

MÔ HÌNH CẤP PHÁT TÀI NGUYÊN ĐIỆN TOÁN Đám Mây CÔNG BẰNG DỰA TRÊN LÝ THUYẾT TRÒ CHƠI

Bùi Thanh Khiết^{1,3}, Nguyễn Thị Nguyệt Quế², Hồ Đắc Hưng³, Phạm Trần Vũ¹, Trần Công Hùng²

¹ Khoa Khoa học & Kỹ thuật máy tính, Đại học Bách Khoa TP HCM

² Học viện Công nghệ Bưu chính Viễn thông, Tp Hồ Chí Minh

³ Khoa Kỹ thuật Công nghệ, Đại học Thủ Dầu Một

khietbt@tdmu.edu.vn, nguyetquehtv@gmail.com, hunghd@tdmu.edu.vn,

ptvu@hcmut.edu.vn, conghung@ptithcm.edu.vn

TÓM TẮT: Quản lý, sử dụng tài nguyên trên Điện toán đám mây là một thách thức lớn cho nhà cung cấp dịch vụ cũng như người sử dụng dịch vụ đầu cuối. Việc cấp phát tài nguyên nhằm xác định yêu cầu tài nguyên và chất lượng dịch vụ của người dùng sẽ được cấp phát ở đâu trong hệ thống. Tài nguyên vật lý trong môi trường Điện toán đám mây đa số không đồng nhất với nhau, bên cạnh đó yêu cầu tài nguyên từ người dùng cũng không đồng nhất với nhau về loại tài nguyên. Điều đó gây khó khăn cho quá trình cấp phát tài nguyên nhất là trong giai đoạn khan hiếm tài nguyên. Trong nghiên cứu này, chúng tôi xây dựng mô hình cấp phát tài nguyên dựa trên lý thuyết trò chơi để đảm bảo sự công bằng giữa các máy chủ vật lý cũng như đảm bảo sử dụng hiệu quả tài nguyên. Chiến lược cấp phát tài nguyên công bằng được tìm dựa trên giải thuật luyện kim. Các kết quả thực nghiệm trình bày việc điều chỉnh tham số cho thuật toán để tìm ra chiến lược cấp phát máy ảo công bằng. Các tham số này phụ thuộc vào các mục tiêu của nhà cung cấp dịch vụ điện toán đám mây.

Từ khóa: Điện toán đám mây, cấp phát tài nguyên, lý thuyết trò chơi, giải thuật luyện kim.

I. GIỚI THIỆU

Điện toán đám mây (ĐTĐM) là một mô hình điện toán sử dụng công nghệ máy tính và phát triển dựa vào internet. Thuật ngữ “đám mây” là một cách nói ẩn dụ nhằm để chỉ mạng internet mà ở đó cung cấp môi trường tính toán là nơi cung cấp phần mềm, nền tảng, tài nguyên theo yêu cầu. Mô hình dịch vụ cơ sở hạ tầng (IaaS) cung cấp cho người dùng cơ sở hạ tầng như mạng, máy chủ, CPU, bộ nhớ, không gian lưu trữ và các tài nguyên tính toán dưới dạng máy ảo (VM) bằng công nghệ ảo hóa máy chủ. Công nghệ ảo hóa máy chủ cho phép tạo ra nhiều máy ảo trên một máy chủ vật lý, mỗi máy ảo cũng được cấp phát tài nguyên phân cứng như máy thật với RAM, CPU, card mạng, ổ cứng, hệ điều hành và các ứng dụng riêng. Người dùng gửi yêu cầu tài nguyên (CPU, bộ nhớ, không gian lưu trữ,...) đến nhà cung cấp dịch vụ ĐTĐM để xây dựng máy ảo. Đa số nhà cung cấp dịch vụ ĐTĐM xây dựng hệ thống với một số lượng lớn các máy chủ vật lý và tính chi phí sử dụng dịch vụ cho khách hàng theo thời gian sử dụng. Tùy vào nhu cầu sử dụng tài nguyên của người dùng theo thời gian mà nhà cung cấp dịch vụ ĐTĐM có thể điều chỉnh việc phân phối tài nguyên để đảm bảo chất lượng dịch vụ cũng như là lợi nhuận. Do vậy, việc quản lý, sử dụng tài nguyên trên ĐTĐM một cách hiệu quả là một thách thức lớn. Cấp phát tài nguyên tối ưu là rất cần thiết trong việc sử dụng hiệu quả tài nguyên trong ĐTĐM, bài toán tối ưu dạng này thường thuộc lớp NP-Hard hoặc NP-Complete [1]. Bên cạnh đó, đa số tài nguyên vật lý trong môi trường ĐTĐM không đồng nhất với nhau đồng thời yêu cầu từ người dùng cũng không đồng nhất đối với các loại tài nguyên. Các yêu cầu tài nguyên không đồng nhất đã gây ra hiện tượng xuất hiện phân mảnh tài nguyên cũng như gây ra việc lãng phí không gian tài nguyên. Đa số nhà cung cấp dịch vụ muốn tối đa hóa lợi nhuận với chi phí đầu tư là thấp nhất điều đó dẫn đến phải tối đa hóa việc sử dụng tài nguyên. Tuy nhiên, việc tối đa hóa sử dụng tài nguyên – cụ thể ở đây là máy chủ vật lý thường dẫn đến những vi phạm về chất lượng dịch vụ đối với khách hàng.

Để giải quyết vấn đề có tính tương tác giữa nhiều bên liên quan đa phần được tiếp cận từ lý thuyết trò chơi. Lý thuyết trò chơi được dùng để phân tích các tình huống dành cho hai hoặc nhiều cá nhân mà kết cục của một hành động được một trong số họ thực hiện không chỉ phụ thuộc và hành động cụ thể của chính cá nhân đó mà còn phụ thuộc vào hành động của những cá nhân khác còn lại trong trò chơi. Kết cục về lý thuyết của trò chơi được biểu diễn dưới dạng những tổ hợp chiến lược có nhiều khả năng đạt được mục tiêu của các người chơi nhất căn cứ vào thông tin sẵn có cho họ. Các tổ hợp chiến lược được gọi là các chiến lược cân bằng (equilibrium strategy) của người chơi nếu không có bất kỳ người chơi nào có động cơ thay đổi kế hoạch hành động của mình. Sự cân bằng của một trò chơi diễn tả những chiến lược mà các người chơi duy lý được tiên đoán là sẽ chọn khi họ tương tác với nhau. Để đảm bảo sử dụng tài nguyên hiệu quả và ổn định cần có chiến lược cấp phát tài máy ảo trong ĐTĐM hợp lý. Trong những trường hợp này, các kế hoạch hoặc chiến lược của những cá nhân liên quan sẽ phụ thuộc vào các dự kiến về những gì mà người khác đang làm. Do đó, những cá nhân trong tình huống này không ra quyết định một cách biệt lập, mà việc ra quyết định của họ có quan hệ phụ thuộc lẫn nhau. Điều này được gọi là sự tương thuộc chiến lược và những tình huống như thế thường được gọi là trò chơi chiến lược, hoặc đơn giản là trò chơi và những người tham gia trong các trò chơi như thế được gọi là những người chơi. Bởi vì những người chơi trong một trò chơi biết rõ rằng kết cục của những hành động của họ bị tác động bởi người chơi khác và ảnh hưởng lên người khác, nên họ cần phải tính đến những hành động khác nhau có thể có của những người chơi khác khi chính họ ra quyết định. Tuy nhiên, khi các cá nhân có thông tin hạn chế về những hành động dự tính (chiến lược) của những cá nhân khác, họ phải phỏng đoán về những gì họ nghĩ rằng những

người kia sẽ làm. Quá trình suy nghĩ kiểu này cấu thành tư duy chiến lược và khi chúng ta dùng đến loại tư duy này, lý thuyết trò chơi có thể giúp chúng ta hiểu được những gì đang xảy ra và dự báo về những kết cục có khả năng xảy ra.

