VM in VMList
6.                If  isFitSituation(Request.RT_new ,
VM.VM_Cluster)
7.                      AllocateRequestToVM(VM, Request);
// Module 3
8.                      isLocated = true;
1.                      break;
2.                End If
3.          End For
4.          If (!isLocated)
5.                VM = VMList.getMinFromMean();  // Module
2
6.                AllocateRequestToVM(VM, Request);
7.          End If
8.    End For
```

Phương pháp đánh giá thuật toán DTLBA

Thuật toán đề xuất cho thấy các kết quả khả quan trong việc cải thiện thời gian phản hồi và xử lí tác vụ của đám mây trung tâm cũng như hạn chế số lượng yêu cầu xếp hàng để phân phối một cách hợp lí. So với các thuật toán cũ như **MaxMin, Round Robin, MinMin** và **FCFS**, thuật toán đề xuất có khả năng tối ưu hóa quá trình cân bằng tải và hiệu năng của điện toán đám mây, thể hiện được sự vượt trội và tính ổn định cao hơn.

3.6 Kết luận chương

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

Trong chương này trình bày về cài đặt mô phỏng thuật toán được đề xuất về cân bằng tải dựa vào công nghệ AI hiện đại. Từ kết quả thực nghiệm mô phỏng cho thấy một phương pháp mới giải quyết vấn đề cân bằng tải, cụ thể là sử dụng thuật toán dự báo Decision Trees phân loại request. Từ đó đưa ra cách giải quyết phân phối tài nguyên một cách hợp lý đến các máy ảo có mức độ hoạt động thấp nhất. Với cơ chế hoạt động này, thuật toán đề xuất DTLBA sẽ tối ưu hoá thời gian xử lý cân bằng tải trên cloud và có khả năng ứng dụng trên môi trường cloud theo thời gian thực. Sau khi kết thúc các bước thực nghiệm, kết quả thu được sẽ được phân tích và so sánh với các thuật toán khác, từ đó chứng minh được tính hiệu quả của thuật toán đề ra.

4.2 Các thông số đầu vào, môi trường thực nghiệm

* Đầu vào là các request của người dùng.
* Sử dụng Java và CloudSim để thực hiện nghiên cứu này.

Dựa vào dữ liệu của các request đã biết, ta có thể sử dụng thuật toán Regression để phân loại request bằng cách tính toán ra bộ Priority = {Power, CPU, RAM}, từ đó biết cách phân bổ 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 có từ 5 đến 15 máy ảo, tạo môi trường request ngẫu nhiên đến các dịch vụ trên cloud. Bao gồm dịch vụ cung cấp máy ảo, dịch vụ cung cấp và đáp ứng thử nghiệm của người dùng CloudSim.

Cài đặt thuật toán Regression, DT trên môi trường mô phỏng được phát triển bởi bộ thư viện Weka và kiểm nghiệm ra kết quả.

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 APACHE NETBEAN IDE để chạy thử, sau đó 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/), kết hợp với bộ thư viện về datamining là WEKA.

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***
- 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)	10000 MB
Mips	512 MB
RAM	250
Bandwidth	1000

Số lượng CPU (pes no.)	1
VMM	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 20 → 1000.

Bảng 4.3: Cấu hình thông số các Request

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

- Thuật toán đề xuất được xây dựng bằng cách tạo ra lớp **DTLBASchedulingAlgorithm** kế thừa từ đối tượng **BaseSchedulingAlgorithm**. Ngoài ra còn cập nhật thêm một số phương thức và thuộc tính liên quan tới predictRequestRegression nhằm điều chỉnh các hàm dựng sẵn để phù hợp với thuật toán đề xuất:

```
@Override
public void run()  // Module 3
public CondorVM getMostFreeVM(String vmClass)
// Module 2
public String predictRequestDT(Cloudlet req)
// Module 1
```

Tiêu chí đánh giá:

Sử dụng thuật toán cân bằng tải có sẵn của CloudSim và thuật toán đề xuất mới cài đặt được để chạy thực nghiệm mô phỏng cloud với các tham số ở trên. Cả hai thuật toán có cùng đầu vào để phục vụ cho quá trình so sánh kết quả đầu ra, đặc biệt là thông số thời gian xử lý (Makespan).

