

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire de Magister

En Mathématiques Appliquées

Option

Modélisation Mathématique et Techniques de Décision

Thème

Evaluation des Performances d'une Architecture pour
la découverte et la composition des Web Services

Présenté par :

M^{elle} Nassima BERNINE

Devant le jury composé de :

Président	M ^r	S. Adjabi	Professeur	U. de Béjaïa
Rapporteur	M ^r	D. Aïssani	Professeur	U. de Béjaïa
Examinatrice	M ^{me}	M. Iouallalen	Professeur	U.S.T.H.B. Alger
Examineur	M ^r	K. Abbas	M. C. A	U. de Béjaïa
Invitée	M ^{me}	H. Talantikite	M. C. B	U. de Béjaïa

Béjaïa, 2012

Remerciements

**Louange A Dieu, le miséricordieux, sans Lui rien de tout cela
n'aurait pu être.**

Je tiens à exprimer ma profonde reconnaissance à M^r D. Aïssani professeur à l'université A. Mira de Béjaïa et M^{me} H. Talantikite, pour leur encadrements, leur qualités tant scientifiques qu'humaines. Ce travail doit beaucoup à leur disponibilité permanente, leur rigueur scientifique et sa patience.

Je tiens également à remercier M^r S. Adjabi, pour l'honneur qu'il a fait en acceptant de présider le Jury de ce mémoire.

Je remercie M^{me} M. Iouallalen, et M^r K. Abbas d'avoir accepté de faire partie du Jury et consacrés leurs temps à la lecture et à la correction de ce mémoire.

J'exprime également toute ma gratitude envers M^{me} H. Talantikite, je la remercie de sa patience, ses conseils et son aide précieuse.

Mes remerciements sont également adressés à M^{me} K. Adel et M^r K. Abbas pour l'aide qu'ils m'ont apporté.

Mes remerciements les plus vifs vont tout particulièrement à mes parents, en qui j'ai puisé tout le courage, la volonté et la confiance, je leurs seront éternellement reconnaissante.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce modeste travail :

A Mes très chers parents qui m'ont soutenu durant tout mon parcours.

A Mon très chère frère Menad.

A Mes sœurs.

A Tous mes amis.

A Toute personne me connaissant de loin ou de près.

Nassima

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	v
Introduction générale	1
1 État de l'art des Web services	4
1.1 Introduction	4
1.2 Définition des Web services	4
1.3 Architecture des Web services	5
1.3.1 Architecture de référence	5
1.3.2 Architecture étendue	7
1.4 Infrastructure de base des Web services	7
1.4.1 Le protocole : SOAP(Simple Object Access Protocol)	8
1.4.2 WSDL (Web Service Definition Language)	10
1.4.3 UDDI (Universal Description, Discovery and Integration)	13
1.5 Découverte des Web services	14
1.6 Composition des Web services	15
1.6.1 Types de composition	15
1.7 Avantages des Web services	16
1.8 Conclusion	18

2	Méthodes d'évaluation des performances	20
2.1	Introduction	20
2.2	Définition de l'évaluation de performances	20
2.3	Concept de l'évaluation de performances	20
2.4	Rôle de l'évaluation de performances	21
2.5	Mesurer et prévoir la performance	21
2.5.1	Méthodes analytiques	22
2.5.2	Simulation	27
2.6	Avantages et inconvénients de la simulation	30
2.6.1	Avantages	30
2.6.2	Inconvénients	31
2.7	Comparaison des techniques d'évaluation des performances	31
2.8	Méthode de stabilité forte	31
2.8.1	Critère de stabilité forte	32
2.9	Troncature	33
2.9.1	Principe de la troncature	33
2.9.2	Différents techniques de la troncature des chaînes de Markov	34
2.10	Conclusion	35
3	Travaux antérieurs d'évaluation de performances des Web services	36
3.1	Introduction	36
3.2	Evaluation des performances par une modélisation analytique	36
3.2.1	Indisponibilité des Web services [63]	36
3.2.2	Modélisation hiérarchique de la disponibilité perçue par l'utilisateur [64]	39
3.3	Evaluation des performances par la simulation	41
3.3.1	Composition des Web services avec un support du réseau actif [65] .	41
3.3.2	Evaluation de performances des Web Services distribués par le Blue- tooth [66]	42

3.4	Comparaison des différents travaux d'évaluation de performances des Web services	43
3.5	Conclusion	47
4	Mesure de performances d'un système de découverte et composition des Web services	48
4.1	Introduction	48
4.2	Description du modèle	48
4.3	Description de la méthode de résolution du modèle	50
4.4	Application	61
4.4.1	Algorithme de stabilité forte	61
4.4.2	Résultats et comparaison	62
4.5	Analyse de performances	63
4.5.1	Intèrprétation des résultats	63
4.6	Positionnement de notre proposition par rapport à la littérature	66
4.7	Conclusion	67
	Conclusion générale	68
	Bibliographie	70
	Liste des acronymes	77
		78

Table des figures

1.1	Architecture des Web services.	6
1.2	Architecture en pile.	7
1.3	Structure d'un message SOAP.	9
1.4	Structure d'un document WSDL.	11
1.5	Structure de données d'UDDI	14
2.1	Étapes d'évaluation de performances d'un système	22
3.1	Modèle de performance.	37
3.2	Chaîne de Markov avec N_W serveurs Web.	39
4.1	Modèle réel.	49
4.2	Modèle idéal.	51
4.3	Courbes des erreurs.	63
4.4	Nombre moyen de clients dans le système en fonction de λ_1	64
4.5	Nombre moyen de clients dans le système en fonction de λ_2	65
4.6	Durée moyenne de séjour en fonction de λ_1	65
4.7	Durée moyenne de séjour en fonction de λ_2	66

Liste des tableaux

3.1	Tableau comparatif des différents travaux d'évaluation de performances des Web services.	46
4.1	Tableau comparatif des erreurs	62

Introduction générale

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.

L'objectif des Web services est de faciliter l'accès aux applications entre entreprises et ainsi de simplifier les échanges de données. Ils poursuivent un vieux rêve de l'informatique distribuée où les applications pourraient interopérer à travers le réseau, indépendamment de leur plate-forme et de leur langage d'implémentation.

L'évaluation de performances est un procédé qui consiste à déterminer les caractéristiques des systèmes obtenus pour ces entrées/sorties dans le cadre de l'accomplissement de la tâche qui leur a été assignée.

La modélisation est un outil de plus en plus utilisé pour configurer et analyser les différents systèmes réels. En tenant compte des facteurs aléatoires, on parle alors de modélisation et de modèles stochastiques. Les processus stochastiques décrivent l'évolution d'une grandeur aléatoire en fonction du temps. Leur étude s'insère dans la théorie des probabilités dont elle constitue l'un des objectifs les plus profonds. Elle soulève des problèmes mathématiques intéressants et souvent très difficiles. Il existe de nombreuses applications des processus aléatoires notamment en physique, biologie, médecine, et bien entendu en informatique. C'est ainsi que les spécialistes font appel de plus en plus, à la modélisation par les processus stochastiques, notamment ceux de type Markovien.

La portée des méthodes analytiques s'avère limitée lors de l'étude de beaucoup de systèmes de files d'attente, en raison de la complexité des résultats connus. En effet, dans la majorité des cas, on se retrouve confronté à des systèmes d'équations dont la résolution

est très difficile ou possèdent des solutions qui ne sont pas facilement interprétables. Pour pallier à toutes ces difficultés, les chercheurs ont recouru à des méthodes d'approximation. Dans ce cas, on est amené à remplacer le système réel, complexe, par un système idéal plus simple dont les résultats analytiques sont exploitables. Le modèle ainsi utilisé représente une idéalisation du système complexe. Cela est généralement réalisé en altérant soigneusement la structure ou les hypothèses du modèle. De plus, les différents paramètres sont estimés par des méthodes statistiques à partir de données empiriques. Le modèle est alors sujet à des perturbations qui pourront naturellement induire une déviation dans ses caractéristiques par rapport à celles d'origine. D'où l'apparition du problème de stabilité.

L'étude de la stabilité consiste à délimiter le domaine dans lequel le modèle idéal peut être utilisé comme une bonne approximation du système réel. En d'autres termes, un modèle est stable s'il résiste suffisamment aux perturbations dans sa structure ou dans ses paramètres (entrées).

Dans ce travail, nous allons modéliser un système de Web services simple et composite, par un modèle de files d'attente à deux stations en tandem, où la première station a une file d'attente infinie et la deuxième station est avec une file d'attente limitée (N), et avec deux taux des arrivées, l'arrivée à la première station avec un taux λ_1 et l'arrivée à la deuxième station avec un taux λ_2 . À la fin du service d'un client dans la première station, s'il est satisfait, il quitte le système avec la probabilité $1 - p$. Sinon, il passera à la deuxième file d'attente avec la probabilité p , pour la recherche d'une réponse dans les Web services composites. À la fin de ce service, s'il est encore non satisfait, il va rejoindre la première file d'attente avec la probabilité $1 - q$.

Notre système n'est pas sous forme produit, ce qui rend son étude difficile. Nous avons alors approximé ce système par un autre système de files d'attente où la première station a une file d'attente limitée (Q). Pour ce faire, nous avons d'abord utilisé la troncature, ensuite la méthode de stabilité forte pour trouver les conditions d'application de cette approximation.

Nous avons fait varier les deux taux des arrivées afin de déterminer l'influence (de ces derniers) sur le nombre moyen de clients dans le système et la durée moyenne de séjour.

Les résultats trouvés montrent clairement que le système de découverte et de composition est toujours sollicité. En effet, l'utilisation du réseau Internet pour satisfaire des services demandés est toujours actuellement très répandue.

Ce mémoire est structuré de la manière suivante :

- Le premier chapitre est consacré à une étude bibliographique sur les Web services.
- Dans le deuxième chapitre, nous illustrons les méthodes d'évaluation des performances.
- Le troisième chapitre comprend une synthèse des résultats antérieurs sur l'évaluation des performances des Web services. Une analyse comparative est présentée.
- Dans le quatrième chapitre, nous présentons et discutons des résultats numériques concernant l'erreur d'approximation de notre système, ainsi que l'influence des deux taux des arrivées sur le nombre moyen de clients dans le système et la durée moyenne de séjour des clients.

Chapitre 1

État de l'art des Web services

1.1 Introduction

Ces dernières années, les systèmes d'information contenus sur Internet ont toujours servi pour un accès rapide et facile à la demande pertinente de l'internaute, selon ses besoins, par l'actualisation des différentes technologies. Au début, la recherche de l'information s'appliquait directement sur le Web, mais l'apparition des Web services a permis de changer cet état, par la création et l'utilisation des applications qui ont permis de réaliser un système d'information rapide et efficace distribué sur l'Internet.

1.2 Définition des Web services

Les Web services sont des applications auto-descriptives, modulaires et faiblement couplées qui fournissent un modèle de programmation et de déploiement d'applications, basé sur des normes, et s'exécutant au travers de l'infrastructure du Web [01].

Le W3C (World Wide Web Consortium) définit un Web service comme étant un système logiciel identifié par un URI (Uniform Resource Identifier), dont les interfaces publiques et les liens sont définis et décrits en XML (eXtensible Markup Language) [02]. Sa définition peut être découverte (dynamiquement) par d'autres systèmes logiciels. Ces autres systèmes peuvent ensuite interagir avec le Web service, en utilisant des messages XML transportés par des protocoles Internet [02].

Tandis qu'IBM donne la définition suivante :

Les Web services sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le Web. Les Web services effectuent des actions allant de simples requêtes à des processus-métiers complexes. Une fois qu'un service est déployé, d'autres applications (y compris des Web services) peuvent le découvrir et l'invoquer.

Nous définissons un Web service comme étant une application (un programme) mise à la disposition des autres, et accessible par les clients à travers des protocoles internet standards. Des exemples de services actuellement disponibles concernent les prévisions météorologiques, la réservation de voyage en ligne, les services bancaires ou des fonctions entières d'une entreprise comme la mise en oeuvre de la gestion de la chaîne logistique.

1.3 Architecture des Web services

1.3.1 Architecture de référence

Les efforts de recherche et de développement récents autour des Web services ont conduit à un certain nombre de spécifications qui définissent aujourd'hui l'architecture de référence des Web services.

L'architecture de référence (Fig1.1) des Web services s'articule autour des trois rôles suivants :

- **Le fournisseur de service** : Il correspond au propriétaire du service. D'un point de vue technique, il est constitué par la plate-forme d'accueil du service ;
- **Le client** : Qui correspond au demandeur du service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service. L'application cliente peut être elle-même un Web service ;
- **L'annuaire des services** : Il correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que

des facilités de recherche de services à l'intention des clients.

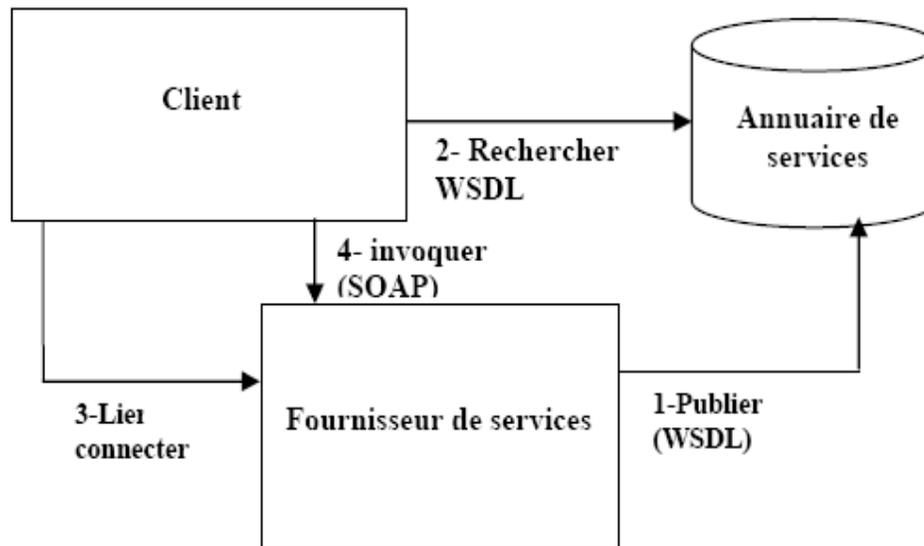


FIG. 1.1 – Architecture des Web services.

Les interactions de base entre ces trois rôles incluent les opérations de publication, de recherche et d'invocation.

Ci-dessous le scénario type d'utilisation de cette architecture :

- Le fournisseur de services définit la description de son service et la publie dans un annuaire de service universel des Web services, UDDI.
- Le client utilise les facilités de recherche disponibles au niveau de l'annuaire universel pour retrouver et sélectionner un service donné.
- Le client examine ensuite la description du service sélectionné pour récupérer les informations nécessaires qui lui permettent de se connecter au fournisseur du service et d'interagir avec l'implémentation du service considéré.

Pour garantir l'interopérabilité ¹ des trois opérations précédentes (publication, recherche et lien), des propositions de standards ont été élaborées pour chaque type d'interaction [04].

1.3.2 Architecture étendue

Une architecture étendue (Fig1.2 [03]) est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. La figure 1.2 décrit un exemple d'une telle pile. La pile est constituée de plusieurs couches, chaque couche s'appuie sur un standard particulier.

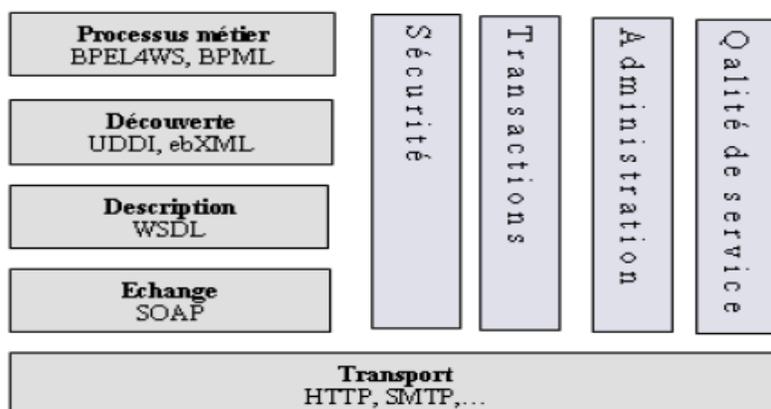


FIG. 1.2 – Architecture en pile.

1.4 Infrastructure de base des Web services

L'infrastructure de base :

Ceux sont les fondements techniques établis par l'architecture de référence.

L'infrastructure de base répond aux problèmes d'intégration technique des applications autour de certains standards.

¹La capacité que possède un système, dont les interfaces sont intégralement connues, à fonctionner avec d'autres systèmes existants ou futurs et ce, sans restriction d'accès ou de mise en œuvre.

