

THUẬT TOÁN DI TRUYỀN TRONG CUNG CẤP TÀI NGUYÊN CHO DỊCH VỤ ẢO HÓA TỪ NỀN TẢNG MÁY CHỦ CHIA SẼ ĐỒNG NHẤT

A GENETIC ALGORITHM IN RESOURCE ALLOCATION FOR VIRTUAL SERVICE BASED ON HOMOGENEOUS SHARED HOSTING PLATFORMS

Phạm Nguyễn Minh Nhật¹, Lê Văn Sơn²

¹Trường Cao đẳng Công nghệ Thông tin Hữu nghị Việt – Hàn; Email: nhut_np@yahoo.com

²Trường Đại học Sư phạm, Đại học Đà Nẵng; Email: levansupham2004@yahoo.com

TÓM TẮT - Tối ưu hóa tài nguyên để cung cấp cho dịch vụ ảo hóa đáp ứng yêu cầu khai thác tài nguyên hiệu quả trong Điện toán Đám mây là vấn đề đang được quan tâm hiện nay. Trong bài báo này, chúng tôi nghiên cứu bài toán cung cấp tài nguyên đa chiều từ nền tảng máy chủ chia sẻ đồng nhất cho dịch vụ ảo hóa, đưa ra công thức tính trên cơ sở bài toán quy hoạch tuyến tính nhằm tối thiểu hóa số máy chủ vật lý, áp dụng các thuật toán di truyền để giải và đánh giá thông qua odaymô phỏng trên nhiều kịch bản thử nghiệm. Tối ưu hóa tài nguyên để cung cấp cho dịch vụ ảo hóa đáp ứng yêu cầu khai thác tài nguyên hiệu quả trong Điện toán Đám mây là vấn đề đang được quan tâm hiện nay.

TỪ KHÓA - cung cấp tài nguyên, điện toán đám mây, thuật toán di truyền, máy ảo, quy hoạch tuyến tính.

ABSTRACT - In cloud computing, optimizing resources for virtual services to meet the requirements of efficient resource exploitation is the concern today. In this paper, we investigate the problem of providing multi-dimensional resources based on homogeneous shared hosting platforms for virtual services. We construct the problem as an optimization formulation that uses a linear programming to minimize the number of physical servers. The solution to this formulation is applying a Genetic algorithm to solve and evaluate through emulation-based program. In cloud computing, optimizing resources for virtual services to meet the requirements of efficient resource exploitation is the issue today.

KEY WORDS - resource allocation; cloud computing; GA; virtual machine; linear programming

1. Đặt vấn đề

Công nghệ ảo hóa cho phép phân vùng tài nguyên của Y máy chủ vật lý ($Y \geq 1$) thành S máy ảo ($S \geq 1$) để thực thi các ứng dụng theo yêu cầu. Kiến trúc phân tầng trong công nghệ ảo hóa được trình bày trong [1].

Hệ thống gồm nhiều máy chủ vật lý có cấu hình giống nhau, được kết nối với nhau để chia sẻ các tài nguyên được gọi là nền tảng máy chủ chia sẻ đồng nhất [3, 5]. Vấn đề cần quan tâm là tối thiểu hóa tài nguyên từ nền tảng đó để cung cấp cho dịch vụ ảo hóa, nhưng đảm bảo yêu cầu chất lượng dịch vụ QoS (Quality of Service). Việc quản lý tài nguyên trong nền tảng máy chủ chia sẻ đã được các nhà chuyên môn quan tâm nghiên cứu, thể hiện trong các công trình [3, 4] nhưng giới hạn máy chủ là nguồn tài nguyên liên khối.

Nội dung bài báo này xem xét nhiều khía cạnh tài nguyên (tài nguyên vật lý) và sử dụng hàm mục tiêu tuyến tính trên cơ sở bài toán quy hoạch tuyến tính nhằm tối thiểu hóa số máy chủ vật lý mà chúng tôi đã nêu ra trong công trình trước đây [5]. Bài toán cung cấp tài nguyên được xem xét trong cả hai trường hợp: tĩnh và động, nhưng chúng tôi tập trung giải quyết bài toán cho trường hợp tĩnh (nghĩa là nhu cầu tài nguyên không thay đổi).

