Dự đoán hiệu năng của Web Services sử dụng mô hình mạng hàng đợi và phân tích hồi quy Gaussian

Web Services Performance Prediction using Queuing Network Model and Gaussian Process Regression Analysis

Huỳnh Quyết Thắng, Phùng Đình Vũ

Abstract: *In this paper, we present a solution to predict the Web Services performance by using the Queuing Network Model to enrich the input data and Gaussian Process regression analysis implementing for data prediction purpose. The detailed analysis results of the proposed solution are also presented and evaluated.*

Keywords*: Performance prediction, Web services, Gaussian Process regression analysis, Queuing Network.*

I. GIỚI THIỆU

Web Services đóng một vai trò quan trọng trong các kiến trúc hướng dịch vụ, cho phép các dịch vụ có sẵn được lựa chọn một cách nhanh chóng để xây dựng một ứng dụng. Có ba tác nhân chính tham gia vào một hệ thống Web Services, đó là Service Provider, Service Broker và Service Requester [1]. Có nhiều khía cạnh để đánh giá hiệu năng của Web Services như thời gian đáp ứng (response time), lượng thông tin truyền qua mạng trong đơn vị thời gian (throughput), số lượng lỗi xảy trong đơn vị thời gian, số lượng giao dịch [5]... Trong bài báo này, ta lựa chọn cách mô hình hóa hiệu năng Web Services thông qua mối quan hệ giữa thời gian đáp ứng của Web Services và tốc độ tới (arrival rate) [3]. Việc thu thập các mẫu trong đánh giá hiệu năng là thực hiện đo đạc thời gian đáp ứng của Web Services ứng với các giá trị tốc độ tới khác nhau. Quá trình thu thập dữ liệu hiệu năng này có thể thực hiện dưới góc độ người dùng [3], hoặc dưới góc độ nhà quản trị hoặc nhà thiết kế hệ thống [2]. Trong nghiên cứu này chúng tôi thực hiện dưới góc độ người sử dụng, xem xét Web Services là một hộp đen và đánh giá thời gian đáp ứng của hệ thống.

Hiện nay, một số hướng tiếp cận dự đoán hiệu năng Web Services đang được triển khai nghiên cứu. Hướng tiếp cận đầu tiên là thực hiện dự đoán thông qua mô phỏng. Phương pháp này sử dụng ngôn ngữ UML mô hình hóa các tác nhân của hệ thống, từ đó xây dựng mô hình ca sử dụng (Use Case) cho Web Services và biểu đồ thực thi tổng quát của Web Services tương ứng. Tiếp đến các tác giả sử dụng công cụ mô phỏng SMTQA (Simulation of Multi-tiered Queueing Applications) cho các hệ thống được thiết kế theo kiến trúc multi-tiered, qua đó có thể dự đoán hiệu năng của Web Services [2]. Ưu điểm của phương pháp này là có thể dự đoán sớm hiệu năng của Web Services bằng mô phỏng, cho ta một ước lượng về hiệu năng của hệ thống, khẳng định tính khả thi của hệ thống trước khi triển khai, khắc phục được lỗi nếu có. Nhược điểm của phương pháp này là ta phải biết được thiết kế chi tiết của hệ thống để đưa vào mô phỏng, hơn nữa không thể kiểm soát một cách tuyệt đối các tham số đầu vào, cũng như trạng thái của hệ thống trong thực tế, do đó đầu vào có thể không chính xác, dẫn tới đầu ra không chính xác. Phương pháp này chỉ áp dụng cho các hệ thống có kiến trúc đa tầng (multi-tiered architecture).

Hướng tiếp cận thứ hai là thực hiện dự đoán hiệu năng của các luồng công việc Web Services. Phương pháp này sử dụng ngôn ngữ thực thi Web Services WS-BPEL để mô hình hóa phía người sử dụng và phía Server. Kết hợp với mô hình mạng hàng đợi (Queuing Network) phương pháp này đưa ra giá trị dự đoán là

cận trên và cận dưới của thời gian đáp ứng và Throughput [9. Việc đánh giá cận là một trong những ưu điểm của phương pháp, nó phù hợp với thực tế, cho phép nhà quản trị có thể hình dung và đánh giá được khoảng hiệu năng, từ đó đưa ra chiến lược tối ưu hóa phù hợp. Hạn chế của phương pháp này là phải chỉ ra được những tài nguyên nào trong hệ thống mà các luồng công việc mô tả trong BPEL truy cập sử dụng, và hệ thống phải đánh giá thời gian sử dụng trên từng tài nguyên đó.

Hướng tiếp cận thứ ba, phương pháp dựa trên góc nhìn từ phía người sử dụng. Thu thập các dữ liệu mẫu về hiệu năng của Web Services, sau đó sử dụng các mô hình Web Services khác nhau như M/M/1, M/G/1, M/M/m, Queuing Network kết hợp với mạng Nơron để phân tích tìm ra các tham số cho mô hình và dự đoán hiệu năng [3]. Ưu điểm của phương pháp này là không cần đòi hỏi phải biết nhiều thông tin về Web Services cần đánh giá.

Hướng tiếp cận thứ 4 thực hiện dự đoán hiệu năng Web Services tại tầng thực thi của hệ thống, được thể hiện qua các thuộc tính về chất lượng dịch vụ (Quality of Service or QoS) của tầng thực thi, kho lưu trữ các dịch vụ, đầu vào,… Phương pháp này xem xét dữ liệu đầu vào có nhiễu, và mô hình hóa dữ liệu đầu sử dụng phân tích hồi quy Gaussian Process, kết hợp với công cụ mô phỏng hệ thống thực thi Web Services để dự đoán hiệu năng [15].

Chúng tôi kế thừa các nghiên cứu công bố trong [2,3,5,9,15] và sử dụng mô hình mạng hàng đợi kết hợp với Gaussian Process trong dự đoán hiệu năng Web Services. Do việc sử dụng mạng Nơron chỉ cho một đầu ra duy nhất, nên kết quả dự đoán hiệu năng cho một giá trị duy nhất. Mặc dù có thể chứng minh tính hiệu quả của mô hình, song giá trị dự đoán chỉ mang tính chất trung bình, không phản ánh được trong một số trường hợp. Cụ thể, trong hai lần đo đạc hiệu năng với cùng một tốc độ tới, cũng thu được kết quả thời gian đáp ứng khác nhau. Lý do là thời gian đáp ứng phụ thuộc vào đặc điểm của hệ thống tại từng thời điểm, các yếu tố phụ thuộc này thường mang tính chất ngẫu nhiên gây nên nhiễu [15]. Do đó thời gian đáp ứng được xem xét như một biến ngẫu nhiên. Khác với việc sử dụng mạng Nơron trong các nghiên cứu trước, ta đưa vào mô hình nhiễu và sử dụng phân tích hồi quy Gaussian Process để đưa ra giá trị dự đoán cho thời gian đáp ứng, đồng thời đánh giá khoảng tin cậy.

Bố cục của bài báo tiếp theo trình bày như sau: trong phần II trình bày giải pháp đề xuất; trong phần III là đánh giá trên cơ sở tính toán khoảng tin cậy cho kỳ vọng; trong phần IV trình bày thử nghiệm thực tế đã cài đặt; phần V là kết luận và hướng nghiên cứu tiếp theo.