Giải pháp cho vấn đề cấp phát tài nguyên thường dựa trên đặc tính cụ thể của từng bài toán từ đó áp dụng các giải thuật như vét cạn (exhaustive algorithm), giải thuật tất định (deterministic algorithm) [2] hoặc giải thuật metaheuristic [3], [4], [5]. Trong thực nghiệm, hầu như các giải thuật tất định tốt hơn các giải thuật vét cạn. Tuy nhiên các giải thuật tất định lại không hiệu quả trong môi trường dữ liệu phân tán từ đó dẫn đến không thích hợp cho các vấn đề cấp phát tài nguyên trong môi trường tính mở rộng [6]. Trong khi đó, ĐTĐM là môi trường có dữ liệu phân tán, đòi hỏi có khả năng mở rộng, khả năng đáp ứng yêu cầu người dùng cao do vậy có thể tiếp cận vấn đề cấp phát tài nguyên máy ảo trên ĐTĐM theo hướng metaheuristic là khả thi mặc dù các giải thuật metaheuristic có thể cho kết quả gần tối ưu trong thời gian chấp nhận được. Trong nghiên cứu này, chúng tôi đưa ra giải pháp cấp phát tài nguyên đảm bảo sự công bằng trong cấp phát tài nguyên cũng như sử dụng hiệu quả tài nguyên dựa trên lý thuyết trò chơi và dùng giải thuật metaheuristic cụ thể là giải thuật luyện kim để tìm ra tìm được giải pháp cấp phát tài nguyên máy ảo tối ưu hoặc gần tối ưu dựa trên cân bằng Nash. Phần còn lại trình bày các nghiên cứu liên quan trong phần II. Phần III trình bày mô hình cấp phát tài nguyên dựa trên lý thuyết trò chơi. Thuật toán luyện kim cho bài toán cấp phát tài nguyên được trình bày trong phần IV. Phần V trình bày kết quả thực nghiệm của mô hình. Phần VI trình bày kết luận.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Tác giả Waldspurger đã ứng dụng chính mô hình tối đa sự công bằng bằng cách xem xét các yếu tố khác nhau như ưu tiên, đặt chỗ và thời hạn [7]. Tuy nhiên, mô hình chỉ xem xét sự công bằng trong việc phân phối nguồn tài nguyên đơn lẻ. Trong nghiên cứu của hai tác giả Deshi Ye và Jianhai Chen [8] xem xét đến hai vấn đề: (1) Vấn đề cân bằng tài máy chủ: một tập các máy ảo VMs được mô tả như một vector nhiều chiều và một lượng máy vật lý PMs cho trước. Mỗi máy vật lý hoạt động dựa trên một hàm thu hoạch (payoff function) và có sự chuyển đổi qua lại giữa các máy nhằm tối đa hóa tài trên bất cứ chiều nào. (2) Vấn đề đặt máy ảo: cho trước một tập VMs, kết quả các hàm lợi ích là lựa chọn được các máy vật lý có tài nguyên cần thiết để cung cấp cho yêu cầu xây dựng máy ảo. Và có sự chuyển đổi máy ảo sang máy vật lý khác nếu có tài nguyên thích hợp hơn. Bài báo đã mô hình hai vấn đề trên dựa vào chiến lược trò chơi không hợp tác giữa các máy ảo. Tuy nhiên, các máy ảo và các máy vật lý không đồng nhất với nhau về công suất, cấu hình,... Việc phân phối máy ảo vào máy vật lý cần đảm bảo sử dụng tài nguyên máy vật lý hiệu quả hơn. Theo nghiên cứu của Zhen Xiao và các cộng sự ứng dụng công nghệ ảo hóa để phân phối tài nguyên trong trung tâm dữ liệu dựa trên nhu cầu ứng dụng và tính toán để tối ưu hóa số lượng máy chủ đang sử dụng. Vấn đề hiệu quả được xét trên hai phương diện: sự quá tải của máy vật lý và sự sử dụng ít máy vật lý để tiết kiệm năng lượng [9]. Bài báo đề xuất thuật toán dự báo nhu cầu sử dụng tài nguyên dựa trên trung bình chuyển động theo cấp số nhân EWMA. Tối ưu sự thay đổi trong sử dụng tài nguyên trên một máy vật lý để đạt sự tối ưu trên toàn hệ thống. Phương pháp này chỉ đạt được tối ưu cục bộ, không đạt được tối ưu toàn cục. Bài báo này chỉ giải quyết vấn đề cấp phát khi có yêu cầu. Khi có yêu cầu đến, nhà cung cấp dịch vụ phải quyết định xem nên chấp nhận phục vụ yêu cầu hay không yêu cầu trên hệ thống của nhà cung cấp mà có thể xử lý yêu cầu này ở hệ thống ngoài (external resources) - một nhà cung cấp liên kết. Nghiên cứu của tác giả Minarolli và các cộng sự đề xuất cấp phát tài nguyên CPU cho các máy ảo ở mức dịch vụ cơ sở hạ tầng ĐTĐM IaaS, xem xét trên 2 khía cạnh: chất lượng dịch QoS và quản lý chi phí [10]. Mô hình quản lý tài nguyên gồm 2 tầng: điều khiển cục bộ (local controller) đảm nhận cấp phát CPU động để chia sẻ cho các máy ảo VMs nhằm đạt được tối ưu cục bộ trên. Điều khiển toàn cục (Global controller) quản lý việc di trú máy ảo đến một máy vật lý khác để đạt được tối đa toàn cục. Các tác giả chỉ mới quan tâm đến tài nguyên CPU chưa đề cập đến các tài nguyên khác như memory và disk. Trong nghiên cứu của mình, tác giả Hassan và các cộng sự đề xuất một phương pháp cho vấn đề phân phối nguồn tài nguyên được phân tán trong cụm đám mây bằng ứng dụng cả trò chơi hợp tác và không hợp tác [11]. Họ cho thấy rằng việc phân phối tài nguyên trong trò chơi hợp tác tạo một động lực mạnh mẽ hơn cho các nhà cung cấp để đóng góp tài nguyên. Nhưng mô hình nghiên cứu của họ được thực hiện trong môi trường đơn tài nguyên. Sự phối hợp chia sẻ tài nguyên cũng là một thách thức cơ bản trong việc phân phối nguồn tài nguyên trong ĐTĐM. Nhiều công trình đã nghiên cứu phân phối tài nguyên công bằng cho đến thời điểm này, ví dụ như cấp phát tài nguyên công bằng của Hadoop - phân chia tài nguyên như là các phân vùng có kích thước cố định, hoặc các khe (slot). Một chính sách công bằng phổ biến khác là tối đa sự công bằng, bằng cách cố gắng tối đa hóa nguồn tài nguyên tối thiểu mà mỗi người dùng nhận được. Để đạt được việc công bằng giữa hệ thống và người dùng trong hệ thống phân tán, Siar và cộng sự đã mô hình hóa bài toán trên lý thuyết trò chơi [12]. Dùng giải thuật di truyền và lai ghép độ phổ biến của giải thuật luyện kim để xấp xỉ được giải pháp tối ưu hoặc gần tối ưu dựa trên cân bằng Nash.

