

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

*Option: Modèle d'intelligence et décision (M.I.D)*

*Thème*

**Composition des services Cloud à base de  
séries temporelles**

**Réalisé par :**

-M<sup>elle</sup>. Bensaha Fatima

-M<sup>elle</sup>. Smail Fatma

*Présenté le 2 juillet 2017 devant le jury composé de MM.*

- Halfaoui Amel (Président)
- Bekkouche Amina (Encadreur)
- Hadjila Fethallah (Co-Encadreur)
- Belabed Amine (Examineur)

# *Remerciements*

*En premier lieu, nous remercions ALLAH de nous avoir donné la force et la patience nécessaire pour achever ce mémoire.*

*Nous tenons à remercier tout particulièrement notre Encadrant Monsieur HADJIL.A.F enseignant à l'université de Tlemcen pour son aide à réaliser ce mémoire ainsi que pour sa disponibilité, son soutien, ses conseils, sa patience et sa générosité. Son ouverture d'esprit et ses analyses pertinentes ont contribué à rendre cette étude agréable et enrichissante.*

*Nos remerciements s'adressent également à : Mme Halfaoui Amel « Président du jury ». et Mr Belabed Amine*

*Pour avoir porté un intérêt à ce modeste travail, et d'avoir accepté de l'examiner.*

*Nous tenons aussi à saluer toute la promotion de Master 2 MID*

*Merci aussi à tous mes enseignants pendant tout le parcours scolaire.*

# *Dédicaces*

*Je dédie ce modeste travail:*

*A mes très chers parents qui n'ont jamais cessé de m'encourager pour entreprendre mes études et atteindre mes objectifs.*

*A Mes très chers frères Mohamed, Sofiane et sa femme Wahiba*

*A ma sœur Amel et mes adorables neveux Ashraf et Akram*

*A tous les membres de ma famille, tantes, oncles, cousins et cousines.*

*A mes très chers Amis*

*A tous mes camarades de promotion.*

*A tous ceux qui m'ont aidé et encouragé pour l'élaboration de ce mémoire.*



***Fatima***

# *Dédicaces*

*Je dédie ce modeste travail:*

*A mes très chers parents*

*A Mes très chers Sœurs Hayet et Halima*

*A mes neveux Islem et Sirine*

*A mes très chers Amis*

*A ma famille*

 *Fatma*

**Table des matières**

Introduction générale.....	4
<b>Chapitre I : Composition des Services Cloud</b>	
I. Introduction .....	9
II. Cloud Computing.....	9
II.1 Historique.....	9
II.2 Définitions.....	11
II.3 Principe du Cloud Computing.....	12
II.4 Usages de Cloud computing.....	12
II.5 Exemples de fournisseurs d'infrastructures de Cloud.....	12
II.5.1 Exemples d'applications.....	12
II.6 Modèles des services de Cloud computing .....	13
II.6.1 SaaS (Software as a Service).....	13
II.6.2 PaaS (Platform as a Service).....	14
II.6.3 IaaS (Infrastructure as a Service).....	14
II.7 Caractéristiques du cloud .....	15
II.8 Avantages et inconvénients des services Cloud.....	16
II.9 Avantages et inconvénients du Cloud Computing .....	16
III. L'Architecture Orienté Service (SOA).....	17
III.1 Définition de SOA.....	17
III.2 Concepts de SOA.....	17
IV Service web.....	17
IV.1 Définition.....	17
IV.2 Caractéristiques.....	18
V. Problèmes de la recherche de composition des services Cloud.....	19

V.1Exemple .....	21
V.2Conclusion.....	21
<b>Chapitre II : Conception &amp; réalisation</b>	
I. Introduction.....	23
II. Présentation de la collection de test.....	23
III. Conception .....	24
III.1 Approche QA (QoS Attribute of TSG).....	24
III.2 Approche QR (QoS Relation for TSG).....	25
III.2.1 Approche de distance QR.....	26
III.3 Approche de distance finale entre QA et QR.....	29
III.4 Approche de génération.....	29
III.5 Recherche de similarité des groupes de séries temporelles.....	31
IV. Présentation de prototype.....	32
IV.1 Outils et environnement de développement.....	32
IV.2 Présentation.....	32
V. Spécification.....	42
VI. Conclusion.....	43
Conclusion générale	
Références bibliographique	