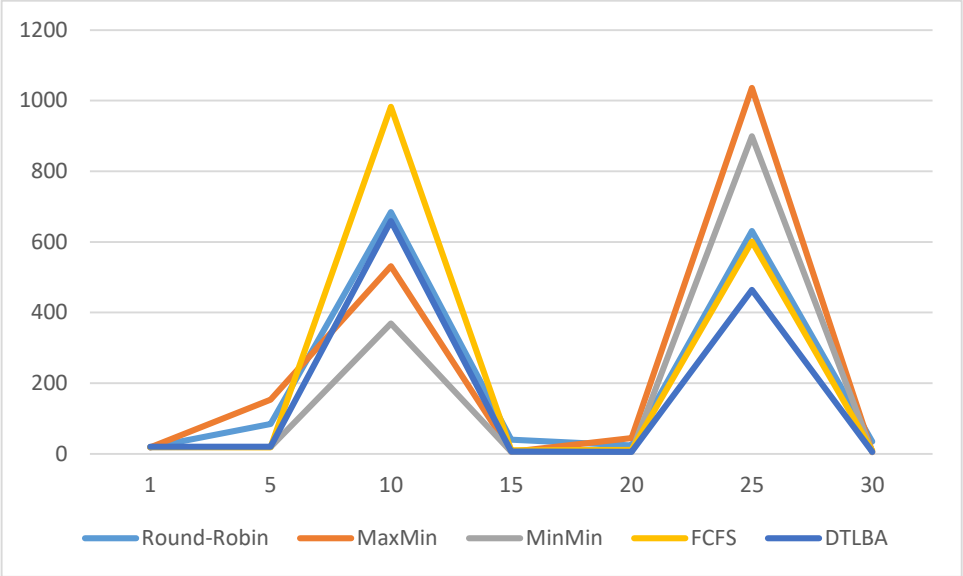
Các máy ảo và cloud có thời gian xử lí với sai số càng thấp thì hiệu quả của thuật toán càng đạt được kết quả tốt.

4.3 Kết quả thực nghiệm của mô hình.

Kết quả chạy thực nghiệm mô phỏng trên CloudSim với 5 máy ảo được dựng sẵn để đáp ứng các yêu cầu. Các yêu cầu được khởi tạo với chiều dài và kích thước ngẫu nhiên. Cùng số lượng Request từ 1-30, đem so sánh với các thuật toán Round-Robin, MaxMin, MinMin và FCFS với thời gian thực hiện là:

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

Số lần Request	Round-Robin	MaxMin	MinMin	FCFS	DTLBA
1	18.1	18.69	18.53	18.87	19.82
5	84.58	153.07	19.22	18.36	20.05
10	684.37	531.03	368.76	982.36	659.43
15	39.63	5.74	5.91	10.62	6.22
20	23.61	45.06	6.07	10.92	5.53
25	630.88	1036.08	899.76	601.5	463.99
30	34.27	4.97	5.11	9.19	5.38

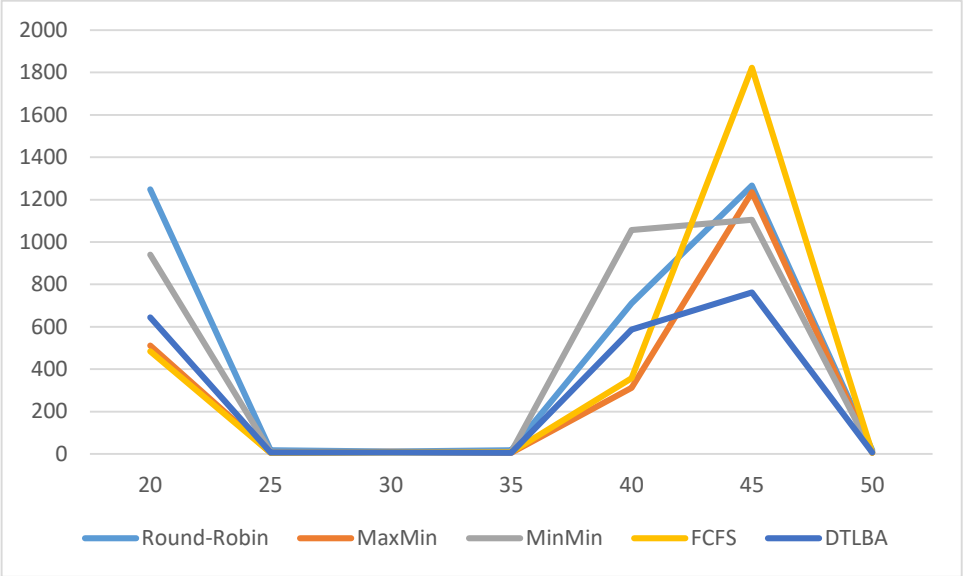


Hình 4.1: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 30 Request

Ở 10 Request đầu tiên, thuật toán MinMin có phần chiếm ưu thế hơn về thời gian xử lý so với các thuật toán còn lại, thuật toán FCFS lại tốn quá nhiều thời gian. Trong khi đó, thời gian xử lý tác vụ của thuật toán DTLBA vẫn duy trì ở mức trung bình. Nhưng từ Request 15 – 20 thì cả 5 thuật toán đã không còn cách biệt quá nhiều về độ chênh lệch thời gian xử lý các tác vụ trên cloud. Tuy nhiên từ Request 20 trở đi, các thuật toán lại có sự biến chuyển mới, thuật toán DTBLA đã bứt phá vươn lên vị trí dẫn đầu, thuật toán MinMin đã tụt hạng xuống gần cuối và thuật toán FCFS đã lên được mức trung bình.

Bảng 4.5: Kết quả thực nghiệm mô phỏng với 50 Request

Số lần Request	Round-Robin	MaxMin	MinMin	FCFS	DTLBA
20	1248.77	511.93	940.91	483.94	645.0
25	17.58	5.9	9.77	5.95	6.28
30	11.7	10.02	9.35	6.51	6.35
35	17.38	5.7	10.15	8.17	6.19
40	710.6	312.47	1056.66	356.97	587.13
45	1266.75	1234.5	1105.19	1822.32	762.82
50	17.56	5.89	9.75	5.94	6.28



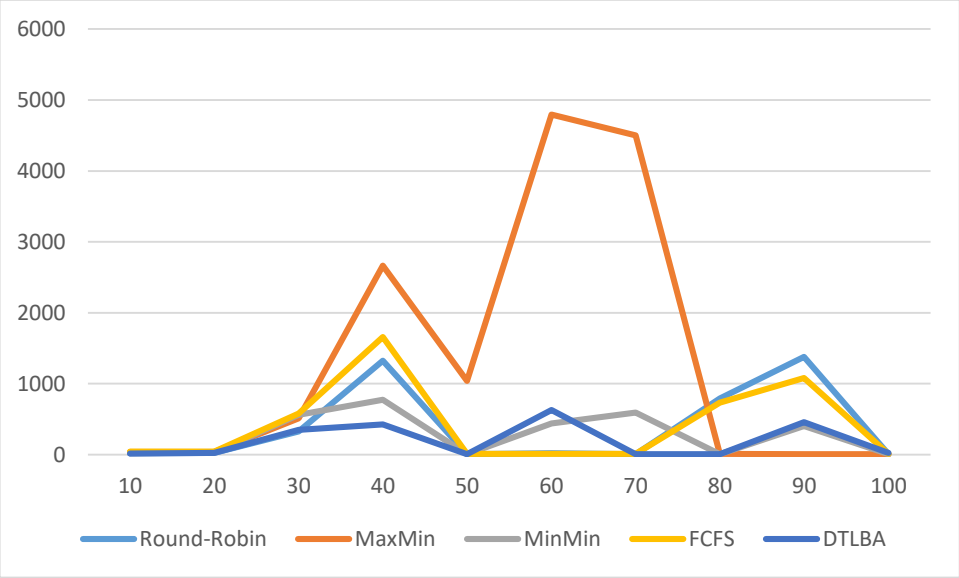
Hình 4.2: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 50 Request

Với sự khởi đầu không quá sai biệt ở Request 20 – 35, các thuật toán gần như có sự đồng nhất về thời gian xử lý. Đặc biệt, các Request từ 25 – 35 khi sự chênh lệch về thời gian xử lý giữa 5 thuật toán là không đáng kể. Nhưng từ Request 35 giữa các thuật toán xử lý tác vụ trên cloud lại có một sự chuyển mình đầy bất ngờ. Cụ thể, thuật toán DTLBA luôn nằm trong top những thuật toán có thời gian xử lý tác vụ hiệu quả, thuật toán FCFS lại bị tụt hậu so với những thuật toán còn lại khi chiếm thời gian gần như gấp đôi thuật toán DTLBA ở Request 45.