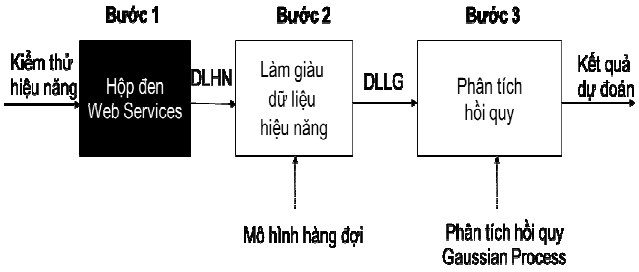
II. GIẢI PHÁP ĐỀ XUẤT

Hình 1 mô tả tổng quan quá trình 3 bước thực hiện dự đoán hiệu năng Web Services theo giải pháp đề xuất.

Bước 1: Mô hình hóa tín hiệu và nhiễu dựa trên các mẫu giá trị hiệu năng thu thập được.

Bước 2: Chia dữ liệu đầu vào thành tập huấn luyện và tập kiểm chứng. Sử dụng mô hình mạng hàng đợi, dựa vào các dữ liệu trong tập huấn luyện để đánh giá các tham số cho mô hình. Khi các tham số của mạng hàng đợi hoàn toàn xác định, ta bắt đầu thực hiện làm giàu dữ liệu. Lấy các giá trị tốc độ tới nằm ngoài khoảng giá trị trong bộ dữ liệu huấn luyện, và tính toán thời gian đáp ứng, để thu được bộ giá trị dữ liệu mới.

Bước 3: Thực hiện phân tích hồi quy Gaussian Process dựa trên dữ liệu đã làm giàu, đưa ra kết quả dự đoán kì vọng cho thời gian đáp ứng và đánh giá khoảng tin cậy.



Hình 1. Phương pháp dự đoán hiệu năng Web Services

II.1. Xây dựng mô hình đầu vào

Kiểm thử hiệu năng hệ thống được sử dụng trong đánh giá hiệu năng Web Services để đảm bảo hiệu năng hệ thống như mong đợi. Trong quá trình kiểm thử hiệu năng, ta có khả năng kiểm soát được tải của hệ thống, có thể thực hiện kiểm thử trong môi trường tải cao (nhiều yêu cầu) hoặc môi trường tải thấp (ít yêu cầu). Việc kiểm thử hiệu năng có thể tiến hành theo hai mô hình, mô hình mở (open model), hoặc mô hình đóng (close model) [3]. Trong mô hình mở, người sử dụng tới hệ thống ở một tốc độ định trước λ gọi là tốc độ tới và rời hệ thống khi yêu cầu được thực hiện. Trong mô hình đóng, số lượng người sử dụng được phục vụ trong hệ thống là cố định. Kế thừa các nghiên cứu trong [2,3,4,5], chúng tôi sử dụng mô hình mở.

Các lí thuyết của Markov và Poisson đưa ra kết quả khẳng định rằng quá trình đến có phân phối xác suất theo hàm mũ (phân phối mũ) là một quá trình Poisson [5]. Trong phương pháp này, quá trình đến thực hiện như sau: Với mỗi tốc độ tới λ_i ta thực hiện M lần, ở mỗi lần thứ $j, (j=\overline{1,M})$ ta lần lượt gửi D_{ij} yêu cầu, sau đó thực hiện đo thời gian đáp ứng của hệ thống với từng yêu cầu thứ k là $R^{i,j,k}$. Ở mỗi lần thứ *j* ta tính giá trị thời gian đáp ứng trung bình theo công thức (1):

$$R^{ij} = \frac{1}{D_{\mathrm{ij}}} \sum_{k=1}^{D_{\mathrm{ij}}} R^{ijk} \qquad (1)$$

Thời gian đáp ứng trung bình ứng với tốc độ tới λ_i cho tất cả các lần đo được tính là:

$$R^{i} = \frac{\sum_{j=1}^{M} D_{\mathrm{ij}} * R^{\mathrm{ij}}}{\sum_{j=1}^{M} D_{\mathrm{ij}}} \qquad (2)$$

Ta thực hiện đo đạc N lần ở các giá trị tốc độ tới khác nhau, giá trị thời gian đáp ứng cho mỗi lần thứ *i* là R^{ij} và R^{i} được cho trong công thức (1) và (2). Kết quả đầu ra của kiểm thử hiệu năng là các bộ số liệu $\{\lambda_i, [\mathrm{R}^{i1}, \mathrm{R}^{i2}, ..., \mathrm{R}^{iM}] \mid \mathrm{i}=\overline{1,N}\}$ và $\{\lambda_i, \mathrm{R}^{i} \mid \mathrm{i}=\overline{1,N}\}$ Dựa vào bộ dữ liệu hiệu năng thu thập ở trên, ta mô hình hóa dữ liệu đầu vào cho tín hiệu và nhiễu. Tín hiệu đầu vào là chung cho tất cả các phương pháp và có dạng:

$$\{(\lambda_i, \mathrm{R}^{i}) \mid \mathrm{i}=\overline{1,N}\} \qquad (3)$$

Nhiễu thường mang tính chất ngẫu nhiên do nhiều nguyên nhân khác nhau, và thường được mô hình hóa bởi phân phối xác suất Gauss. Do tính chất ngẫu nhiên, nhiễu phải có kì vọng 0 [15]. Dựa vào các mẫu số liệu $\{\lambda_i, [\mathrm{R}^{i1}, \mathrm{R}^{i2}, ..., \mathrm{R}^{iM}] \mid \mathrm{i}=\overline{1,N}\}$ ta sẽ mô hình hóa cho nhiễu như sau: Đặt $\overline{R}^{i} = \frac{1}{M}\sum_{j=1}^{M} R^{\mathrm{ij}}$ và $N^{\mathrm{ij}} = R^{\mathrm{ij}} - \overline{R}^{i}$, khi đó ta thu được bộ số là $\{\lambda_i, [\mathrm{N}^{i1}, \mathrm{N}^{i2}, ..., \mathrm{N}^{iM}] \mid \mathrm{i}=\overline{1,N}\}$. Bộ số này thể hiện sai số so với giá trị kì vọng, chúng ta dùng để đặc trưng cho nhiễu của tín hiệu. Giá trị kì vọng của nhiễu là $\mu_{iN} = \frac{1}{M}\sum_{j=1}^{M} N^{\mathrm{ij}} = 0$, và phương sai là:

$$\sigma_{iN}^{2} = \frac{1}{M}\sum_{j=1}^{M} (N^{\mathrm{ij}} - \mu_{iN})^{2} = \frac{1}{M}\sum_{j=1}^{M} (N^{\mathrm{ij}})^{2} \qquad (4)$$

Lí thuyết độ tin cậy phân phối Gauss [10] cho kết quả với xác suất 95%, khoảng tin cậy của kì vọng giá trị nhiễu sẽ là $\mu_{iN} \pm 1.96\sigma_{iN} = \pm 1.96\sigma_{iN}$. Nghĩa là với xác suất 95% thì khoảng tin cậy cho kì vọng thời gian đáp ứng R^{i} sẽ nằm trong khoảng $R^{i} \pm 1.96\sigma_{iN}$. Thực hiện tương tự cho các giá trị tới $\lambda_i, i=\overline{1,N}$ khác ta thu được bộ dữ liệu đầu vào cho nhiễu. Ta sẽ lấy 80% số liệu kiểm thử hiệu năng ở môi trường tải thấp để làm tập huấn luyện. Còn lại 20% dữ liệu kiểm thử hiệu năng ở môi trường tải cao làm tập kiểm chứng. Việc phân chia tập huấn luyện và tập kiểm chứng tương tự như phương pháp [3].