Mặt khác, bài toán cung cấp tài nguyên tối ưu là các bài toán NP-Khó [5], nên việc sử dụng các thuật toán heuristic, trong đó có thuật toán di truyền [2, 6] để giải là cần thiết và được các nhà chuyên môn quan tâm. Trong nội dung bài báo, chúng tôi dựa vào thuật toán di truyền để giải bài toán cung cấp tài nguyên cho các dịch vụ ảo hóa từ nền tảng máy chủ chia sẻ đồng nhất thông qua các kịch bản được xây dựng.

Những nội dung chính của bài báo có thể tóm tắt như sau:

1) Áp dụng bài toán cung cấp tài nguyên dựa trên nền tảng máy chủ chia sẻ đồng nhất dưới dạng bài toán quy hoạch tuyến tính mà chúng tôi đã nêu ra trong [5].

2) Giải quyết bài toán dựa vào thuật toán di truyền bằng việc đưa ra hàm đo độ thích nghi (fitness function) nhằm tối thiểu số máy chủ vật lý.

Phần còn lại của bài báo được tổ chức như sau: Mục 2 trình bày mô hình hóa bài toán cung cấp tài nguyên dưới dạng bài toán quy hoạch tuyến tính. Mục 3 giới thiệu sơ lược thuật toán di truyền và áp dụng để giải bài toán cung cấp tài nguyên. Mục 4 trình bày kết quả thực nghiệm trên nhiều kịch bản mô phỏng và so sánh với kết quả của các thuật toán trong [5]. Kết luận và đề xuất hướng phát triển ở mục 5.

2. Cung cấp tài nguyên cho dịch vụ ảo hóa trong nền tảng máy chủ chia sẻ đồng nhất

2.1. Tài nguyên và nhu cầu tài nguyên

Xét một nền tảng máy chủ chia sẻ đồng nhất [5], gồm cụm các máy chủ vật lý có cấu hình giống nhau, được kết nối bằng thiết bị mạng tốc độ cao để chia sẻ tài nguyên nhằm cung cấp cho dịch vụ ảo hóa. Trong đó, mỗi dịch vụ ảo hóa là một máy ảo và hệ thống đảm bảo rằng các yêu cầu dịch vụ gửi đến máy chủ vật lý thích hợp.

Khi hệ thống nhận được yêu cầu cung cấp *cụm máy ảo*, hệ thống sẽ thiết lập các máy ảo để thực thi yêu cầu đó. Các máy ảo chạy trên máy chủ vật lý dưới sự quản lý của hypervisor [1] và tiêu thụ tài nguyên theo tỷ lệ khác nhau. *Hệ thống quản lý các máy ảo* có nhiệm vụ kiểm soát các hypervisor để xác định tỷ lệ tiêu thụ tài nguyên của các máy ảo. *Bộ cung cấp tài nguyên* có nhiệm vụ ra quyết định từ chối hoặc đáp ứng yêu cầu, phân chia tỷ lệ tài nguyên đến các máy ảo. Mục đích là nghiên cứu xây dựng bài toán như một phần của *bộ cung cấp tài nguyên*, sử dụng thuật toán di truyền để tìm số máy chủ vật lý tối thiểu dựa trên nhu cầu tài nguyên nhằm cung cấp cho dịch vụ ảo hóa.

Để đáp ứng nhu cầu tài nguyên cho dịch vụ ảo hóa, mỗi máy chủ vật lý cung cấp một số loại tài nguyên, như: CPU,

RAM, băng thông I/O,... Trong thực tế, mỗi dịch vụ ảo hóa có hai loại nhu cầu tài nguyên: *nhu cầu chặt* và *nhu cầu lỏng*. *Nhu cầu chặt* biểu thị phần trăm cụ thể của tài nguyên yêu cầu, dịch vụ không hưởng lợi từ phần lớn hơn và không thể hoạt động với phần nhỏ hơn từ tài nguyên được cung cấp. *Nhu cầu lỏng* biểu thị phần trăm tối đa của tài nguyên mà dịch vụ có thể sử dụng, dịch vụ không hưởng lợi từ phần lớn hơn nhưng có thể hoạt động với phần nhỏ hơn với chi phí giảm.

