



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 Master en Recherche Opérationnelle

Option : *Fiabilité et Evaluation des Performances des Réseaux*

Thème

Evaluation des Performances d'un Web Service à l'aide d'un Réseau de Files d'attente



Soutenu devant le jury composé de :

Président : M^{me} K. Adel

Promoteur : P^r Dj. Aissani

Co-Promotrice : M^{me} N. Bernine

Examinatrices : M^{me} N. Belkacem

M^{elle} L. Lakaouer

Présenté par :

M^{elle} Atrouche Sabeha

M^r Oulhaci Massinissa

Promotion 2014/2015

Résumé

Les Web services (WS) sont des applications hétérogènes et réparties exposant leurs fonctionnalités sous forme de services exécutables à distance. Les WS sont basés sur des technologies qui favorisent l'interopérabilité, l'extensibilité, l'indépendance vis-à-vis des plates-formes et des langages de programmation. L'objectif principal de ce mémoire est d'évaluer les performances d'un Web service. Le système considéré est une source Internet et deux types de service simples et composites. Nous l'avons modélisé sous forme d'un réseau de files d'attente en tandem à trois stations. Les caractéristiques de performances ont été obtenues en utilisant le simulateur (Java Modeling Tools). Les résultats de la simulation montrent que le nombre moyen de clients qui utilisent les Web services composites est plus important par rapport aux Web services simples, et que le taux des inter-arrivées influe sur le nombre moyen de clients entrant au système.

Mots clés : *Web services, Évaluation de performances, Files d'attente, Simulation.*

Table des matières

Table des matières	i
Table des figures	i
Liste des abréviations	ii
1 Introduction générale	1
Introduction générale	1
2 État de l’art sur les Web services	3
2.1 Introduction	3
2.2 Définition des Web services	3
2.2.1 Évolution des Web services	4
2.3 Caractéristiques des Web services	5
2.4 Application des Web services	6
2.5 Principaux acteurs dans la technologie des Web services	6
2.6 Architecture générale des Web services	7
2.6.1 Architecture de base	7
2.6.2 Architecture en couche	8
2.6.2.1 L’infrastructure de base (Discovery, Discription, Exchange)	8
2.6.2.2 Couches transversales (Security, Transactions, Administration, QoS)	8
2.6.2.3 La couche Business Processus	9
2.7 Découverte des Web services	9
2.8 Composition de Web services	10
2.9 Types de composition	10

2.9.1	La composition statique	10
2.9.2	Composition dynamique	11
2.10	Classification des Web services selon leurs modèles d'interaction et de composition	11
2.10.1	Modèle de Web service atomique	11
2.10.2	Modèle de Web service comportemental	12
2.11	Principales technologies de communication des Web services	12
2.11.1	Protocole SOAP	12
2.11.2	Langage WSDL	13
2.11.3	Annuaire UDDI	14
2.12	Avantages et Inconvénients des Web services	15
2.12.1	Avantages	15
2.12.2	Inconvénients	15
2.13	Conclusion	16

3 État de l'art sur l'évaluation des performances 17

3.1	Introduction	17
3.2	Définition de l'évaluation de performance	18
3.3	Le rôle de l'évaluation de performance	18
3.4	Les paramètres de performance	18
3.5	Les étapes d'évaluation de performance	19
3.6	Techniques d'évaluation de performance	19
3.6.1	Mesures directes	20
3.6.2	Modélisation	20
3.6.2.1	Les méthodes analytiques	21
3.6.2.2	Simulation	22
3.7	Les chaines markoviennes	23
3.7.1	Les processus stochastiques markoviens	24
3.7.2	Définition d'une Chaîne de Markov	24
3.8	La théorie de files d'attente	24
3.8.1	Définition d'une file d'attente	25
3.8.2	Origine de la théorie des files d'attente	25
3.8.3	Application des files d'attente	26

3.8.4	Le comportement d'une file d'attente	26
3.9	Notation des modèles de files d'attente	27
3.10	Types des files d'attente	28
3.10.1	Systèmes de files d'attente avec un serveur unique	28
3.10.2	Systèmes de files d'attente avec plusieurs serveurs	28
3.10.2.1	Système de files d'attente avec plusieurs serveurs en parallèles	28
3.10.2.2	Système de files d'attente avec plusieurs serveurs en série (tandam)	28
3.11	Les disciplines de service	29
3.12	Les réseaux de files d'attente	29
3.12.1	Définition	29
3.12.2	Les réseaux ouverts	29
3.12.2.1	Processus d'arrivée	30
3.12.2.2	Routage des clients	30
3.12.3	Les réseaux fermés	31
3.12.4	Les réseaux mixtes	31
3.13	Les mesures de performance	31
3.14	Formule de Little	32
3.15	Comparaison des techniques d'évaluation de performances	33
3.16	Conclusion	33
4	Simulation et Analyse de Performance d'un système des Web services	34
4.1	Introduction	34
4.2	Position du probleme (le système)	34
4.3	Présentation du modèle	35
4.4	Caractéristiques du réseau	36
4.5	Description de l'environnement du travail	36
4.6	Les étapes de la simulation avec le Java Modeling Tools	37
4.7	Les paramètres d'entrés de ce modèle	38
4.8	Résultats de la simulation	38
4.9	Interprétation des résultats	39
4.10	Validation du simulateur	43
4.11	Conclusion	43

Conclusion générale

44

Bibliographie

45

Table des figures

2.1	Structure des Web services [24].	7
2.2	Architecture en Pile des Web services [7].	9
2.3	Structure de message SOAP [7].	13
2.4	Les trois types de l'annuaire UDDI [7].	14
3.1	Processus d'évaluation de performance [38].	19
3.2	Technique d'évaluation de performance [34].	20
3.3	Types de simulations [10].	22
3.4	Représentation d'une file d'attente [13].	26
3.5	Un réseau de files d'attente ouvert [23].	30
3.6	Un réseau de files d'attente fermé [23].	31
4.1	Modèle de Web services.	35
4.2	Le modèle du système des Web services.	36
4.3	Modèle a simuler	37
4.4	Choix des paramètres.	37
4.5	Début de la simulation	38
4.6	Le nobmre moyen de clients dans le systeme.	40
4.7	Le temps moyen d'attente dans la file.	41
4.8	La durée de séjour dans le système.	42

Liste des abréviations

CORBA : (Common Object Request Broker Architecture).

DCOM : (Distributed Component Object Model).

RMI : (Remote Method Invocation).

XML : (eXtensible Markup Language).

UDDI : (Universal Description, Discovery and Integration).

SOAP : (Simple Object Access Protocol).

WSDL : (Web Services Description Language).

JMT : (java Modeling Tools).

HTTP : (Hyper Text Transfer Protocol).

W3C : (World Wide Web Consortium).

URI : (Uniform Resource Identifier).

IBM : (International Business Machines).

XMLANG : (XML Business Process Language).

WSFL : (Web Service Flow Language).

URL : (Uniform Resource Locator).

QoS : (Quality of Service).

DES : (Simulation à Événements Discrets).

PAPS : (Premier Arrive Premier Servi).

FIFO : (First In First Out).

LIFO : (Last In First Out).

FIRO : (First In Random Out).

PS : (Processor Sharing).

1

Introduction générale

L'avènement du Web comme support de communication entre les applications, et les organisations a permis l'essor des Web services , qui s'inscrivent dans la continuité des initiatives de l'information distribuée.

Les Web services sont des applications modulaires auto-descriptives pouvant être publiées, localisées et invoquées via le Web grâce à des protocoles basés sur XML tel que UDDI, SOAP, WSDL. Ces protocoles constituent des standards d'interopérabilité qui offrent un accès d'une manière transparente, indépendamment des plateformes et des langages utilisés.

L'objectif des Web services, est de faciliter l'accès aux applications entre les entreprises, ainsi que de simplifier les échanges de données. Ils poursuivent un vieux rêve de l'informatique distribuée.

De nos jours, plusieurs compagnies et organisations implémentent leurs services sur Internet ; En particulier, si aucun Web service ne peut satisfaire une requête d'un client, il devrait y avoir une possibilité de combiner un ensemble de Web services existant pour accomplir cette requête. C'est ce qui est désigné par la composition des Web services.

L'évaluation de performance est un procédé qui consiste à déterminer la qualité des résultats optimaux obtenus, pour les entrées/sorties des systèmes dans le cadre de l'accomplissement de la tâche qui leur a été assignée.

La modélisation et l'évaluation de performance, est une étape indispensable, qui se réalise en fonction de ce que l'on connaît de chacune des composantes du système. Elle est donc devenue un outil de plus en plus utilisée, pour étudier et développer les méthodes mathématiques, et prédire et optimiser les performances du système.

Dans le cadre de ce travail, nous nous intéressons à l'évaluation de performance d'un système des Web services. Le système considéré a une source Internet et deux types de services simples et composites.

A cet effet nous avons modélisé notre modèle par un réseau de files d'attente en Tandem, et à 3 stations. Les résultats sont obtenus en utilisant le simulateur (Java Modeling Tools).

Notre mémoire est organisé comme suit :

- **Le premier chapitre** : est consacré à l'état de l'art sur la technologie des Web services. Nous donnons la définition et l'évolution des Web services ainsi que les caractéristiques et les avantages des Web services et les principaux standards qu'il supporte.
- **Dans le deuxième chapitre** : nous présentons les notions de bases relatives à la théorie des files d'attente, aux réseaux de files d'attente et à la méthodologie d'évaluation de performance.
- **Le troisième chapitre** : est le noyau de notre travail, à savoir l'application de l'outil d'évaluation de performance (simulation) sur les Web services, et l'interprétation des résultats.
- Nous terminons, par une conclusion générale, et une bibliographie.

2

État de l'art sur les Web services

2.1 Introduction

Les Web services prennent leurs origines dans l'informatique distribuée et dans l'avènement du Web. Ils poursuivent un vieux rêve de l'informatique distribuée, où les applications pourraient interopérer à travers le réseau, indépendamment de leur plateforme et de leur langage d'implémentation.

Les Web services regroupent un ensemble de technologies bâties sur des standards XML (eXtensible Markup Language), HTTP (Hyper Text Transfer Protocol). Ils permettent de créer des composants logiciels distribués, d'écrire leur interface et de les utiliser indépendamment de la plateforme sur laquelle ils sont implémentés [9].

2.2 Définition des Web services

Plusieurs définitions des Web services ont été mises en avant par différents auteurs.

- Le consortium W3C définit un Web service comme étant : " une application, ou un composant logiciel qui vérifie les propriétés suivantes " [11] :

- Il est identifié par un URI(Uniform Resource Identifier).
- Ses interfaces et ses liens peuvent être décrits en XML(eXtensible Markup Language).
- Sa définition peut être découverte par d'autres Web services.
- Il peut interagir directement avec d'autres Web services à travers le langage XML en utilisant des protocoles Internet standards [15].

Tandis qu'IBM (International Business Machines) donne la définition suivante :

- " Les services Web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-continues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le Web" [29].
- Un Web service est une application accessible à partir du Web. Il utilise les protocoles Internet pour communiquer, et un langage standard pour décrire son interface [30].

2.2.1 Évolution des Web services

L'une des explications la plus intrigante (étonnante) sur l'émergence des Web services est celle proposée par M. Frank Moss, la poussée technologique des Web services est née de la déconfiture économique (échec) récente du secteur des hautes technologies. Il prétend qu'avec chaque récession de l'économie, une percée technologique majeure de l'informatique s'opère. Il indique que la récession de 60-70 a permis de passer des " mainframes " monolithique à l'informatique distribuée (au niveau départemental). Que celle de 80-81 a vu naître les systèmes d'opérations ouverts (Unix), et le PC (Personal Computer), et que celle 90-91 a permis l'éclosion de l'architecture (client/serveur) [26].

Chaque baisse majeure de la vitalité de l'industrie informatique, permet l'arrivée de nouvelles solutions radicales, qui brisent l'impasse du paradigme informatique précédent, et a pour conséquence d'augmenter la productivité et les profits, tout en décentralisant le pouvoir informationnel vers les utilisateurs. Les Web services sont la solution du paradigme actuel enfermant les entreprises dans une architecture (client/serveur), qui requière (nécessite) des hauts niveaux d'intégrations (et d'important capitaux) de ses différents composants propriétaire [26].

Plusieurs autre contextes favorisent la naissance des Web services. Le phénomène P2P (peer-to-peer) est en croissance exponentielle. cette technologie permet a des utilisateurs individuels de partager des fichiers de tout format et de toute nature, via Intranet [26].

Les services Web sont aussi issus des différentes technologie de programmations distribuées (RMI, DCOM) qui les on précédées. D'autre technologie de programmation, distribuées ont aussi pavé aux

Web services [26].

2.3 Caractéristiques des Web services

Les Web services possèdent les caractéristiques qui leurs permettent une meilleure intégration dans les environnements hétérogènes :

- **Basé sur XML** : Les données dans les protocoles et les technologies des Web services sont représentées en utilisant XML, ces technologies peuvent être interopérables. Comme un transport de données, XML élimine toute dépendance de gestion de réseau, du système d'exploitation, ou de la plateforme liée à un protocole [9] ;
- **Faiblement couplés** : Dans le développement de logiciels, le couplage se rapporte typiquement au degré de dépendance entre les composants/modules logiciels. Contrairement aux composants fortement couplés (tels que CORBA), les Web services sont autonomes et peuvent fonctionner indépendamment les uns des autres [9] ;
- **Auto-descriptif** : Les Web services ont la capacité de se décrire d'une manière qui peut être facilement reconnu. Ainsi, l'interface, les informations de localisation et l'accès au Web service est identifié par n'importe quelle application externe [9] ;
- **Modulaire** : Les Web services fonctionnent de manière modulaire. Cela signifie qu'au lieu d'intégrer dans une seule application globale toutes les fonctionnalités, on crée plusieurs applications spécifiques et on les fait inter-opérer entre elles, et qui définissent chacune de ses fonctionnalités [9] ;
- **Réutilisable** : Une fonctionnalité, développée sous forme de Web service, peut être réutilisée et combinée à d'autres fonctionnalités afin de composer de nouveaux services [9] ;
- **L'interopérabilité** : Elle permet à des applications écrites dans des langages de programmation différents de s'exécuter sur des plateformes différentes, puis de communiquer entre elles, en manipulant différents standards que ce soit XML ou les protocoles d'Internet. Les Web services garantissent un haut niveau d'interopérabilité des applications et ceci indépendamment des plateformes sur lesquelles elles sont déployées et des langages de programmation dans lesquels elles sont écrites [19].

2.4 Application des Web services

L'application des Web services est multiple, on les retrouve dans les domaines du B2C, B2B aussi dans les domaines de gestion, par exemple gestion de stock, gestion commerciale, etc...

B2C (Business to Consumer) : désigne l'ensemble des architectures techniques et logiciels informatiques, permettant de mettre en relation des entreprises directement avec les consommateurs. Les entreprises déterminent leurs "business to consumer" en fonction d'une variété de critères : nature du produit (il peut s'agir d'un bien ou d'un service, d'un bien durable ou éphémère), degrés de finalisation du produit (en fonction de la possibilité qui est offerte, ou non, au client de personnaliser son achat), type de transaction (ce critère permet de distinguer la relation bilatérale (entre l'acheteur et le vendeur) de la relation multilatérale (entre un vendeur et plusieurs potentiels acheteurs). Ces critères ont un impact direct sur les investissements de l'entreprise.

B2B (Business to Business) : représente l'ensemble des relations commerciales entre les entreprises et les professionnels (entreprises, administrations, artisans, professions libérales, associations...). On l'appelle aussi "commerce interentreprises".

Les Web services peuvent être utiles dans la plupart des scénarios applicatifs, lorsque la communication peut être établie sur un modèle bidirectionnel (requête/réponse). C'est néanmoins loin d'être aussi limitatif, beaucoup d'autres modèles peuvent avoir recours aux Web services, sans même que vous en rendiez compte. Les entreprises qui mettent à disposition leurs Web services permettent aux développeurs intéressés par ses fonctionnalités, de les réutiliser sans avoir à les recoder. Le principe des Web services permet d'avoir un partage des fonctionnalités et facilite grandement le développement [39].

2.5 Principaux acteurs dans la technologie des Web services

Les principaux acteurs dans la technologie des Web services sont [15] :

- **Le fournisseur de service** : c'est le propriétaire du service.
- **Le client (ou le consommateur de service)** : c'est un demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service.
- **L'annuaire des services** : c'est 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.

2.6 Architecture générale des Web services

Techniquement, un Web service peut être perçu comme étant une interface décrivant une collection d'opérations accessibles via le réseau, à travers des messages XML standardisés. D'un point de vue technique, la description d'un Web service inclut tous les détails nécessaires à l'interaction avec le service constituant, ce qu'on appelle l'architecture des Web services, par exemples, le format des messages, les signatures des opérations, le protocole de transport et la localisation du Web service [25].

L'interopérabilité est l'objectif premier des Web services. Pour permettre cet échange d'information entre des applications distantes, les Web services sont composés des couches standards. Deux types d'architecture existent pour les Web services : La première dite de **référence**, elle contient trois couches principales. La seconde architecture est plus complète, elle utilise les couches standards de la première architecture en ajoutant au-dessus d'autres couches plus spécifiques, elle est appelé architecture **étendue** ou encore en Pile [25].

2.6.1 Architecture de base

Architecture de base : est l'architecture de référence, elle se base sur les trois concepts (le fournisseur, le client, l'annuaire) [12]. Les interactions de base qui existent entre ces trois éléments sont les opérations de publication, de recherche, d'invocation et de lien . Pour bien comprendre le fonctionnement de cette architecture, nous expliquons ci-dessous le rôle de chacun des éléments précédents donner par la figure 1.1 [24] :

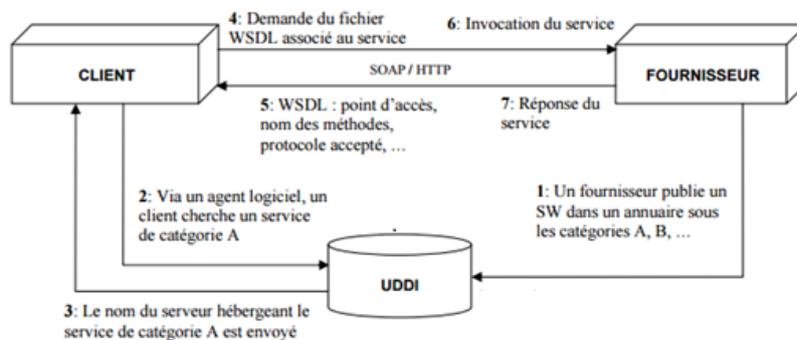


FIGURE 2.1 – Structure des Web services [24].

- Le fournisseur de service crée le Web service , puis publie la description de son service dans un annuaire de service, en vue d'être localisé par des clients [12].
- L'annuaire de service rend disponible la description de son service, pour n'importe quel demandeur de service [12].
- Les clients localisent leurs besoins en terme de services, en effectuant des recherches sur les annuaires de services [12].
- Une fois le service localisé, le client se lie au fournisseur pour invoquer le service [12].
- Sur la base des informations définies dans la description du service, le client entreprend une interaction [12].

Actuellement, SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) et UDDI (Universal Description, Discovery and Integration) sont les trois standards qui constituent l'architecture services Web. Ensemble, ils résolvent les problèmes de l'hétérogénéité des systèmes pour l'intégration d'applications en ligne [12].

2.6.2 Architecture en couche

Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres. La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. Nous apportons une explication de la mise en relief des trois types de couches [8] voir (Figure 1.2) :

2.6.2.1 L'infrastructure de base (Discovery, Discription, Exchange)

Ce sont les fondements techniques établis par l'architecture de référence. Nous distinguons les échanges des messages établis par SOAP, la description de service par WSDL et la recherche de services Web que les organisations souhaitent utiliser via le registre UDDI [8].

2.6.2.2 Couches transversales (Security, Transactions, Administration, QoS)

Ce sont des couches qui rendent viable l'utilisation effective des Web services dans le monde industriel [8].

2.6.2.3 La couche Business Processus

Cette couche permet l'intégration (composition) de services Web, elle établit la représentation d'un " Business Process " comme un ensemble de Web services. De plus, la description de l'utilisation des différents Web services composant ce Processus est disponible par l'intermédiaire de cette couche [8].

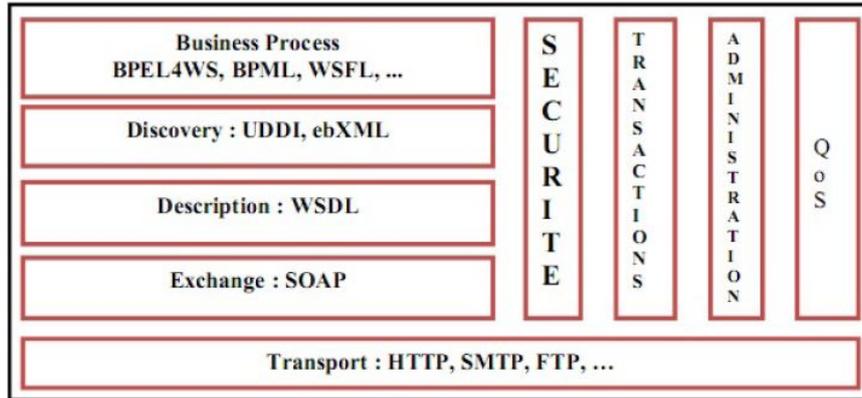


FIGURE 2.2 – Architecture en Pile des Web services [7].

La technologie des Web services offre de fortes potentialités, pour surmonter les problèmes d'interopérabilité des systèmes. Elle constitue un cadre prometteur pour l'intégration des applications, et pour la gestion des interactions entre divers partenaires dans un environnement distribué, hétérogène, ouvert et versatile qui est le Web [8].

2.7 Découverte des Web services

La découverte de Web services, c'est à dire localisation des Web services qui fournissent une fonctionnalité particulière et qui répondent aux propriétés demandées par l'utilisateur. La découverte est une étape importante qui permet la réutilisation des service. En effet, il faudra être en mesure de trouver un service afin de pouvoir en faire usage.

En exploitant les technologies offertes par Internet et en utilisant un ensemble de standards pour la publication, la recherche , l'approche Web services tend à diminuer autant que possible l'intervention humaine en vue de permettre une découverte automatique des services les plus complexes.

En effet, pour réaliser son application, un développeur peut simplement interroger un moteur de recherche de services afin de trouver le service adéquat et à l'aide de langages de coordination appropriés il peut l'intégrer avec le reste des services de son application [19].

2.8 Composition de Web services

La composition ou l'agrégation de Web services, est une opération qui consiste à construire de nouvelles applications, ou services appelés services composites ou agrégats par assemblage de services déjà existants, nommés services basiques ou élémentaires (simples).

La composition spécifie quels services doivent être invoqués, dans quel ordre et sous quelles pré-conditions [17].

Un Web service est dit composite lorsque son exécution implique des interactions avec d'autres Web services, et des échanges des messages entre eux afin de faire appel à leurs fonctionnalités [9].

La composition de Web services vise essentiellement quatre objectifs [17] :

1. Créer de nouvelles fonctionnalités en combinant des services déjà existants.
2. Résoudre des problèmes complexes auxquels aucune solution n'a été trouvée.
3. Faire collaborer plusieurs entreprises ensemble.
4. Optimiser et améliorer une fonctionnalité existante.

2.9 Types de composition

La composition des Web services peut contenir deux types, une composition statique et une composition dynamique.

2.9.1 La composition statique

Est appelée aussi composition off-line, précompilée ou encore proactive. C'est une composition qui utilise des services basiques qui sont au préalable définis d'une façon figée et qui ne peuvent pas changer en fonction du contexte du client [18].

Ce type d'application est celui qui est aujourd'hui le plus utilisé, en particulier pour les industriels. Il existe deux visions de la composition statique qui sont l'orchestration et la chorégraphie [35].

- **L'orchestration** : aborde le problème de façon centralisée, où les collaborations de service Web statiques sont contrôlées par le service composé, tel un chef d'orchestre qui se charge d'ordonner les appels aux services Web et de rattraper les erreurs [35].

- **La chorégraphie** : aborde le problème de façon distribuée, chaque partenaire d'une composition, i.e. chaque fournisseur de Web service, peut réaliser une ou plusieurs tâches, chacun d'eux communicants à l'aide de Web service [35].