II.2. Làm giàu dữ liệu sử dụng Queuing Network

Làm giàu dữ liệu cũng là một quá trình suy diễn, nhưng là suy diễn có căn cứ dựa trên mô hình của

Web Services, do đó đạt độ chính xác cần thiết. Mô hình phổ biến nhất là mô hình mạng hàng đợi (Queuing Network) [4]. Gọi M là số số lượng hàng đợi trong hệ thống; λ là tốc độ tới trong toàn bộ hệ thống; $p_{i,j}$ là xác suất để yêu cầu của người dùng vào hàng đợi j sau khi rời khỏi hàng đợi i ($p_{i,i}=0$); p_{qi} là xác suất để yêu cầu của người dùng rời khỏi hệ thống sau khi chờ ở hàng đợi i; p_{iq} là xác suất để yêu cầu của người dùng đến hệ thống ở hàng đợi *i*. Ta đánh lại chỉ số các hàng đợi để chỉ số cho hàng đợi tương ứng với Web Services cần dự đoán hiệu năng là 1. Vì yêu cầu của người dùng luôn phải qua Web Services này nên xác suất để yêu cầu của người dùng đến hệ thống ở hàng đợi 1 là $p_{1q}=1$; và xác suất người dùng tới hệ thống ở các hàng đợi khác là $p_{iq}=0, i=\overline{2,M}$. λ_i là tốc độ tới tổng thể ở hàng đợi i; v_i là tỉ số thăm (visit ratio) ở hàng đợi i (tỉ số thăm thực chất là giá trị tốc độ tới được chuẩn hóa $v_i=\frac{\lambda_i}{\lambda}$). Tốc độ tới hàng đợi *i* được xem xét trong mối quan hệ với tốc độ rời đi ở các hàng đợi khác tới hàng đợi *i* $\lambda_i=\sum_{j=1}^{M}\lambda_j p_{ji}, i=\overline{1,M}$. Khi đó công thức đánh giá thời gian đáp ứng E[R] của mô hình Queuing Network này là:

$$E[R]=\frac{1}{\lambda}\sum_{i=1}^{M}\frac{\lambda_i}{\mu_i-\lambda_i}=\frac{1}{\lambda}\sum_{i=1}^{M}\frac{\lambda v_i}{\mu_i-\lambda v_i}= =\sum_{i=1}^{M}\frac{1}{\frac{\mu_i}{v_i}-\lambda}=\sum_{i=1}^{M}\frac{1}{\alpha_i-\lambda} \quad (5)$$

Trong đó $\alpha_i=\frac{\mu_i}{v_i}, i=\overline{1,M}$. Trong công thức (5) ở trên đại lượng λ không có chỉ số i, ý chỉ thời gian đáp ứng chỉ phụ thuộc vào tốc độ tới của toàn bộ hệ thống mà không phụ thuộc vào tốc độ tới trong từng hàng đợi [3,4]. Trước khi đưa ra kết quả dự đoán, ta phải đi xác định các tham số α_i sử dụng bộ dữ liệu vào là tập huấn luyện. Gọi θ là bộ tham số trong mô hình hàng đợi. Đặt $f_q(\lambda,\theta)=E[R]$ là công thức tính thời gian đáp ứng theo mô hình hàng đợi, phụ thuộc vào bộ tham số θ. Với một bộ giá trị xác định của tham số θ ta thu được kết quả dự đoán là $\left\{(\lambda_i, f_q(\lambda_i,\theta))\,|\,i=\overline{1,N}\right\}$. Ta đánh giá sai số trung bình bình phương trong kết quả dự đoán:

$$e(\theta)=\frac{1}{N}\sum_{i=1}^{N}(R^i-f_q(\lambda_i,\theta))^2 \quad (6)$$

Ta chọn tham số θ sao cho cực tiểu hóa hàm sai số bình phương $e(\theta)$ trong công thức (6). Để làm được điều này ta áp dụng phương pháp Gradient liên hợp không tuyến tính [11]. Sau khi xác định các tham số của mô hình mạng hàng đợi, ta tiến hành làm giàu bộ dữ liệu vào tại các điểm tốc độ tới như sau. Đặt $\lambda_{t-\min}$, $\lambda_{t-\max}$ lần lượt là giá trị tốc độ tới nhỏ nhất, giá trị tốc độ tới lớn nhất trong tập huấn luyện. Ta sẽ sử dụng mô hình hàng đợi để lấy thêm các mẫu tại tốc độ tới λ nằm ngoài khoảng $[\lambda_{t-\min},\lambda_{t-\max}]$ là chủ yếu, mục đích là để làm giảm sai số ngoại suy của phân tích hồi quy Gaussian Process. Gọi kích thước bộ dữ liệu mới lấy thêm từ mô hình hàng đợi là M, tập dữ liệu mới này có dạng: $\{\lambda_j, f_q(\lambda_j,\hat{\theta})\,|\,j=\overline{1,M}\}$. Kết hợp tập dữ liệu mới này với tập huấn luyện ta có tập dữ liệu làm giàu trước khi đưa vào phân tích hồi quy Gaussian Process.

II.3. Phân tích hồi quy Gaussian Process

Sau khi có toàn bộ dữ liệu hiệu năng và dữ liệu mới được làm giàu ở bước hai, ta thực hiện phân tích hồi quy Gaussian Process. Phân tích hồi quy Gaussian Process là một công cụ mạnh trong giải bài toán dự đoán với nền tảng toán học sử dụng xác xuất Bayes [6,7,8,15]. Phép phân tích hồi quy Gaussian Process sử dụng hàm hiệp phương sai có dạng:

$$k(\lambda_m,\lambda_n,\theta)=k_{signal}(\lambda_m,\lambda_n,\theta)+k_{noise}(\lambda_m,\lambda_n,\theta) \quad (7)$$

Trong đó $k_{signal}(\lambda_m,\lambda_n,\theta)$ là hàm hiệp phương sai đặc trưng cho tín hiệu, còn $k_{noise}(\lambda_m,\lambda_n,\theta)$ là hàm

hiệp phương sai đặc trưng cho nhiễu, và θ là tập các biến tham số của hàm.

Theo [6] ta lựa chọn hàm hiệp phương sai mũ bậc hai cho tín hiệu, công thức có dạng:

$$k_{signal}(\lambda_m, \lambda_n, \theta) = \rho^2 \; e^{-\frac{(\lambda_m - \lambda_n)^2}{2l^2}} \qquad (8)$$

Bộ tham số trong công thức (8) là $\theta = (\rho, l)$, để phân biệt đây là các tham số của hàm hiệp phương sai ta sẽ gọi các tham số này là các siêu tham số (hyperparamaters) [7,15]. Công cụ lí thuyết xác suất Bayes cho ta kết quả tính toán xác suất để có đầu ra là thời gian đáp ứng $\hat{R}_N = [\mathrm{R}^1, \mathrm{R}^2, ..., \mathrm{R}^N]^T$ với điều kiện bộ đầu vào tốc độ tới $\hat{\lambda}_N = [\lambda_1, \lambda_2, ..., \lambda_N]^T$ và bộ siêu tham số $\theta = (\rho, l)$ thỏa mãn công thức sau [6,8,15]:

$$\log p(\hat{R}_N \mid \hat{\lambda}_N, \theta) = -\frac{1}{2}\hat{R}_N^T K_{signal}^{-1} \hat{R}_N - \frac{1}{2}\log|K_{signal}| - \frac{N}{2}\log(2\pi) \qquad (9)$$

trong đó K_{signal} là ma trận hiệp phương sai [8]

