

MỘT CÁCH TIẾP CẬN VIỆC CẤP PHÁT TÀI NGUYÊN MÁY CHỦ ẢO THEO THỜI KHÓA BIỂU TẠI TRƯỜNG ĐẠI HỌC

AN APPROACH TO ALLOCATING VIRTUAL SERVER RESOURCES BASED ON A
TIMETABLE FOR UNIVERSITIES

BÙI MINH PHỤNG^(*) và BÙI XUÂN AN^(**)

TÓM TẮT: *Tại các trường đại học ngày nay, nhu cầu sử dụng tài nguyên tính toán để hỗ trợ cho việc dạy, học và nghiên cứu đang dần trở nên phổ biến. Tuy nhiên, quá trình cấp phát tài nguyên cho máy chủ ảo đang phải làm thủ công và việc cấp phát tài nguyên CPU, RAM, NETWORK cho từng máy chủ ảo chưa được tối ưu về số lượng máy chủ vật lý tối thiểu cần dùng. Cũng đã có một số thuật toán nhằm tối ưu việc cấp phát tài nguyên như Vector Bin Packing, FFD, Heuristic nhưng chưa đáp ứng được việc cấp phát tài nguyên cho một số tổ chức như trường đại học, cần cấp phát - thu hồi máy chủ ảo linh động và liên tục dựa theo một thời khóa biểu. Bài viết đề xuất một thuật toán kết hợp với thuật toán Heuristic Norm-based Greedy (L2) để cấp phát tài nguyên cho máy chủ ảo một cách tự động dựa trên một thời khóa biểu được đưa vào theo từng thời điểm.*

Từ khóa: *máy chủ ảo, tối ưu cấp phát tài nguyên, cấp phát tài nguyên theo lịch biểu.*

ABSTRACTS: *Today in universities, the need to use computational resources to support teaching, learning and studying is gradually becoming widespread. However, the process of allocating virtual machine resources has to be done manually, and allocating CPU, RAM, NETWORK resources for each virtual server has not been optimized for the minimum number of physical servers required. There are several algorithms for for optimizing the allocation of resources such as Vector Bin Packing, FFD, Heuristic, but they do not meet the requirement of universities that the virtual server resource allocation is flexible based on a timetable. The article proposes an algorithm that combines the Heuristic Norm-based Greedy (L2) algorithm to automatically allocate resources to a virtual server based on a timetable that is inserted from time to time.*

Key words: *virtual Server, optimization of resource allocation, resource allocation based on timetable.*

^(*) ThS. Trường Đại học Văn Lang, buiminhpung@vanlanguni.edu.vn, Mã số: TCKH09-19-2018

^(**) ThS. Trường Đại học Văn Lang, buixuanan@vanlanguni.edu.vn

1. ĐẶT VẤN ĐỀ

Điện toán đám mây là một xu hướng phát triển của Công nghệ Thông tin ngày nay bởi những lợi ích cho các tổ chức rất lớn như: giảm chi phí đầu tư hệ thống, chi phí quản trị; tài nguyên cấp phát linh động, co giãn theo nhu cầu, tận dụng được tối đa tài nguyên đã đầu tư; khả năng cấp/phát thu hồi tài nguyên nhanh chóng theo thời điểm, tùy mục đích sử dụng [1]. Một trong những hoạt động quan trọng nhất của điện toán đám mây là tạo, quản lý, thu hồi các máy chủ ảo.

Với các cơ quan có số lượng người sử dụng tài nguyên lớn như các trường đại học, việc cấp phát tài nguyên máy chủ ảo cho sinh viên thực hành, làm đồ án khá phức tạp và khó khăn cho người quản trị hệ thống. Với xu hướng ứng dụng công nghệ thông tin trong giảng dạy, số lượng máy chủ ảo tăng, yêu cầu người quản trị hệ thống phải có giải pháp để quản lý hiệu quả, để có thể tổng hợp các thông tin quản trị trong thời gian ngắn, giải pháp là cấp phát máy chủ ảo. Vì vậy, để tận dụng và tiết kiệm tài nguyên cũng như quản lý hệ thống một cách hiệu quả, việc cấp phát - thu hồi máy chủ ảo cần được triển khai dựa trên thời khóa biểu.

Có những phần mềm nguồn mở cho phép tạo hệ thống đám mây dùng riêng cho tổ chức hoặc dùng để tạo dịch vụ cung cấp dịch vụ đám mây cho khách hàng. Mặc dù những phần mềm này khá linh động, được cộng đồng nguồn mở đông hỗ trợ, nhưng chúng vẫn có những nhược điểm riêng [2], việc tạo và cấp phát tài nguyên cho máy chủ ảo vẫn là cố định và thao tác thủ công.