III. MÔ HÌNH CẤP PHÁT TÀI NGUYÊN TRÊN LÝ THUYẾT TRÒ CHƠI HỢP TÁC

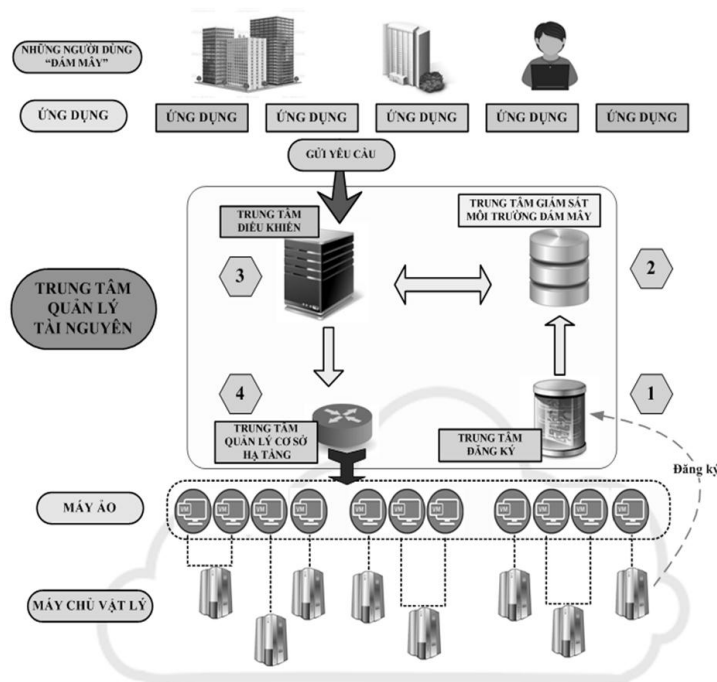
A. Kiến trúc hạ tầng ĐTĐM

Tại trung tâm quản lý tài nguyên của mô hình cấp phát tài nguyên trong Hình 1. bao gồm có 4 thành phần:

- Trung tâm đăng ký: Mỗi máy chủ vật lý trong “đám mây” sẽ đăng ký thông tin tài nguyên của nó đến trung tâm giám sát môi trường đám mây.

- Trung tâm giám sát môi trường đám mây: sẽ nhận những thông tin của các máy chủ vật lý như tên và địa chỉ IP, theo dõi các tình trạng của chúng (các tình trạng có thể là đang khởi động hoặc đang chạy hoặc đã tắt và lượng tiêu thụ của CPU), Bộ nhớ và không gian lưu trữ.
- Trung tâm quản lý cơ sở hạ tầng: có trách nhiệm phát triển và quản lý cơ sở hạ tầng ảo hóa, chẳng hạn như tạo và giải phóng máy ảo.
- Trung tâm điều khiển: là trung tâm tính toán để đưa ra quyết định thích hợp nhất về việc phân phối tài nguyên. Khi một máy chủ vật lý mới bắt đầu tham gia vào đám mây, thông tin như địa chỉ MAC, địa chỉ IP sẽ được đăng ký với trung tâm đăng ký.

Trung tâm giám sát môi trường đám mây sẽ theo dõi tình trạng và nguồn tài nguyên đã sử dụng tại các máy chủ vật lý đã đăng ký trong trung tâm đăng ký. Khi người dùng gửi yêu cầu dịch vụ đến “đám mây”, yêu cầu tài nguyên sẽ được nhận bởi trung tâm điều khiển. Trung tâm điều khiển sẽ đưa ra quyết định phân tài nguyên dựa trên thông tin do trung tâm giám sát môi trường đám mây thu thập được. Quyết định phân bổ tài nguyên được thực hiện bởi trung tâm quản lý hạ tầng để quản lý các máy chủ vật lý cũng như cấp phát các máy ảo trên nó. Trong môi trường đa tài nguyên, người dùng có các yêu cầu không đồng nhất đối với các loại tài nguyên khác nhau, cần cấp phát tài nguyên cho người dùng theo yêu cầu của họ. Để sử dụng tài nguyên một cách hiệu quả cần sử dụng tối ưu nguồn tài nguyên bằng cách giảm độ lãng phí không gian tài nguyên trong môi trường đa tài nguyên [13].



Hình 1. Kiến trúc dịch vụ cơ sở hạ tầng ĐTĐM

B. Trò chơi cấp phát tài nguyên

Trong cơ sở hạ tầng của ĐTĐM, giả sử có k máy chủ vật lý. Nhờ công nghệ ảo hóa, máy vật lý có thể triển khai máy ảo trên chính nó. Tất cả các máy chủ vật lý có thể nhận yêu cầu máy ảo và tạo ra máy ảo đáp ứng yêu cầu của người dùng. Một yêu cầu máy ảo r (cpu, memory, disk) tương ứng với cpu, memory, disk của máy ảo. Để đưa ra được quyết định cấp phát tài nguyên, trung tâm điều khiển lấy những thông tin của tài nguyên từ trung tâm giám sát môi trường đám mây. Chúng ta xem các máy chủ vật lý như những người chơi (player) trong trò chơi (game). Chiến lược của người chơi chính là một quyết định cấp phát tài nguyên của mỗi máy chủ vật lý dựa trên tập yêu cầu tài nguyên của người dùng. Các người chơi biết được các thông tin chiến lược và điểm quyết định của nhau nên ta có thể thiết lập thành trò chơi hợp tác và có thông tin hoàn hảo. Từ đó ta tiếp cận đến khái niệm cân bằng Nash đạt hiệu quả Pareto của trò chơi là điểm mà không có người chơi nhận được thu hoạch nhiều hơn bằng cách thay đổi chiến lược của mình [13]. Một trò chơi cấp phát tài nguyên được quy ước $G = (V, R, A, F)$ trong đó:

- V : là tập các người chơi - các máy chủ vật lý có tài nguyên sẵn sàng đáp ứng yêu cầu của người dùng,
- R : là tập các yêu cầu tài nguyên của người dùng,
- A : là tập các chiến lược của người chơi,
- F : là tập các hàm lợi ích của người chơi.

Tài nguyên sẵn sàng của máy chủ vật lý k được biểu diễn:

$$\vec{N}^{(k)} = (n_1^k, n_2^k, \dots, n_j^k), \tag{1}$$

trong đó n_j^k lượng tài nguyên j sẵn sàng cung cấp yêu cầu người dùng của máy chủ vật lý k.

Các yêu cầu tài nguyên của người dùng được định nghĩa thành một ma trận R trong đó hàng ngang của ma trận mô tả số lượng các loại tài nguyên để xây dựng máy ảo cho từng người dùng, hàng dọc mô tả số lượng yêu cầu khác nhau của mỗi loại tài nguyên cho các người dùng. Ma trận yêu cầu tài nguyên được chuẩn hóa dựa trên tổng số tài nguyên sẵn sàng theo từng loại của tất cả các máy chủ vật lý.

$$R = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \dots & \gamma_{1j} \\ \gamma_{21} & \gamma_{22} & \dots & \gamma_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{s1} & \gamma_{s2} & \dots & \gamma_{sj} \end{pmatrix} = \begin{pmatrix} \frac{r_{11}}{c_1} & \frac{r_{12}}{c_2} & \dots & \frac{r_{1j}}{c_j} \\ \frac{r_{21}}{c_1} & \frac{r_{22}}{c_2} & \dots & \frac{r_{2j}}{c_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{r_{s1}}{c_1} & \frac{r_{s2}}{c_2} & \dots & \frac{r_{sj}}{c_j} \end{pmatrix}, \tag{2}$$

trong đó:

- r_{sj} là tài nguyên j của yêu cầu s,
- $C_j = \sum_i n_j^i$ tổng lượng tài nguyên j sẵn sàng của máy chủ vật lý thứ i,
- γ_{sj} là giá trị tài nguyên loại j của yêu cầu s đã chuẩn hóa.