1.4.1 Le protocole : SOAP(Simple Object Access Protocol)

Le SOAP est un protocole de transmission de messages basé sur XML. C'est un standard recommandé par le W3C(**World Wide Web Consortium**) qui définit un ensemble de règles pour structurer des messages qui peuvent être utilisés dans de simples transmissions unidirectionnelles, mais il est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call). Le SOAP permet une communication dans les systèmes, il est assez léger, simple et facile à déployer, extensible et ouvert mais ne garde pas la trace des requêtes envoyées des différents clients vers les Web services invoqués [04].

Structure d'un message SOAP

Un message SOAP est composé d'une enveloppe, d'un en-tête et d'un corps :

- **Une enveloppe (Envelope)** : Qui est considérée comme la racine d'un message SOAP, elle contient un attribut obligatoire $\prec Body \succ$ qui fait référence au schéma XML décrivant la structure du message SOAP. Elle pourrait aussi contenir un autre attribut optionnel $\prec Header \succ$.
- **Un en-tête optionnel (Optional Header)** : Qui contient des informations spécifiques complémentaires sur ce même message.
- **Un élément corps obligatoire (Mandatory Body)** : Qui contient le message réel destiné au point final du message.

La figure 1.3 représente les différents composants d'une requête SOAP et la façon dont ils y sont classés.



FIG. 1.3 – Structure d'un message SOAP.

Le transporteur idéal de SOAP

SOAP est considéré comme le protocole standard des Web services.

Un message SOAP est assez facile à décoder et éventuellement à modifier puisqu'il est encodé en mode texte, et pour cela on trouve ces deux éléments :

- **Le Content-Type** : Il indique que le corps de la requête est une représentation XML.
- **Le SOAPAction** : Il indique la destination de la requête SOAP. La réponse à la requête SOAP/HTTP² indiquera si elle a finalement été traitée correctement ou pas.

²Le protocole de transfert sur internet le plus courant. C'est par l'intermédiaire de celui-ci que sont transmises les pages Web (au langage HTML) et que le navigateur Web présente le client de façon structurée.

1.4.2 WSDL (Web Service Definition Language)

WSDL est un langage de description de Web services sous format XML.

Il permet de définir la structure des objets d'une manière abstraite et indépendante du langage de développement, et cela quels que soient le protocole ou le codage utilisé par des objets spécifiques à chaque système.

Un fichier WSDL comporte les éléments suivants :

- Définitions abstraites des données qu'on souhaite transmettre ;
- Informations de type adresse afin de localiser le service à spécifier ;
- Informations sur le type de données ;
- Informations sur toutes les fonctions disponibles publiquement ;
- Informations obligatoires sur le protocole de transport qu'il faut spécifiquement utiliser.

WSDL propose une double description du service :

- Une vue dite abstraite qui présente les opérations et les messages des services ;
- Une vue dite concrète qui présente les choix de mise en oeuvre faites par le fournisseur du service.

La figure 1.4 illustre la structure d'un document WSDL.

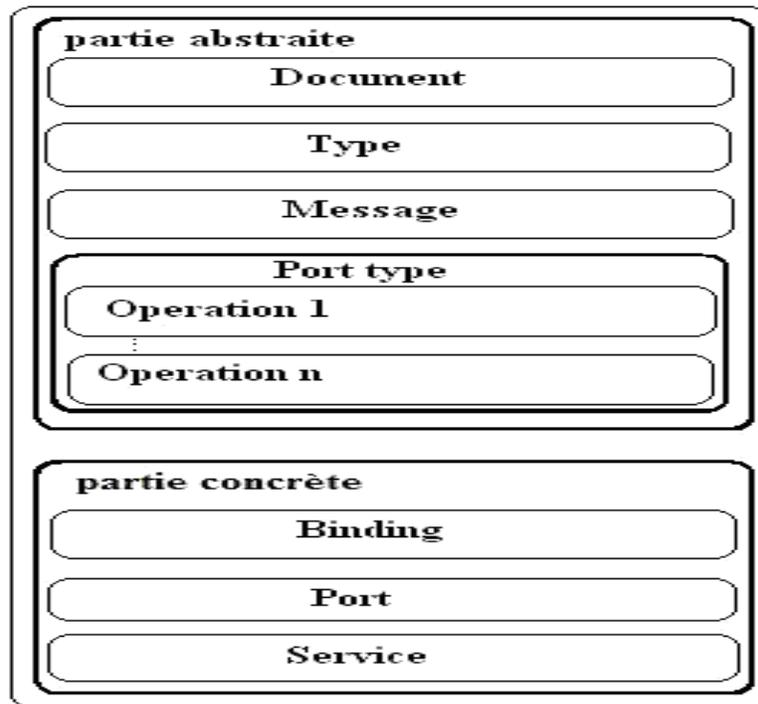


FIG. 1.4 – Structure d'un document WSDL.

La partie abstraite

Elle est composée de six éléments :

Définition, Document, Import, Types, Message et Porttype.

- **Définition** : Cet élément est considéré comme la racine du document WSDL. Il définit le nom du Web service et déclare les différents espaces de nom utilisés dans les documents.
- **Type** : Cet élément contient des types de données importants pour les messages échangés, pour un maximum d'interopérabilité et une neutralité de la plateforme.
- **Message** : Il représente une définition abstraite des données transmises. Il peut être un message d'entrée ou un message de sortie d'un Web service.

- **PortType** : Un Web service peut contenir plusieurs portTypes. Chaque portType définit un ensemble d'opérations abstraites offertes par le Web service. Un portType est identifié par un nom et contient une opération au moins.
Chaque opération a un nom et contient au plus un message d'entrée qui est défini par l'élément "Input" et au plus un message de sortie qui est défini par l'élément "Output". En cas d'exception, l'élément 'Fault' est présent et définit un message d'erreur retourné.
- **Import** : On l'utilise pour importer d'autres documents WSDL, en associant un espace de nom à la location d'un document.
- **Document** : Cet élément est facultatif, le document WSDL : permet de décrire des commentaires.

La partie concrète

Elle comporte deux éléments : liaison et service.

- **Service** : Regroupe un ensemble de ports reliés entre eux. Il a un nom et contient un ou plusieurs ports. Chaque élément "port" a un nom et définit un point d'accès au Web service.
- **Liaison** : Il définit le format d'un message et les protocoles de communication, pour des opérations et des messages définis par un portType. Un fichier WSDL peut contenir plusieurs éléments 'Binding', dont chacun est identifié par un nom et fait référence à un portType.

1.4.3 UDDI (Universal Description, Discovery and Integration)

UDDI est une spécification définissant la manière de publier et de découvrir les Web services sur le réseau.

L'idée de l'UDDI est de normaliser le format des entrées d'entreprises et de services dans un annuaire pour faciliter la découverte des Web services et encourager les échanges d'affaires entre elles.

Rôle de l'UDDI

Les informations dans l'UDDI sont divisés en trois catégories :

- **Pages jaunes** : Elles indiquent ce que fait le service.
- **Pages blanches** : Elles donnent des informations sur le fournisseur du service.
- **Pages vertes** : Elles donnent des détails sur comment utiliser techniquement le Web service.

Le registre UDDI utilise le protocole SOAP ainsi que le protocole HTTP, et suit un modèle d'échange de données fondé sur la notation XML.

Langage de Description

L'information stockée dans le registre UDDI correspond à la figure 1.5 et comprend les quatre types d'informations suivantes :

businessEntity : Qui donne les informations sur l'entreprise offrant le service ;

businessService : Qui donne les informations sur le service offert ;

bindingTemplate : Qui donne l'information permettant d'utiliser un Web service particulier ;

tModel : (technical model) Qui représente toute sorte d'information. Cela peut être une interface de service, une classification, la sémantique d'une opération,...

Le tModel, grâce à ses liens avec les autres éléments, est réellement le coeur des

spécifications UDDI. Le contenu du tModel peut représenter tout type d'information et peut être écrit dans n'importe quel langage. Le tModel est référencé par sa clé unique.

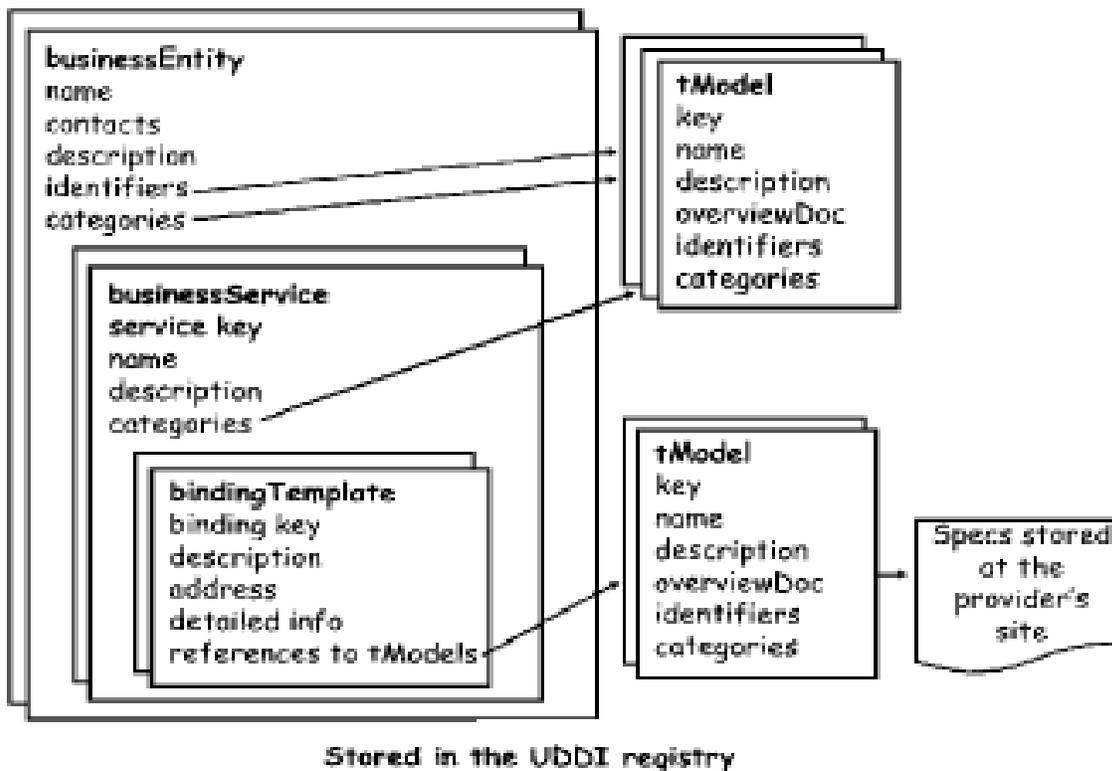


FIG. 1.5 – Structure de données d'UDDI

1.5 Découverte des Web services

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 [05].

1.6 Composition des 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.

La composition des Web services se réfère au processus de création d'un service composite offrant une nouvelle fonctionnalité, à partir de Web services existants, par le processus de découverte dynamique, d'intégration et d'exécution de ces services dans un ordre bien défini afin de satisfaire un besoin bien déterminé [06]. Un système de composition comprend principalement trois types de participants (acteurs) : Le fournisseur, le consommateur et le composeur. Généralement, le composeur se divise en d'autres sous rôles qui peuvent être les suivants : traducteur, générateur de processus et l'évaluateur [07]. Le traducteur traduit la description entre un langage externe utilisé par les participants et un langage interne utilisé par le générateur de processus. Pour chaque requête, ce dernier essaye de générer un plan combinant les services disponibles dans le dépôt de services qui pourront répondre au mieux aux besoins de la requête. Du moment que différents Web services peuvent implémenter des fonctionnalités similaires, plusieurs plans de composition peuvent être ainsi générés. Ces plans seront soumis à l'évaluateur qui va sélectionner le meilleur plan pour l'exécution en se basant sur les préférences du demandeur. Enfin, le moteur d'exécution, exécute le plan ainsi sélectionné et retourne le résultat au service demandeur. L'exécution d'un service composite peut être vue comme l'échange d'une séquence de messages entre les services participants selon le modèle de processus défini [08].

1.6.1 Types de composition

La composition peut être classifiée en trois catégories :

1. Composition manuelle

La composition manuelle des Web services suppose que l'utilisateur génère la

composition à la main via un éditeur de texte et sans l'aide d'outils dédiés.

2. Composition semi-automatique

Les techniques de composition semi-automatiques sont un pas en avant en comparaison avec la composition manuelle, dans le sens qu'ils font des suggestions sémantiques pour aider à la sélection des Web services dans le processus de composition.

3. Composition automatique

La composition totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise.

Malgré les efforts de recherche et de développement autour de la problématique de la composition des services, elle reste une tâche hautement complexe et pose un certain nombre de défis. Sa complexité provient généralement des sources suivantes :

- L'augmentation du nombre des Web services sur le Web rend très difficile la découverte et la sélection des Web services pouvant répondre à un besoin donné ;
- Les services sont créés et mis à jour de façon hautement dynamique ;
- Les Web services sont développés par différentes organisations qui utilisent différents modèles conceptuels pour décrire les caractéristiques des Web services [07] ;
- La modularité constitue une caractéristique importante des Web services. Par conséquent, les Web services composites doivent garder récursivement les mêmes caractéristiques que les Web services simples, à savoir interopérables [09].

1.7 Avantages des Web services

Les Web services [10] :

- Permettent l'interopérabilité dans des environnements applicatifs, cela veut dire que les logiciels et les applications écrits dans différents langages de programmation, et évoluant sous différents systèmes d'exploitation peuvent communiquer et/ou échanger des données entre eux facilement.
- Sont utilisables à distance via n'importe quel type de plateforme, et sont accessibles depuis n'importe quel type d'utilisateurs.
- Peuvent servir au développement d'applications distribuées, appartiennent à des applications capables de collaborer entre elles de manière transparente pour l'utilisateur, et permettent d'avoir un partage des fonctionnalités et facilitent grandement le développement.

Les services Web offrent les solutions suivantes [11] :

- **Architecture de composants** : Les systèmes à base de composants les plus classiques comme Corba [12] présentent souvent une faille au niveau de leur conception. En fait, la pauvreté des systèmes d'interconnexion existants entre leurs différents modules nuit souvent à leur ouverture. En effet, les liens entre les applications sont souvent statiques et un simple changement dans l'un des modules peut avoir de graves répercussions. Cette situation, parfois évitée par un bon niveau d'abstraction dans l'analyse de conception, laisse souvent la place à de trop nombreux cas où la modification de l'implémentation d'un des modules d'une application, entraîne aussitôt la modification du contenu d'autres parties du programme. Les Web services proposent un niveau d'abstraction supérieur qui remplace les modèles d'applications courants par une architecture plus modulaire, plus flexible et qui possède donc une meilleure tolérance aux changements.
- **Facilité de l'interopérabilité** : Les Web services s'appuient sur les protocoles SOAP,

WSDL et UDDI. Ces standards basés sur le langage XML, permettent de simplifier l'utilisation des services auxquels ils sont liés, par l'utilisation de fichiers textuels, donc lisibles par l'utilisateur. Leur contenu se limitant aux éléments d'interopérabilité essentiels, ces standards assurent une grande indépendance de l'implémentation par rapport au système d'exploitation, à l'architecture machine ou au langage utilisé.

- **Intégration temps réel :** L'architecture des Web services montre une évolution dans la conception d'applications. Les applications se résument à l'association dynamique de services en temps réel. Tous les composants de l'application sont donc des services qui encapsulent leur comportement, et possèdent des mécanismes de publication et de découverte. En fait, lors de l'exécution, les besoins de l'utilisateur sont traduits en entrée pour le système de découverte de services, qui va trouver le service correspondant et qui va ensuite gérer la transmission des messages nécessaires à l'exécution de celui-ci. L'avantage, c'est qu'un groupe de services peut lui-même être vu comme un service, ce qu'on appelle "composition de Web services".
- **Ouverture :** Les Web services sont la plus récente technologie utilisée pour supporter le développement de systèmes d'information distribués sur Internet. Aujourd'hui, ils semblent être la solution la plus adaptée pour assurer l'interopérabilité sur Internet entre les plateformes, les systèmes d'exploitation et les langages de programmation.

1.8 Conclusion

Les Web services représentent une technologie qui offre une interaction sur le Web indépendamment de toute plate-forme et de tout langage. Cette partie établit une étude de l'état de l'art des Web services. Pour assurer un bon fonctionnement de cette technologie, et pour avoir ses caractéristiques, on a recours à l'évaluation des performances. Nous exposons dans le chapitre suivant les méthodes d'évaluation des performances.

Chapitre 2

Méthodes d'évaluation des performances

2.1 Introduction

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.

2.2 Définition de l'évaluation de performances

L'évaluation de performances s'intéresse au calcul des paramètres (indices) de performances d'un système [13]. 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 [14].

2.3 Concept de l'évaluation de performances

Pour faire de l'évaluation de performances, on doit disposer de deux éléments [15] :

- **Un système** : C'est l'entité dont on évalue les performances. En général, un système est considéré comme étant un ensemble de ressources partagées entre les différentes tâches. La caractéristique commune pour de tels systèmes est la présence

de temps d'attente pour l'accès à ces ressources partagées.