Bộ siêu tham số θ tìm được sao cho đại lượng $\log p(R \mid \lambda, \theta)$ thể hiện được giá trị lớn nhất. Có thể áp dụng các giải thuật tối ưu như Gradient liên hợp, Nelder-Mead simplex. Chúng tôi sử dụng phương pháp Gradient liên hợp không tuyến tính (Nonelinear Conjugate Gradient) [11] để tìm các siêu tham số $\theta = (\rho, l)$ trong công thức (8). Tương tự, ta xây dựng ma trận hiệp phương sai K_N cho dữ liệu đầu vào, thay hàm hiệp phương sai tín hiệu bởi hàm hiệp phương sai chung nêu trong công thức (7). Mục đích bây giờ của ta là dự đoán tại điểm đầu vào với tốc độ tới λ_{N+1}, gọi giá trị thời gian đáp ứng thực tế tại điểm mới này là R^{N+1}. Ta xem xét (λ_{N+1}, R^{N+1}) như một phần của bộ dữ liệu vào, kích thước bộ dữ liệu vào lúc này là N+1. Ta xác định được phân phối xác suất để xảy ra R^{N+1} với điều kiện tồn tại bộ dữ liệu $\hat{R}_N$ là:

$$P(R^{N+1} \mid \hat{R}_N) = \frac{P(R^{N+1}, \hat{R}_N)}{P(\hat{R}_N)} \qquad (10)$$

Đại lượng $P(R^{N+1} \mid \hat{R}_N)$ gọi là phân phối hậu nghiệm. Theo [7] công thức (10) có thể được viết lại dưới dạng:

$$P(R^{N+1} \mid \hat{R}_N) = \frac{1}{C}\exp\left[-\frac{1}{2}[\hat{\mathrm{R}}_N^T \; \mathrm{R}^{N+1}]\mathrm{K}_{N+1}^{-1}\begin{bmatrix}\hat{R}_N \\ R^{N+1}\end{bmatrix}\right] \qquad (11)$$

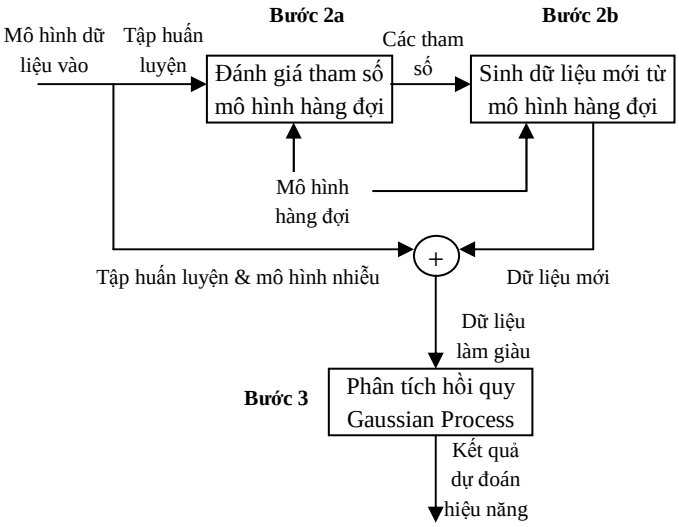
Trong công thức (11) ta có thể biểu diễn ma trận nghịch đảo K_{N+1}^{-1} theo các ma trận K_N và K_N^{-1} dưới dạng công thức nghịch đảo thành phần và thực hiện các biến đổi, sẽ thu được:

$$P(R^{N+1} \mid \hat{R}_N) = \frac{1}{C'}\exp\left[-\frac{(R^{N+1} - k^T K_N^{-1}\hat{R}_N)^2}{2(\kappa - k^T K_N^{-1} k)}\right] \qquad (12)$$

Trong công thức (12) phân phối hậu nghiệm $P(R^{N+1} \mid \hat{R}_N)$ cũng là một phân phối Gauss. Ta sẽ lấy giá trị kì vọng và phương sai của phân phối hậu nghiệm là kết quả dự đoán. Hai giá trị đó lần lượt là:

$$\bar{R}^{N+1} = k^T \; K_N^{-1} \; \hat{R}_N \qquad (13)$$

và $$\bar{\sigma}_{N+1}^2 = \kappa - k^T K_N^{-1} \, k \qquad (14)$$



Hình 2. Mô hình mạng hàng đợi và Gaussian Process

Công thức (13) được sử dụng để đánh giá giá trị kì vọng cho kết quả dự đoán thời gian đáp ứng tại điểm tốc độ tới λ_{N+1}. Công thức (14) được dùng để đánh giá khoảng tin cậy cho giá trị kì vọng ở trên. Đây là hai kết quả của phân tích hồi quy Gaussian Process. Phương pháp kết hợp mô hình hàng đợi và phân tích hồi quy Gaussian Process được mô tả trong Hình 2.

III.ĐÁNH GIÁ PHƯƠNG PHÁP ĐỀ XUẤT

III.1. Đánh giá khoảng tin cậy cho kì vọng

Để đơn giản trong các công thức chứng minh ở dưới ta đặt f thay cho xác suất hậu nghiệm $P(R^{N+1} \mid \hat{R}_N)$, μ thay cho giá trị kì vọng $\bar{R}^{N+1}, \sigma^2$ thay cho phương sai $\bar{\sigma}^2_{N+1}$, và X thay cho biến R^{N+1}.

Đặt biến $Z = \dfrac{X-\mu}{\sigma}$; Kì vọng của biến Z là [8] :

$$E[Z] = E\left[\frac{X-\mu}{\sigma}\right] = \frac{E[X]\text{-}\mu}{\sigma} = \frac{\mu-\mu}{\sigma} = 0 \quad (15)$$

Phương sai của biến Z là [8]:

$$\sigma_Z^2 = \frac{1}{N}\sum_{i=1}^{N}(Z(X_i) - E[Z])^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{X_i-\mu}{\sigma} - 0\right)^2 = \frac{1}{N}\sum_{i=0}^{N}\frac{(X_i-\mu)^2}{\sigma^2} = \frac{N\sigma^2}{N\sigma^2} = 1 \quad (16)$$

Hàm phân phối của Z xác định bởi công thức sau:

$$f_Z(Z) = \frac{1}{\pi}e^{-Z^2} \quad (17)$$

Lí thuyết về xác suất cho ta công thức xác định xác xuất để biến Z nhân giá trị trong khoảng (-n, n) là:

$$P(-n < Z < n) = \int_{-n}^{n} f_z(Z)dZ = \int_{-n}^{n}\frac{1}{\pi}e^{-Z^2}dZ \quad (18)$$

Đặt $\phi(n) = \int_{-n}^{n}\frac{1}{\pi}e^{-z^2}dz$; hàm $\phi(n)$ gọi là hàm mật độ xác suất của biến phân phối chuẩn với kì vọng 0 và phương sai 1. Cho giá trị của n thì giá trị của hàm xác xuất $\phi(n)$ hoàn toàn xác định bởi tích phân trên. (xem Bảng 1).

Bảng 1. Mối quan hệ giữa giữa đại lượng n và khoảng tin cậy

N	1.281	1.644	1.96	2.326	2.576	2.807
Xác suất	0.80	0.90	0.95	0.98	0.99	0.995

Ta có nhiều lựa chọn các giá trị khác nhau cho n, trong khuôn khổ của bài báo này ta chọn khoảng tin cậy 95%, tương ứng với n = 1.96. Chuyển các biến X, μ, σ về các đại lượng ta đặt lúc đầu.