Ce type de composition statique s'appuie sur des langages de composition de Web services tels que :

1. XLANG (XML Business Process Langage) de Microsoft [35];
2. WSFL (Web Service Flow Langage) de IBM [35].

2.9.2 Composition dynamique

Appelée aussi composition on-line, poste compilée ou encore réactive. Elle se réfère à la sélection des services basiques "à la volée". Autrement dit, la sélection des services basiques ne peut pas être prédéfinie à l'avance mais elle sera faite au moment de l'exécution en fonction des contraintes imposées par le client. Ceci permet d'élaborer différents scénarios de composition qui offrent les mêmes fonctionnalités et qui tiennent compte de la dynamique de la situation du client [1].

Ce type de composition est encore très peu utilisé ; car il n'est pas encore assez sûr, c'est-à-dire que l'obtention et la qualité du résultat ne sont pas garanties.

2.10 Classification des Web services selon leurs modèles d'interaction et de composition

Les Web services sont classés selon leur modèle d'interaction comme suit :

2.10.1 Modèle de Web service atomique

Un Web service atomique est décrit comme une boîte noire, c'est-à-dire qu'il est spécifié en terme d'entrées/sorties ainsi que des pré-conditions et effets sans prendre en considération le fonctionnement du service [9].

Le genre de composition associée à ce modèle d'interaction est une composition séquentielle, c'est-à-dire l'exécution de chaque service composant, ne peut commencer seulement qu'après que tous les services précédents auront terminé leurs exécutions [9].

2.10.2 Modèle de Web service comportemental

Les Web services basés sur un modèle comportemental sont souvent connus sous le nom de boîte grise, ils sont décrits par l'ordre d'exécution de leurs opérations [9].

Le genre de composition dans cette classe sera concurrentielle. C'est-à-dire qu'à un moment donné plusieurs service peuvent être actifs et s'exécutent en concurrence simultanément.

2.11 Principales technologies de communication des Web services

Les Web services sont basés sur des technologies qui ont émergé comme standards Internet pour assurer l'interaction entre les opérations de recherche, de lien et de publication des Web services.

Un ensemble de spécifications considérées comme des standards ont été définies par le consortium W3C. Nous décrivons ces standards [9].

2.11.1 Protocole SOAP

SOAP " Simple Object Access Protocol " SOAP est le protocole qui assure l'échange de messages. il est basé sur XML, il permet l'échange de données structurées indépendamment des langages de programmation ou des systèmes d'exploitation. SOAP permet l'échange d'informations dans un environnement décentralisé et distribué, comme Internet. Il peut être employé dans tous les styles de communication : synchrones (sont deux processus qui se déroulent en même temps) ou asynchrones (désigne le caractère de ce qui ne se passe pas à la même vitesse), point à point ou multi-point. SOAP utilise principalement les deux standards HTTP et XML [21] :

- **HTTP** : comme protocole de transport des messages SOAP. Il constitue un bon moyen de transport en raison de sa popularité sur le Web [21].
- **XML** : pour structurer les requêtes et les réponses, indiquer les paramètres des méthodes, les valeurs de retours, et les éventuelles erreurs de traitements [21].

Le message est englobé dans une enveloppe et divisé en 2 parties : l'entête et le corps [7].

- **L'entête (Header)** : offre des mécanismes flexibles pour étendre un message SOAP sans aucune préalable connaissance des parties communicantes. Les extensions peuvent contenir des informations concernant l'authentification, la gestion des transactions, le paiement, etc [7].
- **Le corps (Body)** : offre un mécanisme simple d'échange des informations destinées au receveur du message SOAP. Cette partie contient les paramètres fonctionnels tels que le nom de l'opération à invoquer, les paramètres d'entrés et de sortis ou des rapports d'erreur [7].

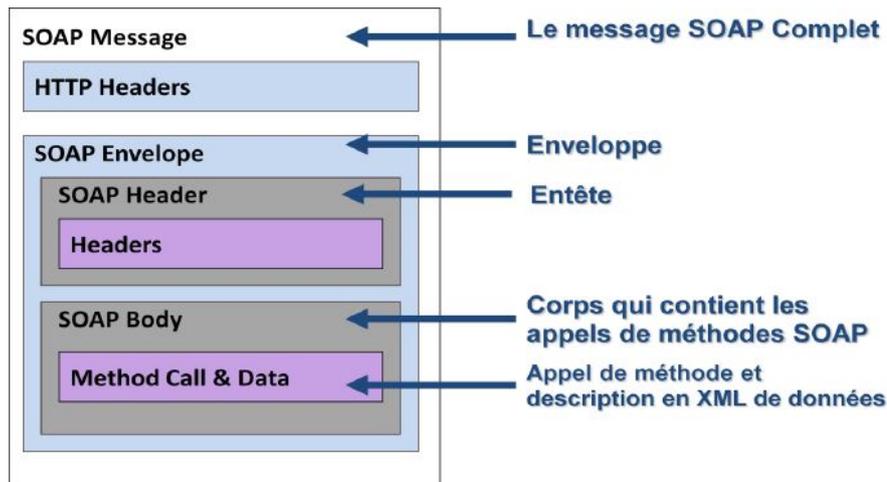


FIGURE 2.3 – Structure de message SOAP [7].

2.11.2 Langage WSDL

WSDL " Web Services Description Language ", basé sur XML, permet de décrire le Web service, en précisant les méthodes disponibles, les formats des messages d'entrée et de sortie, et comment y accéder. L'élément racine d'une description WSDL est une définition. Chaque document définit un service comme une collection de points finals ou ports. Chaque port est associé à un rattachement spécifique qui définit la manière avec laquelle les messages seront échangés. Chaque rattachement établit une correspondance entre un protocole et un type de port. Un type de port se compose d'une ou plusieurs opérations qui représentent une définition abstraite des capacités fonctionnelles du service. Chaque opération est définie en fonction des messages échangés au cours de son invocation. La structure du message est définie par des éléments XML associés à un schéma de type spécifique. Ainsi, un document WSDL utilise les éléments suivants pour la définition des services [21].

- ✓ **Types** : " Data types " est l'élément qui définit les types de données utilisées dans les messages échangés par le Web service. Une fois définie, les "Data types", ou type peuvent être référencés dans n'importe quel message.

- ✓ **Message** : qui définit d'une manière abstraite des données transmises.
- ✓ **Opération** : décrit d'une manière abstraite les actions supportées par le service. Il permet d'incorporer une séquence de messages corrélés sans avoir à spécifier les caractéristiques du flux de données.
- ✓ **Port Type** : le " PortType " est un groupement logique ou une collection d'opérations supportées par un ou plusieurs protocoles de transport, il est analogue à une définition d'un objet contenant un ensemble de méthodes.
- ✓ **Binding (Rattachement)** : est un protocole de communication, qui spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP...).
- ✓ **Port** : est une adresse d'accès au service.
- ✓ **Service** : qui regroupe un ensemble de ports, indique les adresses de port de chaque liaison.

2.11.3 Annuaire UDDI

UDDI "Universal Description, Discovery and Integration" est une norme d'annuaire de Web services appelée via le protocole SOAP. Pour publier un nouveau service Web, il faut générer un document appelé Business Registry, il sera enregistré sur un UDDI Business Registry Node. Le Business Registry comprend 3 parties [21] :

- **Pages blanches** : cette page contient toutes les informations jugées pertinentes pour identifier l'organisation (noms, adresses, contacts, identifiants des entreprises enregistrées).
- **Pages jaunes** : elle contient les informations permettant de classer les entreprises, notamment l'activité, la localisation, etc.
- **Pages vertes** : elles comportent les informations techniques sur les services proposés.

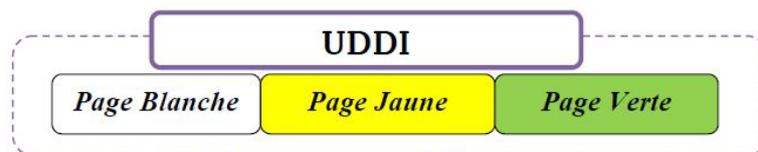


FIGURE 2.4 – Les trois types de l'annuaire UDDI [7].

Le protocole d'utilisation de l'UDDI contient trois fonctions de base :

1. **Publish** : pour enregistrer un nouveau service,
2. **Find** : pour interroger l'annuaire,
3. **Bind** : pour effectuer la connexion entre l'application cliente et le service.

Comme pour la certification, il est possible de constituer des annuaires UDDI privés, dont l'usage sera limité à l'intérieur de l'entreprise [21].

2.12 Avantages et Inconvénients des Web services

L'utilisation des Web services engendre plusieurs avantages dont on peut citer [1] :

2.12.1 Avantages

- Les Web services fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les Web services utilisent des standards et protocoles ouverts (XML).
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les Web services peuvent fonctionner au travers de nombreux pare-feux sans nécessiter des changements sur les règles de filtrage.

Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les Web services existants.

2.12.2 Inconvénients

- Les normes de Web services dans certains domaines sont actuellement récentes.
- Les Web services ont de faibles performances par rapport à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.

- lors de l'utilisation du protocole HTTP, les Web services peuvent contourner les mesures de sécurité mises en place à travers les firewalls (pare-feu) [1].

2.13 Conclusion

Les Web services constituent une technologie idéal pour l'intégration et l'interopérabilité des systèmes répartis, ils sont des applications accessibles par l'échange de documents XML entre deux URL.

Les Web services utilisent des protocoles Internet normalisés comme infrastructure de base qui repose sur les standards : SOAP, UDDI et WSDL .

Ils permettent une souplesse d'utilisation, et une accélération du développement d'applications. Ils offrent une interaction sur le Web indépendamment de toute plate-forme et de tout langage ainsi que les systèmes d'exploitation, ce qui implique leur adoption par les différentes organisations commerciales et industrielles offrant leurs services à travers le Web.

Dans le chapitre suivant, on présentera un état de l'art sur l'évaluation de performance, on donnera les différentes techniques d'évaluation de performance.

3

État de l'art sur l'évaluation des performances

3.1 Introduction

En raison de leur évolution continue et rapide le développement de nouveaux systèmes de communication devient de plus en plus complexe. Il est alors judicieux d'étudier le comportement de ces systèmes, avant leur déploiement sur le terrain, a fin de comprendre et de régler d'éventuels problèmes qui pourrait affecter ces systèmes.

La modélisation permet de représenter un système, par des paramètres et de considérer les relations entre ces différents paramètres.

L'évaluation de performance permet d'analyser l'effet de ces paramètres. Alors, l'évaluation de performance est nécessaire pour la conception, et le choix de la politique à adapter au système, pour atteindre les performances souhaitées malgré les aléas [37].

3.2 Définition de l'évaluation de performance

L'évaluation de performance d'un système est un domaine qui prend de plus en plus d'importance, il devient en effet inconcevable de construire un système (système informatique, système de production, etc) sans avoir auparavant mené une analyse de performance afin de construire un système adapté, conforme aux objectifs du cahier des charges [16].

L'évaluation de performance s'intéresse au calcul des paramètres (indices) de performance d'un système [38].

Les paramètres de performance que l'on souhaite obtenir sont de différents ordres, en fonction du système considéré [5].

3.3 Le rôle de l'évaluation de performance

L'évaluation de performance d'un système s'avère indispensable dès l'instant où l'on souhaite obtenir les performances de celui-ci.

En effet, si l'on peut mesurer simplement les paramètres de performances requis sur le système, pourquoi recourir à des techniques coûteuses et compliquées? L'évaluation de performance peut intervenir à deux niveaux [2] :

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

3.4 Les paramètres de performance

Parmi les paramètres de performance les plus fréquents, on trouve [6] :

- **Le débit (de sortie) X** : La vitesse à laquelle les clients quittent le système ;
- **Le temps de séjour R** : Le temps écoulé entre l'instant d'arrivée d'un client et la fin de son service ;
- **Le nombre de clients Q** : Le nombre de clients présents dans le système ;
- **Taux d'utilisation d'une ressource U** : La proportion de temps pendant laquelle la ressource est occupée ;
- **la probabilité de rejet Pr** : Le taux de clients refusés par le système.

3.5 Les étapes d'évaluation de performance

L'évaluation de performance d'un système se résume en trois étapes [2] :

1. **Etape1** : Comprendre le fonctionnement du système.
2. **Etape2** : Elaborer un modèle plus fidèle aux caractéristiques et fonctionnements du système.
3. **Etape3** : Évaluer les performances du système selon le formalisme du modèle.