Mỗi quyết định cấp phát tài nguyên trên máy chủ vật lý i có thể được định nghĩa thành một ma trận A^i , trong đó mỗi dòng của ma trận tương ứng với tài nguyên được cấp phát cho các yêu cầu:

$$A^i = \begin{pmatrix} a_{11}^{(i)} & a_{12}^{(i)} & \dots & a_{1j}^{(i)} \\ a_{21}^{(i)} & a_{22}^{(i)} & \dots & a_{2j}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1}^{(i)} & a_{s2}^{(i)} & \dots & a_{sj}^{(i)} \end{pmatrix}, \tag{3}$$

trong đó, $a_{sj}^{(i)}$ là tài nguyên cấp j được cấp cho yêu cầu s tại máy chủ vật lý i.

Tập hợp tất cả quyết định cấp phát tài nguyên dựa trên ma trận yêu cầu tài nguyên của các máy chủ vật lý được biểu diễn:

$$A = \{A^1, A^2, \dots, A^k\} \tag{4}$$

Sự công bằng trong cấp phát tài nguyên được áp dụng từ khái niệm DRF (Dominant Resource Fairness) do A. Ghodsi và các cộng sự đã nghiên cứu trong môi trường đa tài nguyên [14]. Thêm vào đó, Xu và đồng nghiệp [13] đã đưa ra mô hình cấp phát tài nguyên công bằng mà vẫn đảm bảo sử dụng tài nguyên hiệu quả - hay nói cách khác là giảm thiểu mức độ lãng phí tài nguyên. Theo đó, sự công bằng của quyết định cấp phát tài nguyên được tính toán theo công thức như sau:

$$v(A) = \left(\sum_i \sum_j \left(\frac{x_{ij}}{c_j} - \tau * d_{ij} \right)^{\alpha-1} \right)^{\frac{1}{\alpha}} \quad (\alpha \in \mathcal{R}), \tag{5}$$

trong đó:

- x_{ij} là số lượng tài nguyên j được cấp phát cho yêu cầu i,
- $\tau = \frac{1}{\max_i \sum_j d_{ij}}$ là chia sẻ ưu thế (dominant share) là tỉ lệ lớn nhất của bất kỳ một tài nguyên nào đó được cấp phát cho người dùng,
- $d_{ij} = \frac{\gamma_{ij}}{\max_j \gamma_{ij}}$ là yêu cầu được chuẩn hóa.

Mức độ sử dụng phí tài nguyên $w(n)$ của máy chủ vật lý n được tính toán theo công thức sau:

$$w(n) = \sqrt{\sum_j \left(\frac{u_j^{(n)}}{u^{(n)}} - 1 \right)^2}, \tag{6}$$

trong đó:

- $u_j^{(n)}$ là tài nguyên j được sử dụng của máy chủ vật lý thứ n ,
- $\bar{u}^{(n)}$ là trung bình các tài nguyên đã sử dụng của các máy chủ vật lý.

Hàm lợi ích được xây dựng dựa trên việc cấp phát công bằng và hiệu quả tài nguyên. Mỗi người chơi sẽ chọn ra một chiến lược để tối đa hóa lợi ích của chính họ. Hàm lợi ích của một quyết định cấp phát tài nguyên được tính như sau:

$$F^n(A) = \beta * v(A) + (1 - \beta) * w(n), \quad (7)$$

trong đó $\beta \in [0,1]$ là trọng số chỉ mức độ quan tâm đối với mức độ công bằng hay hiệu quả sử dụng tài nguyên.

Trong trò chơi này, hàm lợi ích của trò chơi có ảnh hưởng quan trọng đến một quyết định chiến lược của người chơi và kết quả của trò chơi. Mỗi người chơi sẽ chọn chiến lược nhằm tối đa hóa thu hoạch của họ, do vậy hàm mục tiêu như sau:

$$\text{Max } (F^n(A)).$$

$$\text{sao cho } \sum_s \sum_n a_{sj}^n < C_j$$

$$a_{sj}^n \geq 0.$$

Điểm cân bằng Nash của trò chơi là chiến lược mà ở đó không một người chơi nào có thể tăng lợi nhuận khi những người chơi khác đã cố định chiến lược. Khi đó, nếu chiến lược của người chơi thứ i là chiến lược tối ưu được kí hiệu p_i^* , chiến lược tối ưu của những người chơi khác được kí hiệu là p_{-i}^* thì cân bằng Nash của chiến lược p_i^* sẽ tuân thủ theo điều kiện [15], như sau:

$$F_i(p_{-i}^*, p_i^*) \geq F_i(p_{-i}^*, p_i), \quad (8)$$

từ đó tìm ra được một quyết định công bằng và hiệu quả cũng chính là cân bằng Nash của trò chơi $A^* = A^{1*}, A^{2*}, \dots, A^{k*}$ có nghĩa là $F(A^1, A^2, A^{n*}, \dots, A^k) > F(A^1, A^2, A^{n'}, \dots, A^k)$.

Trong môi trường multi agent system, có thể điểm cân bằng sẽ không ổn định (stable) [16]. Ngoài ra, khó có thể tìm được Pareto-efficiency của cân bằng Nash. Để giải quyết vấn đề này đa phần các giải thuật được dựa trên các giải thuật metaheuristic. Từ tập phương án khả thi đó dựa vào điều kiện cân bằng Nash sẽ chọn ra phương án tốt nhất. Điều kiện dừng của thuật toán được tính theo Siar [12] như sau:

$$\sum_{i=1}^n (F^{itr} - F^{itr-1})^2 < \varepsilon \quad (9)$$

IV. GIẢI THUẬT LUYỆN KIM CHO VẤN ĐỀ CẤP PHÁT TÀI NGUYÊN CÔNG BẰNG

Simulated Annealing (SA) là một thuật toán dựa trên các heuristic về xác suất, dùng cho một lớp các vấn đề để tìm một số điều kiện tốt nhất trong một không gian tìm kiếm lớn. Thuật toán được nêu ra độc lập bởi Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi vào năm 1983 và bởi Vlado Černý năm 1985 [17]. Trong thuật toán SA, mỗi điểm s của không gian tìm kiếm là tương tự với một trạng thái của một số hệ thống vật lý và hàm $E(s)$ tương trưng cho nội năng của hệ thống trong trạng thái đó. Tại mỗi bước, quy tắc heuristic sẽ xem xét và đưa ra một số trạng thái s' của trạng thái hiện tại s và hàm xác suất quyết định giữa thay đổi hệ thống sang trạng thái s' hay là vẫn tiếp tục giữ trạng thái s . Xác suất này đưa hệ thống tới trạng thái có năng lượng thấp hơn. Bước này sẽ được lặp đi lặp lại cho đến khi hệ thống đạt được một trạng thái đủ tốt, hoặc cho đến khi không thể tiếp tục.