Tỷ số giữa phần trăm tài nguyên được cung cấp và phần trăm tài nguyên nhu cầu lỏng gọi là năng suất dịch vụ (NSDV). Ví dụ, khi dịch vụ có nhu cầu lỏng CPU là 60%, nhưng chỉ được cung cấp 30%, thì NSDV là $30/60 = 0.5$. Việc sử dụng tài nguyên đối với nhu cầu lỏng thường là quan hệ tuyến tính. Chẳng hạn, trong ví dụ trên, nếu dịch vụ được cung cấp 30% CPU (tức là chỉ một nửa so với nhu cầu) thì khả năng nó chỉ sử dụng 20% băng thông I/O (tức là chỉ một nửa so với nhu cầu). Điều này phù hợp với thực tế vì khi phần trăm công suất CPU cần cung cấp cho các ứng dụng giảm, dẫn đến tiêu hao tài nguyên khác cũng bị giảm (trong trường hợp này là băng thông I/O).

Như vậy, để đơn giản NSDV của tất cả *nhu cầu lỏng* có thể biểu diễn cùng một giá trị và giá trị của nó nằm trong khoảng 0 và 1. Trường hợp đặc biệt, nếu NSDV bằng 0 thì dịch vụ không được cung cấp tài nguyên (lỗi do thủ tục cung cấp tài nguyên), nếu NSDV bằng 1 thì tài nguyên được cung cấp bằng với tài nguyên được yêu cầu. Tuy nhiên, cần xét đến trường hợp có ràng buộc NSDV theo quy định đáp ứng yêu cầu QoS, ràng buộc đó được biểu diễn bởi tích số giữa *nhu cầu lỏng* với yêu cầu QoS và được gọi là *nhu cầu lỏng ràng buộc*. Chúng tôi cũng giả định rằng *nhu cầu chặt* hoàn toàn độc lập *nhu cầu lỏng*, cung cấp tài nguyên cũng bị lỗi nếu *nhu cầu chặt* của dịch vụ không được đáp ứng.

2.2. Hàm mục tiêu và các ràng buộc

Giả định mỗi dịch vụ ảo hóa là một máy ảo riêng lẻ và có nhu cầu tài nguyên không đổi (trường hợp tĩnh). Bài toán cung cấp tài nguyên (VSMSA) trên cơ sở bài toán quy hoạch tuyến tính được mô tả như sau:

Xét S_i dịch vụ, với $i = 1, \dots, n$; $S_i > 0$. Các cụm máy chủ có Y_j máy vật lý giống nhau (đồng nhất), với $j = 1, \dots, m$; $Y_j > 0$. Mỗi máy chủ cung cấp D_k loại tài nguyên, với $k = 1, \dots, d$.

Gọi r_{ik} là biến biểu thị nhu cầu tài nguyên của dịch vụ thứ i với loại tài nguyên k , có giá trị giữa 0 và 1. x_{ij} là số nhị phân có giá trị 1 nếu dịch vụ i chạy trên máy chủ vật lý j và bằng 0 nếu ngược lại.

Gọi α_{ik} là số nhị phân, bằng 1 nếu r_{ik} là một *nhu cầu chặt*, và bằng 0 nếu r_{ik} là một *nhu cầu lỏng*, β_{ij} biểu thị năng suất dịch vụ (NSDV) của dịch vụ S_i trên máy chủ Y_j , y_j số máy chủ vật lý để cung cấp tài nguyên cho dịch vụ i . Bài toán cung cấp tài nguyên được biểu diễn dưới dạng bài toán quy hoạch tuyến tính với các ràng buộc và hàm mục tiêu như sau:

$$x_{ij} \in \{0,1\}, \beta_{ij} \in Q, \quad \forall i, j \quad (1)$$

$$\sum_j x_{ij} = 1, \quad \forall i \quad (2)$$

$$y_j \geq x_{ij}, \quad \forall i, j \quad (3)$$

$$\sum_i (\beta_{ij}(1 - \alpha_{ik}) + \alpha_{ik}) r_{ik} x_{ij} \leq 1, \quad \forall k, j \quad (4)$$

