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

TÓM TẮT LUẬN VĂN THẠC SĨ

TP. HCM – NĂM 2021

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

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

(Ghi rõ học hàm, học vị)

Phản biện 1: ……

Phản biện 2: ……..

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

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).

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.

Phần mở đầu:

2. Tổng quan về vấn đề nghiên cứu

2.1 Lợi ích của điện toán đám mây

- Giúp tiết kiệm chi phí.
- Truy cập tức thì mọi lúc mọi nơi
- Khả năng biến đổi vô tận.
- Khả năng thích ứng.
- Hợp tác bền vững, không xáo trộn
- Bảo mật dữ liệu

2.2 Các mô hình dịch vụ [3]

Mô hình dịch vụ của điện toán đám mây 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.

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

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 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.

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.

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.

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.

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.

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.

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.

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. 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.

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.

Tăng độ tin cậy và khả năng dự phòng cho hệ thống.

Tăng tính bảo mật cho hệ thống.

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. 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.

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.

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

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.

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. 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 trí tuệ nhân tạo để tiến hành cân bằng tải.

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.

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.

3.3 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 (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 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.

3.4 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.

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.

3.5 Kết luận chương

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).

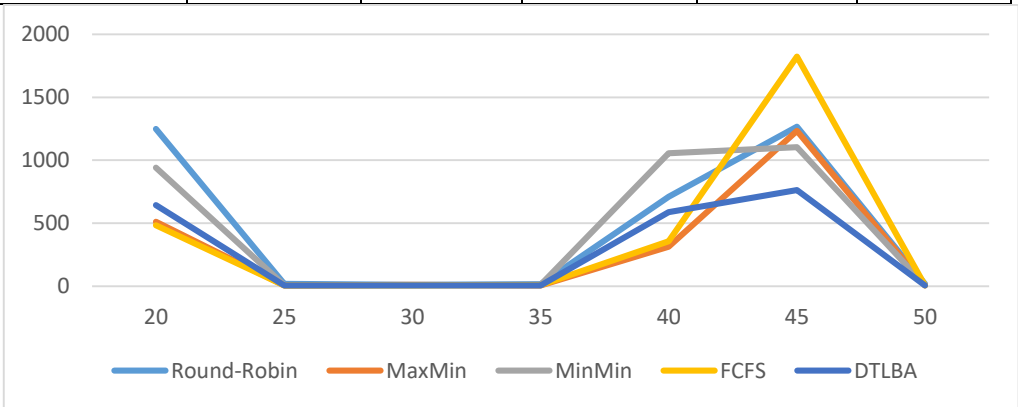
Các máy ảo và cloud có thời gian xử lí với sai số càng thấp thì hiệu quả của thuật toán càng đạt được kết quả tốt.

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.

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



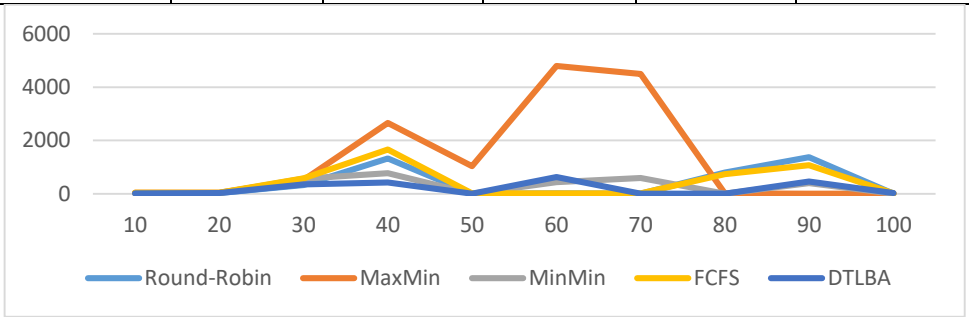
Hình 4.2: Biểu đồ so sánh thời gian thực hiện của 5 thuật toán với 50 Request

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.

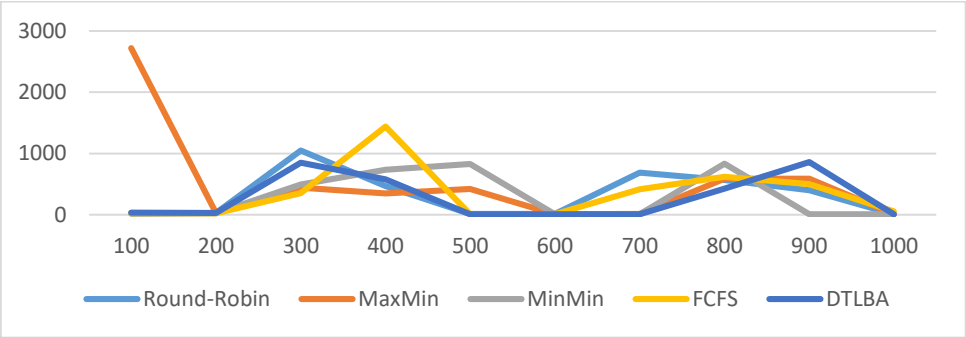
Các Request từ 70 – 80 chính là sân chơi của thuật toán DTLBA khi đối thủ là thuật toán Round-Robin đương kim vô địch ở Request 70 đã nhanh chóng bị đánh bại bởi thuật toán MinMin ở Request 80. Thuật toán FCFS đang ở vị trí thứ ba cũng phải nhường ngôi vị của cho thuật toán MaxMin.

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

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 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. 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.