# Introduction générale

## **Introduction générale**

## *Introduction générale*

### **Contexte de travail :**

Face à l'augmentation continue des coûts de mise en place et de maintenance des systèmes d'informations, les entreprises externalisent de plus en plus leurs services informatiques en les confiant à des entreprises spécialisées comme les fournisseurs de Cloud.

L'importance de l'accès abordable à des ressources matérielles et logicielles performantes et évitant les coûts de maintenance et les problèmes de sécurité ont favorisé les gestionnaires d'institutions et acteurs de la technologie de l'information. Les entreprises migrent vers le Cloud computing.

L'intérêt principal et le plus important de cette stratégie pour les entreprises réside dans le fait qu'elles ne paient que pour les services effectivement consommés.

Le Cloud Computing est aujourd'hui le sujet phare dans le domaine des systèmes d'information et de communication, devient de plus en plus la technologie de choix comme prochaine plateforme de génération pour la conduite d'affaires. Après la virtualisation, le Cloud paraît être la révélation qui va permettre aux entreprises d'être plus performantes et adéquates et de gérer le coût des systèmes d'information plus sereinement.

Le terme Cloud Computing, est un nouveau modèle informatique qui consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui. Cette nouvelle technologie permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosse quantité d'informations.

Le Cloud computing présente une technologie prometteuse et favorable qui facilite l'exécution des applications scientifiques et commerciales. Il fournit des services flexibles et évolutifs, à la demande des utilisateurs, via un modèle de paiement à l'usage.

**Problématique :**

Le développement rapide de l'utilisation du Cloud computing conduit à Publie plus de services Cloud sur le pool de services mondial. En raison de la présence de services complexes et diversifiés, Un service simple ne peut pas satisfaire les exigences fonctionnelles existantes pour de nombreux cas du monde réel. Il est essentiel d'avoir un lot de services atomiques simples qui fonctionnent L'un avec l'autre, pour compléter un service complexe. Par conséquent, l'intégration d'un service Système de composition (SC) est fortement nécessaire dans le Cloud computing.

Les services composites (SC) peuvent être construits en combinant et en coordonnant un ensemble de services indépendants dans un processus performant appelé composition de service. En raison de leurs dépendances sur les composants externes, les services composites dépendent en particulier d'un environnement d'hébergement crédible et fiable. La plate-forme de livraison de services de prochaine génération, le Cloud computing, est considérée comme un environnement d'hébergement qui peut soutenir ces compositions de service mieux qu'un environnement d'hébergement traditionnel.

La composition du service Cloud est habituellement basée sur la QOS à long terme et une fonction économique objective.

Notre recherche se concentre sur la sélection des plans de composition basés uniquement sur des attributs non fonctionnels (Quality of-Service ou QoS).

Nous modélisons les exigences des utilisateurs finaux en tant que série de séries temporelles, en parallèle. Les fournisseurs de services cloud commercialisent leurs services (SaaS ou IaaS) à l'aide d'un ensemble de séries temporelles. Chaque série temporelle représente les valeurs d'un attribut QoS correspondant sur une longue période.

Le problème de la composition du service Cloud devient un Problème de recherche de similarité dont la requête est un ensemble de séries temporelles souhaitées, nous utilisons 2 mesures de similarités basées sur la distance euclidienne entre les valeurs de QOS et la corrélation entre les critères de QOS

Pendant la composition du service la corrélation est prévalente, où chaque attribut QOS est corrélé avec plusieurs autres attributs QOS, nous allons sélectionner les meilleurs plan de composition en utilisant une fonction objective.