Kết hợp với lựa chọn n = 1.96 ta có:

$$P(\bar{R}^{N+1} - 1.96\bar{\sigma}_{N+1} < R^{N+1} < \bar{R}^{N+1} + 1.96\bar{\sigma}_{N+1}) = 0.95 \quad (19)$$

Công thức (19) đánh giá khoảng tin cậy 95% của giá trị kì vọng $\bar{R}^{N+1}$. Khoảng tin cậy này có ý nghĩa là xác suất để thời gian đáp ứng tại điểm N+1 nằm trong khoảng $\bar{R}^{N+1} \pm 1.96\bar{\sigma}_{N+1}$ là 95%.

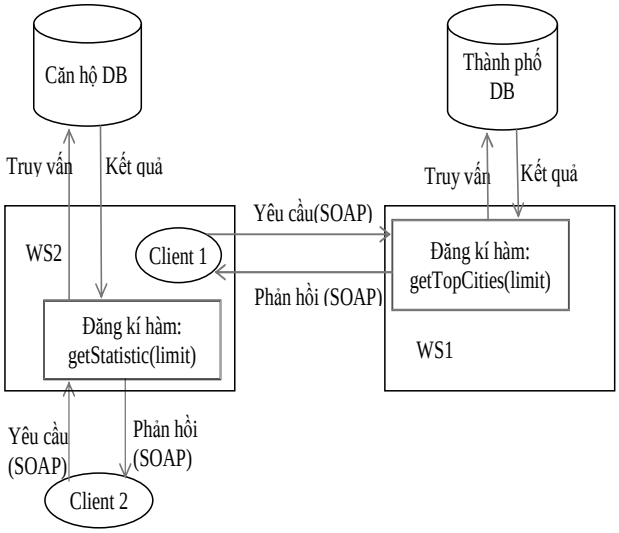
III.2. Đánh giá nhận xét về phương pháp đề xuất

Phương pháp dự đoán này là sự kết hợp của phân tích hồi quy Gaussian Process và mô hình mạng hàng đợi. Sự kết hợp Gaussian Process và mô hình hàng đợi này khai thác tối đa những ưu điểm của cả hai phương pháp. Phương pháp này xây dựng tín hiệu đầu vào giống với tất cả các phương pháp dự đoán hiệu năng Web Services khác. Còn xây dựng bộ dữ liệu nhiễu là hợp lí, phù hợp với một hệ thống Web Services thật. Tiếp theo, sử dụng phân tích hồi quy Gaussian Process, áp dụng lí thuyết xác suất của Bayes trong quá trình dự đoán, với giả thiết mô hình dữ liệu đầu vào có nhiễu ngẫu nhiên. Kết quả phân phối hậu

nghiệm $P(R^{N+1} \mid \hat{R}_N)$ cũng là một phân phối Gauss. Giá trị kì vọng của phân phối hậu nghiệm được sử dụng là giá trị kì vọng cho kết quả dự đoán về thời gian đáp ứng. Lí thuyết về khoảng tin cậy của phân phối Gauss đưa ra kết quả về khoảng tin cậy 95% cho kết quả dự đoán, kì vọng thời gian đáp ứng và khoảng tin cậy 95% cho kết quả dự đoán được chứng minh về mặt lí thuyết. Khoảng tin cậy 95% cho ta biết rằng xác suất để giá trị kì vọng thời gian đáp ứng nằm trong khoảng $\bar{R} \pm 1.96\bar{\sigma}$ là 95%.

IV.CÀI ĐẶT VÀ ĐÁNH GIÁ THỰC NGHIỆM

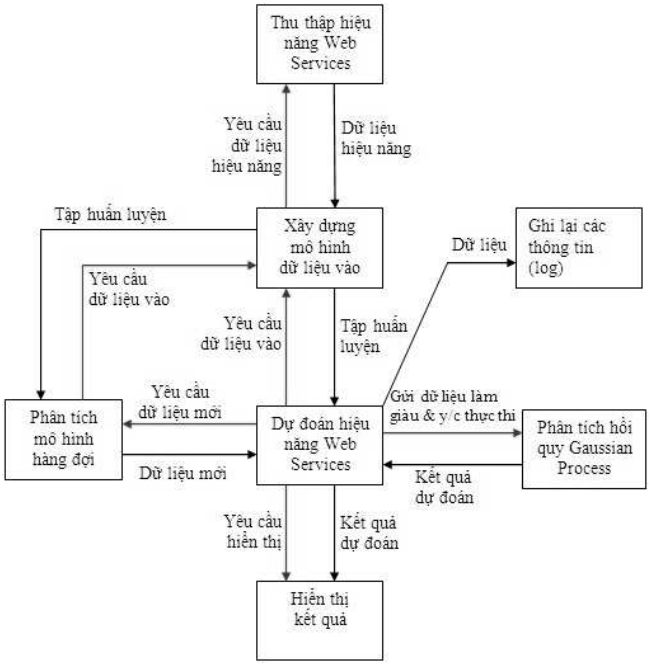
IV.1. Hệ thống Web Services



Hình 3. Hệ thống Web Services về căn hộ

Phần thực nghiệm làm việc với hệ thống Web Services về căn hộ với hai hàng đợi, được xây dựng trên Server công ty Cazoodle [12], trên các dịch vụ về thông tin căn hộ của nước Mỹ (Hình 3). Web Services số 1 có chức năng cung cấp các thành phố tại Mỹ có số dân đông nhất trong giới hạn của yêu cầu đầu vào. Dịch vụ này đăng kí một hàm có tên là getTopCities, hàm này nhận đầu vào là giới hạn số thành phố cần lấy, xây dựng câu truy vấn tới cơ sở dữ liệu "cities" lưu trữ thông tin dân số các thành phố của Mỹ (trong bảng cities có 19.476 thành phố). Kết quả trả về từ câu truy vấn sẽ được xử lí và trả về cho người dùng nào kết nối để sử dụng dịch vụ 1. Web Services số 2 thống kê về giá nhà cho thuê trung bình của một số thành phố đông dân nhất. Web Services 2 sẽ sinh ra một Client 1 để gửi yêu cầu lấy thông tin các thành phố đông dân nhất ở Mỹ. Web Services 2 đăng kí một hàm có tên là getStatistic để các client thông qua đó kết nối vào sử dụng dịch vụ này. Kết quả trả về từ Web Services 1 sẽ được tiếp tục xử lí trong hàm getStatistic, xây dựng câu truy vấn phù hợp, kết nối vào cơ sở dữ liệu về các căn hộ cho thuê tại Mỹ, với khoảng 7.000.000 bản ghi được cập nhật thường xuyên liên tục. Kết quả trả về từ câu truy vấn sẽ được tiếp tục xử lí và trả về cho người dùng sử dụngWeb Services 2. Các quá trình truyền thông giữa các Client với Server dịch vụ theo giao thức SOAP. Framework mã nguồn mở cho việc xây dựng Web Services là NuSOAP (dùng cho WS1) và PEAR-SOAP (dùng cho WS2).

