

GIẢI PHÁP KỸ THUẬT TRÊN CƠ SỞ LOẠI TRỪ TƯƠNG HỒ HỆ THỐNG MÁY CHỦ ẢO

ĐOÀN VĂN THẮNG¹, NGUYỄN HÀ HUY CƯỜNG², VÕ QUANG HOÀNG KHANG¹

¹ Đại học Công nghiệp TP Hồ chí minh;;

² Trường Đại học Quảng Nam Trường;

vanthangdn@gmail.com - nguyenhahuycuong@gmail.com

Tóm tắt. Hệ thống máy chủ ảo được xây dựng trên nền của hệ thống các máy chủ vật lý phân tán trên bình diện toàn cầu và kết nối với nhau thông qua hệ thống viễn thông có vai trò ngày càng quan trọng trong việc vận hành, cung cấp và khai thác tài nguyên ảo hóa. Để đảm bảo việc cung cấp tài nguyên này một cách chính xác, kịp thời, tránh tình trạng chồng chéo, xung đột, bế tắc và thiếu thốn vô hạn cho các tiến trình có nhu cầu tài nguyên, đặc biệt tài nguyên ở xa, nhất thiết, phải nghiên cứu và đề xuất các giải pháp đủ mạnh, tin cậy làm nền tảng cơ sở cho hệ điều khiển bên trong các đám mây. Trong bài báo này, chúng tôi nghiên cứu thuật toán loại trừ lẫn nhau theo hướng tiếp cận Token nhằm tìm ra những giải pháp cung cấp tài nguyên phân tán hữu hiệu trong đó nhấn mạnh các giải pháp loại trừ lẫn nhau và đề ra các biện pháp tránh được các vấn đề thiếu thốn tài nguyên.

Từ khóa. Máy chủ ảo; cung cấp tài nguyên.

A NEW TECHNICAL ON BASIS MUTUAL EXCLUSION FOR VIRTUALIZATION SYSTEMS

Abstract. The virtualization server system is built on the basis of a distributed physical machine system on a global scale and interconnected through a telecommunication system that plays an increasingly important role in the operation and provisioning and exploits virtualization resources. To ensure the timely and accurate resource allocation of this resource, avoid overlapping, conflicting, deadlock and infinite depletion of processes that require resources, particularly remote resources, Necessarily, it is important to research and propose robust, reliable solutions that underpin the control system within the cloud. In this paper, we study the mutually exclusive algorithm in the Token approach to finding effective distributed resource allocation solutions that emphasize mutually exclusive solutions and measures. Avoid the problem of not enough resource hardware from a physical machine.

Keywords. Virtualization Server; Mutual Exclusion, Physical Machine, Resource Allocation.

1 MỞ ĐẦU

Điện toán Đám mây (ĐTĐM) ngày nay dựa trên sự thay đổi từ mô hình máy tính lớn đến mô hình máy khách chủ. Các chi tiết về cơ sở hạ tầng được trừu tượng hóa từ phía người sử dụng, họ không cần biết về hạ tầng công nghệ và nguồn tài nguyên được truy cập một cách dễ dàng trong các đám mây. Hầu hết các cơ sở hạ tầng ĐTĐM bao gồm dịch vụ cung cấp thông qua trung tâm dữ liệu và xây dựng trên các máy chủ ảo. Với sự xuất hiện của nền công nghiệp 4.0 nhu cầu sử dụng nguồn tài nguyên ảo hóa này dự báo sẽ tăng lên trong thời gian tới đây. Vấn đề là nguồn tài nguyên ĐTĐM thường xuất hiện như là điểm truy cập duy nhất của tất cả các máy chủ ĐTĐM. Thế nhưng, ở thời điểm hiện tại mạng vẫn giữ được vai trò truyền thống của nó như là một phương tiện trao đổi thông tin và đồng thời, nó cũng được cảm nhận như là một phương tiện chia sẻ tài nguyên. Xu hướng hiện nay cho thấy cần phải xây dựng được các hạ tầng cung cấp dịch vụ phải có tính năng vượt trội linh hoạt ở khả năng mở rộng, khả năng phục hồi an ninh và tắc nghẽn mạng, cùng với đó là khả năng về tương tranh xung đột tài nguyên hữu hạn. Công nghệ ảo hóa cung cấp sự trừu tượng và cô lập các chức năng cấp thấp hơn, cho phép khả năng di động cao hơn và tập