L'évaluation de performance d'un système peut être schématisée de la façon suivante [5] :

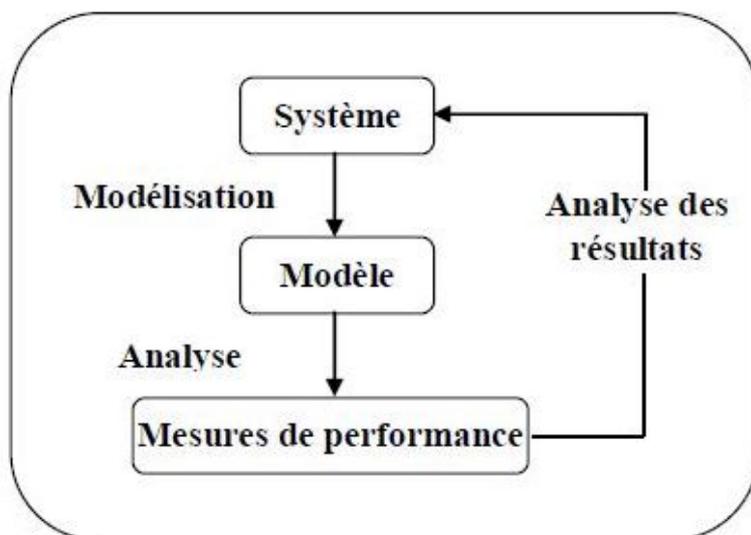


FIGURE 3.1 – Processus d'évaluation de performance [38].

Ce schéma se décompose en une étape de modélisation permettant de passer du système au modèle, et une étape d'analyse de performance du modèle.

3.6 Techniques d'évaluation de performance

Les différentes techniques d'évaluation de performance d'un système sont schématisées dans la figure 2.2 [34]. Elles peuvent être classées en trois grandes catégories : l'obtention des mesures directes sur le système (par la détermination des valeurs directes des critères de performance), les méthodes analytiques et la simulation [34].

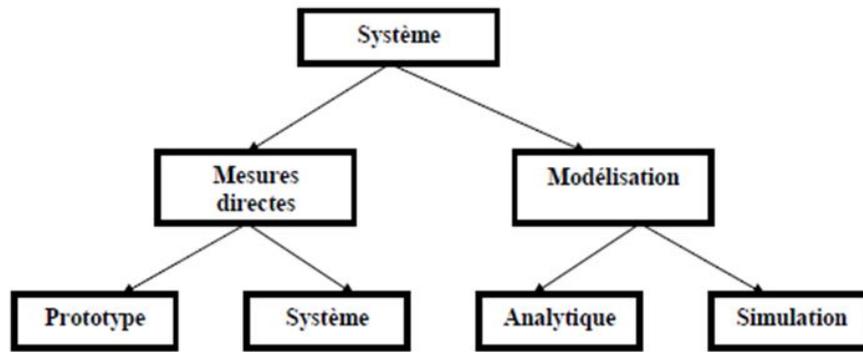


FIGURE 3.2 – Technique d'évaluation de performance [34].

3.6.1 Mesures directes

La méthode des mesures directes est la seule technique qui peut offrir " l'image réelle ", l'état d'un système réel, en tenant compte de toutes les caractéristiques de celui-ci. Elle a cependant plusieurs inconvénients. En effet les caractéristiques du système sont extrêmement variables et imprévisibles et par conséquent, les données obtenues à un moment donné sur le système ne permettent pas toujours de prévoir le comportement du système dans d'autres conditions.

On ne peut pas utiliser cette première technique l'orsque une partie du système ou bien le système dans son intégralité n'existe pas encore [34].

3.6.2 Modélisation

La modélisation a un objectif bien déterminé, qui consiste à décrire de façon simple les parties du système que l'on veut analyser.

Le processus de la modélisation consiste à décomposer le système à étudier en plusieurs tâches, dans le but de simplifier son analyse. Ce symbolisme de représentation s'appuie également sur des outils théoriques et mathématiques [31].

La modélisation analytique est un formalisme mathématique pour créer un modèle traduisant le comportement et intégrant les paramètres du système réel. Il existe plusieurs méthodes de modélisation, nous citons par exemple :

- Les chaînes de Markov.
- Les files d'attente.
- Les réseaux de files d'attente qui sont orientés évaluation de performance.

- Les réseaux de Petri.

L'utilisation de modèles markoviens pour l'analyse de performance des systèmes est souvent limitée car cette technique nécessite l'identification de tous les états du système et la connaissance des vitesses ou des probabilités de changement d'état.

Les deux formalismes les plus utilisés dans un but d'évaluation de performance sont les réseaux de files d'attente et les réseaux de Petri. Leur avantage est qu'ils permettent une construction plus naturelle du modèle à partir de la description des composants du système et de leur comportement. Plutôt que d'utiliser une forme mathématique, les modèles sont alors décrits sous forme graphique, plus proche des mécanismes utilisés par le système réel. Les réseaux de files d'attente et les réseaux de Petri peuvent être analysés en étudiant leur modèle markovien [33].

Remarque : Il est intéressant d'utiliser parfois plusieurs techniques pour vérifier et valider les résultats [33].

La modélisation consiste parfois à poser une ; ou une série d'hypothèses simplificatrices afin que le modèle mathématique puisse être résolu. elle permet de donner un schéma des interactions entre les élément d'un système, et de l'exprimer par un formalisme mathématique.

La réalisation d'un modèle mathématique signifie que tous les composants du système et toutes les connexions entre les composants, sont exprimées sous la forme de relations mathématiques [31].

La sélection des caractéristiques du système devant être modélisé, est une étape très importante dans le processus de modélisation. Elle doit assurer l'adéquation de la description avec le système réel, sans introduire de complexité inutile ; ceci dépend des objectifs de l'analyse de performance [31].

L'évaluation de performances peut être divisée en deux méthodes : **Simulation** et **Les méthodes analytiques**.

3.6.2.1 Les méthodes analytiques

Permettent de dériver les paramètres de performance en fonction des paramètres temporels du modèle, tout en utilisant des solutions sous forme de formules mathématiques prédéfinies.

Ces méthodes sont très efficaces et beaucoup plus rapides que la simulation, mais la solution n'existe que pour une classe très restreinte des systèmes.

Dans les méthodes analytiques, l'évaluation de performance est alors réalisée, par la détermination des équations analytiques, qui l'est suivi d'une résolution numérique [27].

3.6.2.2 Simulation

La simulation permet d'imiter le comportement d'un système en traitant des modèles généralement plus proches de la réalité, et elle sert souvent à valider les modèles de ces derniers.

La simulation à événements discrets (DES) est généralement utilisée pour les modèles stochastiques sans aucune restriction temporelle (distribution générale ou non-Markovienne), où les probabilités d'exécution des actions sont remplacées par des échantillons des lois de distributions associées.

La simulation est répétée plusieurs fois (typiquement plusieurs millions de fois), pour que les variables aléatoires représentent bien la distribution de probabilité, et son coefficient de variation. En plus, la simulation a l'avantage d'être insensible à la taille de l'espace d'état. Mais, l'utilisation directe de la simulation est difficile et sa complexité temporelle croit avec les détails investis dans le modèle et le degré de précision (ou l'intervalle de confiance) requis.

Pour faciliter la mise en place d'une simulation, il sera plus judicieux de construire un modèle formel, pour s'assurer du bon fonctionnement du modèle avant sa mise en œuvre dans le simulateur [20].

a- Différents types de simulation

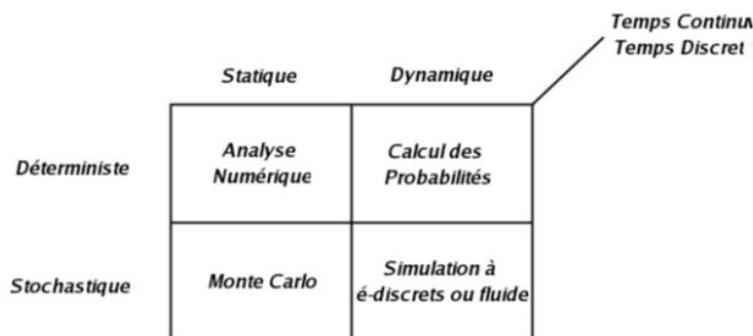


FIGURE 3.3 – Types de simulations [10].

- **Simulations statiques** : Méthode applicable seulement si le temps n'a pas d'influence, utilise des tirages aléatoires et souvent uniformes.

On donne de façon arbitraire des valeurs aux variables et on effectue les évaluations et les mesures sous ces conditions [10].

- **Simulations dynamiques** : Système qui change dans le temps. Les variables du système peuvent changer de valeurs à l'apparition de certains événements; il existe deux types de simulation dynamique : avec horloge à incrémentation fixe, et variable [10].
- **Simulations à événements discrets** : Les variables ne changent qu'à un temps défini (temps d'événements) et restent constantes sur les intervalles entre ces événements. La simulation à événements discrets permet de modéliser un système réel dont le comportement peut changer en fonction de l'apparition d'événements au cours du temps.
Par conséquent, dans une simulation à événements discrets, et au contraire d'une simulation continue, le temps évolue de manière discrète en fonction de l'occurrence des événements. Il illustre l'évolution du temps dans une simulation à événements discrets. Chaque événement représente un changement d'état, couplé avec une variable qui indique le temps auquel l'événement aura lieu [10].

Il existe quatre méthodes utilisées dans la simulation à événements discrets [10] :

1. Méthode basée sur les activités.
2. Méthode basée sur les événements.
3. Méthode des trois phases (sur les événements et les activités).
4. Méthode basée par interaction de processus.

b- Avantages et Inconvénients de la simulation

Avantages :

- Observations des états du système.
- Études des points de fonctionnement d'un système.
- Études de systèmes à échelles de temps variable.
- Études de l'impact de variables sur les performances du système.

Inconvénients :

- La conception de modèle peut nécessiter des compétences spéciales.
- Une autre forme d'analyse plus proche de la réalité est peut être nécessaire.
- Résultats difficilement interprétables.
- Résultats pas forcément généralisables.

3.7 Les chaînes markoviennes

L'étude mathématique d'un système d'attente, se fait le plus souvent par l'introduction d'un processus stochastique approprié, ainsi que la présentation de la chaîne de Markov qui sert comme outil le plus important dans l'analyse de ces phénomènes d'attente.

3.7.1 Les processus stochastiques markoviens

Un processus stochastique $X(t)$ $t \in T$, est une fonction du temps dont la valeur à chaque instant dépend de l'issue d'une expérience aléatoire. A chaque instant $t \in T$, $X(t)$ est donc une variable aléatoire. Un processus stochastique peut être donc considéré comme une famille de variables généralement non indépendantes. L'ensemble des temps T peut être discret ou continu. $X(t)$ définit l'état du processus à un instant donné t . l'ensemble noté E des valeurs que peut prendre le processus à chaque instant est appelé espace d'état et peut de même que T , être discret (fini ou infini) ou continu [2].

En fonction des valeurs possibles de T et de E , on classe les processus stochastiques, de la façon suivante [2] :

1. Processus à temps discret et à espace d'état discret.
2. Processus à temps continu et à espace d'état discret.
3. Processus à temps discret et à espace d'état continu.
4. Processus à temps continu et à espace d'état continu.

3.7.2 Définition d'une Chaîne de Markov

Une chaîne de Markov est une suite de variables aléatoires $(X_n, n \in N)$ qui permet de modéliser l'évolution dynamique d'un système aléatoire : X_n représente l'état du système à l'instant n . La propriété fondamentale des chaînes de Markov, dite propriété de Markov, est que son évolution future ne dépend du passé qu'à travers de sa valeur actuelle [22].

3.8 La théorie de files d'attente

Le formalisme de files d'attente est la technique la plus largement utilisée pour l'évaluation de performances des systèmes. Ceci s'explique par le fait qu'elle permet d'abstraire le comportement de ces systèmes de façon assez réaliste.

Dans les systèmes, de nombreuses entités partagent les ressources communes. par exemple, les messages partagent les bus de communication. En général, la ressource utilisée ayant une capacité limitée, toutes les entités ne peuvent donc pas utiliser la ressource en même temps. Ainsi, lorsqu'une première entité accède à la ressource, toutes les autres doivent attendre leur tour en file d'attente, ou alors être rejetées suivant la politique de gestion choisie.

La théorie des files d'attente, permet de représenter les ressources et les mécanismes de gestion assez fidèlement, mais permet également d'obtenir un certain nombre de résultats assez intéressants

concernant les performances du système étudié.

L'étude d'un système par la théorie des files d'attente fait appel à la notion de (serveur) de "file d'attente" et de (clients) Cette terminologie s'adapte quel que soit le domaine concerné [28].