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

Số lần Request	Round-Robin	MaxMin	MinMin	FCFS	DTLBA
10	19.8	38.36	21.85	45.17	17.73
20	22.99	38.01	31.18	45.44	23.64
30	327.99	514.68	564.03	580.27	351.49
40	1323.85	2663.58	774.97	1657.55	426.93
50	5.75	1041.66	6.55	12.56	5.74
60	20.95	4795.18	439.17	13.02	629.79
70	5.15	4500.71	595.03	13.55	5.25

80	793.39	10.15	5.43	735.17	5.77
90	1378.11	6.42	405.95	1078.82	460.56
100	5.56	8.22	6.49	12.08	26.24



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

Với kết quả thực nghiệm ở 30 Request trở lại, ta thấy thời gian thực hiện của 5 thuật toán là gần như tương đương nhau, trong đó thời gian của thuật toán Round-Robin và DTLBA bám đuổi nhau để giành vị trí dẫn đầu. Các Request từ 30 – 50 chính là lợi thế của thuật toán DTLBA đồng thời cũng là sự bất lợi của thuật toán MaxMin. Cụ thể, khi ở Request 40, thời gian thực hiện của thuật toán MaxMin chiếm gấp 3 lần thời gian thực hiện của thuật toán DTLBA.

Ở Request 50, các thuật toán đồng loạt giảm mạnh thời gian xử lý tác vụ trên cloud, trong khi đó thời gian xử lý của thuật toán MaxMin có sự chênh lệch rất lớn so với các thuật toán còn lại. Đặc biệt, so với thuật toán đang chiếm vị trí đầu bảng là DTLBA thì thời gian của thuật toán MaxMin gấp 200 lần.

Kết quả thực nghiệm từ Request 50 – 70, các thuật toán vẫn giữ được sự bền vững trong thời gian thực hiện của mình, đặc biệt là thuật toán Round-Robin và FCFS luôn chiếm giữ 2 vị trí đầu bảng. Tuy nhiên, ở Request 70, thời gian xử lý của thuật

toán MaxMin đã đến mức tối đa, gấp tới 900 lần thời gian của thuật toán đứng đầu Request 70 hiện giờ là Round-Robin.

Các Request từ 70 – 80 chính là sân chơi của thuật toán DTLBA khi đối thủ là thuật toán Round-Robin đương kim vô địch ở Request 70 đã nhanh chóng bị đánh bại bởi thuật toán MinMin ở Request 80. Thuật toán FCFS đang ở vị trí thứ ba cũng phải nhường ngôi vị của cho thuật toán MaxMin.

Như đã biết, nếu cái gì đã lên đến đỉnh điểm thì sẽ bắt đầu đổ dốc xuống. Thuật toán MaxMin cũng không ngoại lệ, thời gian thực hiện của thuật toán luôn nằm trong top 3 thuật toán tối ưu nhất, bằng chứng là các Request từ 80 – 100. Hơn hết, ở Request 90, thuật toán MaxMin đã ngồi chễm chệ ở ngôi vị đầu bảng.

