

Thuật toán lập lịch tác vụ cho máy ảo trên nền tảng điện toán đám mây

Lê Thị Thu Hương*

*Học viện Hành chính Quốc gia

Received: 21/8/2023; Accepted: 27/8/2023; Published: 5/9/2023

Abstract: Virtualization technology has been widely used to virtualize one server into multiple servers, which not only creates an operating environment for virtual machine-based cloud computing platforms but also has the ability to improve efficiency. Currently, most task scheduling algorithms used in cloud computing environments converge slowly or easily fall into local optimal states. This article introduces the greedy particle swarm optimization algorithm (G&PSO) to solve the task scheduling problem. The greedy algorithm is used to solve the initial element value of the swarm algorithm originating from a cloud-based virtual machine. Therefore, the performance and resources of the virtual machine used are improved compared to the traditional particle swarm optimization algorithm (PSO).

Keywords: Greedy algorithm, particle swarm optimization algorithm, cloud computing.

1. Giới thiệu

Cùng với sự phát triển của điện toán đám mây, công nghệ truyền tải lưu trữ hiệu suất cao, công nghệ ảo hóa đã trở thành một công cụ công nghệ thương mại phổ biến để cung cấp cho người dùng cơ sở hạ tầng, nền tảng và dịch vụ phần mềm từ trung tâm dữ liệu. Nguyên lý của công nghệ ảo hóa là ảo hóa phần cứng máy tính để chạy nhiều hệ điều hành độc lập trong cùng một môi trường phần cứng. Do đó, mỗi hệ điều hành có thể chạy nhiều ứng dụng cùng lúc trong không gian vật lý độc lập, nâng cao hiệu quả đáng kể của nền tảng điện toán đám mây. Công nghệ ảo hóa máy chủ là một trong những công nghệ chủ chốt. Trong công nghệ này, một máy vật lý duy nhất có thể được khởi tạo thành nhiều máy ảo và tài nguyên còn lại của mỗi máy vật lý có thể được ánh xạ và ảo hóa thành một máy ảo mới cho những người dùng khác. Nói chung, mức sử dụng thực tế của máy chủ vật lý chỉ là 7% đến 12%. Vì vậy, việc chạy nhiều máy chủ ảo trên một máy chủ sẽ không chỉ làm giảm toàn cục chi phí kinh doanh mà còn cải thiện đáng kể hiệu quả sử dụng máy chủ. Trên thực tế, tiềm năng lớn nhất của ảo hóa là tích hợp các máy chủ thành một đám mây riêng với nhiều máy chủ ảo độc lập nhằm mang lại hiệu quả sử dụng cao hơn nguồn lực tài nguyên có sẵn.

Khi môi trường điện toán đám mây cần mở rộng quy mô cho một số lượng lớn người dùng và tác vụ, cần thiết kể một thuật toán lập kế hoạch có thể phân phối hiệu quả các tác vụ và tài nguyên. Các thuật toán lập lịch tác vụ trên nền tảng đám mây hiện nay

được triển khai trên hệ thống nhiều máy chủ quy mô lớn. Tuy nhiên, do các doanh nghiệp vừa và nhỏ chỉ sử dụng một máy chủ duy nhất để xây dựng nền tảng đám mây riêng nên các thuật toán đó chưa đáp ứng được yêu cầu của họ. Vì thế, ứng dụng thuật toán lập lịch tác vụ dựa trên máy ảo là cần thiết để cải thiện hiệu quả toàn cục và chi phí vận hành của nền tảng đám mây như vậy. Do đó, bài viết này giới thiệu một thuật toán kết hợp thuật toán tham lam và thuật toán bầy đàn để tạo thành thuật toán lập lịch tác vụ (G&PSO) cho máy ảo dựa trên nền tảng điện toán đám mây. Trong một môi trường đám mây được triển khai bởi một máy chủ duy nhất, sử dụng thuật toán sẽ không chỉ giảm tổng thời gian thực hiện tác vụ mà còn cân bằng tải hệ thống và nâng cao hiệu quả của việc lập kế hoạch công việc và sử dụng tài nguyên của nền tảng điện toán đám mây.