Áp dụng thuật toán luyện kim để tìm ra được một quyết định công bằng và hiệu quả của trò chơi cụ thể như sau:

- Không gian trạng thái: là tập hợp tất cả các chiến lược (hay nói cách khác là tập hợp các quyết định) của người chơi.
- Hàm năng lượng E : là hàm lợi ích của quyết định cho mỗi trạng thái được tính theo công thức (7).
- Trạng thái kề: Ta tìm một trạng thái kề của một trạng thái bằng cách đổi quyết định của người chơi một cách ngẫu nhiên. Ở đây, trạng thái kề cụ thể là chiến lược cấp phát tài nguyên của người dùng một cách ngẫu nhiên. Tại thời điểm t cụ thể của thuật toán, yêu cầu tài nguyên của người dùng có thể được cấp phát hoặc không cho một máy chủ vật lý – người chơi cụ thể. Thuật toán 2. trình bày cách tìm trạng thái kề. Thuật toán tìm trạng thái kề cụ thể ở đây là chiến lược cấp phát tài nguyên của người dùng một cách ngẫu nhiên. Ý tưởng của thuật toán sẽ duyệt hết các yêu cầu tài nguyên máy ảo được tính toán trong công thức (2), tại mỗi yêu cầu tài nguyên máy ảo sẽ tìm ra ngẫu nhiên một máy chủ vật lý có tài nguyên sẵn sàng phục vụ yêu cầu đó. Đầu vào của thuật toán là giá trị ngưỡng exploitation biểu diễn xác suất chọn một yêu cầu máy ảo được cấp phát ở một máy vật lý cụ thể có tài nguyên sẵn sàng phục vụ yêu cầu máy ảo đó. Ở đây, xác suất chọn ngẫu nhiên cho việc cấp phát máy ảo trên một máy vật lý cụ thể tuân theo phân bố chuẩn.

- Nhiệt độ T: nhiệt độ ban đầu ta đặt giá trị 10000 (T càng lớn thì độ chính xác của lời giải càng cao).
- Hàm xác suất được định nghĩa bằng công thức:

$$P = \text{Exp}\left(\frac{\text{Giá trị lợi ích hiện tại} - \text{Giá trị lợi ích mới}}{\text{Nhiệt độ}}\right)$$

Thuật toán 1: SAPVM

```

Input: iterationsmax, tempmax, terminal, anpha, beta
Output: Sbest//giải pháp cấp phát máy ảo tốt nhất

Scurrent ← CreateInitialSolution();// khởi tạo giải pháp hiện tại
Sbest ← Scurrent; //khởi tạo giải pháp tốt nhất
while (i = 1 to iterationsmax || ε <= terminal) do
    Si ← CreateNeighborSolution(Scurrent);// sinh giải pháp khả thi mới – trạng thái
    kề
    tempcurr ← CalculateTemperature(i, tempmax); // tính toán nhiệt độ
    if F(Si) ≤ F(Scurrent) then // so sánh hàm lợi ích của các giải pháp
        Scurrent ← Si;
        if F(Si) ≤ F(Sbest) then // so sánh hàm lợi ích của các giải pháp
            Sbest ← Si;
        end
    else if Exp(F(Scurrent) - F(Si)/tempcurr) > Rand() then
        Scurrent ← Si;
    end
end
return Sbest;
    
```

Thuật toán 2: CreateNeighborSolution

```

Input: exploitation
Output: Sneighbor//giải pháp cấp phát máy ảo

rand = random();//sinh ra giá trị ngẫu nhiên từ [0,1] theo phân bố chuẩn
for i = 1 to R do
    if rand > exploitation
        Sneighbor ← Ri;
    end
end
return Sneighbor;
    
```

Ví dụ

Bảng 1. Cấu hình tài nguyên sẵn sàng phục vụ của máy chủ vật lý

Id_host	cpu	ram	disk
1	4	8	40
2	4	6	50

Bảng 2. Cấu hình yêu cầu tài nguyên máy ảo

id_vm	cpu	ram	disk
1	2	4	20
2	1	1	10
3	2	2	10

Thông qua thuật toán 2 ở trên có thể tìm ra được các chiến lược có thể cấp phát cho yêu cầu người dùng như sau:

Chiến lược 1

$$A^1 = \begin{matrix} & 2 & 4 & 20 \\ 1 & 1 & 1 & 10 \\ & 0 & 0 & 0 \end{matrix}$$

Chiến lược 2

$$A^1 = \begin{matrix} & 2 & 4 & 20 \\ 1 & 1 & 1 & 10 \\ & 0 & 0 & 0 \end{matrix}$$

$$A^2 = \begin{matrix} & 0 & 0 & 0 \\ 1 & 1 & 1 & 10 \\ & 2 & 2 & 10 \end{matrix}$$

$$A^2 = \begin{matrix} & 2 & 4 & 20 \\ 0 & 0 & 0 & 0 \\ & 2 & 2 & 10 \end{matrix}$$

Thông qua thuật toán 1 có thể tìm ra giải pháp cấp phát tài nguyên tốt dựa trên việc tính toán hàm lợi ích nhờ công thức (7).