hợp được các nguồn tài nguyên vật lý [2]. Trong một môi trường như ĐTĐM các nguồn tài nguyên phục vụ cho các nhu cầu của khách hàng ngày một tăng. Hiện tại các khách hàng đang chia sẻ các nguồn tài nguyên hữu hạn, chẳng hạn các hệ đa server lồng vào nhau, hệ điều hành và các loại client khác, mỗi phần có cách thức hoạt động cụ thể, xác định thứ tự mà trong đó các nguồn tài nguyên được giao cho từng phần. Tuy nhiên, tiến trình hoạt động phức tạp trong môi trường phân tán hỗn tạp khách hàng phải cạnh tranh cho một tập hợp hữu hạn của nguồn tài nguyên.

Từ những vấn đề này đã thúc đẩy các nhà nghiên cứu, các chuyên gia trong lĩnh vực khoa học máy tính luôn nghiên cứu tìm tòi các giải pháp nhằm đáp ứng tốt hơn khả năng yêu cầu sử dụng dịch vụ công nghệ thông tin của nhân loại ngày nay. Trong nội dung bài báo, chúng tôi trình bày giải pháp với mô hình máy chủ ảo đáp ứng nhu cầu cung cấp tài nguyên thông tin, ngăn chặn được những vấn đề tranh chấp trong cung cấp tài nguyên. Vấn đề tranh chấp trong cung cấp tài nguyên trên nền tảng phân tán luôn được sự quan tâm giải quyết của các nhà nghiên cứu đi trước. Tuy nhiên, vẫn còn nhiều vấn đề nghiên cứu giải quyết cho thích hợp xu thế tương lai.

Trong nội dung nghiên cứu này, chúng tôi cung cấp giải pháp kỹ thuật mới trong cung cấp tài nguyên cho hệ thống máy chủ ảo dựa trên máy chủ vật lý. Giải pháp dựa trên kỹ thuật loại trừ tương hỗ của tác giả **Ricart – Agrawala**. Chúng tôi phát triển từ ý tưởng thuật toán của tác giả, từ đây đề xuất thuật toán mới gồm ba pha riêng biệt.

Bố cục bài báo gồm các phần như sau: phần 2: Trình bày một số thuật toán học nữa sát và phân cụm dữ liệu; Phần 3: Ứng dụng một số thuật toán phân loại đối tượng vay tín chấp, cuối cùng là kết luận và hướng phát triển.

2 CƠ SỞ LÝ THUYẾT

2.1 Petri Nets

Petri lưới (PN) đã được giới thiệu vào năm 1962 bởi các luận án tiến sĩ của Carl Adams Petri [16], tại Đại học Kỹ thuật Darmstadt, Đức. Lý thuyết ban đầu được phát triển như một cách tiếp cận để mô hình và phân tích hệ thống thông tin liên lạc. Petri Nets (PNs) [14] là một công cụ mô hình đồ họa và toán học có thể được áp dụng trong một số loại hệ thống và cho phép các mô hình song song, đồng thời, không đồng bộ và hệ thống không xác định. Kể từ khi mô hình này ra đời, nhiều nhóm chuyên môn nghiên cứu và đã tiếp tục cải tiến phần mở rộng được đề xuất cho phép giới thiệu ngắn gọn hơn và để trình bày cho các hệ thống tính toán lớn. Như vậy, mạng Petri đơn giản sau đó đã được áp dụng và mở rộng các hướng khác nhau, stochastic, high-level, object-oriented and coloured nets là một vài ví dụ về các phần mở rộng được đề xuất.

