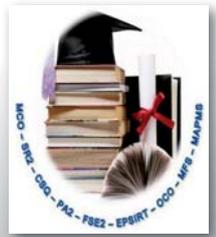


Modélisation d'un système des Web services avec les réseaux de Petri et évaluation de ses performances



N. BERNINE, H. NACER, D. AISSANI

Doctoriales de Recherche Opérationnelle, le 12 et 13 Décembre 2018

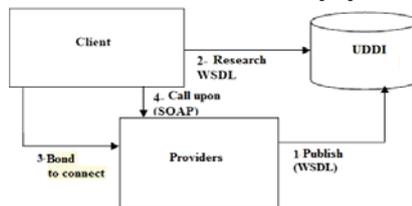
Introduction

Les Web services sont des applications accessibles sur Internet réalisant chacune une tâche spécifique. Pour fournir une solution à une tâche complexe, on peut regrouper des Web services pour n'en former qu'un seul; on parle alors de composition de Web services. Un Web service est dit composé ou composite lorsque son exécution implique des interactions avec d'autres Web services afin de faire appel à leurs fonctionnalités. La composition de Web services spécifie quels services ont besoin d'être invoqués, dans quel ordre et comment gérer les conditions d'exception [01], [02]. La découverte des Web services consiste à trouver les mises en correspondance adéquates entre les éléments de besoin des utilisateurs et les éléments des Web services existants. Ces besoins peuvent être couverts par un Web service simple, ou un Web service complexe issu de la tâche de la composition [03]. Pour assurer la qualité d'un système, on fait appelle à l'évaluation de performances. L'évaluation des performances fait l'objet de discussions sur l'espace informatique. Elle détermine la qualité d'un système à partir des résultats obtenus pour les entrées/sorties de ce système dans le cadre de l'accomplissement de la tâche qui leur a été assignée. L'évaluation de performances s'intéresse au calcul des paramètres (indices) de performances d'un système. Ces derniers sont représentés sous forme de valeurs quantitatives, comme le débit, le temps d'attente, le temps de réponse, le nombre moyen d'une entité donnée, le taux d'utilisation [04],[05],[06],[07]. Les méthodes d'évaluation de performance peuvent être basées sur la simulation ou utiliser des méthodes analytiques. Ce travail présente un modèle analytique basé sur les réseaux de Petri, permettant d'évaluer les performances d'un système de Web services où le service des demandes des clients et les Web services suivent la loi exponentiel. Le processus d'arrivée des demandes des utilisateurs et des Web services suit une distribution de Poisson. Lorsque les Web services terminent le service, ils rejoignent la file d'attente. Nous utilisons une méthode analytique pour résoudre le modèle. De plus, nous calculons le temps de réponse du système et le nombre moyen de clients dans le système en termes de taux d'arrivée des demandes des clients. Nous trouvons le nombre limite de clients dans le système à partir duquel il commence à être saturé.

Méthodologie

POSITION DU PROBLÈME

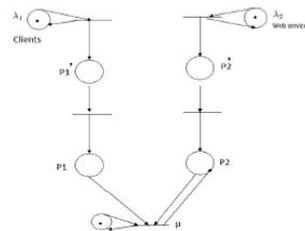
Nous allons contribuer à l'évaluation de performances dans un système des Web services, ce système peut être représenté par l'architecture suivante [08] :



Architecture du système des Web services

MODÉLISATION DU PROBLÈME

Le modèle décrit le système des Web services, tenant en compte l'arrivée des demandes des clients et l'arrivée des Web services, pour évaluer ces performances. Nous considérons que le système est composé de 2 stations; la première représente une demande de clients et la seconde une demande de services Web, avec un serveur exponentiel. Les arrivées sur le réseau sont de Poisson et les taux d'arrivée sont $\lambda_i, i \in [1, 2]$, et sont indépendants. Une fois le service (Découverte et composition) est terminé, les Web services rejoignent leur station. Le taux de service (découverte et composition) est de μ .



Modèle de réseau de Petri d'un système des Web services.

Résultats

La forme générale des probabilités stationnaires:

$$\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_{(n+1)^2})$$

For $n=m=1$, we have

$$\begin{cases} \pi_1 = 0, \\ \pi_2 = 0, \\ \pi_3 = \frac{\mu}{\lambda_1 + \mu}, \\ \pi_4 = \frac{\mu}{\lambda_1 + \mu}. \end{cases}$$

For $n = m \geq 2$, we have

$$\begin{cases} \pi_i = 0 & , \text{ if } 1 \leq i \leq (n+1)^2 - (n+1), \\ \pi_i = \frac{(n\mu)^{(n+1)^2 - i} \lambda_1^{i - [(n+1)^2 - i]}}{(\lambda_1 + \mu)^i + \sum_{j=1}^{n-1} (n\mu)^j \lambda_1^{i-j}} & , \text{ if } (n+1)^2 - n \leq i \leq (n+1)^2. \end{cases}$$