- **Une charge** : La charge du système représente généralement le trafic en entrée, qui pourrait être l'ensemble des messages servis par un dispositif du réseau, le nombre de tâches à exécuter par un processeur. Le trafic est généralement décrit par des lois probabilistes (Poisson, Exponentielle, ...).

2.4 Rôle de l'évaluation de performances

L'évaluation des performances d'un système s'avère indispensable dès l'instant où l'on souhaite obtenir une sûreté de fonctionnement du système réel dans des cas précis, et de permettre d'atteindre le but pour lequel le système est conçu. L'étape de l'évaluation de performances peut intervenir à deux niveaux [18] :

- a) Lors de conception : Cela signifie que le système n'existe pas encore et qu'il s'agit de le créer et de le dimensionner ;
- b) Lors de l'exploitation : A ce niveau, on évalue souvent les performances d'un système dans le but de le modifier (extension) ou de l'expérimenter au delà de son point de fonctionnement normal.

2.5 Mesurer et prévoir la performance

La performance est une grandeur mesurable et prévisible. La mesure de la performance peut être réalisée sur un système en cours de fonctionnement. Elle consiste principalement à recueillir des compteurs de performances disponibles dans les systèmes d'exploitation. Ces informations doivent ensuite être interprétées pour en faire ressortir un diagnostic cohérent et signifiant.

La prévision de la performance nécessite un certain effort. Les activités qui lui sont liées sont donc réservées aux systèmes qui sont considérés critiques par leurs propriétaires[19].

La figure 2.1 [13] montre les étapes d'évaluation de performances d'un système :

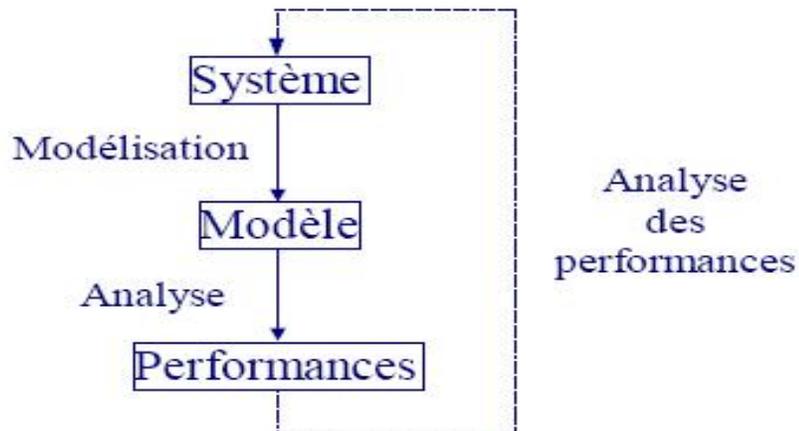


FIG. 2.1 – Étapes d'évaluation de performances d'un système

Les méthodes d'évaluation de performance peuvent être basées sur la simulation ou utiliser des méthodes analytiques.

2.5.1 Méthodes analytiques

L'intérêt des méthodes analytiques est leur grande vitesse de résolution.

En effet, lorsqu'on envisage de faire un changement sur le système, les décisions prises vont conduire à des dépenses très élevées. Les méthodes analytiques peuvent être d'une grande utilité avec leurs temps de calcul très réduits [20].

Parmi ces méthodes, on distingue [21] :

1. Modélisation par les réseaux de Petri

Les réseaux de Petri ont été introduits vers le années soixante, mais ce n'est que récemment que l'on a compris leur importance pour la description des systèmes parallèles, comme par exemple dans le schéma industriel [22] [23] [24], synchroni-

sation des protocoles de communication dans un réseau informatique [22] [23], et aux systèmes d'information. C'est une technique beaucoup plus liée à la théorie des graphes.

2. Méthode des processus stochastiques

Les méthodes probabilistes élémentaires utilisées pour les systèmes non réparables s'avèrent insuffisantes pour étudier des systèmes réparables. Pour l'analyse de ces derniers, on fait appel à une modélisation par des processus stochastiques. Ceci implique que l'on associe à un système réparable l'ensemble \mathbb{S} de tous les états dans lesquels il peut se trouver. Le choix de \mathbb{S} dépend de la structure du problème en question. Si l'on admet qu'un système comprend n éléments et qu'il existe trois états distincts pour chacun de ces éléments (en fonctionnement, en réparation et en attente d'être réparé), l'ensemble \mathbb{S} comprendrait théoriquement 3^n états. Dans la pratique, on utilise cependant souvent des ensembles d'états de taille moins considérable.

3. Modélisation par les processus de naissance et de mort

Pour de nombreux systèmes réels, l'état du système est complètement défini si l'on connaît le nombre d'éléments non défectueux à un instant donné ou, ce qui revient au même, le nombre d'éléments en panne en cet instant. Ceci est notamment le cas si toutes les durées de fonctionnement obéissent à la même distribution exponentielle et si les temps de réparation suivent également une loi exponentielle. On peut alors modéliser l'évolution du système par un processus de naissance et de mort $\{\mathbb{X}(t); t \geq 0\}$, où $\mathbb{X}(t)$ est le nombre d'éléments défectueux à l'instant t . Les systèmes réparables susceptibles d'être modélisés par un processus de naissance et de mort sont essentiellement ceux du type k parmi n , dont la durée de vie de processus de vie commune des composants suit une loi exponentielle de paramètre λ et la durée de réparation une loi exponentielle de paramètre μ .

4. Modélisation par les chaînes de Markov à temps continu

Ce type de modélisation est recommandé quand le système est toujours de type exponentiel (c'est à dire sans mémoire) mais composé d'éléments d'importance et de fiabilité différentes. De même, on peut être en présence de différents taux de réparation, comme il peut comprendre des priorités relatives. Si les durées de vie et/ou les durées de réparation n'obéissent pas à des lois exponentielles, il faut recourir à des méthodes non markoviennes [25] [26].

5. Modélisation par les systèmes de files d'attente

Le calcul de la fiabilité de certains systèmes complexes se ramène à des problèmes de phénomènes d'attente. Supposons par exemple que la capacité d'un atelier de réparation est $N \geq 1$, ce qui signifie que N éléments défectueux au maximum peuvent être remis en service à la fois. Si N est inférieur au nombre d'éléments du système (ce qui est souvent le cas), une file d'attente de composants tombés en panne peut se former. On est alors en présence d'un **phénomène d'attente** [25] [27][22][28] [29].

Les calculs des performances par la méthode analytique utilisent la théorie des files d'attente a/b/m/K/M où

a : désigne le type de la loi des arrivées.

b : celui de la loi de services [19].

m : désigne le nombre de serveur.

K : la capacité du système.

M : nombre de sources de clients.

- **Distributions inter-arrivées et service [19] :**

Pour spécifier les distributions a et b, les symboles suivants sont définis :

GI : loi générale indépendante; **G** : loi générale; H_k : loi hyper exponentielle d'ordre k; E_k : loi Erlang-k; **M** : loi exponentielle; C_k : loi de Cox-k; **D** : loi constante.

- **Analyse mathématique d'un système de files d'attente [19] :**

L'étude mathématique d'un système de files d'attente se fait généralement par l'introduction d'un processus stochastique, défini de façon approprié. On s'intéresse principalement au nombre $\mathbf{X}(t)$ de clients se trouvant dans le système à l'instant t ($t \geq 0$). En fonction des quantités qui définissent le système, on cherche à déterminer :

- Les probabilités d'états $P_n(t) = P(X(t) = n)$, qui définissent le régime transitoire du processus stochastique $\{X(t); t \geq 0\}$. Il est évident que les fonctions $P_n(t)$ dépendent de l'état initial ou de la distribution initiale du processus.

- Le régime stationnaire du processus stochastique est défini par :

$$P_n(t) = \lim_{t \rightarrow +\infty} P_n(t) = P(X(+\infty) = n) = P(X = n) \quad (n = 0, 1, 2, \dots).$$

$\{P_n\}$ est appelée distribution stationnaire du processus $\{X(t); t \geq 0\}$.

Il existe des systèmes pour lesquels l'évolution temporelle ne peut être déterminée par le processus $\{X(t); t \geq 0\}$ (M/G/1), (G/M/1).

- **Caractéristiques d'un système de files d'attente [30] :**

La théorie des systèmes d'attente a comme objectif d'en étudier les structures et de calculer les valeurs caractéristiques permettant de décrire les performances d'un tel système à partir de la distribution stationnaire du processus $\{X(t); t \geq 0\}$:

L_s : nombre moyen de clients dans le système.

L_q : nombre moyen de clients dans la file.

W_s : temps moyen de séjour d'un client dans le système.

W_q : temps moyen d'attente d'un client dans la file.

Ces valeurs permettent de juger le comportement opérationnel d'un système d'attente. Elles sont liées entre elles par les relations suivantes :

$$L_s = \lambda W_s ;$$

$$L_q = \lambda W_q;$$

$$W_s = W_q + 1/\mu;$$

$$L_s = L_q + \lambda/\mu.$$

avec :

λ : est le taux d'arrivée dans le système.

$1/\lambda$: est l'intervalle de temps moyen séparant deux arrivées consécutives.

μ : est le taux de service.

$1/\mu$: durée moyenne de service.

Les deux premières formules (L_s, L_q) sont appelées formules de Little [30].

Les thèmes majeurs traités par les méthodes analytiques sont :

○ **Evaluation de performance et optimisation :**

- Cas industriels : Dans ce cas là, les études qui en découlent ont souvent pour but d'optimiser le réseau. Les demandes industrielles les plus fréquentes consistent à trouver le réseau qui minimise les coûts, à trouver le réseau le plus rapide et établir un rapport vitesse/coût, à trouver celui qui obtient le meilleur taux de service, d'étudier les performances du réseau actuel par rapport au réseau optimal ou bien de changer son mode d'approvisionnement pour voir les effets en termes d'argent et de temps ([31] [32] [33] [34] [35]).

- Cas académique : D'autres méthodes d'évaluation de performances et d'optimisation sont développées de manière académique, comme la minimisation des coûts de stockage sous contrainte d'un taux de service minimal ([36] [37] [38] [39]).

○ **Transport :** Le problème est généralement de minimiser les coûts tout en livrant un certain nombre de clients. Les auteurs étudient la manière d'optimiser leur réseau de

distribution en modifiant l'emplacement et les caractéristiques des sites ou encore le nombre de camions utilisés et on réfléchit aux quantités à emporter sachant que les coûts ne sont pas linéaires ([40] [41] [42] [43] [44]).

- **Analyse de paramètres** : les auteurs effectuent leurs études visant à montrer quels paramètres influencent le plus leurs indicateurs de performance. Les études de paramètres peuvent être empiriques ou analytiques. Ces études montrent les paramètres les plus importants et quantifient leur effet sur le système. Cela permet ainsi de comprendre le fonctionnement du réseau et les interactions entre ses différentes composantes ([45] [46]).
- **Internet** : Les réseaux informatiques, et en particulier internet, ouvrent la voie à de nouvelles pratiques commerciales comme l'étudie Aldin [47]. De leur côté, Chiang et al. et Lee et Whang [48] étudient les effets de l'ouverture d'un réseau de vente directe par internet.
- **Modélisation** : Nous retrouvons les études dont les questions de modélisation représentaient le cœur du sujet. Les discussions peuvent être axées en deux sens : l'outil avec lequel on va modéliser le système ou alors la manière dont doit être modélisé un réseau particulier ([48] [49] [50]).

2.5.2 Simulation

On utilise des simulateurs de charge qui simulent l'accès simultané au système de plusieurs utilisateurs et recueillent des informations de performances [19].

La simulation est une technique largement utilisée dans l'évaluation de performances des systèmes informatiques et les réseaux de communication. Il s'agit d'implanter un modèle simplifié du système à l'aide d'un programme de simulation adéquat. Elle présente l'avantage par rapport aux méthodes analytiques (par calcul) de traduire le comportement

du système à évaluer d'une manière plus réaliste et représente un moyen utile pour prédire les performances d'un système [15].

- **Définition d'un système** [51]

Le système est l'interaction d'une collection d'objets dans un environnement fermé. Il est affecté par des changements dans son environnement. On parle aussi de système réel ou phénomène que l'on veut observer. On distingue dans ce système, d'une part les centres d'activités et d'autre part les objets qui circulent dans le système.

- **Définition d'un modèle** [51]

Le modèle est une représentation d'un système dans un autre environnement : mathématique, logique, langage,... La modélisation est l'étape qui consiste à passer du système à un modèle. La modélisation n'a de sens que si le comportement du modèle peut être comparé au comportement du système.

Les étapes de la simulation :

Les différentes étapes à suivre pour faire une simulation d'un système sont [52] :

1. **Formulation du modèle** : cette étape consiste à identifier et à analyser le problème, en déterminant ses composantes, leurs relations et les frontières entre le système et son environnement.
2. **Elaboration du modèle** : cette étape consiste à extraire un modèle aussi fidèle que possible du système réel.
3. **Identification du modèle et collecte de données** : la collecte de données

est indispensable pour l'estimation des paramètres du modèle. Ceci requiert une connaissance des méthodes statistiques et des tests d'hypothèses.

4. **Validation du modèle** : cette étape consiste à évaluer les performances du modèle puis de les comparer à celles du système réel.
5. **Exécution de la simulation** : pour mettre à l'épreuve le modèle, le concepteur doit effectuer plusieurs exécutions et recueillir les résultats.
6. **Analyse et interprétation des résultats** : une fois les résultats obtenus, le concepteur passe à l'analyse et à l'interprétation de ces résultats pour donner des recommandations et des propositions.
7. **Conclusion** : cette dernière étape consiste à évaluer les perspectives d'exploitation du modèle pour d'autres préoccupations.

Les techniques de la simulation

On peut distinguer deux types de simulation, la simulation par "trace" et la simulation par "événements discrets".

- **La simulation par trace**

C'est la simulation d'un modèle déterministe (pas de tirage de nombres aléatoires) qui est dirigé par une conséquence ou trace obtenue à partir de mesures effectuées sur un système existant [53].

- **La simulation à événements discrets**

La simulation à événements discrets désigne la modélisation d'un système réel tel

qu'il évolue dans le temps par une représentation dans laquelle les caractéristiques du système réel (variable) ne changent qu'en un nombre fini ou dénombrable de poids isolés dans le temps [53].

La conception d'un simulateur dans la simulation à événements discrets consiste à représenter le temps par une variable appelée " horloge de la simulation " qui est avancée au fur et à mesure que les événements sont rangés et ordonnés dans l'échéancier par ordre croissant selon leur instant d'exécution. Il existe quatre méthodes de la simulation à événements discrets.

Chaque approche est caractérisée par une méthode différente pour faire progresser le système simulé dans le temps :

- Méthode basée sur les activités " Approche Activité " [54].
- Méthode basée sur les événements " Approche Événement " [55].
- Méthode basée sur les activités et les événements " Méthode des 3 phases " [54].
- Approche par interaction des processus [55].

2.6 Avantages et inconvénients de la simulation

Comme toute approche scientifique, la simulation présente des avantages et des inconvénients [56].

2.6.1 Avantages

- Seule alternative technologique quand le système à étudier est physiquement difficile à déployer ;
- Répétitions d'expériences ;
- Permet d'adresser des systèmes très complexes ;

- Permet de répondre à des questions de type, qu'est-ce qui se passe si ... ;
- Moins d'hypothèses simplificatrices.

2.6.2 Inconvénients

Quant à ses inconvénients, on cite :

- Coûteux en terme du temps de calcul ;
- Résultats orientés au pire-cas, généralement sans validité ;
- Elle ne fournit que des estimations de ce que l'on cherche.

2.7 Comparaison des techniques d'évaluation des performances

Le tableau suivant [56] montre la comparaison entre la simulation et les méthodes analytiques :

Simulation	Méthode analytique
Résultats approximatifs	Résultats exacts si c'est possible
Appliqué aux systèmes complexes	Pas toujours
Gourmande en temps de calcul	Peut coûteuse en temps de calcul

2.8 Méthode de stabilité forte

La méthode de stabilité forte, connue également sous le nom de méthode des opérateurs de la théorie de la stabilité, est applicable à tous les modèles stochastiques de la recherche opérationnelle pouvant être régis par une chaîne de Markov. Elle a été introduite au début des années quatre vingt par Aïssani et Kartashov [57]. Elle suppose que la perturbation du noyau de transition du processus aléatoire décrivant le système étudié est petite par rapport à une certaine norme d'opérateurs. Cette condition, beaucoup plus stricte que les conditions habituelles, permet d'obtenir essentiellement de meilleures approximations pour les distributions stationnaires perturbées. De plus, sur la base de cette méthode, il

est possible d'obtenir des inégalités de stabilité avec un calcul exact des constantes. Les résultats fondamentaux de cette méthode ont fait l'objet de la publication en 1996 de la monographie de Kartashov [58].

2.8.1 Critère de stabilité forte

Considérons une chaîne de Markov homogène $X = (X_t, t \geq 0)$ à valeurs dans un espace mesurable (E, \mathcal{E}) , (où \mathcal{E} est une σ -algèbre engendrée par une partie dénombrable de E), donnée par un noyau de transition régulier $P(x, A)$, $x \in E, A \in \mathcal{E}$. Par ailleurs, on suppose que la chaîne X admet une unique mesure invariante π tel que $\pi(E) = 1$.