La théorie des files d'attente est une technique de la Recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques, pour aider les décideurs dans leurs prises de décisions [23].

Les files d'attente peuvent être considérées comme un phénomène caractéristique de la vie contemporaine. On les rencontre dans les domaines d'activité les plus divers. L'étude mathématique des phénomènes d'attente, constitue un champ d'application important des processus stochastiques [23].

On parle de phénomène d'attente à chaque fois que certaines unités appelées "client" se présentent d'une manière aléatoire à des "stations" afin de recevoir un service dont la durée est généralement aléatoire.

La théorie des files d'attente est un formalisme mathématique qui permet de mener des analyses quantitatives à partir de la donnée des caractéristiques du flux d'arrivées et des temps de service[23].

3.8.1 Définition d'une file d'attente

Un système stochastique est une famille $X(t), t \in I$ de variables aléatoires définies sur un même espace de probabilité, il peut être à temps discret $I = 1, 2, \dots$ ou continu $I = [0, +\infty]$.

Une file d'attente est composé d'un certain nombre de places d'attente, d'un ou plusieurs serveurs et de clients qui arrivent, attendent, se font servir selon des règles de priorité, puis quittent le système [40].

Cette description de la file d'attente est schématisée dans la figure 2.4 [13] :

3.8.2 Origine de la théorie des files d'attente

- La théorie des files d'attente fut développée pour fournir des modèles permettant de prévoir le comportement de systèmes répondant à des demandes aléatoires [14].
- Les premiers problèmes étudiés en utilisant la théorie des files d'attente et sur la congestion du trafic téléphonique en 1909, grâce à l'article du mathématicien danois A.K. Erlang "The theory

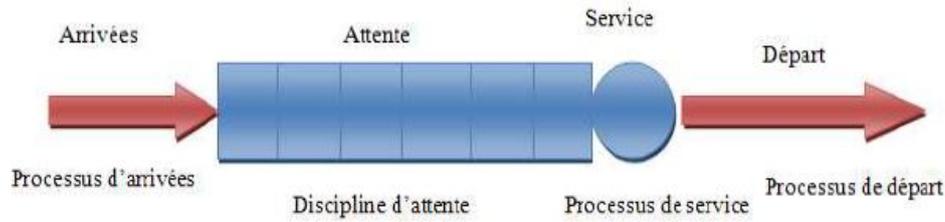


FIGURE 3.4 – Représentation d’une file d’attente [13].

of probabilities and telephone conversations”. Les premiers résultats sont variés : Erlang observe le caractère poissonnien des arrivées des appels à un central téléphonique, et le caractère exponentiel des durées des appels ; il réussit à calculer de manière relativement simple la probabilité d’avoir un appel rejeté. La notion d’équilibre stationnaire d’un système d’attente est introduite [4].

- A partir des années 30, les travaux de plusieurs mathématiciens tels que Molina, Fry, Pollaczek aux Etats-Unis, Kolmogorov et Khintchine en Russie, Palm en Suède, ou Crommelin en France permettent à la théorie des files d’attente de se développer lentement. Ce sont ensuite les années 50 qui verront l’essor important de la théorie [4].
- Actuellement ce sont les applications dans le domaine de l’analyse de performance des réseaux (téléphone mobile, Internet, multimédia, ...) qui suscitent le plus de travaux [4].

3.8.3 Application des files d’attente

La théorie des files d’attente a de nombreuses applications dans [14] :

- La gestion de trafic (réseaux de communication, compagnies aériennes, embouteillages , ...).
- La planification (opérations sur des machines de production, programmes sur des ordinateurs, l’ordonnancement, par exemple les patients dans les hôpitaux, ...).
- Le dimensionnement d’infrastructures (usines, ...).

3.8.4 Le comportement d’une file d’attente

Cinq caractéristiques permettent de décrire le comportement d’une file d’attente [40] :

- **Le processus d’arrivée des clients** : Elle caractérise l’arrivée des clients dans le système.

- **Le processus de service.**
- **La discipline de service de la file d'attente :** C'est-à-dire la manière dont les clients sont choisis pour recevoir leur service.
- **La capacité du système :** C'est-à-dire le nombre total de clients pouvant se trouver dans le système à une instant donnée.
- **Le nombre de serveurs :** Qui représente le nombre de clients qui peuvent être servis simultanément.
- **La taille de la population :** C'est le nombre total de clients pouvant potentiellement accéder au système.

3.9 Notation des modèles de files d'attente

Dans la théorie des files d'attente, la notation de Kendall (premièrement proposée par D.G. Kendall en 1953) est un système standard pour décrire les caractéristiques essentielles d'un modèle de files d'attente [3].

La notation de Kendall : a la forme $A/B/C/K/N/D$ où

- A : Distribution des temps d'inter-arrivées.
- B : Distribution du temps de service.
- C : Nombre de serveurs.
- K : Capacité du système.
- N : Nombre de clients existant dans l'univers considéré.
- D : Discipline de service.

Les différents symboles utilisés pour la caractérisation de la distribution des temps d'inter-arrivées et du temps de service sont : M (Markovien) pour une distribution exponentielle, G (Général) pour une distribution quelconque, PH pour une loi de type phase, E_k pour une distribution d'Erlang d'ordre k , D pour un temps déterministe, etc. Quand un système de file d'attente est caractérisé en utilisant seulement trois champs $A/B/C$, on sous-entend que le système est à capacité infinie, que la population est infinie et que la discipline de service est FIFO [3].

Prenons comme exemple, $M/M/1$, qui dénote une file d'attente avec un seul serveur, le processus des arrivées suit un processus de poisson, temps de service est exponentiel et le serveur est infini, la notation $M/G/K/K$ désigne une file d'attente possédant K serveurs avec un nombre fini de places, les inters-arrivées sont exponentielles et le temps des services est distribué selon une distribution générale [32].

3.10 Types des files d'attente

On distingue deux types des systèmes d'attentes : Systèmes de files d'attente qui fonctionnent avec serveur unique ou serveurs multiples.

3.10.1 Systèmes de files d'attente avec un serveur unique

Ce type de systèmes comprend un seul serveur qui offre le service. Les exemples de systèmes de files d'attente avec serveur unique sont nombreux : les petits magasins avec une seule caisse, tels que les dépanneurs, certains cinémas, certains lave-autos et établissements de restauration rapide avec guichet unique [23].

3.10.2 Systèmes de files d'attente avec plusieurs serveurs

Ce type de systèmes comprend plusieurs serveurs qui fournissent le service. Les serveurs sont en parallèle ou en série [23].

3.10.2.1 Système de files d'attente avec plusieurs serveurs en parallèles

Chaque client ne requiert le service que d'un seul serveur et tous les serveurs sont capables de fournir ce service. Dans ce type de files d'attente, on a S serveurs en parallèle. Le client entrant au système n'est pas obligé de visiter tous les S serveurs. Si chaque serveur est doté d'une file, au moment de son arrivée le client choisit l'une d'entre elles ; bien sûr il doit choisir la file la moins longue ; ou bien il rejoint une position donnée dans la file si elle est unique, puis il sera sélectionné de cette file selon la politique adoptée dans le système. Si n le nombre de clients présents dans le système est inférieur à S , ce qui revient à dire tant que les serveurs ne sont pas tous occupés, alors la file ne se constitue pas et tout client arrivant est immédiatement pris en charge par l'un des serveurs libres .

Si $n = S$, la file commence à se constituer, et tout client qui arrive doit rejoindre la file [23]

.

3.10.2.2 Système de files d'attente avec plusieurs serveurs en série (tandam)

Dans ce type de files d'attente, on a S serveurs en série. Le client entrant au système doit visiter plusieurs serveurs successifs, dans un ordre fixe pour recevoir le service. Ce type est appelé aussi système de files d'attente en cascade, on les rencontre dans une chaîne de fabrication, dans la circulation des dossiers dans une administration... etc [23] .

3.11 Les disciplines de service

La discipline de service : c'est l'ordre dans lequel les clients en attente sont traités. Parmi ces discipline, on trouve [41] :

- **Les clients sont servis dans leur ordre d'arrivée** : c'est la discipline PAPS (premier arrive premier servi) ou FIFO (first in first out).
- **Dernier arrive premier servi ou LIFO (last in first out)** : c'est ce qui est pratique en informatique pour les piles de données.
- **Aléatoire ou FIRO (first in random out)** : le serveur choisit au hasard parmi les clients présents le prochain qui sera servi.
- **Temps partage ou PS (processor sharing)** : quand plusieurs programmes sont présents simultanément dans une unité centrale, ils reçoivent chacun à leur tour une partie de leur traitement, jusqu'à achèvement de toutes les tâches.

3.12 Les réseaux de files d'attente

Pour arriver à modéliser des systèmes beaucoup plus complexes, une file d'attente simple n'est pas suffisante, il faut faire appel à des réseaux de files d'attente [23].

3.12.1 Définition

Un réseau de files d'attentes est la donné de plusieurs files d'attente entre lesquels circulent des flots de clients.

Le réseau peut être ouvert, fermé, ou mixte [23].

3.12.2 Les réseaux ouverts

Dans un réseau de files d'attente ouvert, les clients arrivent de l'extérieur, circulent dans le réseau à travers les différentes stations, puis quittent le réseau.

Le nombre de clients pouvant se trouver à un instant donnée dans un réseau ouvert, n'est donc pas limité. Afin de spécifier complètement un réseau ouvert, il faut bien sûr caractériser chaque station, mais également le processus d'arrivée des clients et le routage (cheminement) des clients, dans le réseau [23].

3.12.2.1 Processus d'arrivée

Le processus d'arrivée des clients dans le réseau est décrit, comme pour une file simple, à l'aide d'un processus de renouvellement (il est donc caractérisé par la distribution du temps d'inter-arrivée).

Si l'arrivée des clients suit un processus de Poisson, les inter-arrivées sont exponentielles et sont caractérisées par un unique paramètre : le taux d'arrivée.

Dans le cas d'un processus d'arrivée non Poissonien, ce paramètre reste intéressant, puisqu'il indique le nombre moyen de clients qui arrivent dans le système par unité de temps, mais devient insuffisant pour caractériser parfaitement l'arrivée des clients [23].

3.12.2.2 Routage des clients

Lorsqu'un client termine son service à une station, il faut préciser où ce client va se rendre : soit à une autre station, soit à l'extérieur (le client quitte alors le réseau). Il existe cependant d'autres types de routages :

- **Le routage vers la file la plus courte (routage dynamique)** : un client quittant une station choisira, parmi toutes les destinations possibles, la station qui comporte le moins de clients.
- **Le routage cyclique (routage déterministe)** : les clients quittant une station choisiront à tour de rôle chacune des stations parmi toutes les destinations possibles [23].

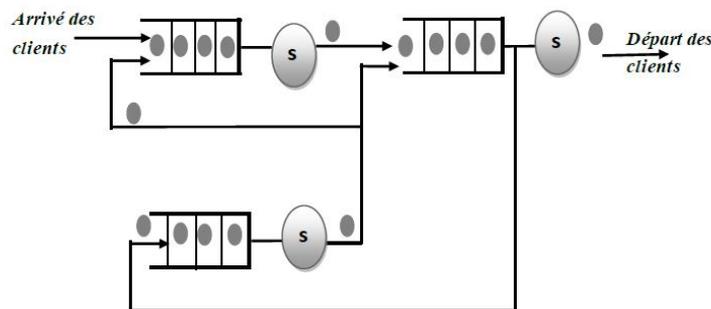


FIGURE 3.5 – Un réseau de files d'attente ouvert [23].

3.12.3 Les réseaux fermés

Dans un réseau de files d'attente fermé, les clients sont en nombre constant. Soit N le nombre total de clients du système. Il n'y a donc pas d'arrivée ni de départ de clients. La spécification d'un réseau fermé se réduit donc à celle des différentes stations et à celle du routage des clients [23].

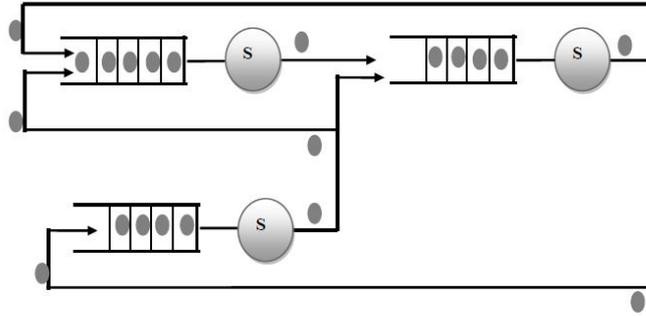


FIGURE 3.6 – Un réseau de files d'attente fermé [23].

