

MỘT MÔ HÌNH MẠNG HÀNG ĐỢI MỞ CHO ỨNG DỤNG ĐA TẦNG TRÊN HỆ THỐNG TÍNH TOÁN Đám Mây

Huỳnh Xuân Chung, Đặng Thanh Chương

Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế

Email: haichung1964@gmail.com, dtchuong@husc.edu.vn

Ngày nhận bài: 19/9/2020; ngày hoàn thành phản biện: 22/9/2020; ngày duyệt đăng: 23/9/2020

TÓM TẮT

Các nền tảng tính toán đám mây được mô hình hóa với một mạng hàng đợi Jackson mở có thể được sử dụng để xác định và đo lường QoS đảm bảo đám mây có thể cung cấp về thời gian phản hồi. Việc phân tích, đánh giá có thể được thực hiện theo các thông số khác nhau, như tốc độ đến của khách hàng yêu cầu dịch vụ và tốc độ phục vụ của các máy chủ xử lý, cùng một số thông số khác. Trong bài báo này, chúng tôi xem xét mô hình mạng hàng đợi mở cho ứng dụng đa tầng trên nền tảng hệ thống tính toán đám mây gồm có bốn tầng với 6 nút mạng, trong đó cấu trúc các nút có thể khác nhau. Theo đó, mô hình hệ thống phân bố như một hệ thống hàng đợi gồm nhiều hàng đợi M/M/1. Việc phân tích và đánh giá mô hình được thực hiện thông qua tính toán một số thông số đo hiệu năng hệ thống, như thời gian đáp ứng trung bình, số yêu cầu trung bình, Một số kết quả số được chỉ ra cũng như mô tả qua đồ thị theo các tham số vào quan trọng như tốc độ đến, tốc độ phục vụ đã cho thấy tính đúng của mô hình phân tích.

Từ khóa: Tính toán đám mây, Mạng hàng đợi mở Jackson.

1. GIỚI THIỆU

Tính toán đám mây (Cloud computing) là sự phát triển của tính toán phân tán, tính toán song song và tính toán lưới. Theo Rajkumar Buyya [10]: “Tính toán đám mây là một loại hệ thống phân bố và xử lý song song gồm các máy tính ảo kết nối với nhau và được cung cấp động cho người dùng như một hoặc nhiều tài nguyên đồng nhất dựa trên sự thỏa thuận dịch vụ giữa nhà cung cấp và người sử dụng”. Trong hệ thống tính toán đám mây, khái niệm cloud tượng trưng cho 1 tổ hợp các tài nguyên có thể cấu hình như mạng, máy chủ, bộ nhớ, ứng dụng và dịch vụ, cung cấp quyền truy cập các tài nguyên này theo yêu cầu. Kiến trúc tính toán đám mây [1], đại bộ phận hạ tầng cơ sở của tính toán đám mây hiện nay là sự kết hợp của những dịch vụ đáng tin cậy được phân phối thông qua các trung tâm dữ liệu (data center) được xây dựng trên những

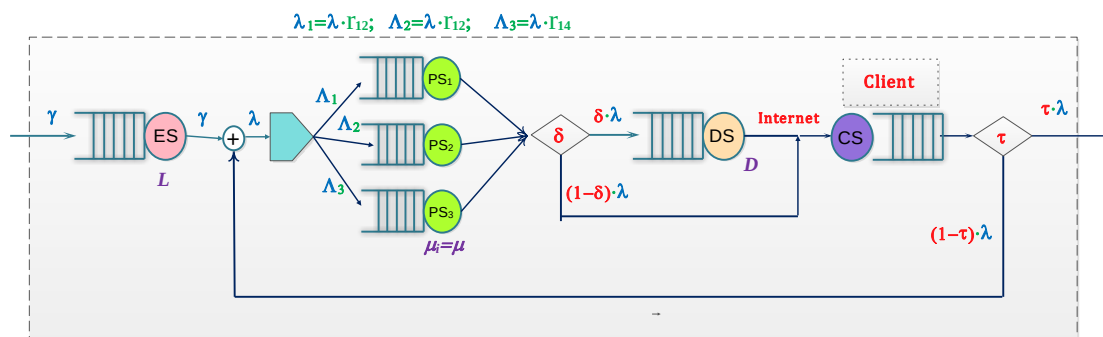
máy chủ với những cấp độ khác nhau của các công nghệ ảo hóa. Những dịch vụ này có thể được truy cập từ bất kỳ đâu trên thế giới, trong đó đám mây là một điểm truy cập duy nhất cho tất cả các máy tính có nhu cầu của khách hàng. Các dịch vụ thương mại cần đáp ứng yêu cầu chất lượng dịch vụ từ phía khách hàng và thông thường đều đưa ra các mức thỏa thuận dịch vụ (Service level agreement). Dựa vào mô hình cung ứng dịch vụ mà đám mây được chia làm 3 tầng: tầng cơ sở hạ tầng (Infrastructure), tầng nền tảng (Platform) và tầng phần mềm (Software).