**Contribution :**

La principale contribution de ce projet est de proposer une distance entre les besoins de l'utilisateur et les services Cloud offerts. Spécifiquement cette approche exploite les corrélations des séries temporelles pour calculer efficacement la distance entre deux groupes de séries temporelles (TSG).

Nous proposons des structures de données pour le traitement des séries temporelles qui exploitent les relations des séries temporelles.

Des expériences analytiques sont présentées pour montrer que l'approche proposée a des performances acceptables.

**Plan du mémoire :**

Le reste de ce mémoire est organisé comme suit :

**Le premier chapitre :**

Ce chapitre est consacré à la présentation de quelques notions de Cloud computing, et nous détaillons un peu de plus la composition des services Cloud.

**Le deuxième chapitre :**

Dans ce chapitre, nous avons décrit notre contribution à résoudre le problème de la recherche de composition des services Cloud, après nous montrons le prototype ainsi que les résultats expérimentaux, et enfin nous terminons par une conclusion.

# Chapitre I :

## Chapitre I :

# Composition des Services Cloud

## I. Introduction

Indéniablement, la technologie de l'internet se développe d'une manière exponentielle depuis sa création. Actuellement, une nouvelle "tendance" a fait son apparition dans le monde de technologies de l'information et de la communication, il s'agit du Cloud Computing, il s'appuie sur le WEB participatif 2.0, offre des opportunités aux sociétés de réduire les coûts d'exploitation des logiciels par leurs utilisations directement en ligne. [15]

Dans ce chapitre nous allons présenter quelques définitions et quelques concepts fondamentaux du Cloud Computing, ses enjeux, ses évolutions et son utilité ainsi que la technologie qui la constitue et les différents acteurs du secteur.

## II. Cloud Computing

### II.1 Historique

Techniquement, le concept de Cloud Computing est loin d'être nouveau, il est même présent depuis des décennies. On en trouve les premières traces dans les années 1960, quand John McCarthy<sup>1</sup> affirmait que cette puissance de traitement informatique serait accessible au public dans le futur. Le terme en lui-même est apparu plus couramment aux alentours de la fin du XXe siècle et il semblerait qu'Amazon.com soit l'un des premiers à avoir assemblé des data-center et fournit des accès à des clients. Les entreprises comme IBM et Google ainsi que plusieurs universités ont seulement commencé à s'y intéresser sérieusement aux alentours de 2008, quand le Cloud Computing est devenu un concept à la mode. Réalisant ce qu'ils pourraient faire de toute cette puissance, de nombreuses compagnies ont ensuite commencé à montrer un certain intérêt, puis à échanger leurs anciennes infrastructures et applications internes contre ce que l'on appelle : les pay per-use service (services payés à l'utilisation). [14]