3.12.4 Les réseaux mixtes

Un réseau est dit mixte s'il est ouvert pour certains clients et fermé pour d'autres.

3.13 Les mesures de performance

L'analyse théorique d'un modèle de files d'attente a pour objet de saisir quantitativement et qualitativement le fonctionnement du système en question. Pour cela il faut définir les critères et les mesures afin d'atteindre cet objectif. En particulier, nous allons présenter dans cette section les mesures de performances les plus répandues [22].

- **Taux d'utilisation** : si le système de files d'attente se compose d'un seul serveur, alors le taux d'utilisation est la fraction du temps pendant laquelle le serveur est occupé. Dans le cas où le nombre de clients dans le serveur est illimité, le taux d'utilisation du serveur est donné par :

$$\rho = \frac{\lambda}{\mu}$$

Avec : ρ est la probabilité que le serveur de la file soit occupé [5].

λ est le taux des arrivées.

μ est le taux de service [22].

Condition de stabilité : La condition de stabilité d'une file simple :

$$\lambda < \mu$$

exprime le fait que le taux des arrivées des clients qui arrivent à la file par unité de temps, λ doit être inférieur au taux de service de la station est capable de traiter par unité de temps μ [5].

- **Temps moyen de réponse (T) :** Le temps de réponse ou temps de séjour d'un client dans une file d'attente, est la différence entre sa date de départ et sa date d'arrivée. Le temps de réponse R_n d'un client est donné par cette formule :

$$R_n = d_n - a_n$$

Avec : d_n l'instant de départ et a_n est l'instant d'arrivée.

ou bien : Temps de réponse = Temps d'attente + Temps de service.

- **Temps moyen d'attente (ϖ) :** C'est le temps ou le client est en file d'attente en attendant d'être servi.
On a : Temps d'attente = Temps de réponse – Temps de service.
- **Longueur moyenne de la file d'attente (Q) :** La longueur moyenne de la file Q est le nombre de clients en attente dans la file.
- **Nombre moyen de clients dans le système (K) :** Le nombre moyen de clients dans le système (file d'attente + serveur) est représenté par K.

3.14 Formule de Little

Les mesures de performance : le nombre moyen de clients dans le système (attente + serveur), et le temps moyen de séjour des clients, sont liées par le théorème fondamental de Little [22].

Considérons un système d'attente quelconque en régime stationnaire , alors :

$$K = \lambda T$$

$$K_q = \lambda T_q$$

$$T = T_q + \frac{1}{\mu}$$

$$K = K_q + \frac{\lambda}{\mu}$$

Avec : λ : taux d'arrivée des clients dans le système.

K : Nombre moyen de clients dans le système.

K_q : Nombre moyen de clients dans le file.

T : Temps moyen de séjour des clients dans le système.

T_q : Temps moyen de séjour des clients dans la file.

3.15 Comparaison des techniques d'évaluation de performances

Les résultats avec les formules analytiques ne sont pas toujours disponibles. Mais s'ils existent ils seront exacts, contrairement à la simulation qui donne des résultats approximatifs [38].

Simulation	Méthodes analytiques
Résultats approximatifs	Résultats exacte si possible
Appliqué aux systèmes complexes	Pas toujours
Gourmande en temps de calcul	Peu couteuse en temps de calcul

TABLE 3.1 – Tableau de comparaison entre la simulation et les méthodes analytiques.

3.16 Conclusion

Pour évaluer les performances d'un système, la modélisation et la simulation sont des outils importants pour évaluer les délais moyens d'attente, le temps de réponse, et autre mesure d'un système donné.

Ce chapitre a été axé sur le formalisme d'attente, chaque file d'attente est caractérisée par son processus d'arrivée, taux de service et la discipline de service. Dans certains systèmes, un client accomplit son travail en passant par plusieurs stations d'où la notion des réseaux de files d'attente.

Le chapitre suivant, sera consacré au calcul des performances de notre modèle ainsi qu'à l'interprétation des résultats obtenus par la simulation.

4

Simulation et Analyse de Performance d'un système des Web services

4.1 Introduction

Dans ce chapitre, nous avons réalisé une modélisation du système des Web services par un réseau de file d'attente en tandem, la simulation nous permet de rendre visible le fonctionnement de ce réseau.

Pour accomplir notre objectif et pouvoir modéliser le trafic en réseau nous avons simulé le fonctionnement des files d'attente qui sont géré par le système FIFO. Après visualisation nous avons affiché les graphes correspondant a chacune d'entre elle, en suite nous avons discuté les différents résultats obtenus.

4.2 Position du probleme (le système)

Notre objectif dans ce mémoire est d'évaluer les performances d'un système des services Web afin d'avoir les caractéristiques de ce réseau. Dans ce système, on a une source Internet et deux types de Web services simples et composites, voir la figure 3.1.

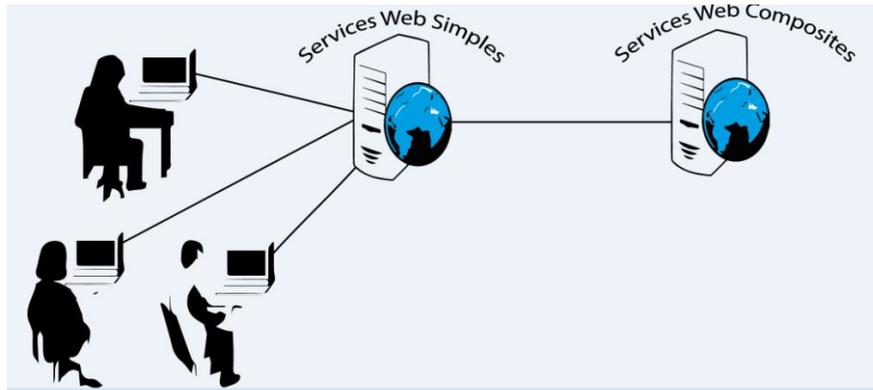


FIGURE 4.1 – Modèle de Web services.

Si le client (requête) arrive à la 1^{ère} station (Internet) il trouve le serveur libre il va extraire les informations de la requête. Ensuite, il passe à la 2^{ème} station pour faire une recherche à la réponse de sa demande dans les Web services simples. Si la requête est satisfaite, elle quitte le système avec une probabilité $1-p$ ou bien elle passe à la 3^{ème} station avec une probabilité p . A ce niveau, on remarque l'existence d'une autre source d'arrivées de clients. Ils arrivent au système lors de la composition, tel que la requête cherche une réponse à sa demande dans les Web services composite. Si la requête est satisfaite, elle quitte le système avec une probabilité q . Dans le cas d'échec, la requête tente une autre tentative avec une probabilité $1-q$ (la probabilité de retour au système). En simulant avec un simulateur à événements discrets (Java Modeling Tools), on a calculé le nombre moyen de clients dans le système, le temps d'attente et le temps de séjour dans ce réseau en faisant varier les taux des inter-arrivées au système.

4.3 Présentation du modèle

On a modélisé notre système par un réseau de files d'attente ouvert, (réseau de Jackson) en Tandem à 3 stations. Tel que on a la 1^{ère} station représente (Internet), la 2^{ème} station représente les Web services simples et la 3^{ème} station représente les Web services composites. Notre modèle est Markovien car on a les deux quantités stochastiques principales (les temps des inter-arrivées) et (la durée de service), sont des variables aléatoires indépendantes, exponentiellement distribuées. Le modèle est présenté dans la figure 3.2 :

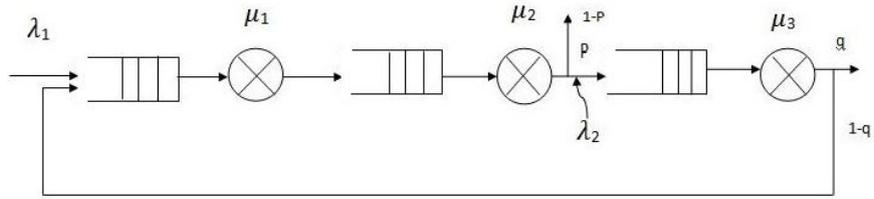


FIGURE 4.2 – Le modèle du système des Web services.

4.4 Caractéristiques du réseau

Les caractéristiques de ce réseau sont les suivantes :

- ✓ Le système comprend $S = 3$ stations en série.
- ✓ Les arrivées sont représentés par un ensemble de clients (qui représentent les requêtes).
- ✓ Ce réseau comporte trois files d'attente M/M/1 en Tandem.
- ✓ Dans chaque station, la discipline de service est : premier arrivée premier servi (FIFO).
- ✓ Nombre de clients, dans chaque file d'attente est illimité.
- ✓ Le temps des inter-arrivées et le temps de service sont indépendants.

4.5 Description de l'environnement du travail

JMT Java Modeling Tools : est un logiciel récent, libre et gratuit, développé par polytechnique de Milan, orienté à l'analyse, l'approximation et à la simulation de réseau de files d'attente.

JMT est composé de six outils de performance, qui ont comme rôle de communiquer et de modéliser les systèmes informatiques, Ces outils sont [36] :

- JSIMwiz : Cet interface décrit les évènements de simulation, destinés pour l'analyse des réseaux de files d'attente, aide à définir, les propriétés des réseaux, les probabilités des inters-arrivées et donne aussi les résultats des intervalles de confiance, et fait le contrôle des paramètres au cours de la simulation.
- JSIMgraph : Cet interface graphique utilise le même simulateur, et intègre les mêmes fonctions que JSIMwiz. Les modèles à simuler sont créés dans cette interface.
- JMVA : Cet interface est Utilisé pour l'analyse exacte ou approximative des réseaux ouverts, fermés ou mixtes des files d'attentes.
- JMCH : Cet interface est appliquée, dans la simulation des chaine de Markov.
- JABA : identifie les vraies performances des stations saturée, c'est-à-dire l'exécution des clients mixtes qui saturent simultanément plus d'une ressource.
- JWAT : Les informations importantes peuvent être analysées en utilisant les techniques de la description statistiques (histogramme ...). [36]

4.6 Les étapes de la simulation avec le Java Modeling Tools

Nous avons simulé le modèle de réseau de files d'attente avec JMT. Ces étapes préconisent, le calcul des paramètres de performance, tel que le nombre moyen de clients dans le système et le nombre moyen de clients qui sont satisfait par les deux types des services Web.

Étape1 : Pour faire la simulation, on choisit l'outil de performance JSIMgraph, on clique sur file new une fenêtre s'ouvre pour construire le modèle à simuler. Après la construction, on obtient ce schéma voir la Figure 3.3

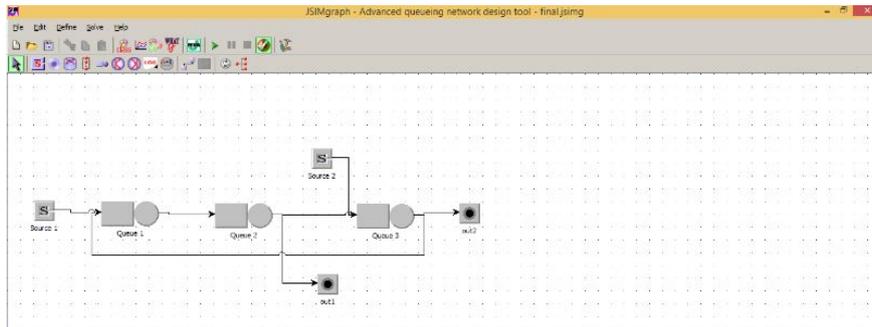


FIGURE 4.3 – Modèle a simuler .

Étape 2 : Elle est consacrée à la définition des paramètres de notre modèle tel que les taux d'arrivées, et les taux de services et les différentes probabilités. Dans cette étape, on définit toutes les performances qu'on souhaite calculer voir la figure 3.4.