Tính toán đám mây đã phát triển như là cách tốt nhất để chia sẻ và cung cấp tài nguyên qua internet. Khả năng để cung cấp chất lượng dịch vụ (QoS) được đảm bảo là rất quan trọng đối với thành công thương mại của các nền tảng đám mây. Quá trình truy nhập vào đám mây thường có thể tổ chức ở mô hình hàng đợi, vì vậy mỗi người dùng cần đợi cho đến khi người dùng hiện tại được phục vụ xong. Lý thuyết xếp hàng là nghiên cứu toán học về các hàng đợi và đặc tính vận hành khác của hàng đợi. Trong lý thuyết hàng đợi, một mô hình được xây dựng sao cho có thể dự đoán độ dài hàng đợi và thời gian chờ. Các nghiên cứu liên quan đều xem lý thuyết hàng đợi là nghiên cứu về các hệ thống xếp hàng, nơi một số khách hàng yêu cầu một số dịch vụ từ một số máy chủ. Lý thuyết hàng đợi rất hữu ích trong máy tính và hệ thống truyền thông để xác định thông lượng, thời gian đáp ứng, thời gian sử dụng tài nguyên, xác suất cuộc gọi bị mất và yêu cầu tài nguyên. Trên cơ sở này, bài báo tìm hiểu về mô hình hàng đợi cho tính toán đám mây và đánh giá hiệu suất mô hình hàng đợi được sử dụng thông qua thông số thời gian đáp ứng của hệ thống.

Nhiều nghiên cứu tương tự liên quan đến mô hình này đã được thực hiện và đề xuất trong một số bài báo [1-9]. Các tác giả trong [1][5] trình bày một mô hình phân tích dựa trên lý thuyết hàng đợi để nghiên cứu ảnh hưởng của QoS trong tính toán đám mây. Theo [1][5], các nền tảng đám mây được mô hình hóa với một mạng Jackson mở có thể được sử dụng để xác định và đo lường QoS đảm bảo đám mây có thể cung cấp về thời gian phản hồi. Việc phân tích, đánh giá có thể được thực hiện theo các thông số khác nhau, như tốc độ đến của khách hàng yêu cầu dịch vụ và tốc độ phục vụ của các máy chủ xử lý, cùng một số thông số khác. Mô hình nghiên cứu trong [2], sẽ xem xét các đơn vị xử lý (máy chủ) và đơn vị lưu trữ (bộ đệm) được tách biệt, đồng thời đưa ra một cơ chế mà thời gian tìm kiếm được khởi động ngay sau một vài khoảng thời gian rỗi của máy chủ. Việc áp dụng mô hình hàng đợi cho hệ thống tính toán đám mây để ứng dụng vào bài toán thực tiễn cũng đã được trình bày trong [3]. Theo đó, các tác giả trong [3] đã cung cấp một giải pháp y tế tính tử được thành lập trên các hệ thống thông tin chăm sóc sức khỏe kết hợp với tính toán đám mây nhằm nhấn mạnh vào xử lý thông tin y tế. Mô hình đề xuất ở đây cũng phân tích theo thời gian chờ được xác định như là tiêu chí chất lượng dịch vụ (QoS). Mô hình dịch vụ đám mây đã được mô hình hóa bằng lý thuyết hàng đợi theo 2 dạng hai hàng đợi, đó là mô hình hàng đợi Tandem $M/M/s$ và mô hình hàng đợi retrial với 1 máy chủ đơn (SSRQ).

Nội dung tiếp theo của bài báo theo đó có cấu trúc như sau: Phần II sẽ là giới thiệu mô hình phân tích. Kết quả đánh giá mô hình được thực hiện theo một số thông số độ đo, bao gồm cả việc tính toán các xác suất trạng thái ổn định của mô hình được trình bày trong phần III. Cuối cùng là phần Kết luận và hướng phát triển.

Mô hình phân tích ở đây ứng với hệ thống tính toán đám mây dạng đa server, đa tầng (Hình 1), được mô hình hóa ở dạng mô hình hàng đợi, cụ thể là với mô hình mạng hàng đợi Jackson mở.