Tối ưu trong quá trình cấp phát máy chủ ảo là với số lượng máy vật lý (n), ta cấp phát tài nguyên cho (m) máy chủ ảo gồm RAM, CPU, NETWORK sao cho số máy vật lý (n) là nhỏ nhất.

Để tối ưu quá trình cấp phát tài nguyên cho máy chủ ảo, Rina Panigrahy và các cộng sự đã đưa ra bài toán Vector Bin Packing, trong đó, tác giả so sánh thuật toán First Fit Decreasing (FFD), Dot Product và Norm-based Greedy (L2). Kết quả so sánh giữa 3 thuật toán, Norm-based Greedy (L2) cho ra kết quả tốt nhất [3], nghĩa là giải thuật Norm-based Greedy sẽ cho ra các thông số tài nguyên cấp phát tối ưu nhất.

Tuy nhiên, các thuật toán trên cũng chỉ tập trung vào việc cấp phát tài nguyên tối ưu nhất, chưa đề cập đến cách thức cấp phát tài nguyên linh động, phù hợp với từng thời điểm. Cụ thể là cấp phát tài nguyên linh động theo thời khóa biểu.

Một số thuật toán phân bố tài nguyên theo thời khóa biểu như thuật toán First-Fit [4] hay thuật toán Round-Robin [5] là các thuật toán thường được sử dụng để cấp phát tài nguyên theo lịch biểu. Đây là các thuật toán dạng kinh điển, dễ sử dụng, dễ hiểu nhưng có nhược điểm là không được tối ưu về việc cấp phát tài nguyên.

Vì vậy, làm sao để có một thuật toán vừa phân phối tài nguyên cho máy chủ ảo theo lịch biểu, vừa tối ưu các tài nguyên phân phối. Đây chính là nội dung mà bài báo đề cập tới, đó là xây dựng một thuật toán có thể nhận biết được các lịch biểu đưa vào và kết hợp với một số thuật toán tối ưu heuristic Norm-based Greedy (L2)

để có thể cấp phát tài nguyên các máy chủ ảo một cách tự động.

2. ĐỀ XUẤT GIẢI THUẬT

Đặt vấn đề là các dữ liệu đầu vào gồm:

Thời khóa biểu sử dụng các máy chủ ảo gồm các thông tin: thời gian bắt đầu, thời gian kết thúc, tài nguyên yêu cầu cho mỗi máy chủ ảo.

Số máy chủ vật lý và cấu hình về tài nguyên các máy chủ vật lý.

Kết quả đầu ra: sắp xếp các máy chủ ảo vào các máy chủ vật lý sao cho số máy chủ vật lý sử dụng là ít nhất.

Tuy nhiên, trong khuôn khổ bài báo, chúng tôi thực hiện dựa trên các ràng buộc, quy định như sau:

Xét máy chủ vật lý, máy ảo với 3 loại tài nguyên: số CPU; dung lượng RAM; băng thông mạng (NETWORK). Tài nguyên về đĩa cứng (DISK) sẽ được lưu trữ trên bộ lưu trữ tập trung nên sẽ không xét ở đây.

Các máy chủ vật lý có tài nguyên giống nhau.

Không có máy ảo nào yêu cầu tài nguyên nhiều hơn tài nguyên máy chủ vật lý.

Đơn vị tính tài nguyên của các máy chủ ảo là (%) – tỷ lệ giữa tài nguyên máy chủ ảo yêu cầu với tài nguyên của máy chủ vật lý.

Thời khóa biểu được xây dựng theo dạng bảng với định dạng được quy ước trước (Hình 1).

	Thứ 2				Thứ 3			
	Tiết 1	Tiết 2	Tiết 3	Tiết 4	Tiết 5	Tiết 6	Tiết 7	Tiết 8
Sáng		VM1		VM3		VM6		VM8
			VM2		VM4		VM7	
				VM5				VM9

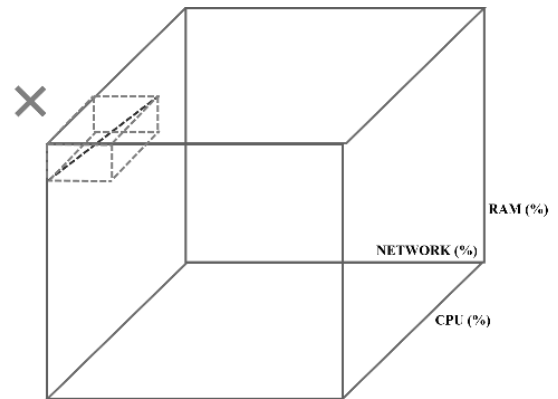
Hình 1. Thời khóa biểu sử dụng máy chủ ảo