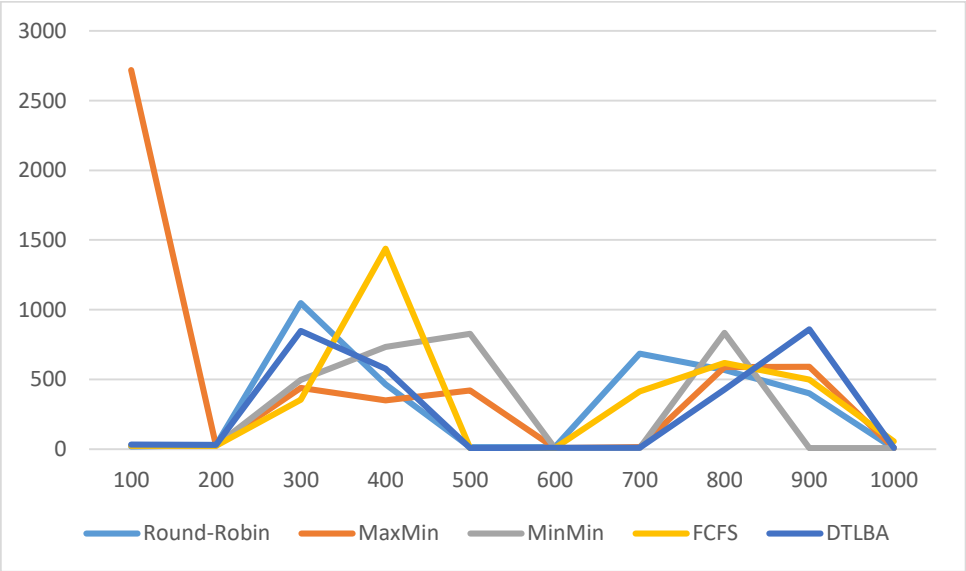
Với kết quả thực nghiệm của 100 Request đổ lại, ta thấy thuật toán MaxMin gặp khá nhiều khó khăn khi luôn tốn nhiều thời gian nhất để xử lý các tác vụ trên cloud, càng nhiều request thì độ ổn định càng cao hơn. Các thuật toán MinMin, FCFS và DTLBA vẫn luôn nhất quán duy trì sự ổn định của mình qua các thời kỳ Request. Thuật toán Round-Robin vẫn luôn chiếm ưu thế và xử lý nhanh nhất so với các thuật toán còn lại.

Kết quả chạy thực nghiệm mô phỏng trên CloudSim với 5 máy ảo được dựng sẵn để đáp ứng các yêu cầu, các yêu cầu được khởi tạo với chiều dài và kích thước ngẫu nhiên, số lượng Request lần lượt là 100 đến 1000:

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

Số lần request	Round-Robin	MaxMin	MinMin	FCFS	DTLBA
100	19.02	2718.88	26.1	22.45	32.8
200	28.58	26.13	24.74	23.88	30.86
300	1048.1	438.76	495.82	353.0	846.62
400	465.03	348.02	733.45	1437.28	576.93
500	13.74	420.48	827.17	6.9	7.63
600	13.72	6.51	7.83	6.71	9.11

700	684.63	14.59	7.66	412.79	8.64
800	567.92	588.5	833.3	617.17	427.73
900	398.96	589.51	7.48	497.46	858.53
1000	5.66	6.47	6.22	54.11	9.7



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

Từ Request 100 trở đi, thuật toán DTLBA vượt trội hơn hẳn so với các thuật toán còn lại do sự ổn định bền vững trong suốt quá trình xử lý các tác vụ trên cloud. Tuy không có những lúc thời gian xử lý ngắn nhất như thuật toán Round-Robin, MaxMin hay FCFS nhưng cũng không những lúc thời gian xử lý tăng đột biến như thuật toán MaxMin hay thuật toán FCFS. Qua đó ta có thể thấy thuật toán MaxMin và FCFS chưa thể hiện được sự thông minh và tính tự nhiên khi xây dựng thuật giải.