Theo đó, chúng tôi sử dụng mô hình như trong Hình 1 xem xét mô hình mạng hàng đợi mở cho ứng dụng đa tầng gồm có bốn tầng với 6 nút mạng để mô hình cho ứng dụng đa tầng với cấu trúc các nút có thể khác nhau. Mô hình hệ thống phân bố như một hệ thống hàng đợi gồm nhiều hàng đợi $M/M/1$ (mỗi nút được mô hình hóa là một hàng đợi $M/M/1$). Tại bộ cân bằng tải (nút ES) sẽ nhận các yêu cầu đến từ người dùng bên ngoài và sau đó xử lý, điều phối yêu cầu đến các nút PS_i trong tầng xử lý nghiệp vụ dựa trên cấu hình của chúng. Trong tầng cơ sở dữ liệu DS , xác suất δ là xác suất một yêu cầu từ nút PS_i trong tầng xử lý nghiệp vụ đến truy cập vào tầng lưu trữ cơ sở dữ liệu. Khi thực hiện mô hình hóa một hệ thống dựa trên web, không phải tất cả

yêu cầu sẽ yêu cầu truy cập vào máy chủ cơ sở dữ liệu [5]. Tuy nhiên, lưu ý rằng xác suất này thường sẽ tương đối cao. Tại tầng cuối cùng (tầng CS), khách hàng có thể rời khỏi hệ thống với xác suất τ hoặc quay lại hệ thống để yêu cầu phục vụ tiếp với xác suất $(1 - \tau)$. Khi đó, ta có $\lambda = \gamma/\tau$ [1].

1. Mô tả mô hình

Mô hình trên Hình 1 được mô tả qua các thành phần sau:

- *ES* (Entering Server): là một server vào, đóng vai trò như là bộ cân bằng tải (load balancer), có nhiệm vụ chuyển tiếp các yêu cầu phục vụ từ các user đến một trong các PS_i , ($i = 1 \dots 3$), sao cho đảm bảo hệ thống hoạt động không quá tải. Bộ cân bằng tải *ES* có thể được mô hình hóa như là mô hình hàng đợi $M/M/1$, với tốc độ đến và tốc độ phục vụ theo phân phối mũ biểu diễn qua các tham số γ và L một cách tương ứng (trong đó $\gamma < L$). Mục đích của *ES* là xác định nút PS_i để phân công phục vụ. Để làm được điều này, *ES* sẽ sử dụng giải thuật phân phối các yêu cầu phụ thuộc vào khối lượng công việc trung bình (averaged workload) tại mỗi nút PS_i [1].
- PS_i (Processing Server): *PS* đóng vai trò là các server xử lý, trong đó mỗi nút PS_i có thể là một vi xử lý trung tâm hoặc 1 lõi vi xử lý ứng với các tài nguyên về mặt vật lý trong mô hình kiến trúc đám mây, mà ở đó các dịch vụ sẽ được tính toán, thực hiện. Các nút PS_i sẽ được lựa chọn để thực hiện tất cả các yêu cầu phục vụ của các user. Trong mô hình này, giả thiết các PS_i là như nhau và mỗi PS_i có tốc độ phục vụ bằng nhau, và bằng μ , do đó, $\mu = \mu_i$, $i = 1 \dots 3$.
- *DS* (Database Server): mỗi PS_i sẽ truy nhập *DS* với xác suất δ . *DS* đại diện cho một máy chủ cơ sở dữ liệu và phục vụ để mô hình hóa quyền truy cập vào các file, thư mục và cơ sở dữ liệu hoặc bất kỳ loại truy cập vào/ra (I/O) nào vào bộ nhớ thứ cấp trong suốt quá trình dịch vụ đối với kiến trúc đám mây. Ngoài ra, để tránh sự không thống nhất (inconsistencies) dữ liệu, mô hình ở đây chỉ lựa chọn 1 máy chủ cơ sở dữ liệu để phân tích [1]. Theo đó, *DS* có thể được mô hình hóa như mô hình hàng đợi $M/M/1$ với tốc độ đến và tốc độ phục vụ cũng tuân theo phân phối mũ có giá trị lần lượt là $\delta\lambda$ (theo quy luật của mô hình hàng đợi Jackson mở) và D .
- *CS* (Client Server): *CS* nhận phản hồi từ mô hình đám mây và rời khỏi hệ thống. Nó cũng có thể gửi lại yêu cầu theo phân phối mũ với tham số λ đến server vào *ES* nếu công việc chưa hoàn thiện (*CS* nhận các tập tin hoặc một phần của tập tin cho đến khi yêu cầu được thỏa mãn hoàn toàn). Đặt τ là xác suất rời khỏi hệ thống mạng hàng đợi Jackson mở, ta có $\lambda = \Lambda/\tau$ [1]. Nút *CS* cũng có thể được mô hình hóa như mô hình hàng đợi $M/M/1$. Tốc độ đến

của nút theo phân phối mũ với giá trị là λ . Tốc độ phục vụ giả thiết cũng tuân theo phân phối mũ với giá trị là μ .