Từ đề bài và ràng buộc trên, dựa vào thuật toán Norm-based Greedy L2:

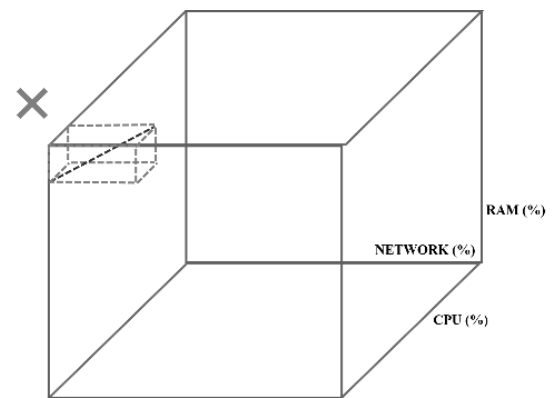
Máy vật lý, máy ảo là một hình hộp chữ nhật với độ lớn của các cạnh là độ lớn về tài nguyên.

Giá trị $\sum_i (l_i^l - r(t)_i)^2$ trong Norm-based Greedy L2 tương ứng với giá trị bình phương độ lớn đường chéo của hình hộp đại diện cho phần tài nguyên còn lại (phần tài nguyên chưa được cấp phát).

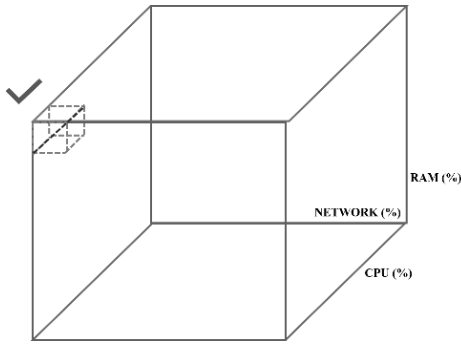
Việc xếp máy ảo vào máy vật lý sẽ xét đường chéo của phần tài nguyên còn lại (Hình 2, Hình 3, Hình 4): máy vật lý được chọn là máy có đường chéo của phần tài nguyên còn lại là nhỏ nhất sau khi xếp máy ảo vào (Hình 4).



Hình 2. Đường chéo khối còn trống của máy chủ vật lý (dài)



Hình 3. Đường chéo khối còn trống của máy chủ vật lý (dài)



Hình 4. Đường chéo khối còn trống của máy chủ vật lý (ngắn)

Chúng tôi đề xuất thuật toán Best Fit by Diagonal (BFD) như sau:

Thuật toán. Xếp máy ảo vào máy vật lý với BFD

```

1:  GETHOST( $PM_{List}$ ,  $VM$ )
2:   $min\_distance \leftarrow 1000$ ;
3:  while  $PM_{List} \neq \emptyset$  do
4:     $RemainResource \leftarrow PM_i.RemainResource - VM.Resource$ ;
5:    if  $RemainResource < 0$  then
6:      skip
7:    end if
8:     $distance \leftarrow get\_instance\_diagonal\_line(RemainResource)$ ;
9:    if  $distance < min\_distance$  then
10:      $min\_distance \leftarrow distance$ 
11:      $tempID \leftarrow i$ ;
12:   end if
13: end while
14:  $PM_{tempID}.RemainResource \leftarrow PM_{tempID}.RemainResource - VM.Resource$ ;
15: Return  $PM_{tempID}.ID$ ;
16: end

17: BEST_FIT_BY_DIAGNAL( $PM_{List}$ , FILE)
18:  $VM_{List} \leftarrow READTIMETABLE(FILE)$ ;
19:  $SORT\_VMS\_BY\_STARTTIME(VM_{List})$ ;
20: while  $VM_{List} \neq \emptyset$  do
21:   if  $VM_i.StopTime < now$  then
22:     while  $PM_{List} \neq \emptyset$  do
23:       if  $PM_i.ID = VM_i.SelectedHost$  then
24:          $PM_i.RemainResource \leftarrow PM_i.RemainResource + VM_i.Resource$ ;
25:       end if
26:     end while
27:   else
28:      $VM_i.SelectedHost \leftarrow GETHOST(PM_{List}, VM_i)$ ;
29:   end if
30: end while
31: End

```

Hàm GETHOST dùng để tìm ra máy chủ vật lý vừa nhất để tạo máy chủ ảo sau khi duyệt qua tất cả máy chủ vật lý. Gồm 2 tham số truyền vào là:

PM_{list} : danh sách các máy chủ vật lý.