và hàm mục tiêu là:

$$\min \sum_j y_j \quad (5)$$

Ràng buộc (1) xác định miền của các biến. Ràng buộc (2) biểu thị trạng thái có một dịch vụ S_i chạy trên máy chủ Y_j . Ràng buộc (3) biểu thị trạng thái mà một máy chủ Y_j có được sử dụng hay không. Ràng buộc (4) biểu thị trạng thái mà tổng số phần trăm nhu cầu tài nguyên cho dịch vụ S_i luôn luôn nhỏ hơn hoặc bằng tổng số tài nguyên của máy chủ vật lý Y_j , biểu thức trong phép lấy tổng cho thấy rằng: nếu nhu cầu tài nguyên r_{ik} là nhu cầu lỏng thì $\alpha_{ik} = 0$ và phần trăm tài nguyên D_k được sử dụng trên máy chủ Y_j là $\beta_{ij} \times r_{ik}$; nếu nhu cầu tài nguyên r_{ik} là nhu cầu chặt thì $\alpha_{ik} = 1$ và phần trăm tài nguyên D_k được sử dụng trên máy chủ Y_j là r_{ik} . Cuối cùng, ràng buộc (5) chính là hàm mục tiêu biểu thị số máy chủ vật lý để cung cấp tài nguyên cho dịch vụ ảo hóa. Mục tiêu là tối thiểu hóa y_j .

3. Thuật toán di truyền trong cung cấp tài nguyên cho dịch vụ ảo hóa

3.1. Giới thiệu về thuật toán di truyền

Thuật toán di truyền [2, 6] được xây dựng dựa trên quy luật tiến hóa sinh học của quần thể sống. Các cá thể trải qua quá trình phát triển và sinh sản để tạo ra các cá thể mới cho thế hệ tiếp theo. Trong quá trình tiến hóa những cá thể xấu (dựa trên một tiêu chuẩn nào đó hay còn gọi là độ thích nghi) sẽ bị đào thải. Ngược lại, những cá thể tốt được giữ lại. Một số khái niệm liên quan đến thuật toán di truyền gồm:

Biểu diễn của cá thể: Là việc biểu diễn các cá thể sao cho mỗi cá thể thể hiện một giải pháp của bài toán đang được quan tâm.

Đánh giá độ thích nghi: Là việc đánh giá khả năng thích nghi (phù hợp) của mỗi cá thể (giải pháp) đối với môi trường (bài toán đang xét). Việc đánh giá dựa trên hàm thích nghi (fitness function).

Lai ghép: Là quá trình tạo ra cá thể mới dựa trên cá thể đã có, gọi là cá thể cha-mẹ. Hai cá thể con được tạo ra bằng cách hoán đổi các gen từ cá thể cha-mẹ.

Đột biến: Là quá trình tạo ra cá thể mới từ một cá thể ban đầu bằng cách thay đổi một số gen của nó.

Chọn lọc và thay thế: Là quá trình chọn những cá thể từ quần thể hiện tại để tạo ra thế hệ sau của nó. Trong quá trình này, những cá thể có độ thích nghi lớn hơn hoặc bằng với độ thích nghi tiêu chuẩn sẽ được giữ lại và độ thích nghi của các cá thể trong quần thể sẽ hoàn thiện hơn sau nhiều thế hệ.

Điều kiện dừng: Thuật toán di truyền là một quá trình ngẫu nhiên, nên không thể đảm bảo chắc chắn thuật toán sẽ dừng sau các bước hữu hạn. Vì vậy, để đảm bảo thuật toán di truyền sẽ kết thúc, người ta thường phải định nghĩa điều kiện dừng cho thuật toán.

3.2. Thuật toán di truyền trong cung cấp tài nguyên

Như đã trình bày trong mục 1, thuật toán di truyền cho phép tối ưu các bài toán NP-khó với thời gian cho phép. Phần này tập trung trình bày thuật toán di truyền để giải bài toán cung cấp tài nguyên cho dịch vụ ảo hóa như đã trình bày trong mục 2.