Yêu cầu đến bộ cân bằng tải từ bên ngoài có tốc độ đến theo phân bố Poisson với tham số γ (số lượng yêu cầu trên giây).

Một số giả thiết:

- Các yêu cầu đến từ ngoài hệ thống tính toán đám mây vào một nút i tuân theo phân phối Poisson;
- Thời gian phục vụ tại mỗi nút i trong hệ thống là độc lập và tuân theo phân bố mũ và được phục vụ theo nguyên tắc FCFS;
- Gọi r_{ij} là xác suất chuyển trạng thái sau khi yêu cầu được xử lý xong tại nút i sẽ chuyển đến nút j). Ta có, tổng các xác suất chuyển trạng thái trong mạng hàng đợi mở là 1 (ở đây r_{i0} là xác suất chuyển ra ngoài mạng từ nút i):

$$r_{i0} = 1 - \sum_{j=1, j \neq i}^N r_{ij}, i \in [1, N] \quad (1)$$

- Gọi γ_i là tốc độ đến từ bên ngoài vào nút thứ i và Λ_i là tổng tốc độ đến của nút thứ i (bao gồm cả tốc độ đến từ bên ngoài và tốc độ chuyển đến từ các nút bên trong hệ thống mạng đến nút thứ i), khi đó ta có:

$$\Lambda_i = \gamma_i + \sum_{j=1, j \neq i}^N \Lambda_j r_{ji}, i \in [1, N] \quad (2)$$

2. Mô hình trạng thái

Mô hình hàng đợi Jackson mở cho hệ thống tính toán đám mây ứng dụng 4 tầng như ở Hình 1, gồm $N = 6$ nút mạng, mỗi nút mạng được mô hình hóa là một hàng đợi $M/M/1$:

- Nút ES (Entering Server) đại diện cho tầng 1, đóng vai trò như là bộ cân bằng tải, có tốc độ yêu cầu đến từ bên ngoài vào là γ .
- Tầng 2 là tầng xử lý gồm 3 nút PS (giá trị $m = 3$): PS_1, PS_2 và PS_3 . Tốc độ yêu cầu được chuyển tiếp từ nút ES đến các PS_i trong tầng xử lý tuân theo xác suất chuyển (trạng thái) đến từng nút PS_i tương ứng là $r_{12} = 0.3, r_{13} = 0.3$ và $r_{14} = 0.4$ (trong đó như quy ước ở trên, r_{ij} là xác suất chuyển trạng thái sau khi yêu cầu được xử lý xong tại nút i sẽ chuyển đến nút j). Với mô hình ở đây có thể phân tích tách biệt các nút PS_i ở tầng 2 này ứng với các mô hình hàng đợi $M/M/1$. Tốc độ phục vụ tại mỗi nút PS_i theo phân bố mũ có tham số μ_i .

- Sau khi một yêu cầu được phục vụ xong ở tầng xử lý, có thể chuyển sang truy cập vào tầng 3, tức là tầng lưu trữ cơ sở dữ liệu DS với xác suất $r_{25} = r_{35} = r_{45} = \delta = 0.3$ hoặc là không truy xuất tầng 3 với xác suất là $r_{26} = r_{36} = r_{46} = (1 - \delta) = 0.7$.
- Sau đó kết quả được phản hồi đến khách hàng (client) qua mạng Internet và rời khỏi hệ thống với xác suất $r_{60} = \tau = 0.8$ hoặc có thể quay lại hệ thống để yêu cầu phục vụ tiếp với xác suất $r_{61} = (1 - \tau) = 0.2$.

Khi đó, ta có ma trận xác suất chuyển trạng thái sau:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} & r_{16} \\ r_{21} & r_{22} & r_{23} & r_{24} & r_{25} & r_{26} \\ r_{31} & r_{32} & r_{33} & r_{34} & r_{35} & r_{36} \\ r_{41} & r_{42} & r_{43} & r_{44} & r_{45} & r_{46} \\ r_{51} & r_{52} & r_{53} & r_{54} & r_{55} & r_{56} \\ r_{61} & r_{62} & r_{63} & r_{64} & r_{65} & r_{66} \end{pmatrix} \quad (3)$$

$$R = \begin{pmatrix} 0 & 0.3 & 0.3 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

trong đó, xác suất rời khỏi hệ thống tại mỗi nút thứ i , $i \in [1, 2, 3, 4, 5, 6]$ tương ứng là: $r_{10} = r_{20} = r_{30} = r_{40} = r_{50} = 0$, $r_{60} = 0.8$.