VM: máy chủ ảo cần tạo.

Dòng (4): thử xếp máy ảo vào máy vật lý (i) bằng cách lấy phần tài nguyên còn lại của máy vật lý (i) trừ đi tài nguyên của máy ảo yêu cầu.

Dòng (5): nếu máy vật lý không chứa vừa máy ảo.

Dòng (6): thì bỏ qua máy vật lý (i) để xét tiếp máy vật lý (i) + 1.

Dòng (8): tìm đường chéo của hình hộp tạo ra bởi phần tài nguyên còn lại.

Dòng (9): nếu đường chéo của hình hộp nhỏ hơn đường chéo nhỏ nhất hiện tại thì thực hiện lệnh (10), (11).

Dòng (10): gán đường chéo nhỏ nhất bằng đường chéo hình hộp hiện tại.

Dòng (11): gán số thứ tự máy chủ vật lý đang duyệt vào biến tạm. Biến tempID giữ thứ tự của máy chủ vật lý.

Dòng (13): sau khi kết thúc quá trình lặp, ta xác định được máy chủ vật lý (PM) thứ (i) sẽ chứa máy chủ ảo (VM).

Dòng (14): trừ phần tài nguyên máy chủ vật lý (PM_i) đã bị máy chủ ảo (VM) chiếm.

Hàm BEST_FIT_BY_DIAGNAL nhận tham số truyền vào là danh sách máy chủ vật lý (PM_{list}) và file chứa thông tin thời khóa biểu (FILE).

Dòng (18): đọc thông tin thời khóa biểu từ file vào bộ nhớ.

Dòng (19): sắp xếp các máy chủ ảo (VM) theo thời gian bắt đầu.

Dòng (20): duyệt qua danh sách các máy chủ ảo (VM) trong 1 buổi.

Dòng (21): nếu máy chủ ảo đã dừng,...

Dòng (22): duyệt danh sách máy chủ vật lý (PM) để tìm máy chủ chứa máy chủ ảo (VM_i) đã dừng.

Dòng (24): cộng lại tài nguyên của máy chủ ảo (VM_i) đã dừng vào máy chủ vật lý (PM).

Dòng (28): gọi đến hàm tìm máy chủ vật lý (PM) chứa máy chủ ảo (VM) vừa nhất.

3. KẾT LUẬN

Bài báo dừng lại ở đề xuất một giải pháp cho việc cấp phát tài nguyên cho máy chủ ảo bằng một thuật toán cấp phát tài nguyên theo thời khóa biểu dựa trên thuật toán tối ưu Norm-based Greedy

L2 và chúng tôi đặt tên là BEST_FIT_BY_DIAGNAL (BFD). Sự kết hợp này sẽ giúp cho việc thực hiện cấp phát tài nguyên cho máy chủ ảo một cách tối ưu và tự động hoàn toàn dựa theo dữ liệu đưa vào là thời khóa biểu kèm theo các thông số tài nguyên là CPU, RAM, NETWORK và các thông số tài nguyên hiện có của các máy chủ vật lý.

Trong tương lai, chúng tôi sẽ đưa thuật toán vào chạy thực tế trên hệ thống Trường Đại học Văn Lang để đánh giá cụ thể hơn các thông số, cũng như so sánh với 2 thuật toán thông dụng là First Fit và Round-Robin để kiểm tra mức độ tối ưu của thuật toán do chúng tôi đề xuất.

TÀI LIỆU THAM KHẢO

- [1] Wikipedia. Advantage of Cloud Computing. Available from: https://en.wikipedia.org/wiki/Cloud_computing.
- [2] Openstackabout. Advantage and Disadvantage of Openstack. Available from: <http://openstackabout.blogspot.com/2012/11/openstack.html#advantage>.
- [3] Panigrahy, R., et al. (2011), Heuristics for vector bin packing. research. microsoft. com.
- [4] Chandrakar, V., P. Patel, and M. Roy, Review on virtual machine placement algorithms.
- [5] Mandal, S.K. (2013), On-Demand VM Placement on Cloud Infrastructure.

Ngày nhận bài: 08-3-2018. Ngày biên tập xong: 16-4-2018. Duyệt đăng: 23-7-2018