2. Nội dung nghiên cứu

2.1. Vấn đề lập lịch tác vụ trên nền tảng điện toán đám mây

Bản chất việc giải quyết vấn đề lập lịch tác vụ của điện toán đám mây là thiết lập một chính sách lập lịch, nghĩa là mối quan hệ ánh xạ phù hợp được thiết lập giữa các tác vụ ứng dụng và tính toán nguồn lực nhằm đạt được sự phân bổ hợp lý và hiệu quả tài nguyên máy tính. Trong bài báo này, đề xuất thuật toán lập lịch tác vụ điện toán đám mây ảo hóa một máy chủ thành nhiều máy ảo, sau đó gán T nhiệm vụ độc lập cho M máy ảo không đồng nhất để thực thi (tức là một nhiệm vụ không thể được thực hiện chạy trên hai máy ảo, mỗi máy ảo chỉ có thể xử lý một

nhệm vụ tại một thời điểm và mỗi nhệm vụ đều có thuộc tính khác nhau), do đó giảm thiểu thời gian cần thiết để hoàn thành mọi nhệm vụ. Để đơn giản hóa quá trình mô phỏng, bài báo này sẽ bỏ qua bộ nhớ và các yêu cầu nguồn lực khác của nhệm vụ. Hơn thế nữa, thời gian thực hiện của mỗi nhệm vụ chỉ liên quan đến kích thước của tác vụ và thuộc tính của máy ảo. Tập tác vụ được biểu diễn dưới dạng $TS = \{t_1, t_2, \dots, t_n\}$, và kích thước nhệm vụ được biểu thị bằng MI, hiệu năng của máy ảo được biểu thị bằng MIPS. Thời gian thực hiện dự kiến của tác vụ TS_i chạy trên ảo máy VMS_j có thể được biểu diễn dưới dạng ma trận ETC:

$$ETC = \begin{bmatrix} ETC(11) & ETC(12) & \dots & ETC(1j) \\ ETC(21) & ETC(22) & \dots & ETC(2j) \\ \dots & \dots & \dots & \dots \\ ETC(i1) & ETC(i2) & \dots & ETC(ij) \end{bmatrix} \quad (1)$$

Trong đó $ETC(ij) = MI_{TS_i} / MIPS_{VMS_j}$, $i \in \{1, 2, \dots, T\}$, $j \in \{1, 2, \dots, M\}$, T là số lượng nhệm vụ, M là số lượng máy ảo và tải trên máy ảo, VMS_j là tổng thời gian thực hiện các tác vụ.

$$Load_{VMS_j} = \sum ETC(ij) \quad (2)$$

Hàm của mức độ cân bằng tải hệ thống được xác định là

$$Load_{level} = \frac{\min_{1 \leq j \leq M} Load_{VMS_j}}{\max_{1 \leq j \leq M} Load_{VMS_j}} \quad (3)$$

Trong đó, $\min_{1 \leq j \leq M} Load_{VMS_j}$ là thời gian tối thiểu để

tất cả các máy ảo hoàn thành tất cả các tác vụ trên và $\max_{1 \leq j \leq M} Load_{VMS_j}$ là thời gian tối đa để tất cả các máy ảo

hoàn thành tất cả các nhệm vụ trên. Vậy hàm của mức độ cân bằng tải hệ thống là tỉ số của tải tối thiểu và tải tối đa. Từ công thức (3), có thể được rút ra kết luận:

+ $\max_{1 \leq j \leq M} Load_{VMS_j} = 0$ nghĩa là các tác vụ chưa bắt

đầu lên lịch.

+ $Load_{level} = 0$ và $\max_{1 \leq j \leq M} Load_{VMS_j} \neq 0$ nghĩa là có máy ảo nhàn rỗi.

+ $Load_{level} = 1$ nghĩa là tải tối đa bằng tải tối thiểu và cân bằng tải là tốt nhất, $Load_{level}$ càng gần 1 thì càng tốt.

2.2. Thiết kế thuật toán lập lịch tác vụ trên nền tảng điện toán đám mây

Các mục tiêu lập lịch tác vụ trên nền tảng điện toán đám mây là giảm tổng số thời gian hoàn thành và cân bằng tải của hệ thống. Trước tiên, sử dụng

thuật toán tham lam để tìm ra giải pháp ban đầu G_{ov} và dự kiến tổng thời gian hoàn thành G_{ct} , sau đó khởi tạo toàn cục giải pháp tối ưu gbest của thuật toán bầy đàn (PSO) của G_{ov} và sử dụng $1/G_{ct}$ làm ngưỡng cập nhật tốt nhất vị trí của bầy đàn.

2.2.1. Khởi tạo các phần tử

Giả sử S, T và M lần lượt là kích thước của bầy đàn, số lượng tác vụ và số lượng máy ảo.

Vị trí của phần tử thứ i được biểu diễn bởi $P = \{P_{i1}, P_{i2}, \dots, P_{in}\}$; $1 \leq n \leq T$; $1 \leq i \leq S$, trong đó P_{ij} biểu diễn cho nhệm vụ thứ i được giao cho máy ảo thứ j và $1 \leq P_{ij} \leq M$.