Tiếp theo ta cần tính xác suất ở trạng thái cân bằng của hệ thống, tức là tính $P_{n_1, n_2, \dots, n_k} = P(N_1 = n_1, N_2 = n_2, \dots, N_k = n_k)$, với N_i là số yêu cầu có tại mỗi nút. Để tính $P_{n_1, n_2, \dots, n_k}$ ta thực hiện các bước sau:

- **Bước 1:** Tính tốc độ đến của mỗi nút theo công thức (2).
- **Bước 2:** Tính xác suất trạng thái cân bằng cho mỗi nút tại thời điểm xét, sử dụng công thức dạng $P_i(n_i) = P(X_i(t) = n_i) = (1 - \rho_i) \rho_i^{n_i}$ để tính xác suất có n_i yêu cầu trong hàng đợi $M/M/1$: với $\rho_i = \Lambda_i / \mu_i$
- **Bước 3:** Tính xác suất trạng thái cân bằng cho hệ thống theo công thức:

$$p_{\bar{n}} = p_{n_1, n_2, \dots, n_k} = (1 - \rho_1) \rho_1^{n_1} (1 - \rho_2) \rho_2^{n_2} \dots (1 - \rho_k) \rho_k^{n_k} = \prod_{i=1}^k (1 - \rho_i) \rho_i^{n_i}$$

Ngoài ra, các thông số độ đo liên quan khác cũng cần được tính đến, như:

- Tính số lượng yêu cầu trung bình trong mỗi nút theo công thức $E[N_i] = \frac{\rho_i}{1 - \rho_i}$
- Tính thời gian đáp ứng trung bình tại mỗi nút theo công thức $T_i = \frac{E[N_i]}{\Lambda_i}$

- Tính thời gian đáp ứng trung bình cho toàn hệ thống theo công thức:

$$T = \frac{E[N]}{\gamma} = \frac{1}{\gamma} \sum_{i=1}^6 E[N_i]$$

3. PHÂN TÍCH KẾT QUẢ

Dựa trên phân tích ở trên, phần này mô tả kết quả dựa trên các tham số đầu vào cho trước, như sau:

- Tốc độ đến từ bên ngoài vào bộ cân bằng tải: $\gamma = 20$ yêu cầu/giây; do đó $\lambda = \gamma/\tau = 25$ yêu cầu/giây
- Thời gian phục vụ tại các nút (giả thiết là như nhau với các nút PS_2, PS_3 và PS_4): $1/L = 0.03$ giây, $1/\mu = 0.03$ giây, $1/D = 0.04$ giây, .
- Giả sử phải tính xác suất trạng thái ổn định của trạng thái sau:

$$P(n_1, n_2, n_3, n_4, n_5, n_6) = P(n_1 = 6, n_2 = 2, n_3 = 3, n_4 = 1, n_5 = 1, n_6 = 1)$$

Tức là tính xác suất ở trạng thái ổn định tại thời điểm ở bộ cân bằng tải (PS_1) có 6 yêu cầu, máy chủ xử lý 1 (PS_2) có 2 yêu cầu, máy chủ xử lý 2 (PS_3) có 3 yêu cầu, máy chủ xử lý 3 (PS_4) có 1 yêu cầu, máy chủ cơ sở dữ liệu (PS_5) có 1 yêu cầu đang xử lý và 1 yêu cầu được gởi ra nút Client. Để tính $P(n_1 = 6, n_2 = 2, n_3 = 3, n_4 = 1, n_5 = 1, n_6 = 1)$, thực hiện theo các bước như đã trình bày ở trên:

- Bước 1: Tính tốc độ đến của mỗi nút:

$$\Lambda_1 = \lambda = 25, \Lambda_2 = \gamma_1 = r_{12} \cdot \lambda = 7.5, \Lambda_3 = \gamma_2 = r_{13} \cdot \lambda = 7.5, \Lambda_4 = \gamma_3 = r_{14} \cdot \lambda = 10, \\ \Lambda_5 = \gamma_1 \cdot r_{25} + \gamma_2 \cdot r_{35} + \gamma_3 \cdot r_{45} = 7.5, \Lambda_6 = \lambda = 25.$$

- Bước 2: Tính xác suất trạng thái cân bằng cho mỗi nút tại thời điểm xét:
 - o Tính mức sử dụng tài nguyên trung bình cho mỗi nút $\rho_i = \Lambda_i/\mu_i$ để biết nhu cầu phục vụ tại mỗi nút:

$$\rho_1 = \Lambda_1/L, \rho_2 = \Lambda_2/\mu, \rho_3 = \Lambda_3/\mu, \rho_4 = \Lambda_4/\mu, \rho_5 = \Lambda_5/D; \rho_6 = \Lambda_6/\mu$$