Notons par $m\mathcal{E}$ ($m\mathcal{E}^+$) l'espace des mesures finies (non négatives) sur \mathcal{E} et $f\mathcal{E}$ ($f\mathcal{E}^+$) l'espace des fonctions mesurables bornées (non négatives) sur E . Associons à chaque noyau de transition P , les deux applications linéaires suivantes :

$$\mathcal{L}_P : m\mathcal{E} \rightarrow m\mathcal{E} \text{ et } \mathcal{L}_P^* : f\mathcal{E} \rightarrow f\mathcal{E}$$

dont les valeurs aux points $\mu \in m\mathcal{E}$ et $f \in f\mathcal{E}$ sont respectivement :

$$\begin{aligned} \mu P(A) &= \mathcal{L}_P(\mu)(A) = \int_E \mu(dx) P(x, A), \forall A \in \mathcal{E}, \\ P f(x) &= \mathcal{L}_P^*(f)(x) = \int_E P(x, dy) f(y), \forall x \in E. \end{aligned}$$

Le produit des deux noyaux de transitions P et Q est le noyau

$$PQ(x, A) = \int_E P(x, dy) Q(y, A) \text{ pour } x \in E, A \in \mathcal{E}.$$

Le produit d'une mesure $\mu \in m\mathcal{E}$ et d'une fonction $f \in f\mathcal{E}$, noté μf , est défini par

$$\mu f = \int_E \mu(dx) f(x).$$

Egalement, $f \circ \mu$ désignera le produit direct d'une fonction mesurable $f \in f\mathcal{E}$ par une mesure $\mu \in m\mathcal{E}$:

$$(f \circ \mu)(x, A) = f(x)\mu(A), x \in E, A \in \mathcal{E}.$$

On considère, dans l'espace $m\mathcal{E}$, l'espace de Banach $\mathcal{M} = \{\mu \in m\mathcal{E} : \|\mu\| < \infty\}$ de norme $\|\cdot\|$ compatible avec l'ordre structurel dans $m\mathcal{E}$, c'est à dire :

1. $\|\mu_1\| \leq \|\mu_1 + \mu_2\|$ pour $\mu_1, \mu_2 \in \mathcal{M}^+$,
2. $\|\mu_1\| \leq \|\mu_1 - \mu_2\|$ pour $\mu_1, \mu_2 \in \mathcal{M}^+$ et $\mu_1 \perp \mu_2$,
3. $|\mu|(E) \leq k\|\mu\|$ pour $\mu \in \mathcal{M}$.

où $|\mu|$ est la variation de la mesure μ , k est une constante positive finie et $\mathcal{M}^+ = \mathcal{M} \cap (m\mathcal{E}^+)$.

La norme induite sur $f\mathcal{E}$ est donnée par

$$\|f\| = \sup(|\mu f|, \|\mu\| \leq 1).$$

La norme induite sur l'espace \mathcal{B} des opérateurs bornés est donnée par :

$$\|P\| = \sup(|\mu P|, \|\mu\| \leq 1).$$

Supposons que l'opérateur $P : \mathcal{M} \rightarrow \mathcal{M}$ est borné, i.e :

$$\mathcal{M}P \subset \mathcal{M}, \|P\| < \infty,$$

où $\mathcal{M}P = \{\mu : \mu = \mu_1 P, \mu_1 \in \mathcal{M}\}$.

Notons par $\Pi = \mathbb{I} \circ \pi$ le projecteur stationnaire du noyau P , où \mathbb{I} est la fonction identiquement égale à 1, et considérons la moyenne de Césaro définie par : $P^{(t)} = t^{-1} \sum_{s=0}^{t-1} P^s$, $t \in \mathbf{N}^*$.

2.9 Troncature

2.9.1 Principe de la troncature

La troncature est énoncé comme suit : "Pour résoudre un système infini d'équations linéaires à un nombre infini d'inconnues, on limite le système aux n premières équations, et on néglige le reste des équations" [59].

2.9.2 Différents techniques de la troncature des chaînes de Markov

Soit $P = P(i, j) \quad i, j \geq 1$, une matrice stochastique infinie, irréductible et récurrente positive, supposons qu'elle admet une distribution stationnaire unique $\pi_n = \pi(j) \quad j \geq 1$, le calcul de cette distribution étant en général difficile. Pour cela, une solution consiste à approcher P par une matrice stochastique finie P_n .

"Le Coin Nord-Ouest" d'ordre n de la matrice P :

$$T_n = P = P(i, j) \quad n \geq i, j \geq 1$$

P étant irréductible, il existe au moins une ligne i pour laquelle

$$\sum_{j=1}^n P(i, j) < 1$$

alors, la matrice tronquée T_n n'est pas stochastique.

À partir de T_n , on construit une matrice stochastique $P_n(i, j) \quad n \geq i, j \geq 1$ vérifiant $P_n \geq T_n$ c'est à dire $P_n(i, j) > P(i, j)$ pour $n \geq i, j \geq 1$, cela peut se faire de plusieurs façons, citons les principales :

1. Augmentation linéaire

La masse de probabilité perdue lors de la troncature de P est redistribuée sur les colonnes de P_n , plus précisément, soit

$$A_n = \alpha_n(i, j) \quad n \geq i, j \geq 1$$

une matrice stochastique quelconque, on pose pour $n \geq i, j \geq 1$

$$P_n(i, j) = P(i, j) + \alpha_n(i, j) \sum_{k>n} P(i, k)$$

. En particulier, on obtient :

- L'augmentation de la première colonne, travaux de Kalashnikov et Rachev [60], seulement si $\alpha_n(i, 1) = 1$ pour $n \geq i, j \geq 1$.
 - L'augmentation de la dernière colonne [61], seulement si $\alpha_n(i, n) = 1$ pour $n \geq i, j \geq 1$.
 - L'augmentation uniforme, seulement si $\alpha_n(i, n) = n^{-1}$ pour $n \geq i, j \geq 1$.
 - On peut aussi prendre pour A , une matrice dont toutes les lignes sont identiques, c'est un cas considéré par Gibson et Seneta [62]
 - Encore, on peut comme Van Dijk [60], choisir A booléenne.
2. On pose $s(i, n) = \sum_{j=1}^n P(i, j)$, on choisit alors pour $n \geq i, j \geq 1$:

$$P_n(i, j) = \frac{P(i, j)}{s(i, n)}$$

En prenant n assez grand afin que $s(i, n) > 0$.

2.10 Conclusion

Au cours de ce chapitre, nous avons introduit les notions d'évaluation de performances. Nous avons présenté également les méthodes d'évaluation des performances telles que la simulation et les méthodes analytiques qui nous serviront par la suite à modéliser le système de découverte et la composition des Web services.

Chapitre 3

Travaux antérieurs d'évaluation de performances des Web services

3.1 Introduction

Il existe plusieurs travaux dans le domaine du développement des applications et des services via Internet. Il est important de définir des mesures quantitatives et des modèles permettant d'évaluer, dès la phase de conception, la qualité de service. L'évaluation de la qualité de service devient un aspect crucial pour les développeurs, les fournisseurs et les utilisateurs de ces applications, pour avoir la *sûreté* de fonctionnement.

3.2 Evaluation des performances par une modélisation analytique

3.2.1 Indisponibilité des Web services [63]

Pour l'évaluation des performances lors de l'indisponibilité des Web services, les auteurs : Martinello, Kaâniche et Kanoun (2005)[63] ont proposé deux modèles : Un modèle de disponibilité et un modèle de performance.

Le modèle de disponibilité décrit les états du système résultant de la survenance de l'échec et des événements de réparation dans le système et évalue les mesures probabilistes associées à ces états.

Le modèle de performance décrit l'arrivée des requêtes et le processus de services qui

permet d'évaluer les mesures de performances, conditionnées sur les états du modèle de disponibilité, voir la figure 3.2 [63].

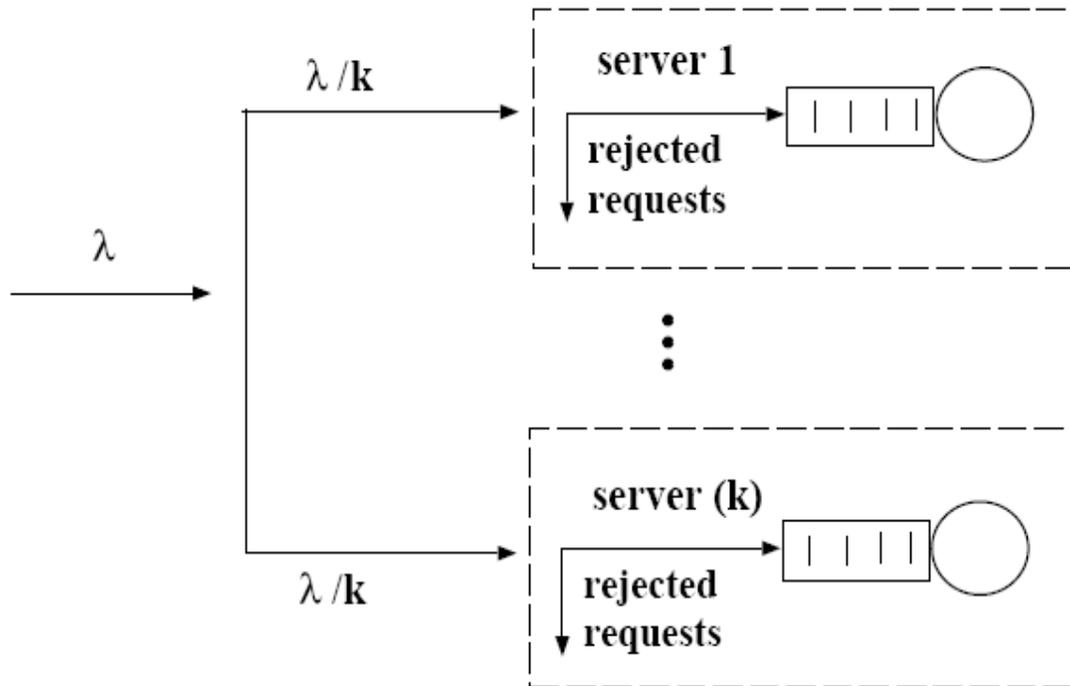


FIG. 3.1 – Modèle de performance.

Les auteurs ont considéré un système de groupe de Web services simple, composé de N nœuds de serveurs et un gestionnaire de charge qui envoie des requêtes entre les serveurs d'une façon Round-Robin.

Le comportement des performances de chaque serveur est modélisé par un système de files d'attente $M/M/1/b$.

1. Calcul des performances

Notons par UA , l'indisponibilité du Web service qui est défini par la probabilité que les requêtes reçues ne sont pas traitées correctement. Cela signifie que les demandes peuvent être perdues :

- A l'arrivée au système en raison de la saturation de la file d'attente,
- ou pendant qu'ils sont en cours de traitement ou en attente de services en raison d'une défaillance d'un nœud du serveur.

Afin d'évaluer l'UA, les auteurs ont calculé :

- La probabilité de perte des requêtes due à la saturation du tampon.
 - La probabilité de perte des requêtes due à une défaillance du nœud du serveur.
 - La probabilité de perte des requêtes pendant le temps de détection des pannes.
- Pour assurer la transmission fiable des données, diverses applications numériques ont été effectuées.

2. Avantages

- Les auteurs ont réalisé cette étude d'évaluation des performances pour les Web services simples et lors de l'indisponibilité des Web services simples.
- La résolution de ce modèle est réalisée par une méthode analytique.

3. Inconvénient

- Ils n'ont pas étudié le cas de composition des Web services.

3.2.2 Modélisation hiérarchique de la disponibilité perçue par l'utilisateur [64]

Pour l'évaluation des performances lors de la disponibilité des Web services, l'auteur Martinello (2005)[64] a proposé deux modèles : un modèle de disponibilité et un modèle de performance.

Le modèle de disponibilité permet de décrire les états du système résultant des événements de défaillance et de réparation et permet d'évaluer les probabilités associées à ces états, voir la figure 3.3.

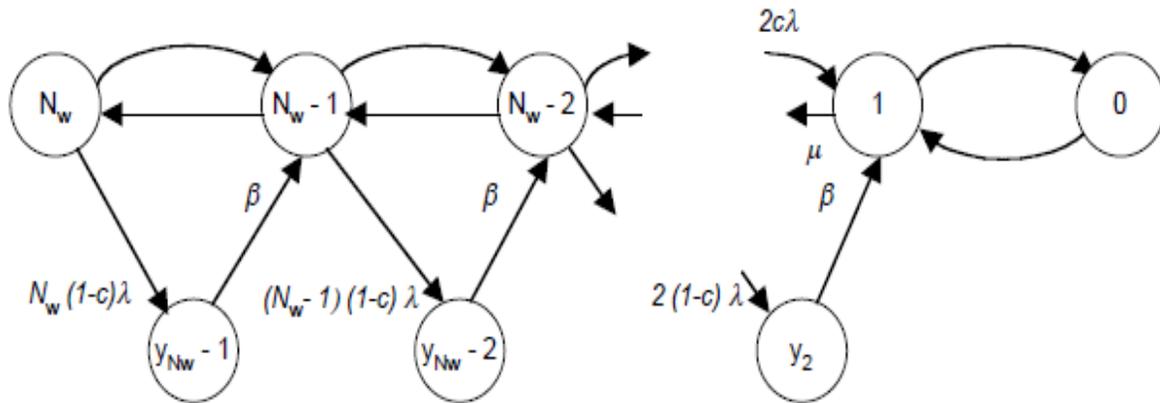


FIG. 3.2 – Chaîne de Markov avec N_w serveurs Web.

Le modèle de performance prend en compte l'arrivée et le traitement des requêtes en évaluant des mesures de performance conditionnées sur l'état du système déterminé par le modèle de disponibilité.

Pour évaluer la probabilité de perte des requêtes Web due à un dépassement de capacité des serveurs, les auteurs ont supposé que l'architecture considérée peut être décrite par un modèle de performance correspondant à une file d'attente de type $M/M/i/K$, où i dénote le nombre de serveurs opérationnels et K est la taille du tampon.

1. Calcul des performances

L'auteur a considéré deux cas d'indisponibilité du Web service :

- les défaillances matérielles et logicielles ;
- les dégradations de performances qui se traduisent par le rejet des requêtes reçues quand la capacité des serveurs est atteinte.

Afin d'évaluer la disponibilité d'un Web service, l'auteur a calculé la probabilité de perte des requêtes due à un dépassement de capacité des serveurs.

Pour assurer la transmission fiable des données, les auteurs ont pris comme exemple, le cas d'une agence de voyage mise en oeuvre sur le Web.

2. Avantages

- L'auteur a réalisé cette étude d'évaluation des performances pour les Web services simples et lors de la disponibilité des Web services simples.
- La résolution de ce modèle est réalisé par une méthode analytique.

3. Inconvénient

- Il n'a pas étudié le cas de composition des Web services.

3.3 Evaluation des performances par la simulation

3.3.1 Composition des Web services avec un support du réseau actif [65]

Pour l'évaluation des performances d'un système de composition des Web services, les auteurs M. Younas, I. Awan et D. Duce (2006)[65] ont proposé un modèle de performance.

Le modèle de performance prend en compte l'arrivée et le traitement des requêtes en évaluant des mesures de performance.

Les auteurs ont supposé que le système considéré est décrit par un modèle de performance pur correspondant à une file d'attente de type $M/M/1/N$, où N est la taille du tampon.

1. Calcul des performances

Les auteurs ont utilisé des réseaux de files d'attente pour modéliser plusieurs scénarios afin de simuler le processus des Web services composites et les messages de communications pendant le processus de composition des Web services.

Le système de files d'attente $M/M/1/N$ a été utilisé pour modéliser les stations de files d'attente dans le réseau.

Pour assurer la transmission fiable des données, une simulation a été mise en application. Diverses expériences ont été effectuées pour calculer les performances suivantes :

- Le temps moyen de réponse des requêtes ;
- La propagation de retard ;
- Le coût de traitement.

2. Avantages

- Les auteurs ont réalisé cette étude d'évaluation des performances pour les Web services composites et lors de la disponibilité des Web services ;
- La résolution de ce modèle est réalisé par la simulation.

3. Inconvénients

- Ils n'ont pas étudié le cas de Web services simples ;
- Ils n'ont pas utilisé la méthode analytique.

3.3.2 Evaluation de performances des Web Services distribués par le Bluetooth [66]

Les dispositifs mobiles devraient permettre à des utilisateurs d'exploiter des services à n'importe quel moment, sans aucune restriction d'endroit et d'une manière transparente. La technologie de Bluetooth réalise ce dispositif en fournissant des services à un bas coût et avec une basse puissance d'énergie.

Dans ce travail, les auteurs Auletta (2005) [66] a évalué les performances d'un framework permettant à une application Java d'accéder d'une manière directe à un Web service à partir d'un dispositif mobile en utilisant une connection Bluetooth.