2.2 Place-transition nets

Pertri Net còn được gọi là Place/Transition Networks (mạng vị trí /chuyển tiếp) được hiển thị bằng đồ thị có hướng gồm 2 node. Node Transition (node chuyển tiếp) có dạng hình chữ nhật hoặc hình vuông được dùng để biểu diễn các sự kiện rời rạc có thể xảy ra. Node Place (node vị trí) có dạng hình tròn dùng để biểu diễn trạng thái các điều kiện. Pertri là một trong những lớp nổi bật và nghiên cứu tốt nhất của Petri lưới, và đôi khi nó được gọi là mạng Petri (PN). Place/Transition mạng Petri là một đồ thị có hướng, thường được định nghĩa như sau:

Định nghĩa 1. Petri net [14] là một 5-tuple:

$$PN = (P, T, F, W, M_0)$$

Trong đó:

1. $P = \{p_1, p_2, \dots, p_m\}$ là tập hữu hạn các place.
2. $T = \{t_1, t_2, \dots, t_n\}$ là tập hữu hạn của transition.
3. $F: (P \times T)$ đường nối từ các input place tới các transition, trong đó N là một tập hợp các số nguyên (là 0 hoặc 1). Trong trường hợp tổng quát nó là các số nguyên không âm. N là hàm xác định hướng đường nối từ các transition tới các output place.
4. $W: (P \times T)$

N trạng thái (marking) ban đầu.

5. M0: P

Mạng Pertri được nghiên cứu một cách rộng rãi trên toàn thế giới, hiện nay có hơn 15 loại Pertri Net khác nhau thường được phân loại vào một trong ba nhóm sau :

- Nhóm thứ nhất : là mạng Pertri mô tả bởi các vị trí có khả năng biểu diễn đúng sai, mỗi vị trí được đánh dấu bởi một thẻ không có cấu trúc.
- Nhóm thứ hai : là mạng Pertri mô tả bởi các vị trí có khả năng biểu diễn giá trị là một số nguyên.
- Nhóm thứ ba : là mạng Pertri có khả năng biểu diễn giá trị ở mức độ cao, chúng được đánh dấu bởi các thẻ dấu có cấu trúc. Các mạng này có thể kể đến như. Mạng Pertri cao cấp với các dữ liệu trừu tượng. Mạng Pertri suy rộng. Trong mạng Pertri suy rộng bao gồm có các mạng :
 - Mạng Pertri tô màu.
 - Mạng Pertri có thời gian.
 - Mạng Pertri có gán nhãn

2.3 Mạng Pertri tô màu Coloured Petri Net (CPN)

Trước khi định nghĩa cho mạng CP – net, chúng ta cần chú thích một số ký hiệu và thuật ngữ cho (net expressions) biểu thức mạng. Các ký hiệu và thuật ngữ được dùng trong biểu thức như là tập màu colour, ban đầu marking (đánh dấu), biểu thức arc, và guards.

- Type(expr) ký hiệu của kiểu một expr cho biểu thức.
- Var(expr) ký hiệu của tập các biến trong biểu thức expr.
- Type(v) ký hiệu kiểu của biến v.
- Type(vars), vars là tập của biến, ký hiệu tập của kiểu $\{Type(v) | v \in vars\}$.
- S_{SM} ký hiệu của tập con của một tập S.
- Bool ký hiệu tập boolean, $Bool = \{true, false\}$.

Định nghĩa 2. Một Coloured Petri net [14] là một 9-tuple :

$$CPN = (\sum, P, T, A, N, C, G, E, I)$$

1. \sum là tập hữu hạn các kiểu không rỗng gọi là Colour Set.
2. P là tập hữu hạn của places
3. T là tập hữu hạn của transitions
4. A là tập hữu hạn của arcs (đường nối) : $P \cap T = P \cap A = T \cap A = \emptyset$.
5. N là chức năng của node. Nó được định nghĩa từ $A \rightarrow P \times T \cup T \times P$.
6. C là chức năng của colour. Nó được định nghĩa từ P và \sum
7. G là chức năng của guard. Nó được định nghĩa từ T công thức như sau:
 $\forall t \in T : [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \sum]$
8. E là chức năng của arc function. Nó được định nghĩa từ A tới biểu thức như sau :
 $\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \sum]$ nơi p(a) là vị trí của N(a).
9. I là chức năng của initialisation. Nó được định nghĩa từ P với biểu thức sau đây:
 $(\forall p \in P : [Type(I(p)) = C(p)_{MS}]$

2.4 Loại trừ tương hồ

Hệ thống phân tán rộng lớn [6,10] sử dụng công nghệ ảo hóa để cho phép việc tạo ra các phạm vi năng động hơn, với nguồn tài nguyên ảo có thể tính toán đáp ứng được nhu cầu của người dùng với các ứng dụng cụ thể. Tuy nhiên, với số lượng người sử dụng ngày càng tăng lên thì vấn đề tranh chấp cùng một tài nguyên gần không thể không xảy ra.