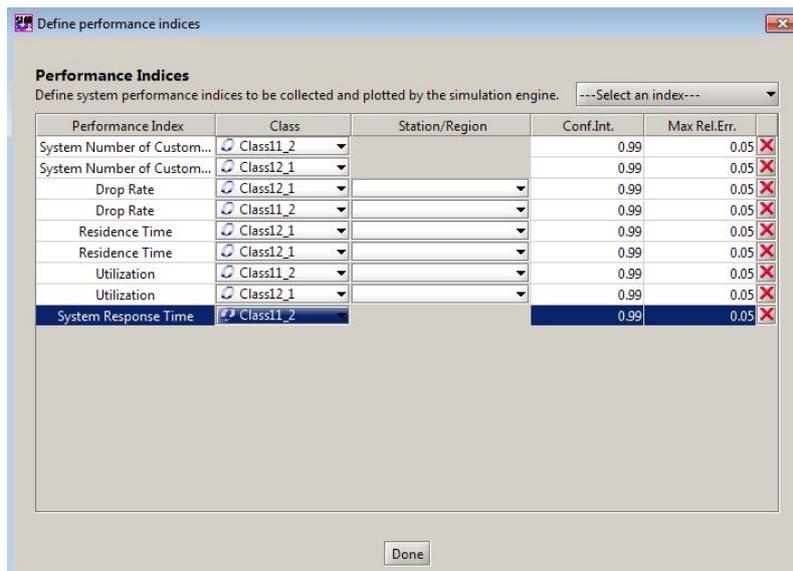


FIGURE 4.4 – Choix des paramètres.

Étape 3 (la simulation) : Au début de la simulation, on clique sur le bouton simuler. Une fenêtre de progression montre le début de la simulation voir la figure 3.5

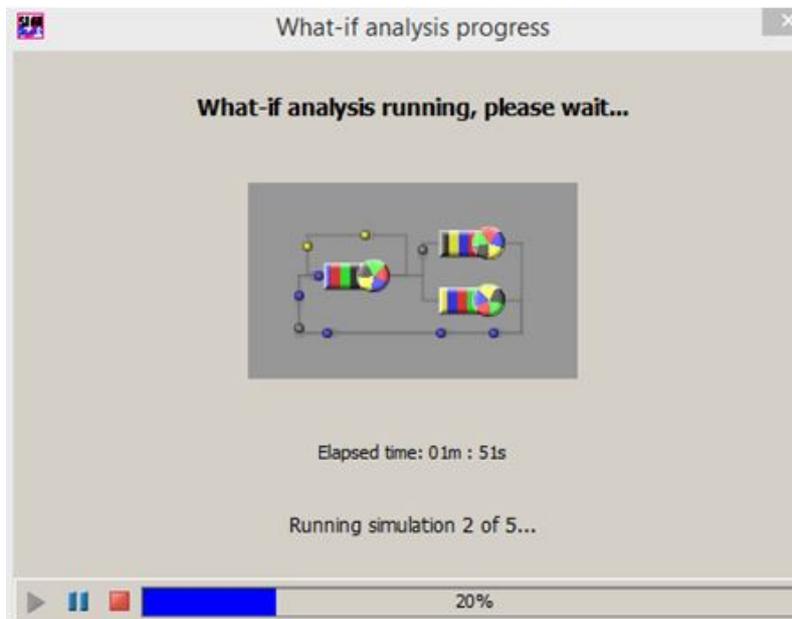


FIGURE 4.5 – Début de la simulation .

4.7 Les paramètres d'entrés de ce modèle

Les paramètres de ce modèle, sont définis par :

T : est la durée de simulation qui est le temps consacré pour faire cette étude.

λ_1 : Le taux des arrivées des clients au serveur 1.

λ_2 : Le taux des arrivées des clients au serveur 3.

μ_1 : Le taux de service du premier serveur.

μ_2 : Le taux de service du deuxième serveur.

μ_3 : Le taux de service du troisième serveur.

p : est la probabilité d'aller de la station 2 vers la station 3.

$1-p$: est la probabilité de satisfaction par les Web services simples.

q : est la probabilité de satisfaction par les Web services composites.

$1-q$: est la probabilité de retour au système.

4.8 Résultats de la simulation

Les résultats de cette simulation établis, après avoir saisi les différents paramètres, on fixe le taux des arrivées λ_2 à la station $N^{\circ}3$, et on varie le taux des arrivées à la station $N^{\circ}1$.

- Le temps maximum de la simulation est : 10000 μs .
- Et pour les différentes valeurs des paramètres d'entrées :
- $1/\lambda_1$ L'intervalle du temps moyen entre deux arrivées successives au réseau (serveur $N^\circ 1$) prend les valeurs suivantes : $1\mu s$, 1 milli.s, 1s, 1minute, 1heure.
- $1/\lambda_2$ L'intervalle du temps moyen entre deux arrivées successives au Web service composite (serveur $N^\circ 3$) égale a : $1\mu s$.
- Le taux de service $N^\circ 1$ (Internet) μ_1 égale : 1 minute.
- Le taux de service $N^\circ 2$ (Web services simples) μ_2 égale : 1 seconde.
- Le taux de service $N^\circ 3$ (Web services composites) μ_3 égale : 2 seconde.
- La probabilité de satisfaction avec les Web services simples est : $1-p = 0.4$.
- La probabilité de satisfaction avec les services composites est : $q = 0.6$.

4.9 Interprétation des résultats

Simulation 1 : Le nombre moyen de clients dans le système :

$1/\lambda_1 = 1\mu s$	$1/\lambda_1 = 1milis$	$1/\lambda_1 = 1s$	$1/\lambda_1 = 1min$	$1/\lambda_1 = 1h$
$S_1 = 0,021$	$S_1 = 3,92 * 10^{-4}$	$S_1 = 1,28 * 10^{-4}$	$S_1 = 0,001$	$S_1 = 8.667 * 10^{-9}$
$S_2 = 1,371$	$S_2 = 0,002$	$S_2 = 7,71 * 10^{-6}$	$S_2 = 0,001$	$S_2 = 5.267 * 10^{-7}$
$S_3 = 50940$	$S_3 = 1289$	$S_3 = 955$	$S_3 = 654$	$S_3 = 6.480 * 10^{-7}$
$Sys = 1467$	$Sys = 0.0011$	$Sys=0.000955$	$Sys = 0.000654$	$Sys = 2.72.480 * 10^{-21}$

TABLE 4.1 – Le nombre moyen de clients dans le système.

La 1^{ère} ligne des tableaux représente les différentes valeurs des inter-arrivées considéré dans notre étude.

On a obtenu les résultats de la dernière ligne des tableaux en utilisant la forme de produit du théorème de Jackson.

Lorsque le système est stable, la distribution stationnaire est à forme produit.

Tout se passe comme si on avait N files d'attente "indépendantes" de charge respectives P_i . Cette forme est donc extrêmement utile car elle permet d'obtenir des résultats analytiques simples sur des modèles Markoviens multi-dimensionnels en général intraitables par des méthodes numériques standard.

S_i : représente le nombre moyen de clients dans la station i , $i=\overline{1,3}$.

Sys : représente le nombre moyen de clients dans le système pour chaque taux des inter-arrivées utilisé.

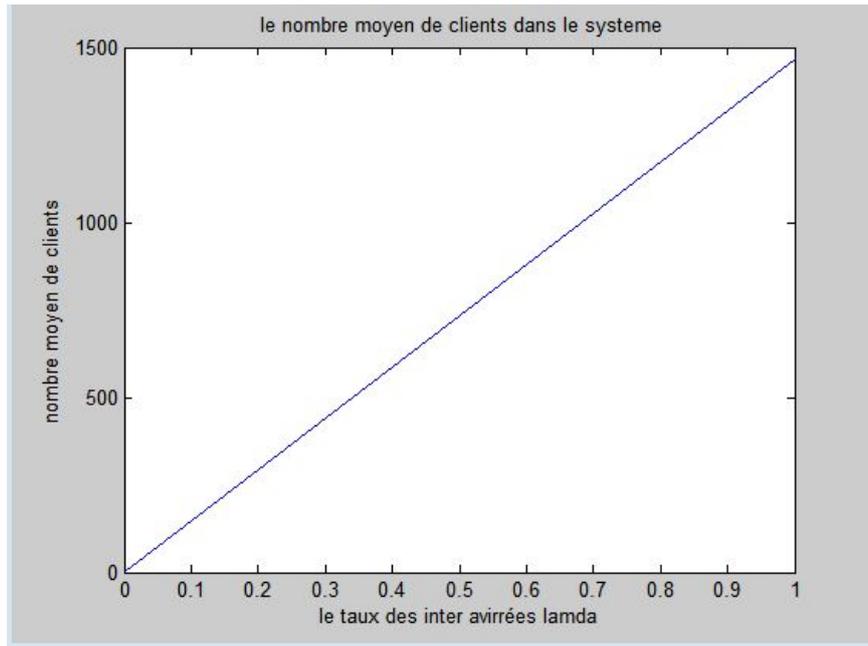


FIGURE 4.6 – Le nobmre moyen de clients dans le systeme.

Les résultats du Tableau 3.1 sont présenté dans la Figure 3.6

D’après le Tableau 3.1 et la Figure 3.6 on constate que plus l’intervalle moyen entre deux arrivées consécutives augmente inversement le nombre moyen de client diminue, il est clair que le taux des arrivées influe sur le nombre moyen de clients entrant au système. Notons que le nombre de clients qui se trouvent dans les services Web composite sont plus important que les clients trouvés dans services Web simple.

On remarque que le nombre moyen de clients qui existent dans la station 3 est plus important par apport au autres stations.

Le nombre de clients maximum existants dans le système vaux 1467 requêtes/ μs .

Simulation 2 : le temps moyen d’attente dans la file :

$1/\lambda = 1\mu s$	$1/\lambda = 1milis$	$1/\lambda = 1s$	$1/\lambda = 1m$	$1/\lambda = 1h$
$S_1 = 0,001$	$S_1 = 0,001$	$S_1 = 0,001$	$S_1 = 0,001$	$S_1 = 0,001$
$S_2 = 0,071$	$S_2 = 0,001$	$S_2 = 0,001$	$S_2 = 0,001$	$S_2 = 0,001$
$S_3 = 10370$	$S_3 = 359$	$S_3 = 196$	$S_3 = 90$	$S_3 = 0;001$
$Sys = 0.7362$	$Sys = 0.00036$	$Sys=0.000196$	$Sys = 0.00009$	$Sys = 10^{-9}$

TABLE 4.2 – Le temps moyen d’attente dans la file.

S_i : represente le temps moyen d'attente dans la file pour chaque station i , $i=\overline{1,3}$.

Sys : represente le temps moyen d'attente dans la file pour chaque taux des inter-arrivées utilisé.

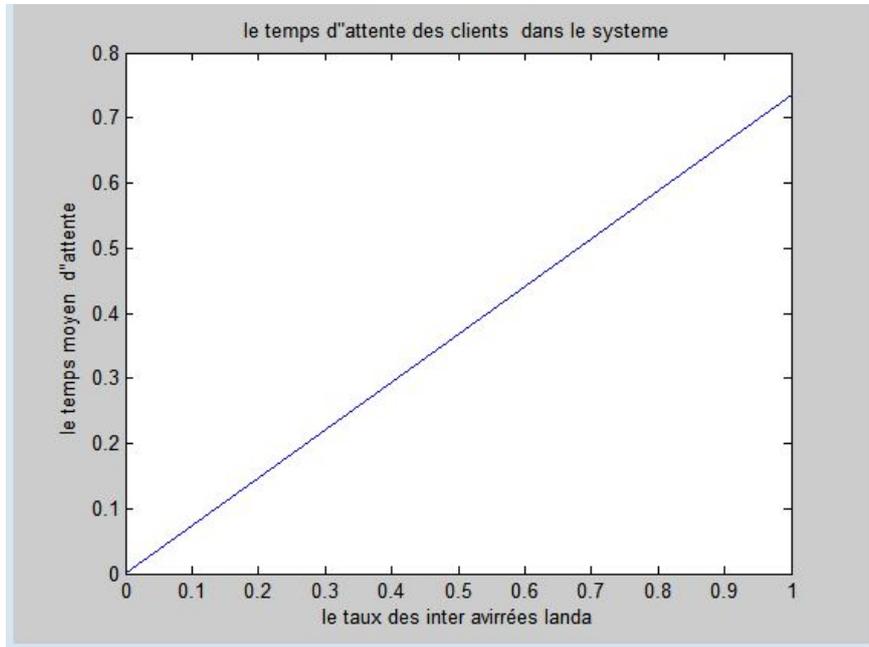


FIGURE 4.7 – Le temps moyen d'attente dans la file.

Selon le Tableau 3.2 et la Figure 3.7 on remarque que le temps d'attente dans la file augmente quand le taux des arrivées est plus grand.