1. Calcul des performances

L'auteur a réalisé quelques expériences pour évaluer les performances de ce système.

Dans ces expériences, l'auteur a évalué :

- Le retard de détection des Web services simples.

- Le temps de réponse.

2. Avantages

- L'auteur a réalisé cette étude d'évaluation des performances pour les Web services simples et lors de la disponibilité des Web services simples.
- La résolution de ce modèle est réalisé par la simulation.

3. Inconvénients

- Il n'a pas étudié le cas de Web services composites.
- Il n'a pas utilisé la méthode analytique.

3.4 Comparaison des différents travaux d'évaluation de performances des Web services

Dans cette section, nous présentons la comparaison entre les différents travaux sur l'évaluation de performances des Web services discutés précédemment.

Nous notons par :

- 1 : Le travail : Indisponibilité des Web services [63].
- 2 : Le travail : Modélisation hiérarchique de la disponibilité perçue par l'utilisateur [64].
- 3 : Le travail : Composition des Web services avec un support du réseau active [65].
- 4 : Le travail : Evaluation de performances des Web Services distribués par le Bluetooth [66].

Cette comparaison est établie selon les différents critères suivants :

- Méthodes de performance utilisées. Nous avons choisi deux méthodes :

1. La méthode analytique qui se base sur les files d'attente. Ces dernières prennent en compte l'arrivée et le traitement des requêtes en évaluant des mesures de performance voulues. Les chaînes de Markov permettent de décrire les états du système résultant des événements de défaillance, de réparation et d'évaluer les probabilités associées à ces états.
2. La simulation peut être réalisée par les applications numériques ou par plusieurs expériences pour un même système.

- Paramètres de performances calculés :

Dans ce critère, nous avons considéré :

1. Probabilité de perte des requêtes due à la saturation du tampon (atteindre la capacité de la file) pendant le temps de détection de pannes, occupation du serveur ou défaillance (la panne) du serveur ;
2. Etat du serveur où nous avons considéré la disponibilité ou l'indisponibilité des serveurs ;
3. Temps moyen de réponse (le temps d'attente + le temps de service) des requêtes et de propagation de retard (accumulation du temps de retard) de détection des Web services ;
4. Nombre moyen de requêtes satisfaites par le serveur.

Les applications sont réalisées pour les Web services simples ou composites.

Le tableau 3.1 illustre la comparaison des différents travaux d'évaluation de performances des Web services discutés précédemment dans ce chapitre, où "×" veut dire que

le travail $i, i = \overline{1,4}$ a pris en considération cette méthode utilisée et ce paramètre de performance. Les cases vides dans les méthodes utilisées, veulent dire que le travail $i, i = \overline{1,4}$ ne prend pas en considération cette méthode et les cases vides dans les paramètres de performances, veulent dire que le travail $i, i = \overline{1,4}$ ne prend pas en considération cette performances.

TAB. 3.1 – Tableau comparatif des différents travaux d'évaluation de performances des Web services.

			1	2	3	4
Les méthodes utilisées	Méthodes analytiques	File d'attente	X	X		
		Chaine de Markov	X	X		
	Simulation	Application numérique	X	X		
		Simulation	X	X	X	X
Paramètres de performances	Probabilité de perte des requêtes	Saturation de tampon	X			
		Défaillance du serveur	X			
		Pendant le temps de détection de pannes	X			
		Occupation du serveur		X		
	Etat du serveur	Disponibilité		X	X	X
		Indisponibilité	X			
	Temps moyen de	Réponse des requêtes			X	X
		Propagation de retard de détection des Web services			X	X
	Nombre moyen de	Requêtes dans le système				
	Application utilisée	Web services	Simple	X	X	
Composites					X	

3.5 Conclusion

Dans ce chapitre, nous avons présenté quelques travaux d'évaluation de performances des Web services. À partir de ces travaux, on constate que les auteurs n'ont pas considéré les Web services simples et Web services composites simultanément. C'est pourquoi, dans notre travail, nous proposons une modélisation pour la découverte et la composition des Web services par un modèle de files d'attente à deux stations en tandem, et avec deux taux d'arrivées, l'arrivée à la première station avec un taux λ_1 et l'arrivée à la deuxième station avec un taux λ_2 . À la fin du service du client dans la première station, s'il est satisfait il quitte le système avec la probabilité $1 - p$. Sinon, il passera à la deuxième file d'attente avec la probabilité p , pour la recherche d'une réponse dans les Web services composites. À la fin de ce service, s'il est encore non satisfait, il va rejoindre la première file d'attente avec la probabilité $1 - q$.

La méthode analytique n'est pas toujours facile à mettre en œuvre pour les systèmes qui correspondent à notre travail, à savoir la découverte et la composition des Web services. D'où l'utilisation des méthodes d'approximation. De plus, nous réalisons une application numérique.

Chapitre 4

Mesure de performances d'un système de découverte et composition des Web services

4.1 Introduction

L'objet de ce présent chapitre est d'évaluer les performances du modèle d'un système de découverte et de composition des Web services. Pour cela, nous avons construit, à partir des résultats de la méthode de stabilité forte, un simulateur permettant d'estimer l'erreur due à l'approximation et d'avoir l'influence des deux taux des arrivées des clients sur le nombre moyen de clients dans le système et la durée moyenne de séjour.

4.2 Description du modèle

À partir du tableau comparatif des différents travaux d'évaluation de performances des Web services, on constate que les auteurs n'ont pas considéré les Web services simples et Web services composites simultanément. D'où, dans notre travail, on propose une modélisation pour la découverte et la composition des Web services par un système de files d'attente à deux stations. La première station a une file d'attente infinie et la deuxième station est avec une file d'attente limitée (N), car on a la capacité de la base de donnée qui est limitée.

La figure 4.1 illustre le modèle du système de découverte et de composition des Web

services.

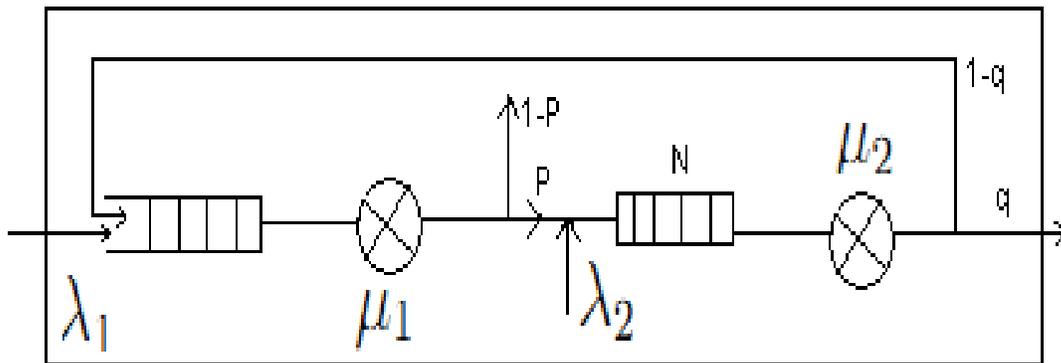


FIG. 4.1 – Modèle réel.

Dans ce modèle, on a deux types d'arrivées :

- ✓ Arrivées des requêtes à la première station : Une inter-arrivée est générée selon la loi exponentielle.

L'arrivée ainsi obtenue est affectée une par une au premier serveur.

Le temps de ce service est alors généré selon la loi exponentielle.

Si la requête n'est pas satisfaite, elle passera à la deuxième file d'attente avec la probabilité p , pour la recherche d'une réponse dans les Web services composites.

À la fin de ce service, si elle est encore non satisfaite, elle va rejoindre la première file d'attente avec la probabilité $1 - q$.

Le temps de ce service est alors généré selon la loi exponentielle.

- ✓ Arrivées des requêtes à la deuxième station : Une inter-arrivée est générée selon la loi exponentielle.

La matrice de transition du système :

Soit : $M = \lambda_1 + \lambda_2 + \mu_1 + \mu_2$.

$$\left\{ \begin{array}{l} P_{(i,j) \rightarrow (i+1,j)} = \frac{\lambda_1}{M} \\ P_{(i,j) \rightarrow (i,j+1)} = \frac{\lambda_2}{M} \\ P_{(i,j) \rightarrow (i-1,j)} = \frac{(1-p)\mu_1}{M} \\ P_{(i,j) \rightarrow (i-1,j+1)} = \frac{p\mu_1}{M} \\ P_{(i,j) \rightarrow (i+1,j-1)} = \frac{(1-q)\mu_2}{M} \\ P_{(i,j) \rightarrow (i,j-1)} = \frac{q\mu_2}{M} \\ P_{(i,j) \rightarrow (i,j)} = 1 - \frac{\lambda_1 + \lambda_2 + \mu_1 + \mu_2}{M} \end{array} \right. \quad (4.1)$$

4.3 Description de la méthode de résolution du modèle

L'évaluation des performances se base sur le calcul des probabilités stationnaires, et comme le modèle n'est pas sous forme produit, on a recours aux méthodes d'approximations.

Grassmann et Tavakoli (1980) [67] ont présenté ce modèle mais le serveur de leur modèle était mobile. C'est-à-dire que si le serveur de l'une des stations est inoccupé, il peut aider le serveur de l'autre station, en augmentant le taux de son service.

Le modèle a été résolu par la méthode spectrale. Les auteurs ont présenté la matrice infinitésimale (génératrice) Q . Le vecteur des probabilités stationnaires π est obtenu en résolvant le système d'équation $\pi Q = 0$, telle que la somme des probabilités stationnaires est égale à 1.

$$0 = \pi_0 Q_{00} + \pi_1 Q_{-1}$$

$$0 = \pi_0 Q_{01} + \pi_1 Q_0 + \pi_2 Q_{-1}$$

$$0 = \pi_{n-1} Q_1 + \pi_n Q_0 + \pi_{n+1} Q_{-1}, n > 1 \dots (a)$$

L'équation (a) est une équation différentielle, où sa solution est similaire à la solution d'une équation différentielle standard, telle que $\pi_n = g x^{n-1}, n \geq 1 \dots (b)$

où : $g = [g_0, g_1, \dots, g_N] \neq [0, 0, \dots, 0]$.

En remplaçant (b) dans (a), les auteurs ont trouvé :

$$0 = gx^{n-2}Q_1 + gx^{n-1}Q_0 + gx^2Q_{-1} \cdots (c)$$

Ils ont supposé que $Q(x) = Q_1 + Q_0x + Q_{-1}x^2$.

D'où (c) devient $0 = gQ(x)$ qui est un problème connu sous le nom de problème de généralisation de valeurs propres [68] et en utilisant un algorithme, ils ont pu avoir les vecteurs propres ($g^{(i)}$).

Une fois que $g^{(i)}$ et x_i ont été obtenus, les auteurs ont calculé $\pi_n = \sum_{i=0}^N c_i g^{(i)} x_i^n$, tel que :
 x_i : Valeurs propres.

c_i : Coefficient la valeur propre dominante.

Enfin, ils ont calculé l'erreur d'approximation.

Dans notre travail, on utilisera la méthode des troncatures qui est énoncé comme suit :
 "Pour résoudre un système infini d'équations linéaires à un nombre infini d'inconnues, on limite le système aux n premières équations, et on néglige le reste des équations" [59].

Cette méthode a pour but de limiter la capacité de la première file d'attente à Q .

La méthode de stabilité forte est utilisée pour l'estimation de l'erreur entre les caractéristiques du modèle réel et le modèle tronqué.

Le modèle tronqué est présenté dans la figure 4.2.

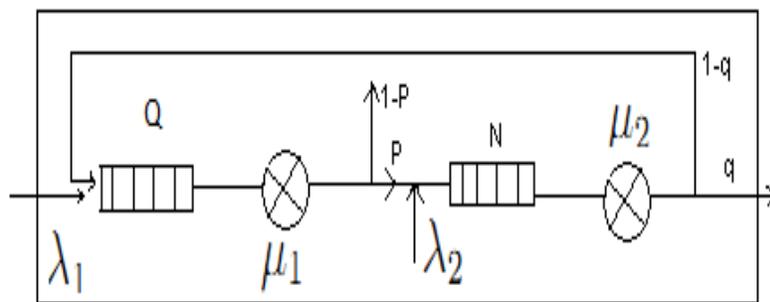


FIG. 4.2 – Modèle idéal.

La matrice de modèle tronqué est donnée par :

$$\begin{cases} \bar{P}_{(Q,j) \rightarrow (0,j)} = 1 - \frac{\lambda_2 + \mu_1 + q\mu_2}{M} & , \text{ Si } i \geq Q \\ \bar{P}_{(i,j) \rightarrow (m,n)} = P_{(i,j) \rightarrow (m,n)} & , \text{ Si } i < Q \end{cases} \quad (4.2)$$

Elle est basée sur l'augmentation selon la première colonne [59].

Pour le modèle définit en (4.2), on introduit les conditions suivantes :

$$\mu_1 > \lambda_1 \cdots (A)$$

$$\mu_2 > \lambda_2 \cdots (B)$$

Lemme 1. Sous les conditions (A) et (B), il existe $\rho(\alpha, \beta)$ telle que : $\rho(\alpha, \beta) < 1$. Avec, $\alpha \in]1, a[$

$\beta \in]1, b[$, tels que :

$$a = \text{Min} \left\{ 1 + \frac{\beta\lambda_2 + (\beta-1)\mu_2 - \beta^2\lambda_2}{\beta\lambda_1 + (1-q)\mu_2}; \frac{\mu_1}{\lambda_1}; \frac{\lambda_2 + \mu_1 + \mu_2 - \beta\lambda_2 - \frac{1}{\beta}\mu_2}{\alpha\lambda_2 + p\mu_1}; 1 + \frac{\mu_2 + (1-p)\mu_1 - \frac{1}{\beta}\mu_2}{\lambda_1 + \frac{1}{\beta}(1-q)\mu_2} \right\}$$

$$b = \text{Min} \left\{ \frac{\mu_2}{\lambda_2}; 1 + \frac{\alpha\lambda_1 + \alpha\mu_1 - \mu_1 - \alpha^2\lambda_1}{\alpha\lambda_2 + p\mu_1}; 1 + \frac{\lambda_2 + \mu_1 + \mu_2 + \sqrt{\lambda_2^2 + \mu_1^2 + \mu_2^2 + 2\mu_1\mu_2 + 2\mu_1\lambda_2 - 2\mu_2\lambda_2}}{2\lambda_2}; 1 + \frac{(1-q)\mu_2 + (1-p)\mu_1}{\mu_2} \right\}$$

Preuve. Nous choisissons la fonction mesurable

$$h(i, j) = \mathbf{1}_{\{i=j=0\}} = \begin{cases} 1 & , \text{ Si } i = j = 0 \\ 0 & , \text{ Sinon} \end{cases} \quad (4.3)$$

et soit la mesure $\sigma_{i,j}$ définit par

$$\sigma_{i,j} = P_{(0,0) \rightarrow (i,j)}$$

On a

$$\pi h = \pi(0, 0) > 0$$

$$\sigma \mathbf{1} = \sum_{i=0}^Q \sum_{j=0}^N P_{(0,0) \rightarrow (i,j)} = 1, \text{ où } \mathbf{1} \text{ est la fonction identité.}$$

$$\sigma h = P_{(0,0) \rightarrow (0,0)} = \frac{\mu_1 + \mu_2}{M} > 0$$

Et soit la matrice auxiliaire T :

$$T_{(i,j) \rightarrow (m,n)} = \bar{P}_{(i,j) \rightarrow (m,n)} - h(i,j) \cdot \sigma_{m,n}.$$

$$T_{(i,j) \rightarrow (m,n)} = \begin{cases} 0 & , \text{ Si } i = j = 0 \\ \bar{P}_{(i,j) \rightarrow (m,n)} & , \text{ Sinon} \end{cases} \quad (4.4)$$

D'où T est une matrice non négative.