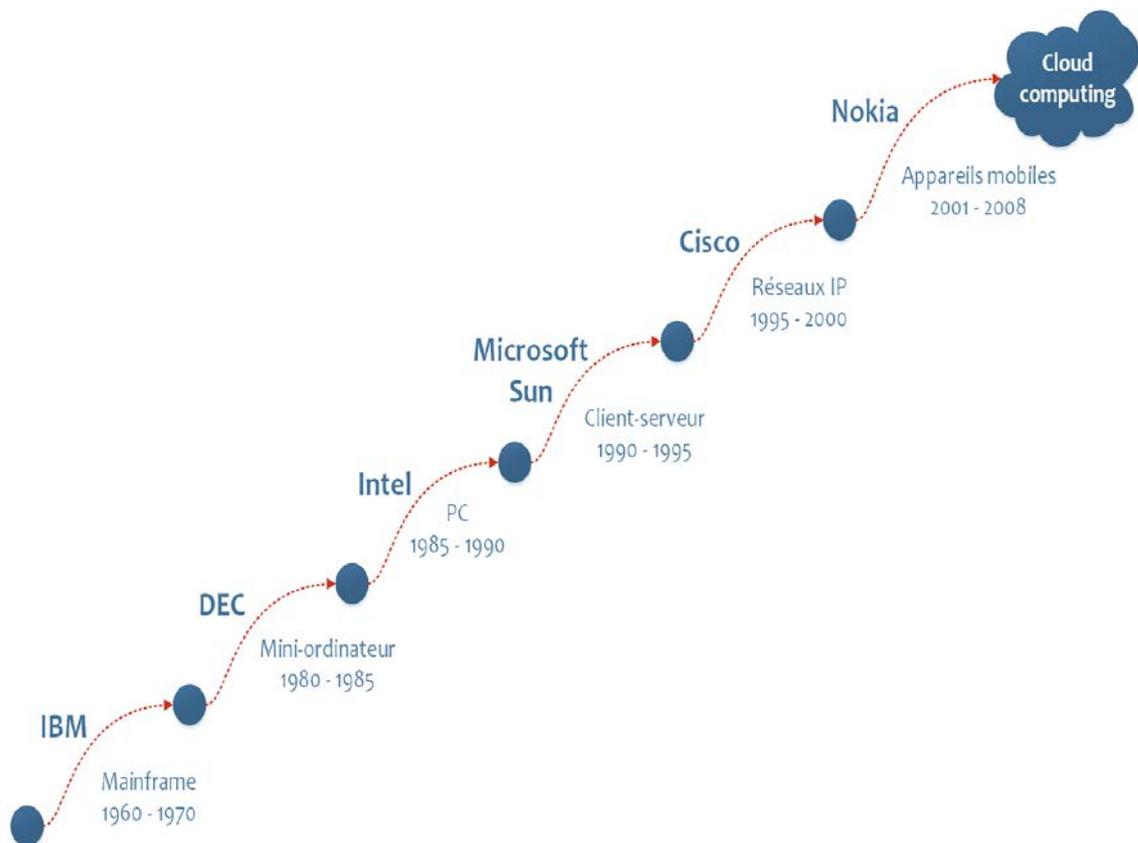
C'est la cinquième génération de l'informatique après les Mainframes, les PCs, les Clients/Serveurs et le Web. [13]

---

<sup>1</sup> *John McCarthy* (né le 4 septembre 1927, à *Boston, Massachusetts*) est le principal pionnier de l'intelligence artificielle. Il est également l'inventeur en 1958 du langage Lisp. A la fin des années 1950, il a créé avec *Fernando Cobarto* la technique du temps partagé, qui permet à plusieurs utilisateurs d'employer simultanément un même ordinateur

Auparavant, seuls les superordinateurs permettaient de fournir cette puissance et étaient principalement utilisés par des gouvernements, des militaires, des laboratoires et des universités pour réaliser des calculs aussi complexes que prédire le comportement d'un avion en vol, les changements climatiques ou la simulation d'explosions nucléaires. Désormais, des entreprises comme Google fournissent des applications qui exploitent le même type de puissance et sont accessibles à tout moment, de n'importe où et par tout un chacun via Internet.

Quelques universités prestigieuses ont également lancé leurs propres programmes de Cloud Computing en fournissant des accès à des maillages de centaines ou milliers de processeurs. Des entreprises comme IBM, ont récemment annoncé leur intention d'utiliser massivement le Cloud Computing à l'avenir. Ces derniers ont récemment dévoilé un système ultra-performant connu sous le nom de « Blue Cloud » qui permettra d'aider les banques et diverses entreprises à distribuer leurs calculs sur un très grand nombre de machines sans posséder d'infrastructure en interne.