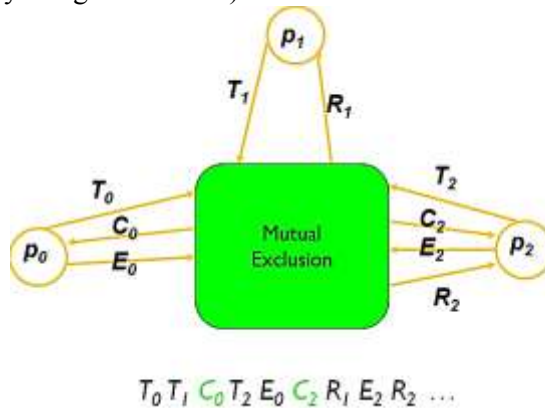
Ví dụ trong vấn đề tranh chấp lẫn nhau:

Các đầu vào là T_i và E_i ($0 \leq i \leq n-1$)

T_i là bộ xử lý thứ i muốn vào đoạn găng
 E_i là bộ xử lý thứ i muốn thoát khỏi đoạn găng
 Các đầu ra là C_i và R_i ($0 \leq i \leq n-1$)
 C_i là bộ xử lý thứ i có thể vào đoạn găng
 R_i là bộ xử lý thứ i có thể vào đoạn còn lại

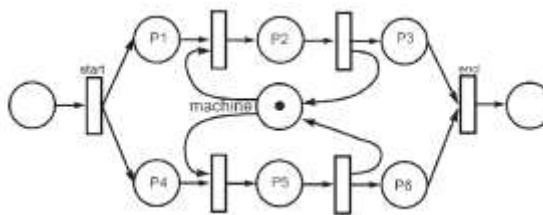
Điều kiện cho chuỗi sự kiện α các đầu vào và đầu ra:

- $\forall i$ ($0 \leq i \leq n-1$): $\alpha|i$ quay vòng qua T_i , C_i , E_i , R_i theo đúng thứ tự (điều kiện 1 - ràng buộc trên các đầu vào đảm bảo yếu tố tuần tự).
- $\forall i, j$ ($0 \leq i \neq j \leq n-1$): đầu vào hay đầu ra của j (C_j) ngay trước C_i trong α (điều kiện 2 – đảm bảo chỉ có 1 bộ xử lý trong 1 thời điểm)



Hình 1. Tiến trình cung cấp tài nguyên cho phép loại trừ lẫn nhau

Việc chia sẻ tài nguyên và / hoặc dữ liệu được phổ biến trong nhiều ứng dụng hệ thống, trong đó hầu hết các nguồn tài nguyên và dữ liệu nên được truy cập một cách độc quyền lẫn nhau. Nguồn (hoặc biến dữ liệu) có thể được mô hình hóa bởi một nơi với thể đại diện cho số lượng tài nguyên. Nơi đây được xem là điều kiện trước cho tất cả các quá trình chuyển đổi cần nguồn đó. Sau khi sử dụng một nguồn tài nguyên, nó phải được phát hành. Hình 2 mô tả một ví dụ về một hệ thống chia sẻ tài nguyên được truy cập một cách độc quyền lẫn nhau.



Hình 2. Tiến trình cung cấp tài nguyên xung đột bằng Petri net

3 GIẢI PHÁP LOẠI TRỪ TƯƠNG HỒ TRONG CUNG CẤP TÀI NGUYÊN CHO HỆ THỐNG MÁY CHỦ ẢO

3.1 Mô hình hệ thống

Hệ thống bao gồm N trạm với $N = \{ S_1, S_2, \dots, S_n \}$. Không mất tính tổng quát, chúng tôi giả định rằng một tiến trình duy nhất đang chạy trên mỗi trạm. Tiến trình tại trạm S_i được ký hiệu là p_i . Tất cả các tiến trình giao tiếp không đồng bộ trên một mạng truyền thông. Một tiến trình có nhu cầu xâm nhập vào miền găng CS yêu cầu tất cả hoặc một tập các tiến trình con bằng cách gửi thông điệp Request, và chờ đợi trả lời thích hợp trước khi vào CS. Trong khi chờ đợi các tiến trình khác không được phép thực hiện các yêu cầu tiếp tục nhập vào miền găng CS. Một trạm có thể ở một trong ba trạng thái sau đây: yêu cầu CS, thực thi trong miền găng CS, hoặc không yêu cầu và cũng không thực thi trong CS (tức là, nhàn rỗi). Trong trạng thái yêu cầu, các trạm bị chặn và không thể thực hiện yêu cầu thêm vào trong CS. Trong trạng thái