Soit la fonction test $v(k, l)$:

$$v(k, l) = \alpha^k \beta^l \text{ avec : } \alpha > 1, \beta > 1$$

Montrons qu'il existe $\rho < 1$ tel que $Tv(k, l) \leq \rho v(k, l)$. En effet, on a :

$$Tv(i, j) = \sum_{m=0}^Q \sum_{n=0}^N v(m, n) \cdot T_{(i,j) \rightarrow (m,n)}$$

• Si $i = 0$

◦ Si $j = 0$

$$Tv(0, 0) = 0$$

◦ Si $0 < j < N$

$$T_{(0,j) \rightarrow (m,n)} = \begin{cases} \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-q)\mu_2}{M} & , \text{ Si } m = i + 1, n = j - 1 \\ 1 - \frac{\lambda_1 + \lambda_2 + \mu_2}{M} & , \text{ Si } m = i, n = j \end{cases} \quad (4.5)$$

$$Tv(0, j) = \alpha^i \beta^{j-1} \frac{q\mu_2}{M} + \alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^i \beta^{j+1} \frac{\lambda_2}{M} + \alpha^{i+1} \beta^{j-1} \frac{(1-q)\mu_2}{M} + \alpha^i \beta^j \frac{\mu_1}{M}$$

$$Tv(0, j) = \alpha^0 \beta^j \left[\frac{1}{\beta} \frac{q\mu_2}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{\mu_1}{M} \right]$$

$$\rho_1 = \frac{1}{\beta} \frac{q\mu_2}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{\mu_1}{M} \quad (4.6)$$

○ Si $j = N$

$$T_{(0,j) \rightarrow (m,n)} = \begin{cases} \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{(1-q)\mu_2}{M} & , \text{ Si } m = i + 1, n = j - 1 \\ 1 - \frac{\lambda_1 + \mu_2}{M} & , \text{ Si } m = i, n = j \end{cases} \quad (4.7)$$

$$Tv(0, j) = \alpha^i \beta^{j-1} \frac{q\mu_2}{M} + \alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^{i+1} \beta^{j-1} \frac{(1-q)\mu_2}{M} + \alpha^i \beta^j \frac{\lambda_2 + \mu_1}{M}$$

$$Tv(0, j) = \alpha^0 \beta^j \left[\frac{1}{\beta} \frac{q\mu_2}{M} + \alpha \frac{\lambda_1}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{\lambda_2 + \mu_1}{M} \right]$$

$$\rho_2 = \frac{1}{\beta} \frac{q\mu_2}{M} + \alpha \frac{\lambda_1}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{\lambda_2 + \mu_1}{M} \quad (4.8)$$

● Si $0 < i < N$

○ Si $j = 0$

$$T_{(i,0) \rightarrow (m,n)} = \begin{cases} \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ 1 - \frac{\lambda_1 + \lambda_2 + \mu_1}{M} & , \text{ Si } m = i, n = j \end{cases} \quad (4.9)$$

$$Tv(i, 0) = \alpha^{i-1} \beta^j \frac{(1-p)\mu_1}{M} + \alpha^{i-1} \beta^{j+1} \frac{p\mu_1}{M} + \alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^i \beta^{j+1} \frac{\lambda_2}{M} + \alpha^i \beta^j \frac{\mu_2}{M}$$

$$Tv(i, 0) = \alpha^i \beta^j \left[\frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{\mu_2}{M} \right]$$

$$\rho_3 = \frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{\mu_2}{M} \quad (4.10)$$

○ Si $0 < j < N$

$$T_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ \frac{(1-q)\mu_2}{M} & , \text{ Si } m = i + 1, n = j - 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 1 - \frac{\lambda_1 + \lambda_2 + \mu_1 + \mu_2}{M} & , \text{ Si } m = i, n = j \end{cases} \quad (4.11)$$

$$Tv(i, j) = \alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^i \beta^{j+1} \frac{\lambda_2}{M} + \alpha^{i-1} \beta^j \frac{(1-p)\mu_1}{M} + \alpha^{i-1} \beta^{j+1} \frac{p\mu_1}{M} + \alpha^{i+1} \beta^{j-1} \frac{(1-q)\mu_2}{M} + \alpha^i \beta^{j-1} \frac{q\mu_2}{M}$$

$$Tv(i, j) = \alpha^i \beta^j \left[\frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} \right]$$

$$\rho_4 = \frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} \quad (4.12)$$

○ Si $j = N$

$$T_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{(1-q)\mu_2}{M} & , \text{ Si } m = i + 1, n = j - 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 1 - \frac{\lambda_1 + (1-p)\mu_1 + \mu_2}{M} & , \text{ Si } m = i, n = j \end{cases} \quad (4.13)$$

$$Tv(i, j) = \alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^{i-1} \beta^j \frac{(1-p)\mu_1}{M} + \alpha^{i+1} \beta^{j-1} \frac{(1-q)\mu_2}{M} + \alpha^i \beta^{j-1} \frac{q\mu_2}{M} + \alpha^i \beta^j \frac{\lambda_2 + p\mu_1}{M}$$

$$Tv(i, j) = \alpha^i \beta^j \left[\frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \alpha \frac{\lambda_1}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} + \frac{\lambda_2 + p\mu_1}{M} \right]$$

$$\rho_5 = \frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \alpha \frac{\lambda_1}{M} + \frac{\alpha}{\beta} \frac{(1-q)\mu_2}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} + \frac{\lambda_2 + p\mu_1}{M} \quad (4.14)$$

● Si $i = Q$

○ Si $j = 0$

$$T_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ 1 - \frac{\lambda_2 + \mu_1}{M} & , \text{ Si } m = 0, n = j \end{cases} \quad (4.15)$$

$$Tv(i, j) = \alpha^i \beta^{j+1} \frac{\lambda_2}{M} + \alpha^{i-1} \beta^j \frac{(1-p)\mu_1}{M} + \alpha^{i-1} \beta^{j+1} \frac{p\mu_1}{M} + \alpha^0 \beta^j \frac{\lambda_1 + \mu_2}{M}$$

$$Tv(i, j) = \alpha^i \beta^j \left[\frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{1}{\alpha^Q} \frac{\lambda_1 + \mu_2}{M} \right]$$

$$\rho_6 = \frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{1}{\alpha^Q} \frac{\lambda_1 + \mu_2}{M} \quad (4.16)$$

◦ Si $0 < j < N$

$$T_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 1 - \frac{\lambda_2 + \mu_1 + q\mu_2}{M} & , \text{ Si } m = 0, n = j \end{cases} \quad (4.17)$$

$$Tv(i, j) = \alpha^i \beta^{j+1} \frac{\lambda_2}{M} + \alpha^{i-1} \beta^j \frac{(1-p)\mu_1}{M} + \alpha^{i-1} \beta^{j+1} \frac{p\mu_1}{M} + \alpha^i \beta^{j-1} \frac{q\mu_2}{M} + \alpha^0 \beta^j \frac{\lambda_1 + (1-q)\mu_2}{M}$$

$$Tv(i, j) = \alpha^i \beta^j \left[\frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} + \frac{1}{\alpha^Q} \frac{\lambda_1 + (1-q)\mu_2}{M} \right]$$

$$\rho_7 = \frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \beta \frac{\lambda_2}{M} + \frac{\beta}{\alpha} \frac{p\mu_1}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} + \frac{1}{\alpha^Q} \frac{\lambda_1 + (1-q)\mu_2}{M} \quad (4.18)$$

◦ Si $j = N$

$$T_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 1 - \frac{(1-p)\mu_1 + q\mu_2}{M} & , \text{ Si } m = 0, n = j \end{cases} \quad (4.19)$$

$$Tv(i, j) = \alpha^{i-1} \beta^j \frac{(1-p)\mu_1}{M} + \alpha^i \beta^{j-1} \frac{q\mu_2}{M} + \alpha^0 \beta^j \frac{\lambda_1 + \lambda_2 + p\mu_1 + (1-q)\mu_2}{M}$$

$$Tv(i, j) = \alpha^i \beta^j \left[\frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} + \frac{1}{\alpha^Q} \frac{\lambda_1 + \lambda_2 + p\mu_1 + (1-q)\mu_2}{M} \right]$$

$$\rho_8 = \frac{1}{\alpha} \frac{(1-p)\mu_1}{M} + \frac{1}{\beta} \frac{q\mu_2}{M} + \frac{1}{\alpha^Q} \frac{\lambda_1 + \lambda_2 + p\mu_1 + (1-q)\mu_2}{M} \quad (4.20)$$

$$\rho = \max(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6, \rho_7, \rho_8)$$

Où : $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6, \rho_7, \rho_8$ sont présentés de cet ordre dans les équations :

(4.6), (4.8), (4.10), (4.12), (4.14), (4.16), (4.18), (4.20).

$$\text{Si } \alpha \in]1; 1 + \frac{\beta\lambda_2 + (\beta-1)\mu_2 - \beta^2\lambda_2}{\beta\lambda_1 + (1-q)\mu_2} [\text{ et } \beta \in]1; \frac{\mu_2}{\lambda_2} [$$

Alors $\rho_1 < 1$

$$\text{Si } \alpha \in]1; \frac{\mu_1}{\lambda_1} [\text{ et } \beta \in]1; 1 + \frac{\alpha\lambda_1 + \alpha\mu_1 - \mu_1 - \alpha^2\lambda_1}{\alpha\lambda_2 + p\mu_1} [$$

Alors $\rho_3 < 1$

$$\text{Si } \alpha \in]1; \frac{\lambda_2 + \mu_1 + \mu_2 - \beta \lambda_2 - \frac{1}{\beta} \mu_2}{\alpha \lambda_2 + p \mu_1} [\text{ et } \beta \in]1; 1 + \frac{\lambda_2 + \mu_1 + \mu_2 + \sqrt{\lambda_2^2 + \mu_1^2 + \mu_2^2 + 2\mu_1 \mu_2 + 2\mu_1 \lambda_2 - 2\mu_2 \lambda_2}}{2\lambda_2} [$$

Alors $\rho_4 < 1$

$$\text{Si } \alpha \in]1; 1 + \frac{\mu_2 + (1-p)\mu_1 - \frac{1}{\beta} \mu_2}{\lambda_1 + \frac{1}{\beta}(1-q)\mu_2} [\text{ et } \beta \in]1; 1 + \frac{(1-q)\mu_2 + (1-p)\mu_1}{\mu_2} [$$

Alors $\rho_5 < 1$

On pose :

$$a = \text{Min} \left\{ 1 + \frac{\beta \lambda_2 + (\beta-1)\mu_2 - \beta^2 \lambda_2}{\beta \lambda_1 + (1-q)\mu_2}; \frac{\mu_1}{\lambda_1}; \frac{\lambda_2 + \mu_1 + \mu_2 - \beta \lambda_2 - \frac{1}{\beta} \mu_2}{\alpha \lambda_2 + p \mu_1}; 1 + \frac{\mu_2 + (1-p)\mu_1 - \frac{1}{\beta} \mu_2}{\lambda_1 + \frac{1}{\beta}(1-q)\mu_2} \right\}$$

$$b = \text{Min} \left\{ \frac{\mu_2}{\lambda_2}; 1 + \frac{\alpha \lambda_1 + \alpha \mu_1 - \mu_1 - \alpha^2 \lambda_1}{\alpha \lambda_2 + p \mu_1}; 1 + \frac{\lambda_2 + \mu_1 + \mu_2 + \sqrt{\lambda_2^2 + \mu_1^2 + \mu_2^2 + 2\mu_1 \mu_2 + 2\mu_1 \lambda_2 - 2\mu_2 \lambda_2}}{2\lambda_2}; 1 + \frac{(1-q)\mu_2 + (1-p)\mu_1}{\mu_2} \right\}$$

Théorème 1. Sous les conditions du Lemme 1, et soit

$$a = \sup \{ \alpha, \rho(\alpha, \beta) < 1 \}$$

$$b = \sup \{ \beta, \rho(\alpha, \beta) < 1 \}$$

Alors, la chaîne de Markov associée au modèle tronqué est fortement v -stable, pour la fonction test $v(k, l) = \alpha^k \beta^l$.

Preuve. Pour prouver la v -stabilité de la chaîne de Markov pour la fonction $v(k, l) = \alpha^k \beta^l$,

où $\alpha \in]1, a[$

$\beta \in]1, b[$, tel que :

$$a = \text{Min} \left\{ 1 + \frac{\beta \lambda_2 + (\beta-1)\mu_2 - \beta^2 \lambda_2}{\beta \lambda_1 + (1-q)\mu_2}; \frac{\mu_1}{\lambda_1}; \frac{\lambda_2 + \mu_1 + \mu_2 - \beta \lambda_2 - \frac{1}{\beta} \mu_2}{\alpha \lambda_2 + p \mu_1}; 1 + \frac{\mu_2 + (1-p)\mu_1 - \frac{1}{\beta} \mu_2}{\lambda_1 + \frac{1}{\beta}(1-q)\mu_2} \right\}$$

$$b = \text{Min} \left\{ \frac{\mu_2}{\lambda_2}; 1 + \frac{\alpha \lambda_1 + \alpha \mu_1 - \mu_1 - \alpha^2 \lambda_1}{\alpha \lambda_2 + p \mu_1}; 1 + \frac{\lambda_2 + \mu_1 + \mu_2 + \sqrt{\lambda_2^2 + \mu_1^2 + \mu_2^2 + 2\mu_1 \mu_2 + 2\mu_1 \lambda_2 - 2\mu_2 \lambda_2}}{2\lambda_2}; 1 + \frac{(1-q)\mu_2 + (1-p)\mu_1}{\mu_2} \right\}.$$

Vérifions les conditions du Lemme 1 et d'après le Théorème de Kartachov [69] :

$$\|T\|_v = \rho(\alpha, \beta)$$

Avec $\rho(\alpha, \beta) < 1$

$$\rho(\alpha, \beta) = \max_{i=1,8} \rho_i \tag{4.21}$$

Théorème 2. Sous les conditions du Lemme 1, nous avons

$$\|P - \bar{P}\|_v = \frac{\alpha\lambda_1}{M} - \frac{\lambda_1 + (1-q)\mu_2}{\alpha^Q M} \quad (4.22)$$

Preuve. $\|P - \bar{P}\|_v = \sup_{0 \leq i \leq Q} \sup_{0 \leq j \leq N} \frac{1}{v(i,j)} \cdot \sum_{i=0} \sum_{j=0} v(m,n) |P_{(i,j) \rightarrow (m,n)} - \bar{P}_{(i,j) \rightarrow (m,n)}|$

- Si $i = 0$

$$\|P - \bar{P}\|_v = 0$$

- Si $0 < i < Q$

$$\|P - \bar{P}\|_v = 0$$

- Si $i = Q$

- Si $j = 0$

$$P_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ 0 & , \text{ Si } m = 0, n = j \end{cases}$$

$$\bar{P}_{(i,j) \rightarrow (m,n)} = \begin{cases} 0 & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ 1 - \frac{\lambda_2 + \mu_1}{M} & , \text{ Si } m = 0, n = j \end{cases}$$

$$\|P - \bar{P}\|_v = \frac{1}{\alpha^i \beta^j} \left[\alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^0 \beta^j \frac{-(\lambda_1 + \mu_2)}{M} \right]$$

$$\|P - \bar{P}\|_v = \frac{\alpha\lambda_1}{M} - \frac{\lambda_1 + \mu_2}{\alpha^Q M} \quad (4.23)$$

◦ Si $0 < j < N$

$$P_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ \frac{(1-q)\mu_2}{M} & , \text{ Si } m = i + 1, n = j - 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 0 & , \text{ Si } m = 0, n = j \end{cases}$$

$$\bar{P}_{(i,j) \rightarrow (m,n)} = \begin{cases} 0 & , \text{ Si } m = i + 1, n = j \\ \frac{\lambda_2}{M} & , \text{ Si } m = i, n = j + 1 \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{p\mu_1}{M} & , \text{ Si } m = i - 1, n = j + 1 \\ 0 & , \text{ Si } m = i + 1, n = j - 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 1 - \frac{\lambda_2 + \mu_1 + \mu_2}{M} & , \text{ Si } m = 0, n = j \end{cases}$$

$$\|P - \bar{P}\|_v = \frac{1}{\alpha^i \beta^j} [\alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^0 \beta^j \frac{-(\lambda_1 + (1-q)\mu_2)}{M}]$$

$$\|P - \bar{P}\|_v = \frac{\alpha \lambda_1}{M} - \frac{\lambda_1 + (1-q)\mu_2}{\alpha^Q M} \quad (4.24)$$

◦ $j = N$

$$P_{(i,j) \rightarrow (m,n)} = \begin{cases} \frac{\lambda_1}{M} & , \text{ Si } m = i + 1, n = j \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ \frac{(1-q)\mu_2}{M} & , \text{ Si } m = i + 1, n = j - 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 0 & , \text{ Si } m = 0, n = j \end{cases}$$

$$\bar{P}_{(i,j) \rightarrow (m,n)} = \begin{cases} 0 & , \text{ Si } m = i + 1, n = j \\ \frac{(1-p)\mu_1}{M} & , \text{ Si } m = i - 1, n = j \\ 0 & , \text{ Si } m = i + 1, n = j - 1 \\ \frac{q\mu_2}{M} & , \text{ Si } m = i, n = j - 1 \\ 1 - \frac{(1-p)\mu_1 + q\mu_2}{M} & , \text{ Si } m = 0, n = j \end{cases}$$

$$\|P - \bar{P}\|_v = \frac{1}{\alpha^i \beta^j} [\alpha^{i+1} \beta^j \frac{\lambda_1}{M} + \alpha^0 \beta^j \frac{-(\lambda_1 + (1-q)\mu_2 + p\mu_1 + \lambda_2)}{M}]$$

$$\|P - \bar{P}\|_v = \frac{\alpha \lambda_1}{M} + \frac{-(\lambda_1 + (1-q)\mu_2 + p\mu_1 + \lambda_2)}{\alpha^Q M} \quad (4.25)$$

D'où :

$$\|P - \bar{P}\|_v = \frac{\alpha\lambda_1}{M} - \frac{\lambda_1 + (1-q)\mu_2}{\alpha^Q M}$$

Théorème 3. Dans les conditions du Lemme 1, et si $\|P - \bar{P}\|_v < \frac{1-\rho}{A}$, alors

$$\|\pi - \bar{\pi}\|_v \leq \frac{A_0(\alpha, \beta).A(\alpha, \beta).d_1(\alpha, \beta)}{1 - \rho(\alpha, \beta) - A(\alpha, \beta).d_1(\alpha, \beta)}$$

Avec :

$$A_0(\alpha, \beta) = \|\pi\|_v \leq \sigma v(1 - \rho)^{-1}.(\pi h)$$

$$\rho = \max_{i=1,8} \rho_i$$

$$A = 1 + A_0(\alpha, \beta)$$

$$d_1 = \|P - \bar{P}\|_v$$

Preuve. On a $\|P - \bar{P}\|_v = \frac{\alpha\lambda_1}{M} - \frac{\lambda_1 + (1-q)\mu_2}{\alpha^Q M}$

En utilisant le Théorème de Kartachov :

$$\|\pi - \bar{\pi}\|_v \leq \frac{A_0(\alpha, \beta).A(\alpha, \beta).d_1(\alpha, \beta)}{1 - \rho(\alpha, \beta) - A(\alpha, \beta).d_1(\alpha, \beta)}$$

Avec :

$$A_0(\alpha, \beta) = \|\pi\|_v \leq \sigma v(1 - \rho)^{-1}.(\pi h) \tag{4.26}$$

$$\rho = \max_{i=1,8} \rho_i$$

On a :

$$\sigma_{(i,j)} = P_{(0,0) \rightarrow (i,j)}$$

$$P_{(0,0) \rightarrow (0,0)} = \frac{\mu_1 + \mu_2}{M}$$

$$P_{(0,0) \rightarrow (1,0)} = \frac{\lambda_1}{M}$$

$$P_{(0,0) \rightarrow (0,1)} = \frac{\lambda_2}{M}$$

D'où :

$$\sigma v = \alpha^0 \beta^0 \left[\frac{\mu_1 + \mu_2}{M} + \alpha \frac{\lambda_1}{M} + \beta \frac{\lambda_2}{M} \right] \quad (4.27)$$

$$A = 1 + A_0(\alpha, \beta) \quad (4.28)$$

$$\pi h = \pi(0, 0) > 0 \quad (4.29)$$

$$d_1(\alpha, \beta) = \|P - \bar{P}\|_v \quad (4.30)$$

4.4 Application

Pour la résolution du système, on a élaboré un algorithme qui est utilisé pour vérifier l'approximation du système réel par le système idéal, et déterminer les erreurs des distributions stationnaires dûent à cette approximation.