**Figure I.1: Evolution de l'informatique [13]**

## II.2 Définitions

Le Cloud Computing, littéralement l'informatique dans les nuages est un concept qui consiste à déporter sur des serveurs distants des stockages et des traitements informatiques traditionnellement localisés sur des serveurs locaux ou sur le poste de l'utilisateur. Il consiste à proposer des services informatiques sous forme de service à la demande, accessible de n'importe où, n'importe quand et par n'importe qui, grâce à un système d'identification, via un PC et une connexion à Internet. Cette définition est loin d'être simple à comprendre, toutefois l'idée principale à retenir est que le Cloud n'est pas un ensemble de technologies, mais un modèle de fourniture, de gestion et de consommation de services et de ressources informatiques. [15]

Selon Wikipedia « Il s'agit d'un concept de déportation sur des serveurs distants des traitements informatiques traditionnellement localisés sur le poste client».

Selon Cisco «Le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité de ressources».

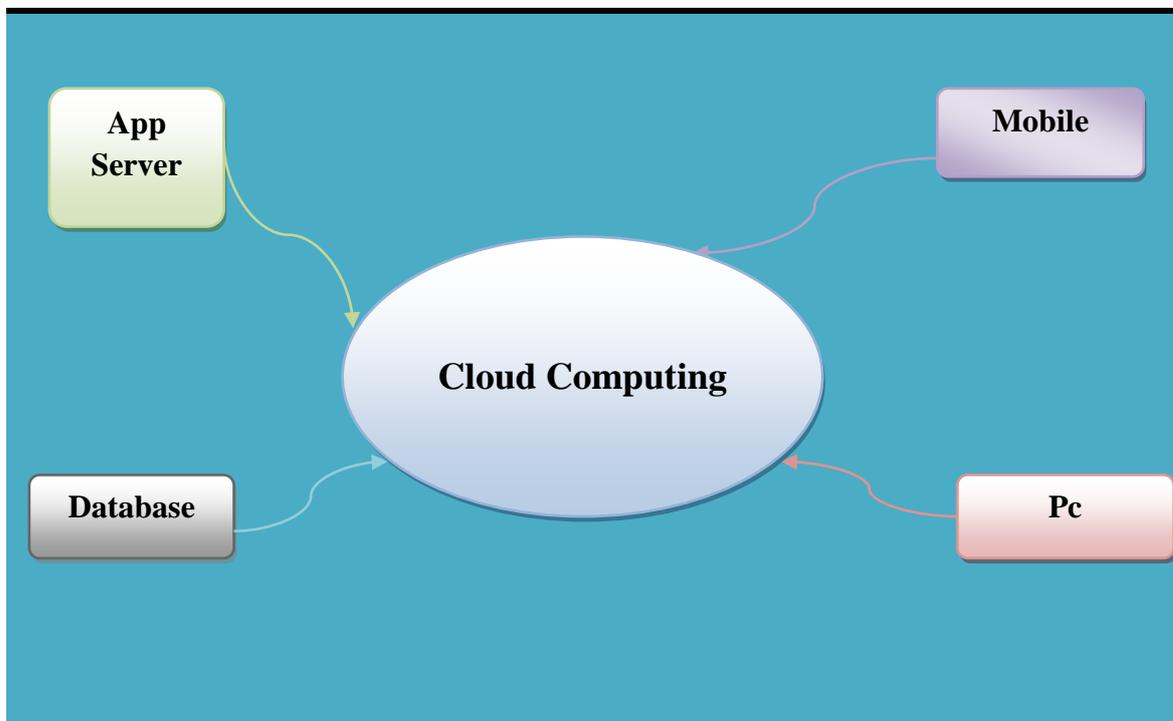


Figure I.2 : Centre de données informatique en nuage [7]

### II.3 Principe du Cloud Computing

Le nuage est un ensemble de matériels, de raccordements réseau et de logiciels qui fournissent des services sophistiqués que les utilisateurs peuvent exploiter à volonté depuis n'importe où dans le monde. Le Cloud computing est un basculement de tendance : au lieu d'obtenir de la puissance de calcul par acquisition de matériel et de logiciel, le consommateur se sert de puissance mise à sa disposition par un fournisseur via internet.

### II.4 Usages de Cloud computing

- Cloud de services : solutions applicatives.
- Cloud d'infrastructures : virtualisation de serveurs ou de réseaux.