Aussi on constate que le temps d'attente au niveau des stations numéro 1 et 2 sont négligeable pour les différentes valeurs de $1/\lambda_1$. car le taux de service de ces deux dernières est considérable. par contre la station numero 3 enregistre un temps d'attente important, parce que la plupart des clients ont utilisés la source numéro 2 indépendamment de la source numéro 1.

Le temps moyen d'attente dans la file idéal vau : $10^{-9}\mu s$

Simulation 3 : La durée de séjour des clients dans le système.

$1/\lambda_1 = 1/$	$1/\lambda_1 = 1\text{millis}$	$1/\lambda_1 = 1s$	$1/\lambda_1 = 1m$	$1/\lambda_1 = 1h$
$S_1 = 0.023$	$S_1 = 0.022$	$S_1 = 0.001$	$S_1 = 0.001$	$S_1 = 8.551 * 10^{-9}$
$S_2 = 1.290$	$S_2 = 1.260$	$S_2 = 0.001$	$S_2 = 0.001$	$S_2 = 5.162 * 10^{-7}$
$S_3 = 8850.726$	$S_3 = 480.823$	$S_3 = 141.617$	$S_3 = 107.775$	$S_3 = 6.399 * 10^{-7}$
$Sys = 262.60$	$Sys = 13.328$	$Sys = 0.00014$	$Sys = 0.00010$	$Sys = 2.63 * 10^{-21}h$

TABLE 4.3 – La durée de séjour des clients dans le système.

S_i : represente la durée moyenne de séjour dans le système pour chaque station $i, i=\overline{1,3}$.

Sys : represente la durée moyenne de séjour dans le système pour chaque taux des inter-arrivées utilisé.

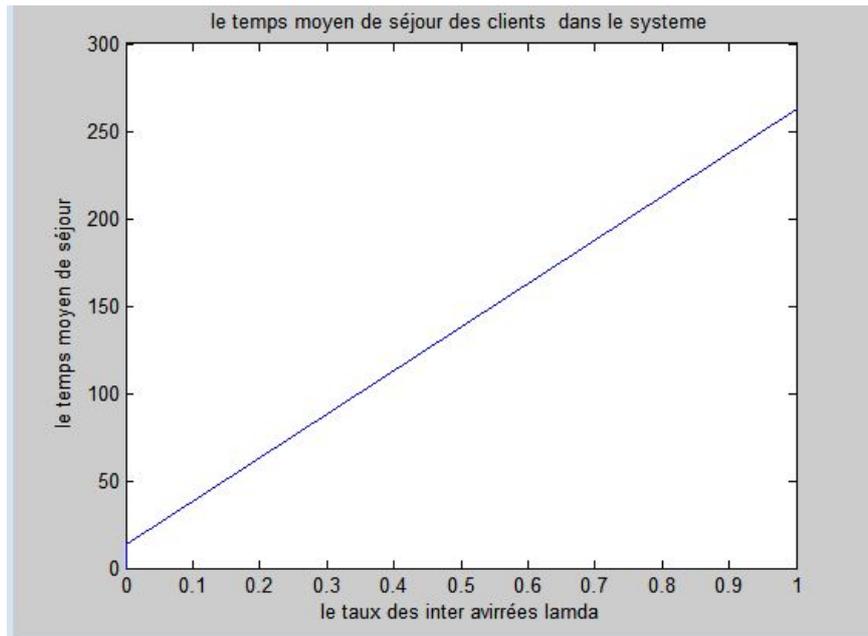


FIGURE 4.8 – La durée de séjour dans le système.

D’après les résultats obtenus dans Tableau 3.3 et dans la Figure 3.8 on constate que la durée de séjour dans le système deviens importante a chaque fois que le nombre de clients devient important.

On remarque d’après ce graphe que la durée de séjour des clients vaux : $2.63 * 10^{-21} \mu s$.

4.10 Validation du simulateur

L'étape la plus importante de la simulation est la validation du simulateur utilisé parce qu'il justifie l'exactitude des résultats obtenus, par cette simulation, on les a effectué dans des intervalles très court, donc on valide le simulateur par L'utilisation de la simulation de Monte-Carlo.

La simulation de Monte-Carlo est une technique mathématique informatisée qui permet de tenir compte du risque dans l'analyse quantitative et la prise de décision (le nombre de fois de la simulation).

4.11 Conclusion

Dans ce chapitre, nous avons étudié les caractéristiques de deux type des Web services simples et composites en termes de nombre moyen de clients, et le temps moyen de séjour et le temps moyen d'attente. Nous pouvons conclure à partir des performances obtenues pour ce modèle :

- Le nombre de clients idéal dans le système est égal : 1467 requêtes/ μs .
- Le temps moyen d'attente idéal des client dans la file est : $10^{-9} \mu s$.
- la durée de séjour idéal des clients dans le systeme vaux : $2.63 * 10^{-21} \mu s$.

Durant toute la simulation, on remarque que : le nombre de clients satisfait par les Web services simples est inférieur au nombre de clients satisfait par les Web services composite. Le taux des arrivées influe sur le nombre moyen des client dans le système, et la durée de séjour dans le système devient importante à chaque fois que le nombre de clients devient important.

Conclusion générale

Notre travail a pour objectif principal d'évaluer les performances d'un Web service. Le système considéré a une source Internet et deux types de service simples et composites. On s'intéresse au nombre moyen de clients entrant dans système, et au nombre moyen de clients qui sont satisfait par les deux types de Web services (simples et composites).

En premier lieu, nous avons modélisé notre système par un réseau de files d'attente en tandem à trois stations et un type des arrivées. Ensuite, nous avons étudié son comportement.

Nous avons résolu le problème en faisant appel à la simulation. On a utilisé un simulateur JAVA Modeling Tools (JMT).

Les résultats de la simulation montrent

, et que le taux des inter-arrivées influe sur le nombre moyen de clients entrant au système.

Les performances du modèle ont été obtenues par la simulation. Comme perspective, on propose une étude du système tel que on fixe $1/\lambda_1$ et on calcule les performances en fonction $1/\lambda_2$ où bien avec une autre technique tel que les réseaux de Petri.

Bibliographie

- [1] ADDAD, N., AND AMARA, M. W. Implémentation d'un simple service web avec axis. Master's thesis, Université de Telemcen, 2013.
- [2] AOUMER, S., AND ASLI, L. Performance de la methode de stabilité forte dans des système d'attente avec priorité. Master's thesis, université de bejaia, 2007.
- [3] ARDA, Y. *Politiques d'approvisionnement dans les systèmes à plusieurs fournisseurs et Optimisation des décisions dans les chaînes logistiques décentralisées*. PhD thesis, l'université de Toulouse, 2008.
- [4] BARTOLI, A. *Le management dans les organisations publiques*, vol. 73. Dunod, Aix en Provance, 1997.
- [5] BAYNAT, B. *Théorie des files d'attente*, vol. 8. HERMES Science publications, Paris, 2000.
- [6] BEGIN, T. *Modélisation et calibrage automatique de systèmes*. PhD thesis, Université Paris 6 Pierre et Marie Curie, 5 Décembre 2008.
- [7] BELKHITER, K. Selection de services web a base l'algorithmes génétique multi-objectifs. Master's thesis, Université de A .b Belkaid Telemcen, (2012).
- [8] BERTINO, KEIDL, M., KEMPER, E., AND FRAMEWORK, A. For contextaware adaptable web services, springer-verlag berlin heidelberg. *EDBT, LNCS 2992 4* (2004), 826–829.
- [9] BOUDJELABA, H. Sélection des web services sémantiques. Master's thesis, Université A/Mira Bejaia, Juin 2012.
- [10] BRAUNSCHWEIG, B. *La simulation sur micro-ordinateur(les modèles de dynamique des système*, vol. 169 pp. Eyrolles, 1985.
- [11] CERAMI, E. *Web services essentials*, vol. 297. edition O'reilly, fevrier 2002.
- [12] CHEMANRE, C. Implémentation d'une infrastructure collaborative asynchrone basée sur les services web. Master's thesis, Université Annaba, 2010.
- [13] COOPER, B. R. Introduction to queueing elsevier north holland united states. *edition America 13* (1981), 347.
- [14] DAD, D. Modélisation et simulation du flux un réseau pour la régulation du trafic. Master's thesis, Université d'Oran, 2011.

- [15] DALI, Y. M. Selection de services web a base de qows. Master's thesis, Université de A .b Belkaid Telemcen, (2010-2011).
- [16] DIVAN, S. Évaluation des performances des architecture logicielles à l'aide d'un réseaux de files d'attente. Master's thesis, Uneversité de Paris, 2008-2009.
- [17] DRISS, M. *Approche multi-perspective centrée exigences de composition de services web*. PhD thesis, Université de Rennes, 2011.
- [18] DUSTDAR, S., AND SCHREINER, W. A survey on web services composition. *International Journal of Web and Grid Services* 1 29, 1 (2005), 1–30.
- [19] ERL, T. Soa principles of service design. *article,Prentice Hall 117* (2007), 63–69.
- [20] GHARBI, S. Les simulateurs réseaux technologie réseaux. Tech. rep., Université Houari Boume-diene, 2013-2014.
- [21] HALIMA, R. B. *Conception implantation et expérimentation d'une architecture en bus pour l'auto-réparation des applications distribuées a base de services web*. PhD thesis, l'Université de Toulouse et de l'Université de Sfax, 14 Mai 2009.
- [22] IVO, A., AND RESING, J. Queuing theory,the netherlands ,department of mathematics and computing science of eindhoven. Tech. rep., 2002.
- [23] KEBBAS, S. Contribution à la correction et l'amélioration de la qualité de service dans une entreprise publique, en utilisant les réseaux de files d'attente. Master's thesis, Université Hadj-Lakhdar -Batna, 2013.
- [24] KEITA, I. A. *Sécurité des service web, restauration d'un message Soap après détection d'un attaque par envelopement sur un élément signé*, 11 mars 2010.
- [25] KREGER, H. *Web service conceptual architecture*, vol. 24. edition IBM Software Group, Mai 2001.
- [26] LEBLANCE, M. Les web services. Tech. rep., Hec Montréal, 2001.
- [27] MARTIN, F. *Modelisation et evaluation de performance prévisionnelle d'architectures avionique modulaire intégrée*. PhD thesis, l'onera toulouse, 1999.
- [28] MARTIN, F. *Modelisation et evaluation de performance prévisionnelle d'architectures avionique modulaire intégrées*. PhD thesis, l'onera toulouse, 1999.
- [29] MEDJAHAD, B., BOUGUETTAYA, A., AND ELMAGARMID, A. K. A semantic web enabled composition of web services. *the VLDB Journal* 18 (Novembre 2003), 333–351.
- [30] MELLITI, T. *Interopérabilité des services web complexes*. PhD thesis, Université Paris IX Dau- phine, 8 décembre 2004.
- [31] NAKANNISHI, T. *The development of aread traffic simulation system in broad areas mathematics and computers insimulation*, vol. 6. 1995, pp. 207–212.
- [32] ROBERT, P. *Réseaux et files d'attente méthodes probabilistes*. volume 85, 2000.
- [33] ROSS, J. W. Stochastic process. *Any Edition 2* (1983), 2–3.

- [34] SAD, S. *Cour évaluation et simulation*, Décembre 2009.
- [35] SCHWIND, B. *Composition automatique et adaptative des services web de météorologie*. PhD thesis, École Nationale Supérieure Des Mines de Paris, 2009.
- [36] SERAZZI G, A. M. B., AND CASALE, G. *JMT Java Modelling Tools*. user manual, October 20th, 2013.
- [37] TURKI, S., AND S HENNEQUIN, A. N. S. Performance evaluation of a failure-prone manufacturing systeme with time to delivery and stochastic demand. *edition INCOM Moscow Russia 6* (2009), 22–27.
- [38] USTHB. *Cour d'évaluation de performance des reseaux de file d'attente*. www.jmct.123.fr, 18/12/2008.
- [39] VAN, V. *Web services Communication inter langage*. PhD thesis, Ecole supérieur d'Informatique de Paris, 8 mars 2006.
- [40] XIE, X. Evaluation des performance des système de production. Tech. rep., Ecole nationale supérieure des mines saint-etienne, 2003.
- [41] YCART, B. Processus stochastiques l'école supérieure de la statistique et de l'analyse de l'information cahier de mathématiques appliquées no 14 files d'attente. *Publications Universitaires de la Tunisie* (2006).