Afin de comprendre le principe de l'algorithme mis en oeuvre pour la méthode, nous allons détailler les différentes étapes suivantes.

4.4.1 Algorithme de stabilité forte

1. Introduire les paramètres du modèle $(\lambda_1, \mu_1, \lambda_2, \mu_2, N, Q, p, q, h)$.

2. Calcul de b tel que :

$$b = \text{Min} \left\{ \frac{\mu_2}{\lambda_2}; 1 + \frac{\alpha\lambda_1 + \alpha\mu_1 - \mu_1 - \alpha^2\lambda_1}{\alpha\lambda_2 + p\mu_1}; 1 + \frac{\lambda_2 + \mu_1 + \mu_2 + \sqrt{\lambda_2^2 + \mu_1^2 + \mu_2^2 + 2\mu_1\mu_2 + 2\mu_1\lambda_2 - 2\mu_2\lambda_2}}{2\lambda_2}; 1 + \frac{(1-q)\mu_2 + (1-p)\mu_1}{\mu_2} \right\}$$

3. $\beta = 1 + h$

4. Tant que $\beta < b$

$$\alpha = \beta + h$$

Calcul de a tel que :

$$a = \text{Min}\left\{1 + \frac{\beta\lambda_2 + (\beta-1)\mu_2 - \beta^2\lambda_2}{\beta\lambda_1 + (1-q)\mu_2}; \frac{\mu_1}{\lambda_1}; \frac{\lambda_2 + \mu_1 + \mu_2 - \beta\lambda_2 - \frac{1}{\beta}\mu_2}{\alpha\lambda_2 + p\mu_1}; 1 + \frac{\mu_2 + (1-p)\mu_1 - \frac{1}{\beta}\mu_2}{\lambda_1 + \frac{1}{\beta}(1-q)\mu_2}\right\}$$

5. Tant que $\alpha < a$

On calcul ρ avec la formule (4.21)

On calcul $\|P - \bar{P}\|_v$ avec la formule (4. 22)

Si $\|P - \bar{P}\|_v < \frac{1-\rho}{A}$

alors $\|\pi - \bar{\pi}\|_v = \frac{A_0(\alpha,\beta) \cdot A(\alpha,\beta) \cdot d_1(\alpha,\beta)}{1-\rho(\alpha,\beta) - A(\alpha,\beta) \cdot d_1(\alpha,\beta)}$

Avec

A_0 est calculé dans la formule (4. 26).

$A = 1 + A_0(\alpha, \beta)$ est calculé dans la formule (4. 28).

6. On détermine α, β qui minimise $\|\pi - \bar{\pi}\|_v$

4.4.2 Résultats et comparaison

En fixant les paramètres d'entrée : $\lambda_1 = 0.001, \mu_1 = 1, \lambda_2 = 0.001, \mu_2 = 1, N = 5, p = 0.1, q = 0.3$, on a obtenu les résultats suivants :

Q	$\ \pi - \bar{\pi}\ _v$ approché	$\ \pi - \bar{\pi}\ _v$ réel
5	0.0019	6.0445×10^{-18}
10	0.0016	1.6289×10^{-19}
15	1.6673×10^{-4}	7.9495×10^{-20}
20	1.6672×10^{-4}	5.6344×10^{-26}
25	1.6671×10^{-4}	0
30	1.6639×10^{-4}	0
35	1.6639×10^{-4}	0
40	1.6639×10^{-4}	0
45	1.6639×10^{-4}	0
50	1.6639×10^{-4}	0

TAB. 4.1 – Tableau comparatif des erreurs

Ces résultats sont représentés par le graphe de la figure 4.3

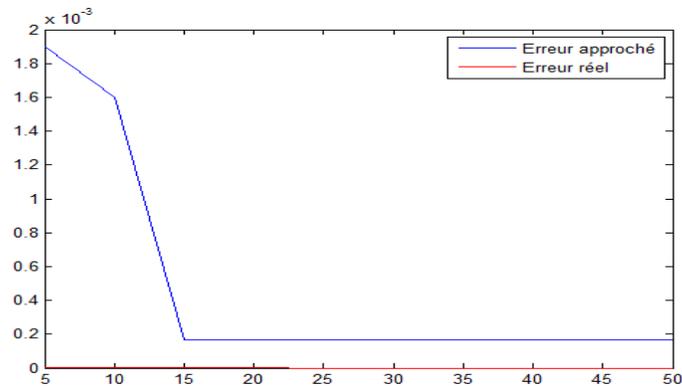


FIG. 4.3 – Courbes des erreurs.

Les courbes nous ont permis d'observer le comportement des erreurs des probabilités stationnaires approchées et celui des probabilités stationnaires réelles. Il est aisé de remarquer que l'augmentation de la taille de la première file d'attente Q induit une diminution des erreurs commises. De plus, l'erreur des probabilités stationnaires réelles sont toujours inférieures à celles des probabilités stationnaires approchées.

4.5 Analyse de performances

Dans ce travail, on s'intéresse :

- au nombre moyen de clients dans le système.
- à la durée moyenne de séjour.

4.5.1 Intèrprétation des résultats

L'implémentation de l'algorithme a permis d'obtenir :

1. La variation du nombre moyen de clients dans le système en fonction de λ_1 , qui est présentée sur le graphe de la figure 4.4

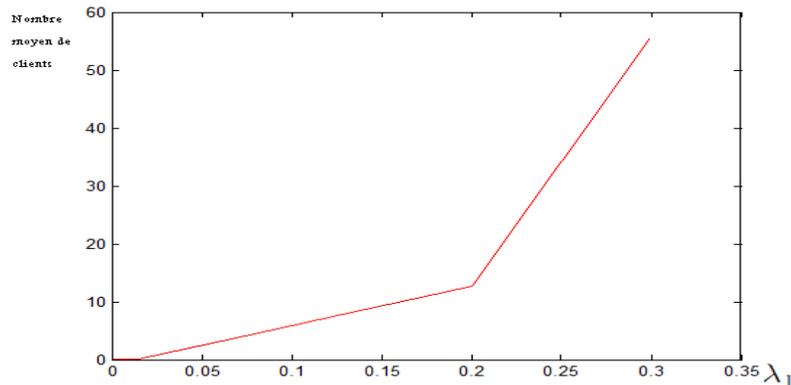


FIG. 4.4 – Nombre moyen de clients dans le système en fonction de λ_1 .

À partir de la courbe, on remarque que, l'augmentation de λ_1 induit à l'augmentation du nombre moyen de clients dans le système. Tant que le taux des arrivées des clients augmente, on donne plus de chance aux utilisateurs d'émettre leurs requêtes. D'où, on aura le nombre de requêtes qui augmente.

Ce résultat valide nos hypothèses qui sont :

Le système est accédé par un nombre élevé de clients, quand la période de temps est grande. Ceci mène à étudier dans le futur le système dans le cas de pannes. Nous précisons que les requêtes englobent les nouvelles arrivées et les requêtes non satisfaites.

2. La variation de nombre moyen de clients dans le système en fonction de λ_2 , qui est présentée sur le graphe de la figure 4.5

À partir de la courbe, on remarque que, l'augmentation de λ_2 induit à l'augmentation du nombre moyen de clients dans le système. Tant que le taux des arrivées des clients augmente, on donne plus de chance aux utilisateurs d'émettre leurs requêtes. D'où, on aura le nombre de requêtes qui augmente.

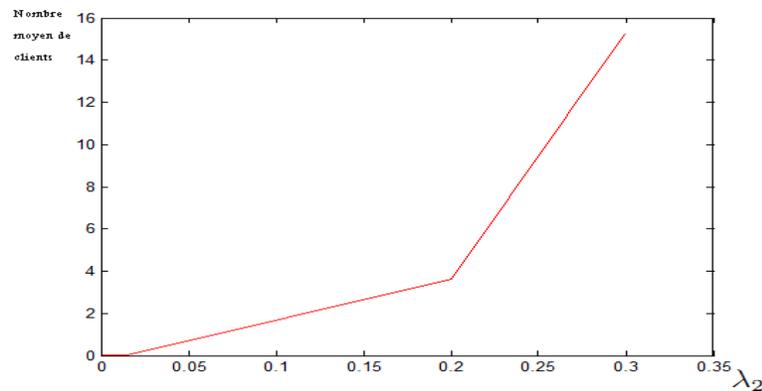


FIG. 4.5 – Nombre moyen de clients dans le système en fonction de λ_2 .

Ce résultat valide nos hypothèses qui sont :

Le système est accédé par un nombre élevé de clients, quand la période de temps est grande. Néanmoins, le nombre de requêtes qui arrivent lors de la composition des Web services est moindre que celui de la figure 4.4, car elle regroupe aussi les requêtes non satisfaites et les requêtes retardées.

3. La variation de la durée moyenne de séjour en fonction de λ_1 , qui est présentée sur le graphe de la figure 4.6

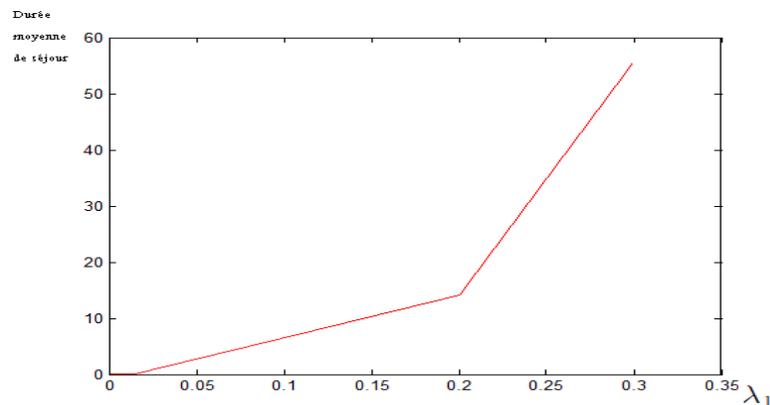


FIG. 4.6 – Durée moyenne de séjour en fonction de λ_1 .

À partir de la courbe, on remarque que, l'augmentation de λ_1 induit à l'augmentation de la durée moyenne de séjour. Tant que le taux des arrivées des clients augmente, on donne plus de chance aux utilisateurs d'émettre leurs requêtes. D'où, on aura la durée moyenne de séjour qui augmente.

4. La variation de la durée moyenne de séjour en fonction de λ_2 , qui est présentée sur le graphe de la figure 4.7

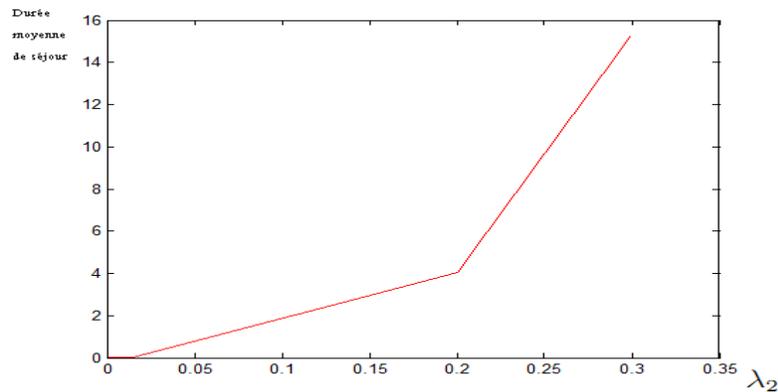


FIG. 4.7 – Durée moyenne de séjour en fonction de λ_2 .

À partir de la courbe, on remarque que, l'augmentation de λ_2 induit à l'augmentation de la durée moyenne de séjour. Tant que le taux des arrivées des clients augmente, on donne plus de chance aux utilisateurs d'émettre leurs requêtes. D'où, on aura la durée moyenne de séjour qui augmente.

4.6 Positionnement de notre proposition par rapport à la littérature

Le modèle proposé par Grassmann et Tavakoli (1980) [67] est adéquat au problème de la découverte et de la composition des Web services.

Néanmoins, les auteurs se sont intéressés aux erreurs d'approximation.

Nous avons adopté ce modèle, résolu par la méthode de troncature et évalué ses performances.

Dans la littérature, il n'existe pas des résultats exacts pour ce modèle complexe. Par contre pour les travaux cités ([63], [64]), on trouve des résultats analytiques qui existent déjà.

Le modèle proposé traite le cas des Web services simples et composites à la différence des autres travaux ([63], [64], [65], [66]) et dans le cas de la disponibilité.

De plus, à notre connaissance, le modèle proposé a été traité par une méthode analytique. Nous avons aussi calculé les performances du modèle, à savoir le nombre moyen de clients dans le système et la durée moyenne de séjour des clients. Le nombre moyen de clients dans le système n'a pas été traité dans les autres travaux, alors qu'il est très important, car nous traitons un cas réel dans le Web qui est très dynamique.

4.7 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'aspect pratique du problème de stabilité forte appliqué au système de composition et découverte des Web services. Nous avons élaboré un algorithme qui nous a permis de déterminer l'erreur d'approximation et d'évaluer les performances de ce système.

Conclusion générale

Dans le cadre de ce travail, nous avons modélisé un système de Web services simples et composites, par un modèle de files d'attente à deux stations en tandem, où la première station a une file d'attente infinie et la deuxième station est avec une file d'attente limitée (N), et avec deux taux d'arrivées, l'arrivée à la première station avec un taux λ_1 et l'arrivée à la deuxième station avec un taux λ_2 . À la fin du service du client dans la première station, si le client est satisfait, il quitte le système avec la probabilité $1 - p$. Sinon, il passera à la deuxième file d'attente avec la probabilité p , pour la recherche d'une réponse dans les Web services composites. À la fin de ce service, s'il est encore non satisfait, il va rejoindre la première file d'attente avec la probabilité $1 - q$.

Notre système n'est pas sous forme produit, ce qui rend son étude difficile. Pour ce faire, nous avons approximé ce système par un autre système de files d'attente où la première station a une file d'attente limitée (Q), et ce, en utilisant la technique de troncature. Nous avons utilisé la méthode de stabilité forte pour trouver les conditions d'application de cette approximation.

Nous nous sommes ensuite intéressés à la mesure des performances. Nous avons calculé quelques performances du système de découverte et de composition de Web services via le réseau Internet afin de statuer sur le nombre de demandes de services qui pourra être pris en charge réellement pour éviter la saturation du système et de satisfaire les demandes des clients efficacement en temps réduit.