Thông qua 02 biểu đồ so sánh thời gian xử lý của các thuật toán với điều kiện như nhau ta có thể thấy sự phân bổ khá ổn định và hợp lý của thuật toán đề xuất DTLBA. Thời gian xử lý của các máy ảo khả quan hơn so với thời gian xử lý của các thuật toán khác trên cloud (ở trường hợp ít và nhiều Request).

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, đòi hỏi máy tính mạnh hơn và bộ xử lý tốt hơn. Vì vậy đây chính là hạn chế của thí nghiệm mô phỏng này.

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. Từ đó, ghi nhận các thông số về thời gian đáp ứng dự báo của các máy ảo và của cloud. Việc chạy thực nghiệm mô phỏng với thông số 5 máy ảo, chịu tải từ 50 đến 1000 request đã cho thấy kết quả tương đối tốt, việc phân bổ các request đến các máy ảo xử lý cũng khá đồng đều và có tính khả thi cao.

KẾT LUẬN

Luận văn **"ĐỀ XUẤT THUẬT TOÁN CÂN BẰNG TẢI TRÊN ĐIỆN TOÁN ĐÁM MÂY BẰNG CÔNG NGHỆ AI HIỆN ĐẠI"** dựa vào các thuật toán [23] đã có đó để phân tích và làm rõ chúng. Sau đó, có thể đánh giá đưa ra nhược điểm và lợi thế của từng thuật toán. Rồi từ các nhược điểm đã phân tích, đề xuất một thuật toán nhằm cải tiến và nâng cao khả năng cân bằng tải so với thuật toán cũ. Thuật toán đề xuất đã đạt được hiệu quả nhất định trong việc phân bổ tác vụ và nâng cao cân bằng tải trong môi trường điện toán đám mây. Quá trình nghiên cứu đã đạt được nhiều mục tiêu đề ra như sau:

- Nghiên cứu tổng quan đám mây và các đám mây với ba mô hình chính (IaaS, PaaS, SaaS) đang được sử dụng. Các kỹ thuật cân bằng tải được dùng trong môi trường điện toán đám mây.

- Mô phỏng lại quá trình nghiên cứu điện toán đám mây thông qua công cụ CloudSim – một trong những công cụ được đánh giá có giao diện thân thiện và dễ dàng sử dụng để thực nghiệm. Cài đặt và mô phỏng các kỹ thuật cân bằng tải với các thuật toán Round Robin, MaxMin, MinMin cũng như thuật toán tự nhiên FCFS. Các giá trị thu được khi mô phỏng đưa ra để phân tích so sánh với nhau. Mục đích là tóm lại được các ưu điểm và nhược điểm của các thuật toán, từ đó có hướng đề xuất một thuật toán sửa đổi để khắc phục mặt hạn chế đó.

- Đề xuất thuật toán DTLBA có khả năng giải quyết được các vấn đề về quản lí tài nguyên, thời gian đáp ứng cho các tác vụ được cải thiện và các yêu cầu cũng được xử lí nhanh chóng bởi máy ảo.

Hạn chế luận văn:

- Chưa được ứng dụng vào môi trường thực tế.

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

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

- Đưa thuật toán đề xuất vào ứng dụng thực tế.

- Áp dụng mô hình năng lượng tiêu thụ của datacenter hoặc cloud tương ứng để xây dựng biểu đồ phân bổ tải cho cloud.

TÀI LIỆU THAM KHẢO

[1] P. Srivastava and R. Khan, "A Review Paper on Cloud Computing," *International Journals of Advanced Research in Computer Science and Software Engineering,* vol. 8, no. 6, 2018.

[2] J. Gao, H. Wang and H. Shen, "Task Failure Prediction in Cloud Data Centers Using Deep Learning," *Transactions on Services Computing,* 2020.

[3] Huu, Tiep Vu, "Machine learning cơ bản," Machine learning cơ bản, 26 12 2016. [Online]. Available: https://machinelearningcoban.com/2016/12/26/introduce/. [Accessed 21 4 2021].