Biểu diễn của các cá thể: Mỗi cá thể là một mảng số nguyên một chiều S_i có độ dài n (S_i : dịch vụ ảo hóa và $i =$

$1, \dots, n$), trong đó giá trị thứ i (tức là, phần tử S_i) bằng Y_j nếu dịch vụ S_i được phân bổ trên máy chủ Y_j (Y_j : máy chủ vật lý và $j = 1, \dots, m$).

Khởi tạo quần thể ban đầu: Sinh ra P (P là tham số thiết kế) cá thể ban đầu một cách ngẫu nhiên. Một cá thể ban đầu được tạo ra bằng cách gán ngẫu nhiên mỗi dịch vụ S_i đến một máy chủ Y_j .

Hàm đo độ thích nghi:

Gọi $F_k(Y_j)$ là tổng phần trăm nhu cầu tài nguyên k của dịch vụ S_i mà máy chủ Y_j cần để cung cấp cho dịch vụ S_i :

$$F_k(Y_j) = \left(\sum_{i \in Y_j} (\beta_{ij}(1 - \alpha_{ik}) + \alpha_{ik}) r_{ik} x_{ij} \right)^K \quad (6)$$

Trong đó hằng số mũ K ($K > 1$) là tham số biểu thị mối quan tâm đến những máy chủ vật lý cung cấp số tài nguyên nhiều hơn khi so sánh với các máy chủ vật lý khác. Trường hợp K lớn sẽ dẫn đến khả năng hội tụ sớm của thuật toán. Thực nghiệm cho thấy giá trị $K=2$ là tốt nhất.

- Hàm thích nghi của thuật toán di truyền cho toàn bộ nhu cầu tài nguyên k là:

$$f_{VMSA} = \sum_{k=1}^d \left(\frac{1}{m} \sum_{j=1}^m F_k(Y_j) \right) \quad (7)$$

Trong đó, m là tổng số máy chủ vật lý Y_j được dùng.

Phép toán lai ghép: Chúng tôi sử dụng toán tử lai ghép một điểm (one-point), trong đó mỗi nhiễm sắc thể cha mẹ cắt thành hai đoạn và hai nhiễm sắc thể con mới được tạo ra bằng cách ghép các đoạn của mỗi nhiễm sắc thể cha mẹ. Quá trình lai ghép được thực hiện với mọi cặp có thể của các cá thể với một xác suất lai ghép là p_c (p_c là tham số thiết kế).

Phép toán đột biến: Toán tử đột biến hoán đổi ngẫu nhiên hai dịch vụ của hai máy chủ khác nhau. Quá trình đột biến được thực hiện với mọi cá thể với giá trị xác suất đột biến là p_m (p_m là tham số thiết kế).

Chọn lọc các cá thể cho thế hệ kế tiếp: Chọn P cá thể cho thế hệ sau, một cá thể mới được tạo ra sau tiến trình: khởi tạo, đột biến, lai ghép có thể không phù hợp với một cấp phát tài nguyên hợp lệ và chúng tôi gọi là các nhiễm sắc thể không hợp lệ. Các cá thể này chúng tôi vẫn sử dụng trong dân số (quần thể) mới. Sau khi một nhiễm sắc thể không hợp lệ được tạo ra, chúng tôi sử dụng một thuật toán tham lam để làm cho chúng trở thành các nhiễm sắc thể hợp lệ. Thuật toán này duyệt qua các máy chủ theo một trật tự tùy ý, với mỗi máy chủ quá tải thuật toán sẽ di chuyển các dịch vụ đến các máy chủ khác được nạp ít dịch vụ hơn.

Phương pháp này đã được kiểm chứng trong thực nghiệm là nó làm giảm sự đa dạng dân số nhiễm sắc thể và dẫn đến xu hướng cấp phát tài nguyên hợp lệ.

Điều kiện dừng: Thuật toán sẽ dừng sau G thế hệ (G là tham số thiết kế) hoặc giá trị trung bình về độ thích nghi của các cá thể không thay đổi.