Les résultats obtenus dans le cadre de ce mémoire ouvrent de nombreuses perspectives de recherche :

- Faire l'étude du système de Web services simples et composites dans le cas de l'indisponibilité des serveurs (pannes).
- Des résultats similaires peuvent être obtenus pour le système de Web services simples et composites, en tenant compte des arrivées des Web services.
- Résoudre le modèle de files d'attente avec d'autres méthodes (réseau de Petri).
- Résoudre le modèle de files d'attente avec la méthode de Van Dijk [59] et faire une étude comparative entre nos résultats et les résultats obtenus.
- Etudier la sensibilité des résultats aux changements des paramètres.

Bibliographie

- [1] H. Kadima. *"Les Web services"*, Edition Eyrolles, 2003.
- [2] T. Bray *"Extensible Markup Language (XML)"*, édition 2, W3C Recommendation, Aout 2006.
- [3] H. Kreger *"Web Services Conceptual Architecture (WSCA 1.0)"*, IBM Software Group, Technical report, May 2001.
- [4] H. Kadima et V. Monfort *"Les services Web, Techniques, Démarches et outils XML, SOAP, UDDI, ROSETTAN, UML"*, Dunod édition, 2004.
- [5] H. Talantikite *"Les Web services complexes : Découverte, Composition, Sélection et Orchestration"*, Thèse de Doctorat, Université de Béjaia, janvier 2010.
- [6] B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid *"Composing Web services on the Semantic Web"*, The VLDB Journal, 12 (4), November 2003.
- [7] J. Rao and X. Su *"A Survey of Automated Web Service Composition Methods"*, Norwegian University of Science and Technology Department of Computer and Information Science N-7491, Trondheim, Norway, 2004.
- [8] N. Temglit, H. Aliane, M. A. Nacer *"Un modèle de composition des services web sémantiques"*, Volume 11, Editeur : Marc Kokou Assogba, pp. 115-133, ARIMA, 2008.
- [9] T. Melliti *"Interopérabilité des services Web complexes, Application aux systèmes multi-agents"*, Thèse de Doctorat, Université Paris IX Dauphine, décembre 2004.
- [10] D. Chappell, T. Jewell *"Java Web Service"*, édition O'Reilly, Mars 2002.
- [11] S. Chatterjee, J. Webber *"Developing Enterprise Web Services"*, édition Prentice Hall PTR, 2003.

-
- [12] J. Siegel " *Corba 3 fundamentals and programming*", Université du Michigan, Édition 2, illustrée, pp. 899, 2000.
- [13] B. Baynat " *Théorie des files d'attente :des chaines de Markov aux réseaux à forme produit*", Edition Hermes science publications, 2000.
- [14] A. Lahmadi " *Evaluation de performance des architectures de gestion de reseau*", Rapport de recherche, hal.archives-ouvertes.fr/docs/00/07/04/09/PS/RR-5598.ps, Juin 2005.
- [15] A. Koubaa " *Introduction à l'évaluation de performances des systèmes informatiques et de communication*", Rapport de recherche, University of Minnesota, [www.amazon.com/Performance Evaluation of Computer and Communication Systems/3540](http://www.amazon.com/Performance-Evaluation-of-Computer-and-Communication-Systems/3540), Janvier 2004.
- [16] " *Loi de Poisson*", URL du document : [http : //fr.WiKipedia.org/WiKi/Loi – de – poisson](http://fr.WiKipedia.org/WiKi/Loi-de-poisson), dernière consultation 2 Septembre 2011.
- [17] " *Loi Exponentielle*", URL du document : [http : //fr.WiKipedia.org/WiKi/Loi – exponentielle](http://fr.WiKipedia.org/WiKi/Loi-exponentielle), dernière consultation 2 Septembre 2011.
- [18] L. Bouallouche " *L'approche de stabilité forte dans l'évaluation de performance des systèmes informatiques*", Thèse de doctorat, Université Farhat Abbas de Sétif, 2006.
- [19] P. Cong-Duc " *Files d'attente*", Cours de Modélisation et d'Evaluation de Performance, Département d'informatique Université de Sherbrooke,
- [20] L. Cheyroux " *Sur l'évaluation de performances des chaines logistiques* ", Thèse de doctorat, Institut national polytechnique de grenoble, 2003.
- [21] F. Aoudia " *Développement en série et stabilité forte dans les systèmes d'attente avec vacances du serveur*", Thèse Doctorat, Université de Bejaia, 2008.
- [22] A. Aissani " *Modèles stochastiques de la théorie de fiabilité* ", Office de publications Universitaires, Alger, 1992.

- [23] O. C. Ibe " *Stochastic Petri Net Analysis of Queueing Systems with Vacations*", Internet Report, www.docstoc.com/docs/119754744/STOCHASTIC-PETRI-NET, Digital Equipment Corporation. Andover. MA, septembre 1989.
- [24] O. C. Ibe and K. S. Trivedi " *Stochastic Petri Net Analysis of Finite -Population Vacation Queueing Systems*", Vol. 8, pp. 111-128, 1991.
- [25] M. Roussingnol and D. Flipo " *Files d'attente et fiabilité.*", DEES, Université des Sciences et Technologies de Lille, 2002-2003.
- [26] A. Ruegg " *Processus Stochastiques (avec Applications aux Phénomènes d'Attente et de Fiabilité)*", Presses Polytechniques Romandes, Première Edition, Vol. 6, 1989.
- [27] J. Sztrik " *Finite Source Queueing Systems and their Applications* ", Technical report, irh.inf.unideb.hu/user/jsztrik/education/Slides/fsqs.pdf, University of Debrecen, Institute of Mathematics and Informatics, Departement of Information Technology, August 2001.
- [28] A. Aissani and D. Aissani " *Fiabilité des Systèmes et Systèmes de Files d'Attente non Fiabiles* ", U.E.R Mathématique- Informatique, ENITA, Bordj-El-Behri, 1986.
- [29] T. K. Das " *Analysis of Patrolling Repairman Systems* ", Thèse Doctorat, Université Texas, 1989.
- [30] S. Kendi, S. Touati, L. Bouallouche et al " *Evaluation des performances d'un réseau informatique : Application au Réseau de la D.R.G.B. Sonatrach*", Proceedings of the International Conference MOAD'S2007 (Decision Aid Methods and Tools), Béjaia, November 2007, pp. 735- 740. ISBN : 978-9947-0-1958-0.
- [31] B. Arntzen, G. Brown " *Global Supply Chain Management at Digital Equipment Corporation*", Interfaces, vol. 25, n° 1, pp. 69-93, 1995.
- [32] J. D. Camm, T. E. Chorman " *Blending OR/MS, Judgment and GIS : Restructuring P and G's Supply Chain*", Interfaces, vol. 27, n° 1, pp. 128-142, 1997.
- [33] H. L. Lee, C. Billington " *Material Management in Decentralized Supply Chains*", Operations Research, vol 41, n° 5, pp. 835-847, 1993.

- [34] H. L. Lee, C. Billington, B. Carter "Hewlett-Packard Gains Control of Inventory and Service through Design for Localization ", Interfaces, vol. 23, n° 4, pp. 1-11, 1993.
- [35] H. L. Lee, C. Billington "The Evolution of Supply-Chain-Management Models and Practice at Hewlett-Packard", Interfaces, vol. 25, n° 5, pp. 42-63, 1995.
- [36] M. A. Cohen, H. L. Lee "Strategic Analysis of Integrated Production-Distribution Systems : Models and Methods", Operations Research, vol. 36, n° 2, pp. 216-228, 1988.
- [37] 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.
- [38] N. R. S Raghavan, N. Viswanadham "Generalized Queueing Network Analysis of Integrated Supply Chains ", International Journal of Production Research, vol.39 n° 2, pp. 205-224, 2001.
- [39] C. J. Vidal, N. Goetschalckx "A global supply chain model with transfer pricing and transportation cost allocation", European Journal of Operational Research, n° 129, pp. 134-158, 2001.
- [40] L. Bertazzi, M. G. Speranza " Minimizing Logistic Costs in multistage supply chains", Naval Research Logistics, vol. 46, pp 399-417, 1999.
- [41] K. L. Cheung, H. L. Lee "The inventory benefit of shipment coordination and stock rebalancing in a supply chain ", Management Science, vol. 48, n° 2, pp. 300-306, 2002.
- [42] J. Grahovac, A. Chakravarty "Sharing and lateral transshipment of inventory in a supply chain with expensive low-demand items", Management Science, vol. 47, n° 4, pp. 579-594, 2001.
- [43] T. J. Van Roy "Multi-Level Production and Distribution Planning with Transportation Fleet Optimization", Management Science, vol. 35, n° 12, pp. 1443-1453, 1989.
- [44] M. Vroblefski, R. Ramesh, S. Zionts "Efficient lot-sizing under a differential transportation cost structure for serially distributed warehouses", European Journal of Operational Research, n° 127, pp. 574-593, 2000.

- [45] G. P. Cachon "Exact Evaluation of Batch-Ordering Inventory Policies in Two- Echelon Supply Chains with Periodic Review", Operations research, vol. 49, n° 1, pp. 79-98, 2001.
- [46] C. E. Riddalls, S. Bennett " The stability of supply chains", International Journal of Production Research, vol. 40, n° 2, pp. 459-475, 2002.
- [47] W. K. Chiang, D. Chhajed, J. D. Hess " Direct marketing, indirect profits : a strategic analysis of dual-channel supply chain design", Management Science, vol. 49, n° 1, pp. 1-20, 2003.
- [48] H. Lee, S. Whang " The impact of the secondary market on the supply chain", Management Science, vol. 48, n° 6, pp. 719-731, 2002.
- [49] Y. H. Lee " Supply Chain Model for the Semiconductor Industry of Global Market", Journal of Systems Integration, n° 10, pp. 189-206, 2001.
- [50] N. C. Simpson, S. S. Erenguc "Modeling the order picking function in supply chain systems : formulation, experimentation and insights ", Interfaces, Volume 33, Number 2, pp. 119-130, 2001
- [51] A. Aissani "Éléments de simulation statistique", Office de publications Universitaires, Alger, 1998.
- [52] D. Aissani "Cours de simulation", Département d'informatique et de recherche opérationnelle, Université de Béjaia, 2001.
- [53] R. Sait, N. Zerrougui, S. Adjabi et al "Évaluation de performances du parc à conteneurs de l'entreprise portuaire de Béjaia", Proceedings of an International Conference Sada'07 (Applied Statistics for Development in Africa), Cotounou (Benin), 2007.
- [54] S. George, Fishman " Concepts and Méthods in Discrete event Digital Simulation", Awiley-Interscience publication, 1973.
- [55] P. J. Erard, P. Déguénon "simulation par événements discrets", Publié par press polytechniques et Université Romandes, 1996.

- [56] F. Valois "Introduction à la simulation de réseaux", Technical report, fvalois.insa-lyon.fr/courses/IRZ_simulation_v3.pdf, Université de Lyon, 2006.
- [57] D. Aissani and N.V. Kartashov "Ergodicity and stability of Markov chains with respect to operator topology in the space of transition kernels", Dokl. Akad. Nauk Ukr. SSR, ser. A 11, 3-5, 1983
- [58] N.V. Kartashov "Strong stable Markov chains", VSP, Utrecht, TbiMC Scientific Publishers, 1996
- [59] N. Van Dijk "Truncation of Markov chains with applications to queuing", Operation Research, Vol. 39, pp. 1018-1026, 1991.
- [60] V. Kalashnikov et S. Rachev "Mathematical methods for construction of queueing models", Wadsworth and brooks cole Operations Research Series, 1990.
- [61] Y.Q. Zhao et D. Liu "The censored Markov chain and the best augmentation", Journal of Applied Probability, 33, 623-629, 1996
- [62] D. Gibon et E. Seneta "Augmented truncations of infinite stochastic matrices", Journal of Applied Probability, 24, 600-608, 1987
- [63] M. Martinello, M. Kaâniche, K. Kanoun "Web service availability-impact of error recovery and traffic model", Elsevier, Volume 89, Pages 6-16, Toulouse, france, July 2005.
- [64] M. Martinello "Modélisation hiérarchique de la disponibilité perçue par l'utilisateur", Elsevier, vol 277, pp. 17-32, Institut National Polytechnique de Toulouse, 2005.
- [65] M. Younas, I. Awan, D. Duce. "An efficient composition of Web services with active network support", Revue Expert Systems with Applications, Vol 13, pp. 859-869, 2006.
- [66] V. Auletta "Performance Evaluation of Web Services Invocation over Bluetooth", Technical report, Université de Di Salerno, 2005.
- [67] K. Winfried. Grassmann et T. Javad "Two-station queueing networks with moving servers, blocking, and customer loss", Journal of Linear Algebra, Volume 13, pp. 72-89, March 2005

- [68] I. Gohberg, P. Lancaster et L. Rodman *"Matrix Polynomials"*, Academic Press, New York, 1982.
- [69] N. V. Kartachov *"Criteria of uniform ergodicity and storing stability of Markov chain with a common phase spase"*, Theor. Prob. and Math. Statist, Vol 30, pp. 71-89, 1984

Liste des acronymes

DAML	DARPA Agent Markup Language
DAML-S	DARPA Agent Markup Language Service
HTTP	Hyper Text Transfer Protocol
OWL	Ontologie Web Language
OWL-S	Ontologie Web Language Service
QOS	Quality Of Service
RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
UDDI	Universal Description Discovery and Integration
W3C	Word Wide Web Consortium
WSDL	Web Services Description Language
XML	Extensible Markup Language
MMPP	Markov modulated Poisson traffic
UA	Indisponibilité du Web service
RPMH	Réseaux Petits Mondes Hiérarchiques
RP	Réseau Possibiliste
TLMs	Top-level managers
MLMs	Mid-level managers
SNMP	Simple Network Management Protocol

Résumé

Dans ce mémoire, nous avons modélisé un système de Web services simple et composite, par un modèle de files d'attente à deux stations en tandem, où la première station a une file d'attente infinie et la deuxième station est avec une file d'attente limitée (N), et avec deux taux d'arrivées, l'arrivée à la première station avec un taux λ_1 et l'arrivée à la deuxième station avec un taux λ_2 . À la fin du service du client dans la première station, si le client est satisfait, il quitte le système avec la probabilité $1 - p$. Sinon, il passera à la deuxième file d'attente avec la probabilité p , pour la recherche d'une réponse dans les Web services composites. À la fin de ce service, s'il est encore non satisfait, il va rejoindre la première file d'attente avec la probabilité $1 - q$. Notre système n'est pas sous forme produit, ce qui rend son étude difficile.

Nous avons approximé les caractéristiques de ce système par celles d'un autre système de files d'attente où la première station a une file d'attente limitée (Q), et ce, en utilisant la technique de la troncature. Nous avons ensuite utilisé la méthode de stabilité forte pour trouver les conditions d'application de cette approximation.

Enfin, en faisant varier les deux taux des arrivées, on a pu déterminer l'influence des deux taux des arrivées sur le nombre moyen de clients dans le système et la durée moyenne de séjour.

Mots clés : Web service, Composition, Evaluation des performances, Troncature, Stabilité forte.

Abstract

We have modelled a simple and compose Web service system, by a queue of two stations in tandem. Where the first station have an infinite buffers and the second, with a limited buffer (N). Customers arrive at first (second) station according to the Poisson processes with rate λ_1 (λ_2 , respectively). After completing service at a first station, customers enter to second (first) station with probability p ($1 - q$) or leave the system with probability $1 - p$ (q). This model is not under a product form, what makes the study of this system harder.

We have approached this system by another queue system where the first station has a limited queue (Q) by using truncation. Then we have used the strong stability method in order to find the condition to applied this approach.

At the end, by varying the two incoming rates, we could determine their effect and influence on the average client level in the system and the average duration of the stay.

Key words : Web service, Composition, performances evaluation, Truncation, Strong stability.

الملخص

في هذه الورقة، ونحن على غرار خدمات ويب بسيطة ومركبة نموذج محطتين جنبا إلى جنب، حيث أول محطة لها قائمة الانتظار اللانهائي والمحطة الثانية لها قائمة انتظار محدودة، ومع معدلين للوصول، معدل وصول إلى المحطة الأولى ومعدل وصول إلى المحطة الثانية. في نهاية خدمة العملاء في المحطة الأولى، إذا اقتنعت العملاء، فإنه يترك النظام و في خلاف ذلك، فإنه يذهب إلى قائمة الانتظار الثانية، والبحث عن إجابة في خدمات ويب مركب. بحلول نهاية الخدمة، إذا كان لا يزال غير راض، يقوم بالانضمام إلى طابور المحطة الأولى. نظامنا ليس من شكل الجداء مما يجعل دراسته صعبة. نقوم بتقريب خصائص هذا النظام مع نظام آخر لديه قائمة انتظار محدودة، وذلك باستخدام تقنية الاقتطاع. ثم استخدام الأسلوب الاستقرار لإيجاد ظروف هذا التقارب وأخيرا، من خلال تغيير كل من معدلي الوصول، قمنا بتحديد تأثير كل من معدلي الوصول على متوسط عدد العملاء في النظام ومتوسط مدة الإقامة

كلمات البحث: خدمة ويب، تركيب، تقييم الأداء، اقتطاع، الاستقرار.