Tốc độ của phần tử thứ i được biểu diễn bởi $V_i = \{V_{i1}, V_{i2}, \dots, V_{in}\}$, $1 \leq n \leq T$; $1 \leq i \leq S$ và V_{ij} phải thỏa mãn điều kiện $1 \leq V_{ij} \leq M$.

Vị trí ban đầu của phần tử là một số nguyên ngẫu nhiên được chọn từ $[1, M]$ và tốc độ của phần tử là một số nguyên ngẫu nhiên được chọn từ $[-(M-1), (M-1)]$. Vị trí tối ưu là gbest được khởi tạo bằng G_{ov} .

2.2.2. Hàm thích nghi

Hàm thích nghi được sử dụng để đánh giá giá trị của các vị trí phần tử. Khi toàn bộ thời gian hoàn thành nhệm vụ là tham số chính để tính toán lập lịch tác vụ trên đám mây, nghịch đảo của tổng thời gian hoàn thành nhệm vụ được sử dụng để biểu diễn hàm thích nghi. Hàm thích nghi được xác định là:

$$F(i) = \frac{1}{SFT_i}, 1 \leq i \leq S \quad (4)$$

$$SFT = \max_{1 \leq m \leq M} \left(\sum_{n=1}^K VM(m, n) \right) \quad (5)$$

Trong công thức (4), SFT_i biểu thị thời gian cần thiết để hoàn thành việc lập kế hoạch nhệm vụ để phân bổ nhệm vụ ở phần tử thứ i. Trong công thức (5), SFT biểu thị thời gian cần thiết để hoàn thành tất cả các nhệm vụ; $VM(m, n)$ biểu thị thời gian để tác vụ thứ n chạy trên máy ảo thứ m và K là số tác vụ được phân phối tới máy ảo này. Mỗi lần lặp lại chọn phần tử có giá trị lớn hơn và một trong những giá trị này được sử dụng làm giải pháp tối ưu toàn cục.

2.2.3. Cập nhật vận tốc và vị trí của phần tử

Trong thuật toán PSO, chỉ khi vị trí phần tử hiện tại có giá trị thích hợp tốt hơn, vị trí được ghi tốt nhất sẽ là vị trí tốt nhất được thay thế bằng vị trí hiện tại. Vị trí tốt nhất mà phần tử thứ i đã trải qua ký hiệu là $pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{in})$. Trong toàn bộ nhóm phần tử, vị trí tốt nhất mà tất cả các phần tử đã trải qua được ghi lại là $gbest_i = (gbest_{i1}, gbest_{i2}, \dots, gbest_{in})$ với n biểu diễn cho vị trí tốt nhất của phần tử

đã trải qua, trong phạm vi của tổng số nhiệm vụ ($1 \leq n \leq T$). Vì mỗi lần lặp, giá trị của hàm thích nghi của phần tử có thể được tính bằng công thức (4) và (5). Các giá trị của hàm thích nghi hiện tại của phần tử được biểu thị dưới dạng $f(p_i(t))$ và trong lần lặp tiếp theo là $f(p_i(t+1))$.

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & \text{khi } f(p_i(t+1)) \leq f(pbest_i(t)) \\ p_i(t+1) & \text{khi } f(p_i(t+1)) > f(pbest_i(t)) \end{cases}$$

$$f(\max(pbest(t)) = \text{getMax}(f(pbest_1(t)), f(pbest_2(t)), \dots, f(pbest_s(t))) \quad (7)$$

$$gbest_i(t) = \begin{cases} \max(pbest(t)) & \text{khi } f(\max(pbest(t))) > f(gbest) \\ gbest(t) & \text{khi } f(\max(pbest(t))) \leq f(gbest) \end{cases}$$

(8)

Trong thuật toán của bài báo này, trong mỗi lần lặp, nếu vị trí hiện tại của phần tử có giá trị thích nghi tốt hơn vị trí cuối cùng, vị trí sẽ được cập nhật. Khi giá trị thích nghi của phần tử lớn hơn $1/G_{ct}$ tương ứng với sơ đồ lập kế hoạch G_{ov} (được tính toán bằng thuật toán tham lam) vị trí của nó sẽ đã được cập nhật. Khi đáp ứng được các điều kiện trên thì vận tốc và vị trí của phần tử sẽ được cập nhật.