Tóm lại, thuật toán di truyền cho việc cung cấp tại nguyên được viết như sau:

Bắt đầu

a. $t=0$;

b. Khởi tạo quần thể ban đầu $P(t)$;

c. Tính độ thích nghi cho các cá thể thuộc $P(t)$;

d. Trong khi (điều kiện dừng chưa thỏa mãn) lặp

i. $T=t+1$;

ii. Lựa chọn $P(t)$ từ $P(t-1)$;

iii. Lai ghép trên $P(t)$ để nhận được $Q(t)$;

iv. Đột biến trên $P(t)$ để nhận được $R(t)$;

v. Chọn lọc từ $P(t-1) \cup Q(t) \cup R(t)$ để nhận được $P(t)$

e. Hết lặp;

Kết thúc;

4. Các kết quả thử nghiệm

Phương pháp thực nghiệm

Để đánh giá thuật toán, chúng tôi tạo ra tập các mẫu thực nghiệm một cách ngẫu nhiên, như sau: Xét S dịch vụ và Y máy vật lý với chiều tài nguyên D . Tương ứng với mỗi dịch vụ, số nhu cầu chặt là $D/2$ và số nhu cầu lỏng là $D/2$. Tất cả các nhu cầu tài nguyên được lấy mẫu từ một phân bố xác suất với trung bình μ và độ lệch chuẩn σ . Mỗi dịch vụ có ρ xác suất để có một yêu cầu QoS.

Chúng tôi giả định giá trị của các tham số như sau: năng suất dịch vụ $\beta_{ij} = 0.5$, số dịch vụ $S = 32, 64, 128, 256, 512$ chiều tài nguyên $D = 6$ (trong đó, số nhu cầu chặt = số nhu cầu lỏng = 3), $\mu = 0.5$, $\sigma = 0.25, 0.5, 1.0$, $\rho = 0.25, 0.5, 1.0$ và tất cả các yêu cầu QoS có giá trị là 0.5 (tức là một nửa nhu cầu lỏng của dịch vụ phải được đáp ứng), thực nghiệm với các giá trị khác hoặc với các giá trị ngẫu nhiên cũng dẫn đến kết quả tương tự. Tương ứng với các giá trị đó sẽ có $1 * 5 * 1 * 1 * 3 * 3 = 45$ kịch bản. Với mỗi kịch bản chúng tôi tạo ra 100 mẫu ngẫu nhiên, như vậy sẽ có 4500 mẫu dữ liệu đầu vào để đánh giá thuật toán. Các mẫu thực nghiệm được tạo ra lưu vào tập tin có cấu trúc như Hình 1.

```
<Số dịch vụ>
<Số nhu cầu lỏng>
<Số nhu cầu chặt>
<Giá trị QoS 1><Nhu cầu lỏng 1 của dịch vụ 1> <nhu cầu lỏng 2 của dịch vụ 1>...<nhu cầu
chặt 1 của dịch vụ 1> <nhu cầu chặt 2 của dịch vụ 1> ...
<Giá trị QoS 2><Nhu cầu lỏng 1 của dịch vụ 2> <nhu cầu lỏng 2 của dịch vụ 2>...<nhu cầu
chặt 1 của dịch vụ 2> <nhu cầu chặt 2 của dịch vụ 2> ...
...
```

Hình 1. Cấu trúc tập tin dữ liệu thực nghiệm

Với mỗi thuật toán ta sử dụng hai thước đo. Thước đo thứ nhất là số máy chủ vật lý tối thiểu tương ứng với 5 giá trị của số dịch vụ là $S = 32; 64; 128; 256; 512$. Thước đo thứ hai là thời gian thực hiện thuật toán tính bằng giây. Giá trị của hai thước đo được lấy trung bình từ 900 (tức là, $3 * 3 * 100$) mẫu thực nghiệm. Chương trình mô phỏng các thuật toán được thực hiện bằng ngôn ngữ C++ và thời gian thực hiện được đo trên máy tính đơn có bộ vi xử lý Intel Core Duo 1.86 GHz, RAM 2Gb.