IV.2. Xây dựng phần mềm thực nghiệm



Hình 4. Biểu đồ luồng dữ liệu và luồng điều khiển

Chúng tôi sử dụng ngôn ngữ lập trình Python, mã nguồn mở cho WS viết bằng ngôn ngữ Python "SOAPpy" [13], thư viện đồ họa bằng Python "matplotlib", thư viện xử lí toán học bằng Python "numpy", thư viện mã nguồn mở phân tích hồi quy

Gaussian Process viết bởi được Marion Neumann, Zhao Xu và Kristian Kersting [14] để xây dựng chương trình thử nghiệm đề xuất trong mục III. Chương trình được chia thành 7 module chính: Mô-đun 1 thu thập hiệu năng Web Services; Mô-đun 2- xây dựng mô hình dữ liệu vào; Mô-đun 3- thực hiện phân tích mô hình hàng đợi; Mô-đun 4 - thực hiện phân tích hồi quy Gaussian Process; Mô-đun 5- dự đoán hiệu năng Web Services; Mô-đun 6- ghi lại thông tin các tham số, các biến; Mô-đun 7- hiển thị kết quả. Hình 4 mô tả luồng dữ liệu (các đường mũi tên đen) và luồng điều khiển (các đường mũi tên đỏ) giữa các modules.

IV.3. Các kết quả thực nghiệm

Chiến lược kiểm thử sẽ là thực hiện tại các tốc độ tới phù hợp, vỡi mỗi tốc độ tới thực hiện 10 lần, trong mỗi lần gửi 1000 yêu cầu tới Server dịch vụ để tính thời gian đáp ứng trung bình.

Bộ dữ liệu đầu vào: Kết quả cho quá trình xây dựng bộ dữ liệu đầu vào của phương pháp theo mô hình nhiễu và mô hình tín hiệu thể hiện trong Bảng 2.

Kết quả sử dụng Gaussian Process: Kết quả sử dụng phân tích hồi quy Gaussian Process trên bộ dữ liệu tập huấn luyện thể hiện trong Bảng 3. Gọi $error_i$ là sai số dự đoán cho tại tốc độ tới λ_i. Ta đánh giá sai số nội suy dựa vào Bảng 3.

$$\mathrm{ERR}_{G-InterR} = \frac{1}{7}\sum_{i=1}^{7}\mathrm{err}_i^2 = 1.57E-3 \qquad (20)$$

Đánh giá sai số ngoại suy là:

$$\mathrm{ERR}_{G-\mathrm{Extra}R} = \frac{1}{2}\sum_{i=7}^{9}\mathrm{err}_i^2 = 50.23 \qquad (21)$$

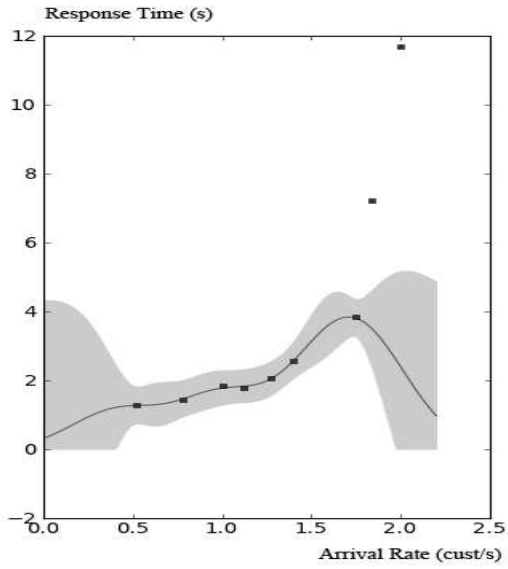
Kết quả của phương pháp dự đoán sử dụng phân tích hồi quy Gaussian Process được mô tả trong Hình 5. Nhìn vào đồ thị trong Hình 5 và công thức (20), (21) ta có thể thấy phương pháp dự đoán Gaussian Process cho ta kết quả với khoảng sai số nội suy là nhỏ, còn khoảng sai số ngoại suy tương đối lớn. Như vậy phân tích hồi quy Gaussian Process rất tốt với nội suy.

Bảng 2. Bảng dữ liệu đầu vào theo mô hình

Arrival Rate	Response Time	Noise
0.52	1.28754	8.00E-05
0.78	1.43226	1.90E-02
1	1.84129	4.93E-01
1.12	1.77317	1.40E-02
1.27	2.0507	1.34E-01
1.4	2.57144	8.12E-03
1.75	3.83371	2.60E-02
1.84	7.22472	9.57E-01
2	11.67351	1.50E-02

Bảng 3. Kết quả sử dụng Gaussian Process

Arrival Rate	Response Time (RT)	Prediction RT	Errror	Confidence Interval
0.52	1.28754	1.2716	0.01594	0.52833
0.78	1.43226	1.46654	0.03428	0.52453
1	1.84129	1.77886	0.06243	0.49309
1.12	1.77317	1.83044	0.05727	0.47468
1.27	2.0507	2.03771	0.01299	0.48568
1.4	2.57144	2.5886	0.01716	0.52406
1.75	3.83371	3.78998	0.04373	0.54417
1.84	7.22472	3.46354	3.76118	1.16117
2	11.67351	2.38324	9.29027	2.7627



Hình 5. Đồ thị kết quả dự đoán Gaussian Process

Kết quả sử dụng mô hình hàng đợi

Phần này trình bày kết quả dự đoán sử dụng mô hình hàng đợi. Hai tham số đánh giá cho mô hình sử dụng tập huấn luyện đầu vào $\alpha_1 = 2.24819$ và $\alpha_2 = 2.24819$. Kết quả trình bày trong bảng 4. Ta đánh giá sai số nội suy của phương pháp dự đoán:

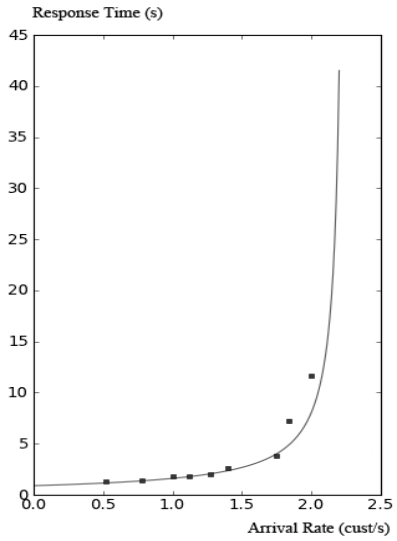
$$\mathrm{ERR}_{Q-InterR} = \frac{1}{7}\sum_{i=1}^{7}\mathrm{err}_i^2 = 22.47E-3 \qquad (22)$$

Đánh giá sai số ngoại suy của phương pháp dự đoán:

$$\mathrm{ERR}_{Q-ExtraR} = \frac{1}{2}\sum_{i=7}^{9}\mathrm{err}_i^2 = 9.237 \qquad (23)$$

Bảng 4. Kết quả sử dụng mô hình mạng hàng đợi

Arrival Rate	Response Time (RT)	Prediction RT	Errror
0.52	1.28754	1.157281	0.13026
0.78	1.43226	1.362223	0.07004
1	1.84129	1.602322	0.23897
1.12	1.77317	1.772753	0.00042
1.27	2.0507	2.044596	0.00610
1.4	2.57144	2.357966	0.21347
1.75	3.83371	4.014544	0.18083
1.84	7.22472	4.899696	2.32502
2	11.67351	8.058388	3.61512



Hình 6. Đồ thị kết quả sử dụng mô hình mạng hàng đợi

Hình 6 là đồ thị cho ta cái nhìn trực quan về kết quả của phương pháp dự đoán dùng mô hình hàng đợi. Nhìn vào đồ thị Hình 6 ta thấy đường cong dự đoán phù hợp hơn với mộ hệ thống Web Services thực tế. Khi mà tốc độ tới lớn, tương ứng với môi trường tải cao, có nhiều yêu cầu gửi tới Server dịch vụ. Vì vậy Server dịch vụ sẽ phải mất nhiều thời gian hơn, yêu cầu của người sử dụng phải đặt ở trong hàng đợi lâu hơn trước khi được phục vụ. Do vậy thời gian đáp ứng với tốc độ tới lớn sẽ lớn hơn (xem đường cong). Một điểm nhận thấy trong kết quả thực nghiệm dự đoán theo mô hình hàng đợi là kết quả sai số ngoại suy giảm đi đáng kể so với phân tích hồi quy Gaussian Process. Phương pháp sử dụng mô hình hàng đợi cho kết quả ngoại suy tốt.