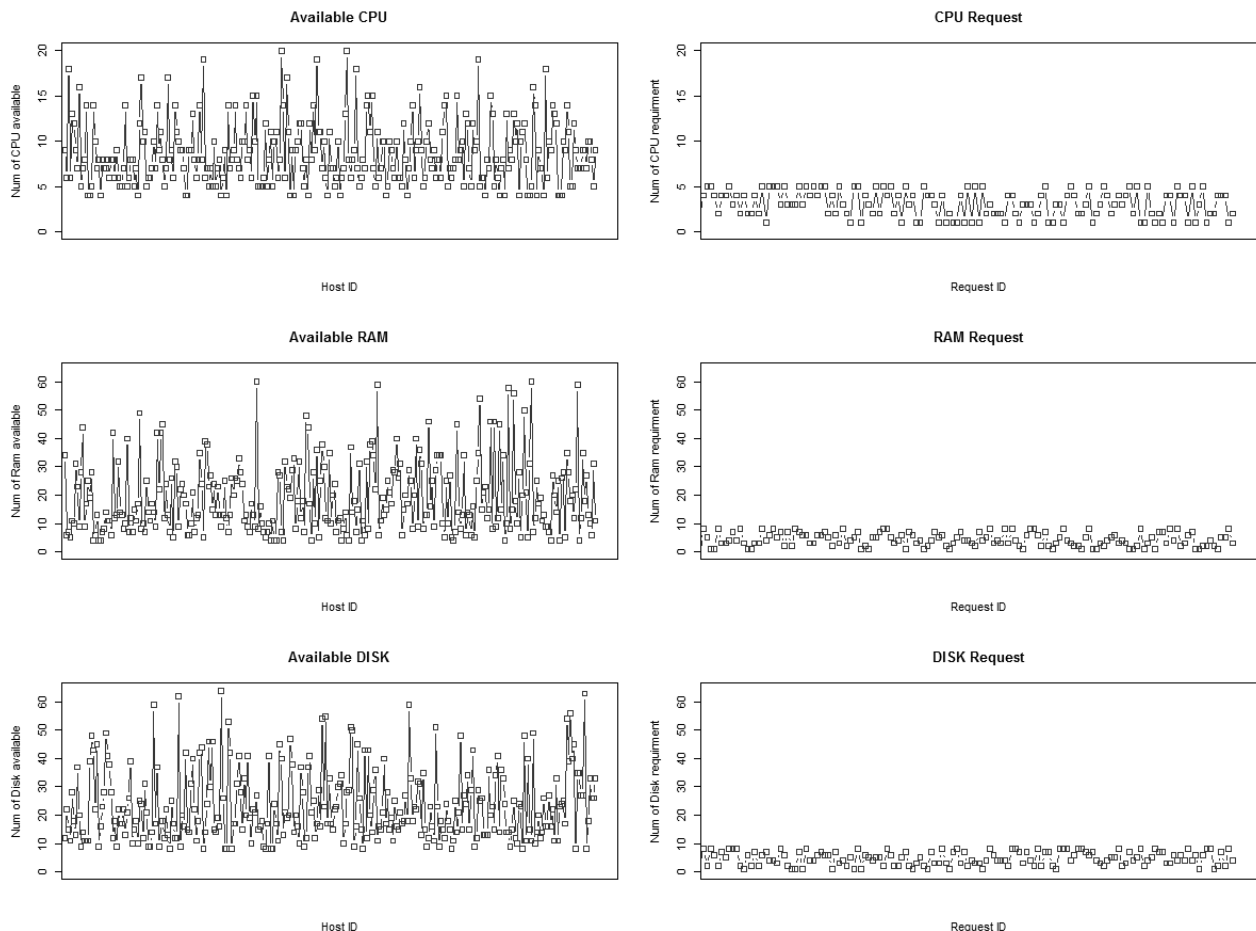
V. KẾT QUẢ THỰC NGHIỆM

A. Thiết lập thực nghiệm

Trong bài báo này, chúng tôi quan tâm với vấn đề cấp phát tài nguyên đảm bảo sự công bằng trong cấp phát tài nguyên cũng như sử dụng hiệu quả tài nguyên dựa trên lý thuyết trò chơi. Để đánh giá hiệu quả của thuật toán, thực nghiệm được thực hiện trên một mô phỏng yêu cầu tài nguyên (CPU, RAM, DISK) một cách ngẫu nhiên với các giá trị tối đa/tối thiểu của tài nguyên như trong Bảng 3. Theo đó, 300 Máy chủ vật lý – người chơi được sinh ra theo số lượng cấu hình tối đa/tối thiểu (CPU, RAM, DISK) của máy chủ sẽ được thiết lập sẵn ban đầu. Các máy chủ có cấu hình ngẫu nhiên không vượt quá cấu hình tối đa. Hình 2. biểu diễn phân bố tài nguyên sẵn sàng phục vụ yêu cầu máy ảo của các máy chủ vật lý. Tổng số lượng tài nguyên yêu cầu của máy ảo phải nhỏ hơn hoặc bằng tổng số lượng tài nguyên sẵn sàng đáp ứng của máy chủ. Môi trường thực nghiệm trên máy tính cấu hình như sau: Windows 10 Ultimate, 8GB RAM, Core i5, 240 GB.

Bảng 3. Cấu hình tối đa/tối thiểu của máy chủ vật lý và máy ảo

	MaxCPU	MinCPU	MaxRam	MinRam	MaxDisk	MinDisk
Host	20	8	64	8	64	16
VM	5	1	6	1	6	1



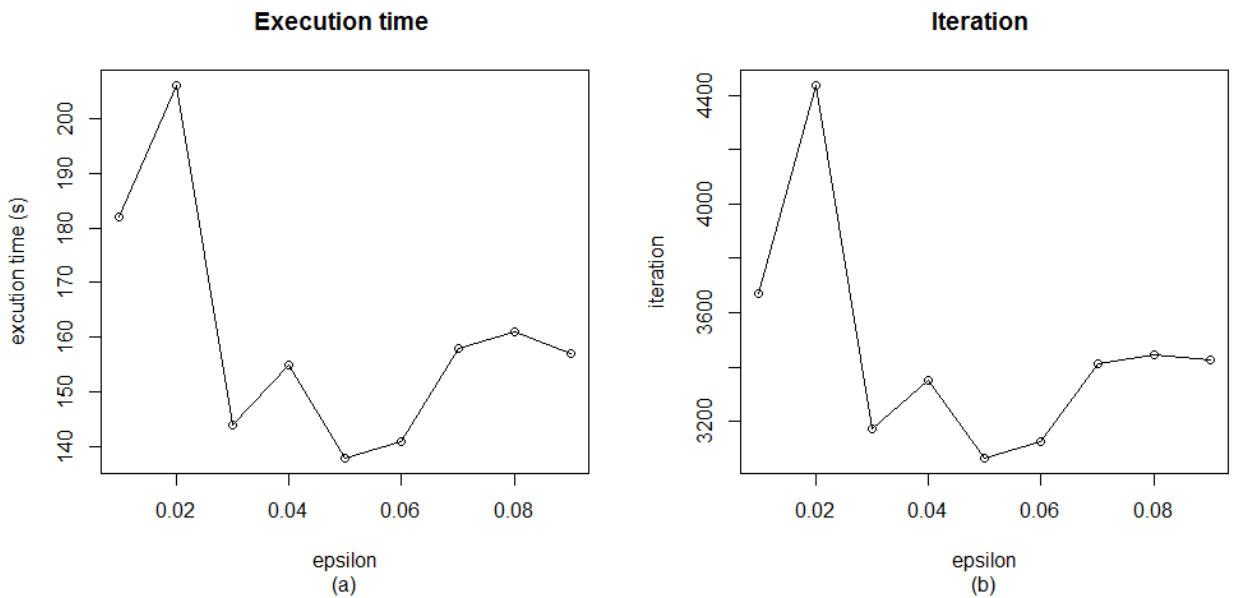
Hình 2. Tình trạng tài nguyên sẵn sàng của các máy chủ vật lý và yêu cầu tài nguyên

B. Hiệu chỉnh tham số thuật toán SAPVM

Thuật toán SAPVM tìm ra giải pháp cấp phát tài nguyên máy ảo tối ưu hoặc gần tối ưu dựa trên cân bằng Nash. Kết quả xấp xỉ phụ thuộc vào các tham số epsilon, cooling rate, exploitation, anpha, beta. Do vậy, trong các thực nghiệm dưới đây, chúng tôi tìm những tham số thích hợp cho hệ thống cũng như đánh giá việc cấp phát tài nguyên máy ảo cho khách hàng thông Sự công bằng trong cấp phát tài nguyên được áp dụng từ khái niệm DRF trong công thức (5) và độ sử dụng phí tài nguyên theo công thức (6).