nhân rồi "idle", các trạm được thực thi bên ngoài CS. Trong các thuật toán theo hướng Token một trạm cũng có thể có thông tin về trạng thái của trạm giữ thông báo thực hiện bên ngoài CS. Tại bất kỳ thời điểm nào, một trạm cũng có thể có một vài yêu cầu cấp phát CS. Một trạm khi yêu cầu được sắp xếp từng tự và chờ phục vụ. Chúng tôi đề xuất giải pháp trên cơ sở của thuật toán Ricart–Agrawala.

Thuật toán sử dụng hai loại thông điệp REQUEST và REPLY. Một tiến trình gửi thông điệp REQUEST cho tất cả các tiến trình khác để yêu cầu cho phép được quyền xâm nhập vào miền găng. Một tiến trình gửi thông điệp REPLY đến một tiến trình để cho phép tiến trình đó. Tiến trình sử dụng đồng hồ logic Lamport để găng nhãn thời gian để yêu cầu tới miền găng (CS). Nhãn thời gian dùng để quyết định ưu tiên các yêu cầu trong trường hợp có xung đột nếu một tiến trình p_i đang chờ thực thi trong miền găng nhận một thông điệp REQUEST yêu cầu từ tiến trình p_j , sau đó nếu thứ tự ưu tiên của p_j là sau, thì p_i trì hoãn gửi REPLY và gửi thông điệp REPLY tới p_j chỉ sau khi thực hiện CS cho yêu cầu cấp phát. Ngược lại, p_i gửi thông điệp REPLY cho p_j ngay khi nó không thực thi trong CS, yêu cầu trước thành công trong việc nhận thông điệp REPLY và được thực hiện trong CS. Mỗi tiến trình p_i duy trì một dãy yêu cầu trì hoãn RD_i , kích cỡ của dãy này tương đương với số tiến trình trong hệ thống. Bắt đầu, $\forall i \forall j: RD_i[j]=0$. Khi mà tiến trình p_i trì hoãn gửi yêu cầu tới p_j , nó xác lập $RD_i[j]=1$ và sau đó nó gửi thông báo REPLY tới p_j , nó xác lập $RD_i[j]=0$

3.2 Giới thiệu về thuật toán loại trừ tương hỗ

Loại trừ tương hỗ có thể được điều khiển trên một trạm trung tâm có nhiệm vụ nhận tất cả các thông điệp và khuyến nghị giải phóng. Trạm này duy trì một hàng đợi, sắp xếp các yêu cầu theo trật tự đến và phục vụ cho từng thông điệp một trong trật tự.

Phân tán giải thuật này kéo theo việc phân tán các chức năng cung cấp mà cần phải điều khiển hàng đợi trên trạm. Do vậy, một trạm chuyên cho việc tiếp nhận các yêu cầu và các khuyến nghị giải phóng từ tất cả các trạm còn lại. Một trật tự giống trên các trạm chỉ đạt được, nếu ta áp dụng dấu trong các thông điệp bởi các đồng hồ logic truyền và đánh số các trạm. Quan hệ trật tự toàn bộ được định nghĩa. Thêm vào đó, để cho một trạm có thể ra quyết định bằng việc tham chiếu duy nhất vào hàng đợi của mình nó còn cần phải được nhận một thông điệp của từng trạm khẳng định rằng không có thông điệp nào trước các thông điệp khác mà còn đang quá cảnh trên đường. Thuật toán loại trừ tương hỗ Ricart – Agrawala có thể mô tả như sau.