[4] Hogarty, Daniel T.; Su, John C.; Phan, Kevin; Attia, Mohamed; Hossny, Mohammed; Nahavandi, Saeid; Lenane, Patricia; Moloney, Fergal J.; Yazdabadi, Anousha, "Artificial Intelligence in Dermatology—Where We Are and the Way to the Future: A Review," *American Journal of Clinical Dermatology,* 2019.

[5] Phi, Nguyen Xuan; Hung, Tran Cong;, "Study the effect of Parameters to load balancing in cloud computing," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 8, 2016..

[6] Phi, Nguyen Xuan; Hieu, Le Ngọc; 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 số 04(CS.01),* pp. 43-48, 2018.

[7] Cuong, Nguyen Ha Huy; Vy, Dang Hung; Thuy, Nguyen Thanh;, "A New Technical Solution Prevention Deadlock for Resource Allocation in Heterogeneous Distributed Platforms," *International Journal of Computer Science and Network,* vol. 4, no. 2, pp. 266-272, 2015.

[8] Mahitha.O; Suma. V, "Deadlock Avoidance through Efficient Load Balancing to Control Disaster in Cloud Environment," in *4th ICCCNT 2013*, Tiruchengode, India, 2013.

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

[10] Sujata Kumari Research Student; Rahul Abhishek Research Student;B S Panda Asst, "Intelligent Computing Relating to Cloud Computing," *Special Issue of International Journal of Mechanical Engineering and Computer Applications (IJMCA)*, vol. 1, no. 1, 2013.

[11] U. Ahmed, M. Aleem, Y. N. Khalid, M. A. Islam and M. A. Iqba, "RALB-HC: A resource-aware load balancer for heterogeneous cluster," *Concurrency and Computation Practice and Experience*, 2019.

[12] Haenlein, Michael; Kaplan, Andreas, "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence," *sage journals*, vol. 61, no. 4, pp. 5-14, 2019.

[13] Zaidi, Taskeen; Rampratap, "Virtual Machine Allocation Policy in Cloud Computing Environment using CloudSim," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 344-354, 2018.

[14] Mukherjee, Tanmoy; Sarddar, Debabrata;, "A Conceptual Framework Towards Implementing a Cloud-Based Dynamic Load Balancer Using a Weighted Round-Robin Algorithm," *International Journal of Cloud Applications and Computing*, vol. 10, no. 2, 2020.

[15] E. Rani and H. Kaur, "STUDY ON FUNDAMENTAL USAGE OF CLOUDSIM SIMULATOR AND ALGORITHMS OF RESOURCE ALLOCATION IN CLOUD COMPUTING," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2017.

[16] Jaykrushna, Avatar; Pate, Pathik; Trivedi, Harshal; Bhatia, Jitendra;, "Linear Regression Assisted Prediction Based Load Balancer For Cloud Computing," in *2018 IEEE Punecon*, Pune, India, 2018.

[17] K. Govindarajan and V. S. Kumar, "An Intelligent Load Balancer for Software Defined Networking (SDN) based Cloud Infrastructure," in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, 2017.

[18] Filho, Manoel C. Silva; Oliveiray, Raysa L.; Monteiro, Claudio C.; Inácioy, Pedro R. M.; Freirey, Mário M.;, "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, 2017.

[19] Filho, M. C. Silva; Oliveira, R. L.; C. C. Monteiro, P. R. M. Inácio; Freire, M. M.;, "Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, 2017.

[20] E. Barbierato, M. Gribaudo, M. Iacono and A. Jakóbik, "Exploiting CloudSim in a multiformalism modeling approach for cloud based systems," *Simulation Modelling Practice and Theory*, 2018.

[21] Hamid Arabnejad; Claus Pah; Giovani Estrada; Areeg Samir, "A Fuzzy Load Balancer for Adaptive Fault Tolerance Management in Cloud Platforms," *IFIP International Federation for Information Processing 2017,* 2017.

[22] "Concury: A Fast and Light-weight Software Cloud load balancer," *SoCC '20: Proceedings of the 11th ACM Symposium on Cloud Computing,* pp. 179-192, 2020.

[23] 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.