Kết quả sử dụng phương pháp mới đề xuất

Phần này trình bày kết quả thực nghiệm của phương pháp đề xuất, sử dụng mô hình đầu vào tín hiệu và mô hình nhiễu, kết hợp với phân tích hồi quy Gaussian Process và mô hình hàng đợi. Các điểm tốc độ tới được chọn để bổ sung vào tập huấn luyện được chỉ ra trong Bảng 5.

Bảng 5. Bộ dữ liệu bổ sung từ mô hình hàng đợi

Arrival Rate	0.1	0.3	1.6	1.9	2.1
Response Time	0.93102	1.02659	3.08552	5.74401	13.49631

Bảng 6. Kết quả dự đoán phương pháp đề xuất

Arrival Rate	Response Time (RT)	Prediction RT	Errror	Confidence Interval
0.52	1.28754	1.27434	0.0132	0.53724
0.78	1.43226	1.46188	0.02962	0.53070
1	1.84129	1.79953	0.04176	0.50822
1.12	1.77317	1.81719	0.04402	0.49220
1.27	2.0507	2.06687	0.01617	0.48990
1.4	2.57144	2.5299	0.04154	0.50570
1.75	3.83371	3.7525	0.08121	0.51312
1.84	7.22472	4.7291	2.49562	0.50082
2	11.67351	9.18637	2.48714	0.67253

Kết quả của phương pháp dự đoán đề xuất được mô tả trong Bảng 6.

Ta đánh giá sai số nội suy của phương pháp:

$$\mathrm{ERR}_{C-InterR} = \frac{1}{7}\sum_{i=1}^{7}\mathrm{err}_i^2 = 1.902E-3 \qquad (24)$$

Đánh giá sai số ngoại suy của phương pháp:

$$\mathrm{ERR}_{C-\mathrm{Extra}R} = \frac{1}{2}\sum_{i=7}^{9}\mathrm{err}_i^2 = 6.207 \qquad (25)$$

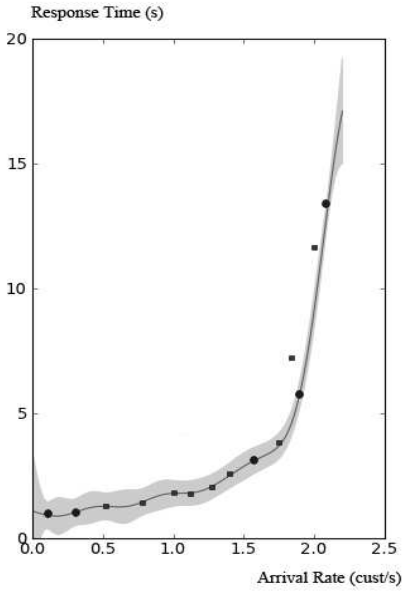
Từ đồ thị Hình 7, ta thấy đường cong dự đoán phù hợp với một hệ thống Web Services thực tế như giải thích ở trên. Sai số nội suy và ngoại suy đều giảm hơn so với hai phương pháp sử dụng phân tích hồi quy Gaussian Process và mô hình hàng đợi riêng lẻ ở trên.

IV.4. Đánh giá sai số của phương pháp

Về sai số nội suy: Điểm đầu tiên có thể thấy là phương pháp đề xuất có sai số nội suy khá nhỏ (cỡ 10^{-3}). Căn cứ vào các công thức (20), (22) và (24) ta có đánh giá sau:

$$E_{C-InterR} \approx E_{G-InterR} < E_{Q-InterR} \qquad (26)$$

Về sai số ngoại suy: phương pháp mới đề xuất đưa ra kết quả dự đoán với sai số ngoại suy nhỏ hơn nhiều so với khi sử dụng riêng lẻ phương pháp phân tích hồi quy Gaussian Process và mô hình hàng đợi.



Hình 7. Đồ thị kết quả phương pháp đề xuất

Phần thực nghiệm này cũng khẳng định rằng phân tích hồi quy Gaussian Process tỏ ra rất kém hiệu quả trong bài toán ngoại suy, nhưng kết quả nội suy lại tốt. Trong khi đó mô hình hàng đợi đạt được kết quả ngoại suy tốt, phù hợp với hệ thống Web Services thực tế. Dựa vào đặc điểm bộ dữ liệu làm giàu là sử dụng mô hình hàng đợi để lấy mẫu thêm tại các điểm có tốc độ tới lớn. Bộ dữ liệu làm giàu này là đầu vào của phân tích hồi quy Gaussian Process. Nên kết quả nội suy của phân tích hồi quy Gaussian Process trên bộ dữ liệu làm giàu chính là kết quả ngoại suy của bài toán trên bộ dữ liệu tập huấn luyện. Phương pháp mới này cho kết quả dự đoán tốt, dựa trên sự kết hợp điểm mạnh của từng phương pháp phân tích Gaussian Process và mô hình hàng đợi một cách riêng lẻ.

$$E_{C-\mathrm{Extra}R} \le E_{Q-\mathrm{Extra}R} \le E_{G-ExtraR} \qquad (27)$$

IV.5. Đánh giá khoảng tin cậy

Như đã chỉ ra trong phần đánh phương pháp về mặt lý thuyết, điểm mới trong kết quả dự đoán của phương pháp đề xuất đó là đánh giá được khoảng tin cậy 95% cho kì vọng thời gian đáp ứng. Trong phần này ta sẽ đánh giá thực nghiệm phương pháp cho khoảng tin cậy này. Ví dụ với tốc độ tới $\lambda_i = 0.52$, theo Bảng 6 khoảng tin cậy sẽ là 1.27434 ± 053424. Dựa vào bảng số liệu hiệu năng trong 10 lần đo cho tốc độ tới này, ta có đánh giá độ thuộc của mỗi lần đo thời gian đáp ứng trung bình vào khoảng tin cậy, thể hiện qua Bảng 9. Từ Bảng 9 ta thấy 100% các giá trị tại các lần đo thuộc vào khoảng tin cậy. Làm tương tự cho các giá trị tốc độ tới khác, ta có tổng kết như trong Bảng 10.

Từ Bảng 10, ta có có một số nhận xét sau:

(i) Có nhiều tốc độ tới mà độ thuộc là 100%.Một số độ thuộc 80%,90% là do tính chất xác suất chỉ đúng khi số lượng sự kiện là lớn, ở đây ta chỉ xét tới 10 sự kiện.

(ii) Ở các giá trị ngoại suy, độ thuộc là 0% bởi vì các điểm ngoại suy lấy mẫu mới phụ thuộc vào mô hình hàng đợi, dự đoán theo mô hình cũng có sai số. Tại các điểm ngoại suy, ta đã chỉ ra được là sai số ngoại

suy là nhỏ nhất trong kết quả sử dụng 3 phương pháp, thể hiện trong công thức (27). Đây cũng là ưu điểm của phương pháp dự đoán hiệu năng đề xuất.