- o Tính xác suất trạng thái cân bằng cho mỗi nút (xác suất có n_i yêu cầu trong hàng đợi $M/M/1$):

$$P_1(n_1) = 0.044494629; P_2(n_2) = 0.019125; P_3(n_3) = 0.002869; P_4(n_4) = 0.16; \\ P_5(n_5) = 0.21; P_6(n_6) = 0.25.$$

- Bước 3: Tính xác suất trạng thái cân bằng cho hệ thống:

$$P(n_1, n_2, n_3, n_4, n_5, n_6) = 2.0506E - 08$$

Tính toán các thông số độ đo liên quan:

Một mô hình mạng hàng đợi mở cho ứng dụng đa tầng trên hệ thống tính toán đám mây

- Tính số lượng yêu cầu trung bình trong mỗi nút:

$$E[N_1] = \frac{\rho_1}{1-\rho_1} = 3; E[N_2] = 0.176471; E[N_3] = 0.176471; E[N_4] = 0.25; E[N_5] = 0.428571; E[N_6] = 1$$

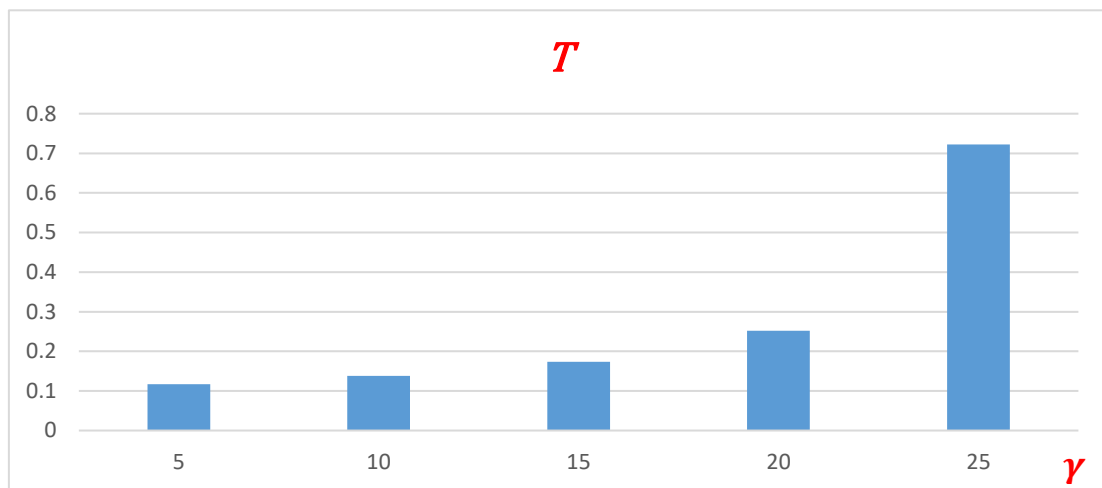
- Tính thời gian đáp ứng trung bình tại mỗi nút:

$$T_1 = \frac{E[N_1]}{\Lambda_1} = 0.12; T_2 = 0.023529; T_3 = 0.023529; T_4 = 0.025; T_5 = 0.057143; T_6 = 0.04.$$

- Tính thời gian đáp ứng trung bình cho toàn hệ thống:

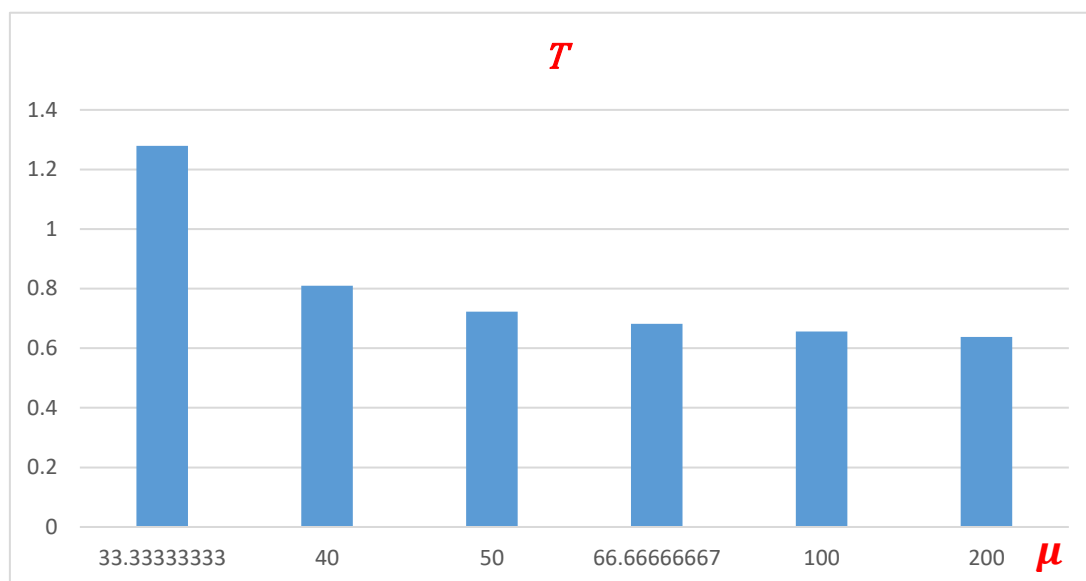
$$T = \frac{E[N]}{\gamma} = \frac{1}{\lambda} \sum_{i=1}^6 E[N_i] = 0.25157563$$

Dựa trên các kết quả minh họa ở trên, chúng tôi biểu diễn một số kết quả ở dạng đồ thị bằng cách thay đổi các tham số vào như tốc độ đến từ bên ngoài vào bộ cân bằng tải γ , tốc độ phục vụ của các nút tại tầng 2.



Hình 2. Thời gian đáp ứng của hệ thống thay đổi theo tốc độ đến γ

Hình 2 mô tả sự thay đổi của thời gian đáp ứng trung bình của toàn hệ thống khi tăng tốc độ đến γ ($\lambda < L$). Khi yêu cầu đến hệ thống tăng, với số server xử lý và tốc độ phục vụ không thay đổi, số yêu cầu cũng như thời gian chờ trong hàng đợi sẽ tăng, dẫn đến thời gian đáp ứng trên toàn hệ thống sẽ tăng, đặc biệt khi tốc độ đến tăng gần đến ngưỡng tốc độ phục vụ.



Hình 3. Thời gian đáp ứng của hệ thống thay đổi theo tốc độ phục vụ μ

Tương tự, Hình 3 mô tả sự thay đổi của thời gian đáp ứng trung bình của toàn hệ thống khi tăng tốc độ phục vụ μ ($\lambda < \mu$) tại các nút ở tầng 2 (tầng xử lý). Rõ ràng, khi tăng tốc độ phục vụ tại các nút xử lý, tức là giảm thời gian được phục vụ của các yêu cầu, thời gian trung bình các yêu cầu trong hệ thống sẽ giảm. Khi tốc độ phục vụ tăng đến 1 ngưỡng thì thời gian trung bình các yêu cầu trong hệ thống sẽ giảm không đáng kể.

4. KẾT LUẬN

Bài báo đã thực hiện việc áp dụng mô hình mạng hàng đợi Jackson mở vào mô hình hóa một kiến trúc tính toán đám mây trên cơ sở kết hợp các hàng đợi cơ bản M/M/1. Việc phân tích và đánh giá mô hình được thực hiện thông qua tính toán một số thông số đo hiệu năng hệ thống, như xác suất trạng thái cân bằng của hệ thống, thời gian đáp ứng trung bình, số yêu cầu trung bình, Một số kết quả số được chỉ ra cũng như mô tả qua đồ thị theo các tham số vào quan trọng như tốc độ đến, tốc độ phục vụ đã cho thấy tính đúng của mô hình phân tích. Hướng tiếp theo của bài báo sẽ là mở rộng mô hình hóa dựa trên các nền tảng mô phỏng tính toán đám mây như OpenStack hay CloudSim. Từ đó, có thể đánh giá, so sánh mô hình một cách thực tế hơn.