Các tham số của thuật toán di truyền

Chúng tôi sử dụng kích thước dân số $P=100$, chạy trên $G=2000$ thế hệ, với giá trị xác suất đột biến là $p_m=0,1$ giá trị xác suất lai là $p_c = 0,25$ và hằng số $K=2$. Các thông số này được ước tính theo kinh nghiệm dựa trên thực nghiệm.

Kết quả và nhận xét

Giá trị số máy chủ vật lý tối thiểu (giá trị hàm mục tiêu) và thời gian thực hiện tương ứng với số dịch vụ khi thực hiện thuật toán được trình bày trong Bảng 1.

Bảng 1. Thời gian thực hiện và số máy chủ vật lý tối thiểu

Số dịch vụ	S=32	S=64	S=128	S=256	S=512
Thời gian thực hiện (s)	0.0010	0.00307	0.02076	0.04126	0.0893
Số máy chủ tối thiểu	24	47	90	174	344

So sánh kết quả thực nghiệm ở Bảng 1 và kết quả 6 thuật toán mà chúng tôi đã trình bày trong [5] trên cùng 1 tập mẫu dữ liệu thực nghiệm, nhận thấy rằng: thời gian thực hiện cao hơn không nhiều, trường hợp số dịch vụ lớn (≥ 255 dịch vụ) thì thời gian thực hiện ngắn hơn thời gian thực hiện của các thuật toán BestFitDesMax, BestFitDesLex, BestFitDesSum. Giá trị hàm mục tiêu (số máy chủ vật lý) giống nhau (trừ thuật toán BestFixDesLex cho giá trị tốt

hơn). Hơn nữa, thời gian thực hiện thuật toán di truyền tương đối nhỏ và có thể áp dụng trong thực tế.

5. Kết luận

Nội dung bài báo trình bày vấn đề cung cấp tài nguyên (tài nguyên vật lý) tĩnh, đa chiều dựa trên nền tảng máy chủ chia sẻ đồng nhất cho dịch vụ ảo hóa với ràng buộc tối ưu và đảm bảo các yêu cầu QoS; mỗi dịch vụ là một máy ảo đơn lẻ. Trên cơ sở bài toán tối ưu trong [5], chúng tôi đưa ra thuật toán di truyền để cài đặt và đánh giá nhằm tối thiểu hóa số máy chủ vật lý. Hướng nghiên cứu mở rộng và phát triển của đề tài là bài toán cung cấp tài nguyên động trong môi trường không đồng nhất, là hướng nghiên cứu cần thiết và cũng đang được các nhà chuyên môn quan tâm.

TÀI LIỆU THAM KHẢO

[1] Lê Văn Sơn, Phạm Nguyễn Minh Nhựt, *Vấn đề cung cấp tài nguyên máy ảo trên cơ sở hạ tầng tính toán đám mây*, Tạp chí khoa học và công nghệ, Đại học Đà Nẵng, 5(11), (2012), 63-71.

[2] Trịnh Thị Thúy Hằng, Lê Trọng Vĩnh, Hoàng Chí Thành, Nguyễn Thanh Thủy, *Tối ưu đa mục tiêu trong việc lập lịch cho hệ thống tính toán lưới*, Tạp chí Tin học và Điều khiển học, 25 (1) 2009, 79-87.

[3] B. Urgaonkar, P. Shenoy and T. Roscoe, *Resource Overbooking and Application Profiling in Shared Hosting Platforms*, SIGOPS Operating Systems Review, 36(SI), (2002) 239-254.

[4] M. Aron, P. Druschel and W. Zwaenepoel, *Cluster Reserves: A Mechanism for Resource Management in Cluster-based Network Servers*, In Proceedings of the 2000 ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems, New York, USA, (2000), 90-101.

[5] Phạm Nguyễn Minh Nhựt, Đoàn Văn Ban, Lê Văn Sơn, *Mô hình nền tảng máy chủ chia sẻ và bài toán Vector Packing trong cung cấp tài nguyên cho dịch vụ ảo hóa*, Tạp chí tin học và điều khiển học, 30(1), (2014), 63-72.

[6] Christian Blum, Andrea Roli, *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*, ACM Computing Surveys, 35(3), (2003), 268-308.

(BBT nhận bài: 24/03/2014, phản biện xong: 06/04/2014)