Bảng 9. Độ thuộc khoảng tin cậy ở tốc độ tới 0.52 cust/s

Lần đo	1	2	3	4	5	6	7	8	9	10
Kết luận	T	T	T	T	T	T	T	T	T	T

Bảng 10. Độ thuộc khoảng tin cậy tại các tốc độ tới

Tốc độ tới (cust/s)	0.52	0.78	1	1.12	1.27	1.4	1.75	1.84	2
Phần trăm thuộc (%)	100	80	100	90	100	100	100	0	0

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi đã trình bày một phương pháp dự đoán hiệu năng của Web Services sử dụng các dữ liệu về kiểm thử hiệu năng thu thập được, làm giàu dữ liệu hiệu năng bằng mô hình hàng đợi và sử dụng phân tích hồi quy Gaussian Process để đưa ra kết quả dự đoán kèm theo đánh giá khoảng tin cậy cho giá trị kì vọng. Chúng tôi cũng trình bày đánh giá bằng lí thuyết và thực nghiệm cho phương pháp. Qua các kết quả, có thể khẳng định rằng phương pháp đề xuất đã kế thừa thành công sử dụng mô hình mạng hàng đợi vào quá trình dự đoán. Kết quả dự đoán là một giá trị kì vọng cho thời gian đáp ứng, giống như ở các phương pháp khác. Về đánh giá sai số, phương pháp mới đề xuất cho ra kết quả sai số nội suy tốt do kế thừa được ưu điểm của phân tích hồi quy Gaussian Process. Kết quả sai số ngoại suy cũng tốt theo đánh giá phần thực nghiệm. Với việc làm giảm sai số ngoại suy ở mức nhỏ nhất (so sánh với hai phương pháp sử dụng Gaussian Process và mô hình hàng đợi riêng lẻ), phương pháp đề xuất đã giải quyết được vấn đề dự đoán tốt hiệu năng cho môi trường tải cao, thay thế kiểm thử hiệu năng thực hiện trong môi trường tải cao vì có chi phí cao.

Một số hướng phát triển có thể đề xuất cụ thể là:

-Thứ nhất: có nhiều cách lựa chọn hàm hiệp phương sai khác nhau, nghiên cứu để tìm ra một hàm hiệp phương sai phù hợp nhất với phương pháp dự đoán cũng là một hướng phát triển. Tiến tới, có thể sử dụng mô hình hàm ẩn (latent function) đa lớp (multi-class) cho hàm hiệp phương sai, để tự động lựa chọn hàm hiệp phương sai phù hợp trong lớp các hàm hiệp phương sai.

-Thứ hai: nghiên cứu sâu hơn để đưa ra quyết định đúng đắn và có căn cứ cho việc lựa chọn giá trị tốc độ tới nào là hợp lí để thu thập thêm bộ dữ liệu mới sử dụng mô hình hàng đợi, góp phần vào quá trình làm giàu dữ liệu hiệu năng đầu vào. Phương pháp đề xuất chưa giải quyết được vấn đề là khi có thêm một bộ dữ liệu mới thu thập từ kiểm thử hiệu năng bổ sung vào bộ dữ liệu đầu vào ta sẽ giải quyết thế nào cho tối ưu nhất. Việc tối ưu hóa quá trình bổ sung, cập nhật dữ liệu hiệu năng mới giúp ta giải quyết bài toán dự đoán hiệu năng online, thích hợp cho các hệ thống thời gian thực.

TÀI LIỆU THAM KHẢO

[1] ERIC NEWCOMER, GREG LOMOW: *Understanding SOA with Web Services.* Addison-Wesley Professional, 2004, (480 pages).

[2] CH RAM MOHAN REDDY, D EVANGELIN GEETHA, KG SR T V SURESH KUMAR, K.RAJANI KANTH, *Early performance prediction of Web Service,* International Journal on Web Service Computing (IJWSC), Vol.2, No.3, September 2011.

[3] LESLIE CHEUNG, LEANA GOLUBCHIK, FEI SHA, *A study of Web Services Performance Prediction: A Client's Perspective,* IEEE 19th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, July 2011.

[4] WILLIAM J.STEWART, *Probability, Markov Chains, Queues, and Simulation,* Priceton University Process, 2009, (776 pages).

[5] RAMESH NAGAPPAN, SAMEER TYAGI, *High Performance Web Service,* Sun's 2004 worldwide Java developer conference, 2004.

[6] M. EBDEN, *Gaussion Process for Regression: A quick introduction*, www.gaussianprocess.org

[7] C. E. RASMUSSEN, C. K.I.WILLIAMS, *Gaussian Processes for Machine Learning*, MIT Press, 2006, (266 pages).

[8] DAVID J.C MACKAY, *Introduction to Gaussian Processes*, www.cs.utoronto.ca/~mackay/gpB.pdf

[9] MERNO MARZOLLA, RAFAELA MIRANDOLA, *Performance prediction of Web Services workflows*. Proceedings of the Quality of software architectures 3rd international conference on Software architectures, components, and applications, Pages 127-144 , Springer-Verlag Berlin, Heidelberg, 2007

[10] Document in the course B01.1305, "*Confidence Interval*", New York University, Stern School of Business

[11] WILLIAM W.HAGER, HONGCHAO ZHANG, *A survey of nonlinear conjugate gradient methods*, Pacific journal of Optimization, Vol. 2, No. 1. (2006), pp. 35-58.

[12] http://www.cazoodle.com/

[13] http://pypi.python.org/pypi/SOAPpy/

[14] https://github.com/marionmari/pyGPs

[15] JAKUB M. TOMCZAK, JERZY SWIATEK, AND KRZYSZTOF LATAWIEC, *Gaussian process regression as a predictive model for Quality-of-Service in Web service systems*, CoRR abs/1207.6910 (2012).

Nhận bài ngày: 11/09/2013

SƠ LƯỢC VỀ TÁC GIẢ

HUỲNH QUYẾT THẮNG

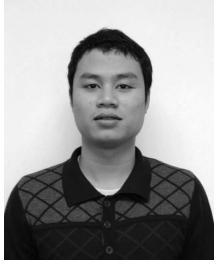
Sinh năm 1967 tại Hà Nội.

Tốt nghiệp đại học và bảo vệ luận án tiến sỹ tại ĐH Tổng hợp kỹ thuật Varna, Cộng hòa Bulgaria, vào các năm 1990 và 1995.

Hiện đang công tác tại Bộ môn Công nghệ phần mềm, Viện CNTT& TT, trường ĐH Bách Khoa Hà Nội.

Hướng nghiên cứu đang triển khai: Software Metrics, Software Reliability Engineering, Secure Coding, Program Analysis.

PHÙNG ĐINH VŨ



Sinh năm 1989 tại Nam Định.

Tốt nghiệp kỹ sư CNTT tại trường ĐH Bách khoa HN năm 2012.

Hiện nay đang công tác tại Công ty TNHH phần mềm đầu tư WorldQuant Việt Nam.

Hướng nghiên cứu chính: dự đoán đánh giá hiệu năng các hệ thống Web, tối ưu hoá các hệ thống Web trên môi trường tải cao và dữ liệu lớn.

Điện thoại liên hệ: 0125 277 7431

Email: phungdinhvu@gmail.com