Thuật toán Ricart – Agrawala

Giai đoạn: Yêu cầu trong miền găng

Bước 1. Khi một trạm S_i muốn vào trong đoạn găng CS, nó gửi gửi quảng bá một thông điệp REQUEST có gán nhãn thời gian tới tất cả các trạm khác. Thông điệp có điệp có dạng (T, H_i, i) , trong đó H_i là dấu của thông điệp có nghĩa là đồng hồ logic của nó và T có thể nhận một trong ba giá trị là REQ, REL, ACQ.

Bước 2. Khi trạm S_j nhận một thông điệp REQUEST từ trạm S_i , nó gửi một thông điệp REPLY tới trạm S_i nếu trạm S_j không yêu cầu hoặc không thực thi trong miền găng CS, hoặc nếu trạm S_j là đang yêu cầu và S_i 's yêu cầu với nhãn thời gian là nhỏ hơn trạm S_j 's. Nếu không, trả lời trì hoãn và S_j xác lập $RD_j[i]=1$.

Giai đoạn: Thực thi trong miền găng

Bước 3. Trạm S_i vào trong miền găng sau khi nó nhận được thông điệp REPLY từ trạm mà nó gửi thông điệp REQUEST.

Giai đoạn: Khôi phục trong miền găng

Bước 4. Khi trạm S_i thoát khỏi miền găng CS, nó gửi tất cả thông điệp REPLY: $\forall j$ nếu $RD_i[j]=1$, thì nó gửi thông điệp REPLY tới S_j và xác lập $RD_i[j]=0$.

Ý tưởng giải thuật

Start

1. $\langle b[i] := \text{true} \rangle;$
2. $\langle x := i \rangle;$
3. If $\langle y \neq 0 \rangle$ then

4. $\langle b[i]:=false \rangle;$
5. $await \langle y=0 \rangle$
6. $goto start;$
7. $F_i;$
8. $\langle y:=i \rangle;$
9. If $\langle x \neq i \rangle$ then
10. $\langle b[i]:=false \rangle;$
11. For $j:=1$ to N
12. Do wait $\langle \neg b[j] \rangle$ od
13. If $\langle y \neq i \rangle$ then
14. Await $\langle y=0 \rangle;$
15. Goto start;
16. $F_i;$
17. $F_i;$
18. Critical section;
19. $\langle y:=0 \rangle;$
20. $\langle b[i]:=false \rangle;$

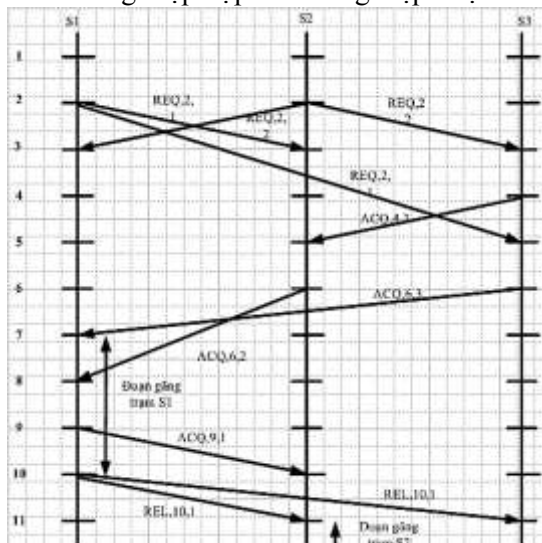
3.3 Kiểm nghiệm giải thuật

Các yêu cầu vào đoạn găng được xử lý theo trật tự FIFO và theo quan hệ \Rightarrow . Để chứng minh cho điều đó ta lưu ý rằng khi trạm S_i quyết định vào đoạn găng, nó không thể đưa và trong mạng yêu cầu REQUEST nào trước trong bối cảnh sau. Thực tế, khi trạm S_i vào đoạn găng có nghĩa là nó đã nhận thông điệp từ tất cả các trạm khác và tất cả các thông điệp đều sau REQUEST của riêng nó.

Ta kiểm tra các đặc tính sau đây:

- Trạm S_i đang ở trong đoạn găng là trạm duy nhất nằm trong đoạn găng ấy. Thực tế cho thấy thông điệp REQUEST được phát bởi S_i vẫn tiếp tục tồn tại trong tất cả các hàng đợi cho đến khi nó được thay thế bởi thông điệp REL.
- Trạm nào đã yêu cầu vào đoạn găng phải đảm bảo thời hạn và phải ra khỏi đoạn găng sau một khoản thời gian xác định.