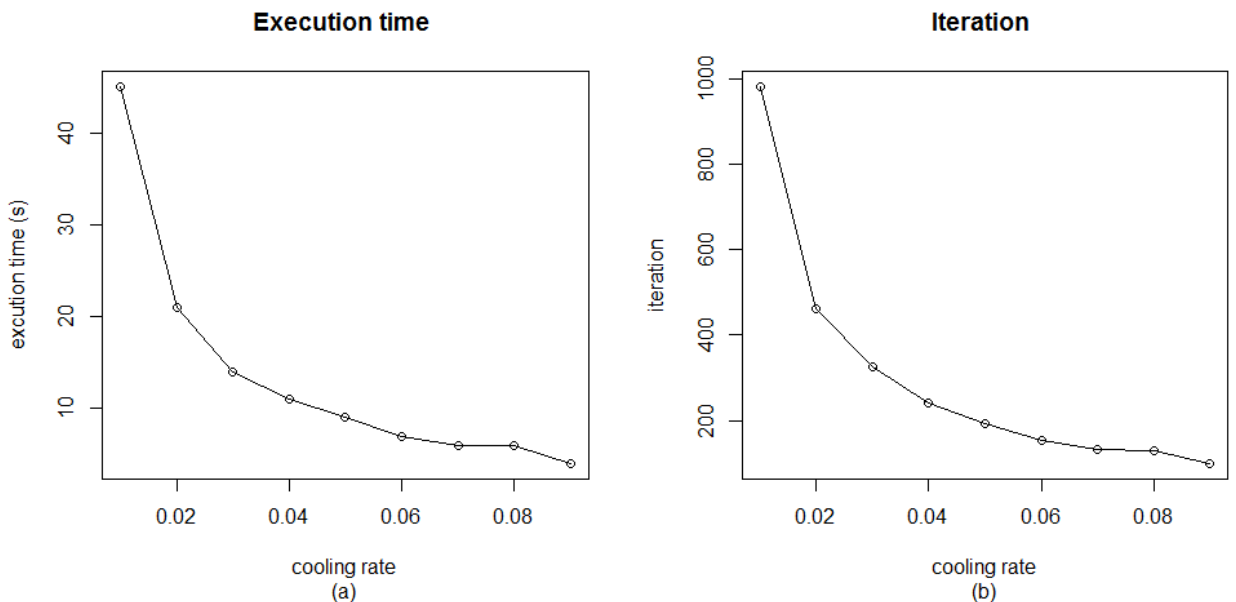
1. Điều kiện dừng của thuật toán

Với 300 máy chủ và 150 yêu cầu máy ảo, chọn tham số $\alpha = 2$, $\beta = \text{exploitation} = 0.5$, $\text{coolingrate} = 0.003$, chúng tôi điều chỉnh ϵ trong công thức (9) từ 0.01 đến 0.09 đo thời gian thực thi và số vòng lặp được thực hiện. Trong Hình 3, ta thấy thời gian thực thi và số lượng vòng lặp lớn khi ϵ nhỏ và ngược lại ϵ càng lớn thì thời gian thực thi và số lượng vòng lặp càng nhỏ. Theo đó tại giá trị $\epsilon = 0.05$ cho thời gian thực thi và số lượng vòng lặp nhỏ do trong các thực nghiệm tiếp theo chúng tôi chọn $\epsilon = 0.05$.



Hình 3. Thời gian thực thi và số vòng lặp với epsilon từ 0.01 đến 0.09

Để tiếp tục đánh giá thời gian thực thi và số lượng vòng lặp, chúng tôi điều chỉnh cooling rate từ 0.01 đến 0.09



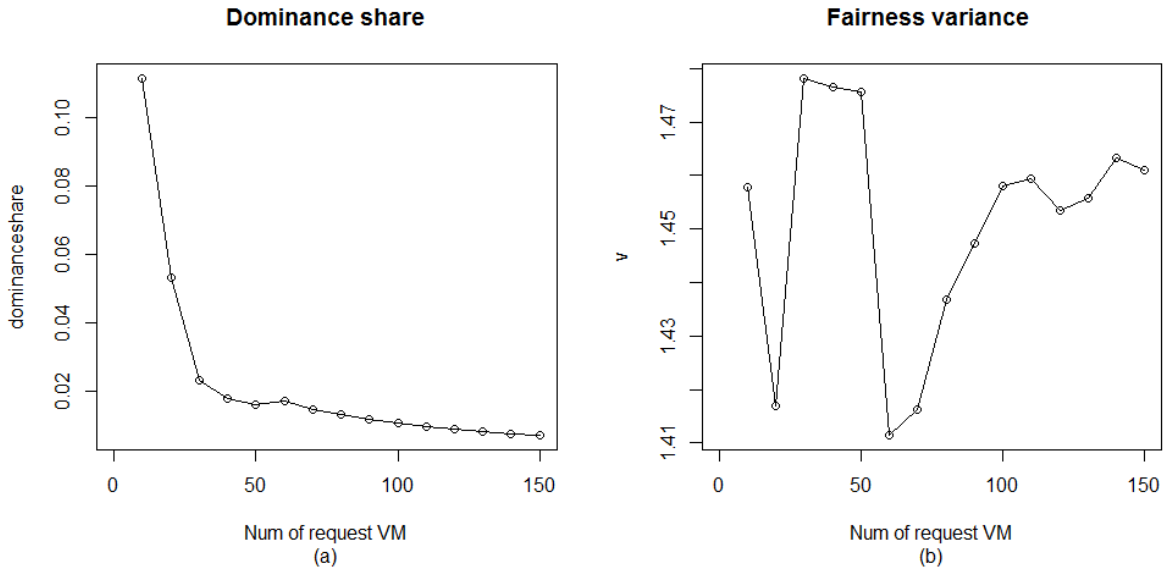
Hình 4. Thời gian thực thi và số vòng lặp với cooling rate từ 0.01 đến 0.09

Hình 4. cho thấy thời gian thực thi và số lượng vòng lặp của thuật toán càng lớn khi tham số cooling rate càng nhỏ và ngược lại. Tham số cooling rate càng nhỏ sẽ giúp cho tìm kiếm cơ hội tìm kiếm điểm tối ưu càng cao nhưng bên cạnh đó nếu chúng ta chọn tham số cooling rate quá nhỏ dẫn đến thời gian thực thi càng lớn. Do vậy, chúng tôi chọn tham số cooling rate = 0.05 cho các thực nghiệm tiếp theo.

2. Hiệu quả cấp phát tài nguyên

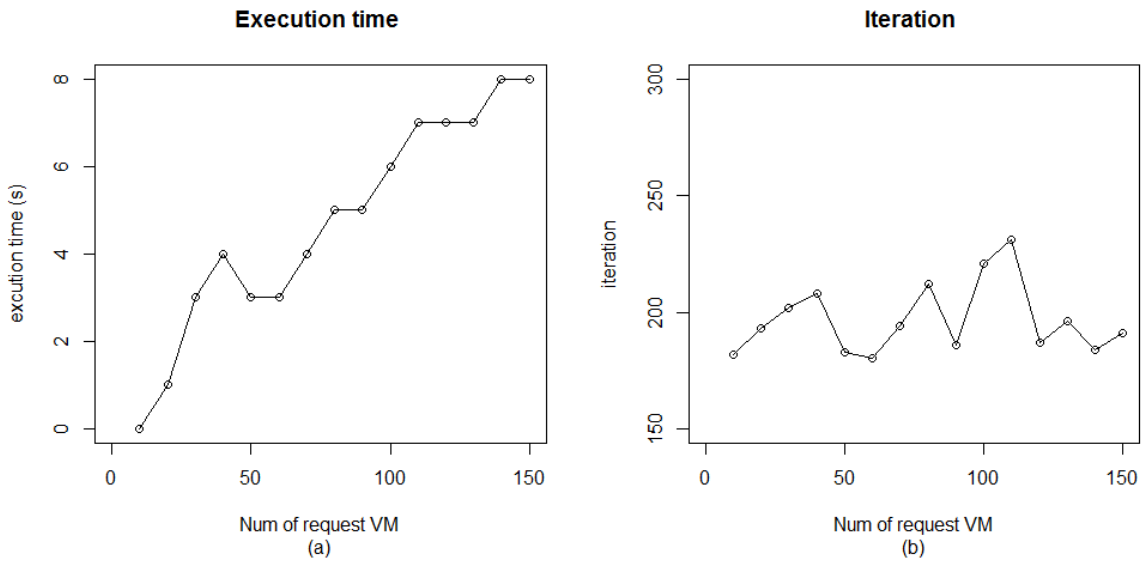
Sự công bằng trong cấp phát tài nguyên được đánh giá thông qua đo mức độ chia sẻ ưu thế (dominant share) và mức độ biến động công bằng cho chiến lược cấp phát tài nguyên tốt nhất của một người chơi. Chúng tôi thực nghiệm

với 300 máy chủ, $\alpha = 2$, $\beta = \text{exploitation} = 0.5$, $\epsilon = 0.05$, $\text{coolingrate} = 0.05$ và số lượng yêu cầu máy ảo của người dùng từ tăng từ 10 đến 150.



Hình 5. Mức độ công bằng trong cấp phát tài nguyên từ 10 đến 150 máy ảo

Hình 5a. cho thấy mức độ chia sẻ ưu thế (dominant share) giảm từ khoảng 0.1 đến 0.01. Theo đó, ban đầu số lượng yêu cầu máy ảo ít trong khi đó số lượng máy chủ vật lý nhiều thì mức độ chia sẻ ưu thế cao và ngược lại. Hình 5b. cho thấy mức độ biến động công bằng cho chiến lược cấp phát tài nguyên giao động trong khoảng 1.40 đến 1.48.