TÀI LIỆU THAM KHẢO

- [1]. Jordi Vilaplana, Francesc Solsona, Ivan Teixidó, Jordi Mateo, Francesc Abella, Josep Rius (2014), "A queuing theory model for cloud computing", The Journal of Supercomputing, Springer. [DOI 10.1007/s11227-014-1177-y].
- [2]. Phung-Duc, T., "Retrial Queue for Cloud Systems with Separated Processing and Storage Units," Proceedings of The 10th International Conference on Queueing Theory and Network Applications (QTNA2015), Hanoi, Vietnam, 17--20, August, 2015. Do, T.V., Takahashi, Y., Yue, W., Nguyen, V.-H. (Eds.), Advances in Intelligent Systems and Computing, Vol. 383, pp. 143-151. [DOI: 10.1007/978-3-319-22267-7_13].
- [3]. Santhi Kannan, Saravanan Ramakrishnan (2017), "Performance Analysis of Cloud Computing in Healthcare System Using Tandem Queues", International Journal of Intelligent Engineering and Systems, Vol.10, No.4, 2017, pp. 256 – 264. [DOI: 10.22266/ijies2017.0831.27].
- [4]. P. Suresh Varma. A. Satyanarayana and M. V. Rama Sundari, "Performance analysis of Cloud Computing using Queueing Models", In: Proc. of International Conf. on Cloud Computing Technique Application and Management, IEEE, pp.12-15, 2012.
- [5]. Chien Nguyen Khac, Khiết Bui Thanh, Hung Ho Dac, Son Nguyen Hong, Vu Pham Tran, Hung Tran Cong (2019), "An open jackson network model for heterogeneous infrastructure as a service on cloud computing". International Journal of Computer Networks & Communications (IJCNC) Vol.11, No.1, January 2019, pp. 63 - 80
- [6]. K. M. Maiyama, A. P. Namanya, Bashir Mohammed, M. Kiran, M. A. Kamala, and D. D. Kouvatsos (2016), "Analytical Performance Evaluation of OpenStack IaaS Cloud Using M/M/1 and M/M/c Queues", 32nd Annual UK Performance Engineering Workshop & Cyber Security Workshop, pp. 107-116.
- [7]. B. Sundarraj, "A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 3, March 2015
- [8]. Sai Sowjanya T, Praveen D, Satish K, Rahiman A (2011), "The queueing theory in cloud computing to reduce the waiting time", IJCSET, vol 1, no 3, pp 110–112.
- [9]. Vilaplana J, Solsona F, Abella F, Filgueira R, Rius J (2013), "The cloud paradigm applied to e-health", BMC Med Inf Decis Making 13:35
- [10]. Xiong K, Perros H (2009), "Service performance and analysis in cloud computing", In: Proceedings of IEEE World Conference Services, pp 693–700.
- [11]. H. Khazaei, J. Misić, and V.B. Misić (2012), "Performance Analysis of Cloud Computing Centers using M/G/m/m+r Queueing System", IEEE transactions on parallel and distributed systems, Vol. 23, no. 5, pp.936-943, 2012.

AN OPEN JACKSON QUEUE NETWORK MODEL FOR MULTI-TIER APPLICATION SYSTEMS BASED ON CLOUD COMPUTING

Huynh Xuan Chung, Dang Thanh Chuong

Faculty of Information Technology, University of Sciences, Hue University

Email: haichung1964@gmail.com, dtchuong@husc.edu.vn

ABSTRACT

Cloud computing platforms modeled with an open Jackson queue network can be used to identify and measure QoS, ensuring the cloud can deliver on response times. The analysis and evaluation can be done according to different parameters, such as the arrival rate of the customer service request and the service rate of the processing server, among other parameters. In this paper, we consider the model of an open queue network for a multilayer application on a cloud computing platform consisting of four layers with six nodes, in which the node structure can be different. Accordingly, the system model is modeled on a classical open network consisting of many $M/M/1$ queues. The analysis and evaluation of the model are done by calculating some parameters to measure system performance, such as average response time, the average number of requests,.... Some numerical results are shown as well as graphically described according to important input parameters such as arrival rate, service rate, which showed the accuracy of the analytical model.

Keywords: Cloud computing, open Jackson queue network.



Huỳnh Xuân Chung tốt nghiệp cử nhân Công nghệ thông tin tại trường Đại học Khoa học, Đại học Huế. Hiện nay, ông đang công tác tại trường THPT Nguyễn Công Trứ, Krông Pắc, Đắk Lắk

Lĩnh vực nghiên cứu: khoa học máy tính.



Đặng Thanh Chương sinh ngày 23/3/1975 tại TP Vinh. Ông tốt nghiệp cử nhân chuyên ngành Vật lý tại trường Đại học Khoa học, ĐH Huế năm 1997, tốt nghiệp thạc sĩ chuyên ngành Tin học năm 2004 tại trường Đại học Khoa học, ĐHH Huế; nhận học vị tiến sĩ về Đảm bảo toán học cho máy tính và hệ thống thông tin năm 2014 tại Viện CNTT - Viện Hàn lâm Khoa học Công nghệ Việt nam. Ông công tác tại Khoa CNTT, trường Đại học Khoa học, ĐH Huế từ năm 1997 đến nay.

Lĩnh vực nghiên cứu: mạng quang, tập trung vào chuyển mạch dựa trên gói/chùm và chất lượng dịch vụ; Lý thuyết hàng đợi; Tính toán đám mây.