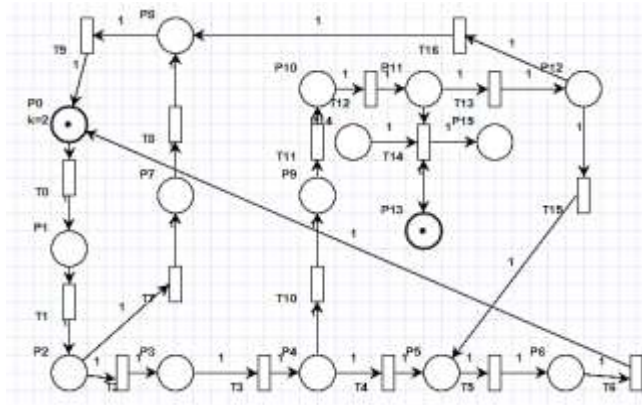
Ví dụ 1: Chúng ta xét một mạng bao gồm 3 trạm, trong đó có hai trạm 1 và 2 yêu cầu vào đoạn găng tại thời điểm 2 của đồng hồ logic của chúng. Tập hợp các thông điệp được mô tả như hình sau:



Hình 3. Loại trừ tương hỗ nhờ dấu

4 MÔ PHỎNG VÀ ĐÁNH GIÁ KẾT QUẢ

Chúng tôi sử dụng phần mềm mô phỏng Coloured Petri net 4.0.0. Được cài đặt trên máy Laptop Lenovo; Chipset Intel core™i5. Một số kết quả mô phỏng với mô hình CPN cho thuật toán loại trừ lẫn nhau với thuật Ricart – Agrawala.



Hình 4. Mô hình mô phỏng với thuật toán Ricart – Agrawala

Petri net simulation results

Place	Average number of tokens	95% confidence interval (+/-)
P0	0.13861	0.02329
P1	0.14851	0.02329
P2	0.14851	0.02329
P3	0.06931	0.01552
P4	0.06931	0.01552
P5	0.0495	0.03247
P6	0.0495	0.02852
P7	0.06931	0.03599
P8	0.08911	0.04493
P9	0.0396	0.01452

Hình 5. Một số yêu cầu mô phỏng Coloured Pertri Net

Trên hình 5 chúng tôi sử dụng 15 vị trí Place với công cụ Pertri Net chúng tôi có được kết quả yêu cầu mô phỏng như trên với tỷ lệ trung bình của các tokens tại các vị trí giao động từ 0,0396 tới 0,14851. Và độ tin cậy tại các vị trí (Place) từ 0,01452 tới 0,04493.

Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	false

Hình 6. Trạng thái của hệ thống phân tích

Yêu cầu phân tích về không gian trạng thái của mạng Pertri net như trên hình 6.

Steady State Distribution of Tangible States

Marking	Value
M0	0.16667
M1	0.16667
M2	0.08333
M3	0.08333
M4	0.08333
M5	0.10417
M6	0.04167
M7	0.04167
M8	0.0625
M9	0.04167

Hình 8. Phân bố trạng thái của hệ thống phân tích

Phân bố trạng thái của hệ thống được đánh dấu từ M0 tới M15. Với dữ liệu M0 là 0.16667 và M15 là 0.10427.

Token Probability Density

	$\mu=0$	$\mu=1$
P0	0.83333	0.16667
P1	0.83333	0.16667
P2	0.91667	0.08333
P3	0.91667	0.08333
P4	0.95833	0.04167
P5	0.9375	0.0625
P6	0.9375	0.0625
P7	0.91667	0.08333
P8	0.89583	0.10417
P9	0.95833	0.04167
P10	0.95833	0.04167
P11	0.95833	0.04167
P12	0.97917	0.02083

Hình 8. Xác suất của Token tại các vị trí

Trên hình 8 mật độ xác suất của các Token tại các vị trí (Place) với $\mu=0$ các vị trí P0,P1 có giá trị thấp nhất là 0.83333 mật độ xác suất cao nhất tại P15, P12 có giá trị là 0,97917. Khi giá trị $\mu=1$ mật độ xác suất tại vị trí P5,P6 là thấp nhất có giá trị là 0.0625 còn giá trị cao nhất tại vị trí P8,P13 là 0.10417