$$v_i(t+1) = \omega \times v_i(t) + c_1 \times \text{Rand}() \times (pbest_i(t) - p_i(t)) + c_2 \times \text{Rand}() \times (gbest(t) - p_i(t)) \quad (9)$$

$$p_i(t+1) = p_i(t) + v_i(t) \quad (10)$$

Trong đó, t đại diện cho số lượng lặp đi lặp lại; ω là trọng lượng quán tính; c_1 và c_2 là các thừa số và nói chung là $c_1 = c_2 = 2$. $\text{Rand}()$ là giá trị ngẫu nhiên trong $[0, 1]$. Trong quá trình lặp lại, vị trí của phần tử được giới hạn trong một phạm vi cụ thể ($1 \leq p_i(t) \leq M$), đồng thời $pbest$ và $gbest$ cũng được cập nhật tương ứng và cuối cùng $gbest$ là đầu ra như là giải pháp tối ưu toàn cục.

2.2.4. Quy trình thực hiện thuật toán G&PSO

Thuật toán G&PSO được thực hiện theo các bước cụ thể như sau:

Bước 1: Khởi tạo nhóm phần tử. Các vị trí và vận tốc của các phần tử được khởi tạo lần đầu tiên, và sử dụng thuật toán tham lam để có giải pháp ban đầu G_{ov} và tổng thời gian hoàn thành nhiệm vụ dự kiến G_{ct} ; thì vị trí tốt nhất mà các phần tử trải qua với G_{ov} được khởi đầu.

Thủ tục tham lam: Thủ tục bắt đầu từ chỉ mục hàng 0 của ma trận ETC; nó phân bổ nhiệm vụ cho máy ảo từ cột cuối cùng của mỗi hàng trong ma trận ETC. Nếu sự lựa chọn được đưa ra tốt hơn những máy khác thì nhiệm vụ được hoàn thành; nếu không nhiệm vụ được gán cho máy ảo để thực hiện kết quả hiện tại tối ưu. Hơn nữa, nếu có nhiều kế hoạch phân

bổ có sẵn thì nhiệm vụ được giao đến máy ảo có ít nhiệm vụ nhất, do đó đạt được cân bằng tải.

Bước 2: Tính giá trị hàm thích nghi của từng phần tử theo công thức (4) và (5).

Bước 3: Cập nhật tối ưu. Cập nhật cá nhân và nhóm tối ưu dựa trên công thức (6)–(8):

+ So sánh giá trị hàm thích nghi của phần tử hoạt động theo $pbest$ tối ưu riêng của nó, nếu giá trị của hàm mục tiêu của phần tử tốt hơn $pbest$, thì thay thế giá trị của $pbest$ bằng vị trí hiện tại của phần tử.

+ So sánh giá trị hàm thích nghi của phần tử với nhóm tối ưu $gbest$ của nó, nếu giá trị hàm thích nghi của phần tử tốt hơn giá trị ban đầu được tính toán bằng thuật toán tham lam, sau đó đặt lại giá trị của $gbest$ với vị trí hiện tại của phần tử.

Bước 4: Cập nhật tốc độ và vị trí của phần tử tương ứng theo công thức (9) và (10).

Bước 5: Điều kiện dừng. Vòng lặp sẽ quay lại Bước 2 cho đến khi thỏa mãn điều kiện dừng.

3. Kết luận

Bài viết này giải quyết vấn đề về lập kế hoạch công việc của các máy ảo trên nền tảng điện toán đám mây bằng thuật toán G&PSO để giảm tổng thời gian hoàn thành và cân đối khối lượng công việc trong mỗi máy ảo. Thuật toán không chỉ làm giảm tổng thời gian hoàn thành nhiệm vụ mà còn cân bằng hệ thống tải và cải thiện hiệu quả toàn diện của toàn bộ nền tảng đám mây. Tuy nhiên, trong khuôn khổ bài viết, chỉ xem xét kích thước nhiệm vụ và khả năng xử lý của máy ảo khi ước tính mức độ hoàn thành nhiệm vụ theo thời gian. Trong các ứng dụng thực tế, cần được xem xét nhiều yếu tố hơn, chẳng hạn như ảnh hưởng của băng thông và quá trình truyền dữ liệu.

Tài liệu tham khảo

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al., *Above the clouds: A Berkeley view of cloud computing*, Technical Report No. UCB/EECS-2009-28, University of California at Berkeley, USA, 2009.

[2] J. E. Smith and R. Nair, *The architecture of virtual machines*, Computer, vol. 38, no. 5, pp. 32–38, 2005.

[3] R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. M. Martins, A. V. Anderson, S. M. Bennett, A. Kagi, F.H. Leung, and L. Smith, *Intel virtualization technology*, Computer, vol. 38, no. 5, pp. 48–56, 2005.