Hình 6. Thời gian thực thi và số vòng lặp trong cấp phát tài nguyên từ 10 đến 150 máy ảo

Hình 6 thể hiện gồm thời gian thực thi tăng từ 1 giây lên 8 giây và số lượng vòng lặp trong khoảng 160 đến 240 khi tăng dần số lượng yêu cầu máy ảo.

Bảng 4. Hiệu quả cấp phát tài nguyên với 300 máy chủ và 150 yêu cầu máy ảo

Tham số	Mức độ chia sẻ ưu thế (dominant share)	Mức độ biến động công bằng cho chiến lược cấp phát tài nguyên	Thời gian thực thi (giây)	Số vòng lặp
$\epsilon = 0.01$, $\text{coolingrate} = 0.003$, $\alpha = 2$, $\beta = \text{exploitation} = 0.5$,	0.007344	1.4611	182	3672
$\epsilon = \text{coolingrate} = 0.05$, $\alpha = 2$, $\beta = \text{exploitation} = 0.5$,	0.00783	1.4633	8	191

Bảng 4. Với thực nghiệm 300 máy chủ và 150 yêu cầu máy ảo cho thuật hiệu quả của thuật toán phụ thuộc vào cách điều chỉnh các tham số đặt biệt là epsilon và cooling rate. Khi epsilon = 0.01, coolingrate = 0.003 mức độ biến động công bằng cho chiến lược cấp phát tài nguyên là 1.4611 và số lượng vòng lặp là 3672. Trong khi đó, với epsilon = coolingrate = 0.05 thì mức độ biến động công bằng cho chiến lược cấp phát tài nguyên là 1.4633 và số vòng lặp 191.

VI. KẾT LUẬN

Từ kết quả thực nghiệm có thể đánh giá mô hình cấp phát tài nguyên dựa trên lý thuyết trò chơi và ứng dụng thuật toán luyện kim đã thực hiện đạt được tiêu chí công bằng và hiệu quả đã đặt ra. Các yêu cầu của máy ảo đã được đáp ứng hợp lý với các tài nguyên có sẵn của các máy vật lý. Giải thuật SAPVM được đề xuất để tìm chiến lược cấp phát tài nguyên tối ưu hoặc gần tối ưu theo tiêu chí công bằng và hiệu quả dựa trên cân bằng Nash. Tùy vào mục tiêu việc khai thác dịch vụ điện toán đám mây mà các tham số của thuật toán sẽ được thiết lập. Tuy nhiên, do cấu hình máy thực hiện mô phỏng có giới hạn nên bộ dữ liệu đầu vào chưa đủ lớn điều này ảnh hưởng ít nhiều đến hiệu quả của thuật toán khi thực thi. Sắp tới, việc so sánh đánh giá với các thuật toán metaheuristic khác như thuật toán đàn kiến, thuật toán di truyền,... sẽ được nghiên cứu thực hiện.

TÀI LIỆU THAM KHẢO

- [1]. Gary, M. R., and Johnson, D. S.: 'Computers and Intractability: A Guide to the Theory of NP-completeness', in Editor (Ed.)^(Eds.): 'Book Computers and Intractability: A Guide to the Theory of NP-completeness' (WH Freeman and Company, New York, 1979, edn.), pp.
- [2]. Morton, T., and Pentico, D. W.: 'Heuristic scheduling systems: with applications to production systems and project management' (John Wiley & Sons, 1993. 1993).
- [3]. Van Laarhoven, P. J., Aarts, E. H., and Lenstra, J. K.: "Job shop scheduling by simulated annealing", *Operations research*, 1992, 40, (1), pp. 113-125.
- [4]. Colomi, A., Dorigo, M., Maniezzo, V., and Trubian, M.: "Ant system for job-shop scheduling", *Belgian Journal of Operations Research, Statistics and Computer Science*, 1994, 34, (1), pp. 39-53.
- [5]. Ghumman, N. S., and Kaur, R.: "Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system", in Editor (Ed.)^(Eds.): "Book Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system" (IEEE, 2015, edn.), pp. 1-5.
- [6]. Tsai, C. W., and Rodrigues, J. J.: "Metaheuristic scheduling for cloud: A survey", *IEEE Systems Journal*, 2014, 8, (1), pp. 279-291.
- [7]. Waldspurger, C. A.: "Lottery and Stride Scheduling: Flexible Proportional-share Resource Management", in Editor (Ed.)^(Eds.): "Book Lottery and Stride Scheduling: Flexible Proportional-share Resource Management" (Massachusetts Institute of Technology, 1995, edn.), pp.
- [8]. Ye, D., and Chen, J.: "Non-cooperative games on multidimensional resource allocation", *Future Generation Computer Systems*, 2013, 29, (6), pp. 1345-1352
- [9]. Xiao, Z., Song, W., and Chen, Q.: "Dynamic resource allocation using virtual machines for cloud computing environment", *IEEE transactions on parallel and distributed systems*, 2013, 24, (6), pp. 1107-1117
- [10]. Minarolli, D., and Freisleben, B.: "Utility-based resource allocation for virtual machines in cloud computing", in Editor (Ed.)^(Eds.): "Book Utility-based resource allocation for virtual machines in cloud computing" (IEEE, 2011, edn.), pp. 410-417.
- [11]. Hassan, M. M., Hossain, M. S., Sarkar, A. J., and Huh, E. N.: "Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform", *Information Systems Frontiers*, 2014, 16, (4), pp. 523-542.
- [12]. Siar, H., Kiani, K., and Chronopoulos, A. T.: "An effective game theoretic static load balancing applied to distributed computing", *Cluster Computing*, 2015, pp. 1-15.
- [13]. Xu, X., and Yu, H.: "A game theory approach to fair and efficient resource allocation in cloud computing", *Mathematical Problems in Engineering*, 2014.
- [14]. Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., and Stoica, I.: "Dominant Resource Fairness: Fair Allocation of Multiple Resource Types", in Editor (Ed.)^(Eds.): "Book Dominant Resource Fairness: Fair Allocation of Multiple Resource Types" (2011, edn.), pp. 24-24.
- [15]. Osborne, M. J., and Rubinstein, A.: "A course in game theory" (MIT press, 1994. 1994).
- [16]. Pendharkar, P. C.: "Game theoretical applications for multi-agent systems", *Expert Systems with Applications*, 2012, 39, (1), pp. 273-279.
- [17]. Aarts, E., Korst, J., and Michiels, W.: "Simulated annealing": "Search methodologies" (Springer, 2014), pp. 265-285.

A FAIR VM ALLOCATION FOR CLOUD COMPUTING BEASED ON GAME THEORY

Bui Thanh Khiết, Nguyen Thi Nguyet Que, Ho Dac Hung, Pham Tran Vu, Tran Cong Hung

ABSTRACT: *Managing and using resources on cloud computing is a major challenge for service providers as well as end-users. Resource allocation solutions will be determined by using the user's resource requirements and quality of service. Physical resources in cloud computing environment are heterogeneous, and resource requirements from users are also not the same in kind of resources. It difficult to allocate resources especially during periods of scarce resources. In this study, we built a resource allocation model to ensure fairness between physical servers as well as ensure efficient use of resources based on game theory. A fair resource allocation strategies will be found by applying Simulated Annealing algorithm. The simulation results show how to use the parameters to achieve a fair VM allocation. These parameters depend on objectives of cloud computing service providers.*

Keywords: *Cloud computing, VM allocation, Game theory, Simulated Annealing.*