5 KẾT LUẬN

Qua kết quả kết quả mô phỏng và phân tích đánh giá ưu nhược điểm của thuật toán đề xuất này chúng ta nhận thấy rằng thuật toán vừa nêu có những ưu điểm hơn so với thuật toán đã đề xuất Lamport trước đây. Với thuật toán Lamport và thuật toán Ricart – Agrawala với yêu cầu (N-1) thông điệp REQUEST và RELEASE và độ trễ của thuật toán là T. Tuy nhiên đối với thuật toán Ricart – Agrawala là 2(N-1) thông điệp cho mỗi CS gọi còn Lamport là 3(N-1).

Như vậy, quá trình mô hình hóa hệ thống phân tán máy chủ ảo giải ứng dụng thuật toán giải quyết vấn đề loại trừ tương hồ với bằng Petri Net chúng tôi nhận thấy kết quả mô phỏng phù hợp với yêu cầu đề ra. Dựa trên kết quả này chúng ta có thể mở rộng cho các hệ thống phức tạp hơn. Và giải quyết được các vấn đề trong cung cấp tài nguyên phân tán.

Hiện nay có nhiều công cụ để chuyển mô hình hệ thống Petri Net sang chương trình với các ngôn ngữ lập trình thông dụng như Java điều này giúp ta dễ dàng phát triển hệ thống sau khi mô hình hóa với Petri Net

TÀI LIỆU THAM KHẢO

- [1] D. P. Mitchell and M. J. Merritt, “A distributed algorithm for deadlock detection and resolution,” in Proc.ACM Symposium on Principles of Distributed Computing, 1984, pp. 282–284.
- [2] E.Knapp. (1987), Deadlock Detection in Distributed Database Systems, ACM Computing Surveys, Vol.19, No. 4 pp.303-327.
- [3] M,Singhal.(1989), Deadlock detection in distributed systems. IEEE Computer, Vol.22, pp. 37–48.
- [4] A.D.Kshemkalyani, and M.Singhal. (1999), A One-Phase Algorithm to Detect Distributed Deadlocks in Replicated Databases, IEEE Trans. Knowledge and Data Eng., vol. 11, No. 6, pp. 880-895.
- [5] W. Voorsluys, S. Garg, and R. Buyya, “Provisioning spot market cloud resources to create cost-effective virtual clusters,” in Proceedings of the 11th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP). Los Alamitos, CA, USA: IEEE Comput. Soc., 2011.
- [6] R. Buyya, R. Ranjan, and R. Calheiros, “Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities,” in Proceeding of the 7th International Conference on High Performance Computing & Simulation (HPCS). Los Alamitos, CA, USA: IEEE Comput. Soc., 2009, pp. 1–11.
- [7] D. Feitelson, “Parallel workloads archive,” <http://www.cs.huji.ac.il/labs/parallel/workload>.
- [8] D. Tsafir, Y. Etsion, and D. G. Feitelson, “Modeling User Runtime Estimates,” in In Processing of the 11th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP) . Springer-Verlag, 2005, pp.1–35.
- [9] Binildas CA, Malhar Barai, Vincenzo Caselli, Service Oriented Architecture with Java (Using SOA and Web Services to build powerfull Java applications), PACKT Publishing, 2008.
- [10] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, Cloud Computing and Grid Computing 360-Degree Compared, Grid Computing Environments Workshop, 2008.
- [11] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal, Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, International Conference on High Performance Computing, 2008.
- [12] M. Andreolini, S. Casolari, M. Colajani, and M. Messori, “Dynamic load management of virtual machines in a cloud architectures,” in CLOUDCOMP, 2009.
- [13] D. Prangchumpol, S. Sanguansintukul, and P. Tantasanaw, “Analyzing User Behavior from Server Logs for Improved Virtualization Management”,2009.

- [14] M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova, “Resource allocation algorithms for virtualized service hosting platforms,” JPDC, vol. 70, no.9, pp. 962-974, 2010.
- [15] A. Kshemkalyani, M. Singhak, “Deadlock Detection in Distributed systems”, in Distributed Algorithms, 2010 – 2011.

Ngày nhận bài: 24/10/2018

Ngày chấp nhận đăng: 10/02/2019