Nombre moyen de Clients dans le système

$$\begin{cases} L_s = \frac{3\mu + 4\lambda_1}{\lambda_1 + \mu} & , \text{ if } n=m=1, \\ L_s = \sum_{i=(n+1)^2 - n}^{(n+1)^2} \frac{i(n\mu)^{(n+1)^2 - i} \lambda_1^{i - (n+1)^2 + i}}{(\lambda_1 + \mu)^i + \sum_{j=1}^{n-1} (n\mu)^j \lambda_1^{i-j}} & , \text{ if } n \geq 2. \end{cases}$$

Temps moyen de réponse

$$\begin{cases} W_s = \frac{3\mu + 4\lambda_1}{\lambda_1(\lambda_1 + \mu)} & , \text{ if } n=m=1 \\ W_s = \sum_{i=(n+1)^2 - n}^{(n+1)^2} \frac{i(n\mu)^{(n+1)^2 - i} \lambda_1^{i - (n+1)^2 + i}}{\lambda_1[(n\mu)^i + \sum_{j=1}^{n-1} (n\mu)^j \lambda_1^{i-j}]} & , \text{ if } n \geq 2 \end{cases}$$

Temps moyen de réponse

| | $\lambda_1=0.00001$ | $\lambda_1=0.0001$ | $\lambda_1=0.001$ | $\lambda_1=0.01$ | $\lambda_1=0.1$ |
|---------|-----------------------|-----------------------|-----------------------|------------------------|----------------------|
| $n=1$ | 3×10^3 | 3.0001×10^3 | 3.001×10^3 | 3.0099×10^3 | 30.909 |
| $n=10$ | 99.9001×10^3 | 99.9010×10^3 | 99.9101×10^3 | 100.0009×10^3 | 1009.183 |
| $n=100$ | 10^3 | 10^3 | 10^3 | 10^3 | 1.0010×10^3 |
| $n=120$ | 1.44×10^3 | 1.44×10^3 | 1.44×10^3 | 1.4401×10^3 | 1.4412×10^3 |
| $n=140$ | 1.96×10^3 | 1.96×10^3 | 1.96×10^3 | 1.9601×10^3 | 1.9614×10^3 |
| $n=141$ | 1.9881×10^3 | 1.9881×10^3 | 1.9881×10^3 | 1.9882×10^3 | 1.9895×10^3 |
| $n=142$ | ∞ | ∞ | ∞ | ∞ | ∞ |

Nombre moyen de clients dans le système

| | $\lambda_1=0.00001$ | $\lambda_1=0.0001$ | $\lambda_1=0.001$ | $\lambda_1=0.01$ | $\lambda_1=0.1$ |
|---------|----------------------|----------------------|----------------------|----------------------|----------------------|
| $n=1$ | 3 | 3.0001 | 3.001 | 3.0099 | 3.0909 |
| $n=10$ | 99.9001 | 99.9010 | 99.9101 | 100.0009 | 100.9183 |
| $n=100$ | 10^3 | 10^3 | 10^3 | 10^3 | 1.0010×10^3 |
| $n=120$ | 1.44×10^3 | 1.44×10^3 | 1.44×10^3 | 1.4401×10^3 | 1.4412×10^3 |
| $n=140$ | 1.96×10^3 | 1.96×10^3 | 1.96×10^3 | 1.9601×10^3 | 1.9614×10^3 |
| $n=141$ | 1.9881×10^3 | 1.9881×10^3 | 1.9881×10^3 | 1.9882×10^3 | 1.9895×10^3 |
| $n=142$ | ∞ | ∞ | ∞ | ∞ | ∞ |

Conclusions

Dans ce travail, nous avons un modèle de performance avec des réseaux de Petri pour analyser les demandes des clients et les Web services, et nous obtenons le nombre limite des Web services qui est de 142 à saturer le système. Cependant, l'élaboration de la stratégie client proposée nécessite des mécanismes de récupération plus complexes et coûteux. concernant la disponibilité ou l'indisponibilité des Web services. Nous allons aborder d'autres problèmes dans nos travaux futurs en complétant la solution proposée avec les réseaux de Petri colorés dans le cloud computing.

Références

- [1]. H. Kadima. "Les Web services", Edition Eyrolles, 2003.
- [2]. B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid "Composing Web services on the Semantic Web", The VLDB Journal, 12 (4), 2003.
- [3]. H. Nacer, D. Aissani, Semantic web services: Standards, applications, challenges and solutions. Journal of Network and Computer Applications, 44:134-151, 2014
- [4]. A. Aissani " Modèles stochastiques de la théorie de fiabilité ", Office de publications Universitaires, Alger, 1992.
- [5]. M. Ettl, G. E. Feigin "A Supply Network Model with Base- Stock Control and Service Requirements", Operations Research, vol. 48, n. 2, pp. 216-232, 2000.
- [6]. C. E. Riddalls, S. Bennett " The stability of supply chains", International Journal of Production Research, vol. 40, n. 2, pp. 459-475, 2002.
- [7]. A. Koubaa "Introduction à l'évaluation de performances des systèmes informatiques et de communication", apport de recherche, University of Minnesota, www.amazon.com/PerformanceEvaluationofComputerandCommunicationSystems/3540, 2004.
- [8]. H. Nacer, D. Aissani, N. Djebari. A distributed authentication model for composite web services. computer and security, 70:144-178, 2017.