### II.5 Exemples de fournisseurs de Cloud

- Amazon
- Google
- Microsoft
- IBM
- Zoho
- Rackspace
- Azure
- Peak 10
- Citrix
- Intel
- Sales force
- Hp

#### II.5.1 Exemples d'applications

- ✓ Gmail (messageries)
- ✓ Hotmail (messageries)
- ✓ Zoho (bureautique)
- ✓ Salesforce (bureautique)
- ✓ Google Apps (bureautique)

- ✓ Dropbox (stockage de données)
- ✓ Skydrive (stockage de données)
- ✓ Flibby (stockage de données)
- ✓ Slideshare (diaporamas)

## II.6 Modèles des services de Cloud computing

### II.6.1 SaaS (Software as a Service)

Concept consistant à proposer un abonnement à un logiciel plutôt que l'achat d'une licence. On oublie donc le modèle client-serveur et aucune application n'est installée sur l'ordinateur, elles sont directement utilisables via le navigateur Web. L'utilisation reste transparente pour les utilisateurs, qui ne se soucient ni de la plateforme, ni du matériel, qui sont mutualisés avec d'autres entreprises. Le SaaS remplace l'ASP, aussi appelé fournisseur d'applications hébergées ou FAH, ou application service provider en anglais ou ASP, qui est une entreprise qui fournit des logiciels ou des services informatiques à ses clients au travers d'un réseau.

Il s'agit de la mise à disposition d'applications sous la forme de service (CRM<sup>2</sup>, outils collaboratifs, messagerie, ERP<sup>3</sup>, ...). Ce concept consiste à proposer un abonnement à un logiciel plutôt que l'achat d'une licence. Plus d'installation, plus de mise à jour à gérer (elles sont continuées chez le fournisseur), plus de migration de données etc. Paiement à l'usage.

---

<sup>2</sup> La gestion de la relation client (GRC), ou gestion des relations avec les clients, en anglais Customer relation ship management (CRM), est l'ensemble des outils et techniques destinés à capter, traiter, analyser les informations relatives aux clients et aux prospects, dans le but de les fidéliser en leur offrant le meilleur service

<sup>3</sup> Enterprise resource planning, signifiant littéralement en anglais, « planification des ressources de l'entreprise », et traduit en français par « progiciel de gestion intégré » (PGI).

### II.6.2 PaaS (Platform as a Service)

Il s'agit des plateformes du nuage, regroupant principalement les serveurs mutualisés et leurs systèmes d'exploitation. En plus de pouvoir délivrer des logiciels en mode SaaS, le PaaS dispose d'environnements spécialisés au développement comprenant les langages, les outils et les modules nécessaires.

### II.6.3 IaaS (Infrastructure as a Service)

Il s'agit de la mise à disposition, à la demande, de ressources d'infrastructures (serveurs, moyens déstockage, réseau ...) dont la plus grande partie est localisée à distance dans des Datacenter. L'IaaS permet l'accès aux serveurs et à leurs configurations pour les administrateurs de l'entreprise. Le client a la possibilité de louer des clusters, de la mémoire ou du stockage de données. Le coût est directement lié au taux d'occupation. Une analogie peut être faite avec le mode d'utilisation des industries des commodités (électricité, eau, gaz) ou des Télécommunications. [12]

SaaS	PaaS	IaaS
<ul style="list-style-type: none"> <li>- Google</li> <li>- offre Google Apps (messagerie et bureautique).</li> <li>-Sales Force</li> <li>- CRM (Customer Relationship Management).</li> </ul>	<ul style="list-style-type: none"> <li>-Microsoft</li> <li>-offre Azur.</li> <li>- Google</li> <li>-offre Google App Engine.</li> </ul>	<ul style="list-style-type: none"> <li>-Amazon</li> <li>-offres EC2 et AWS</li> <li>-Microsoft</li> <li>-offre Azur</li> </ul>

Tableau I.1 : Les grands acteurs mondiaux de Cloud

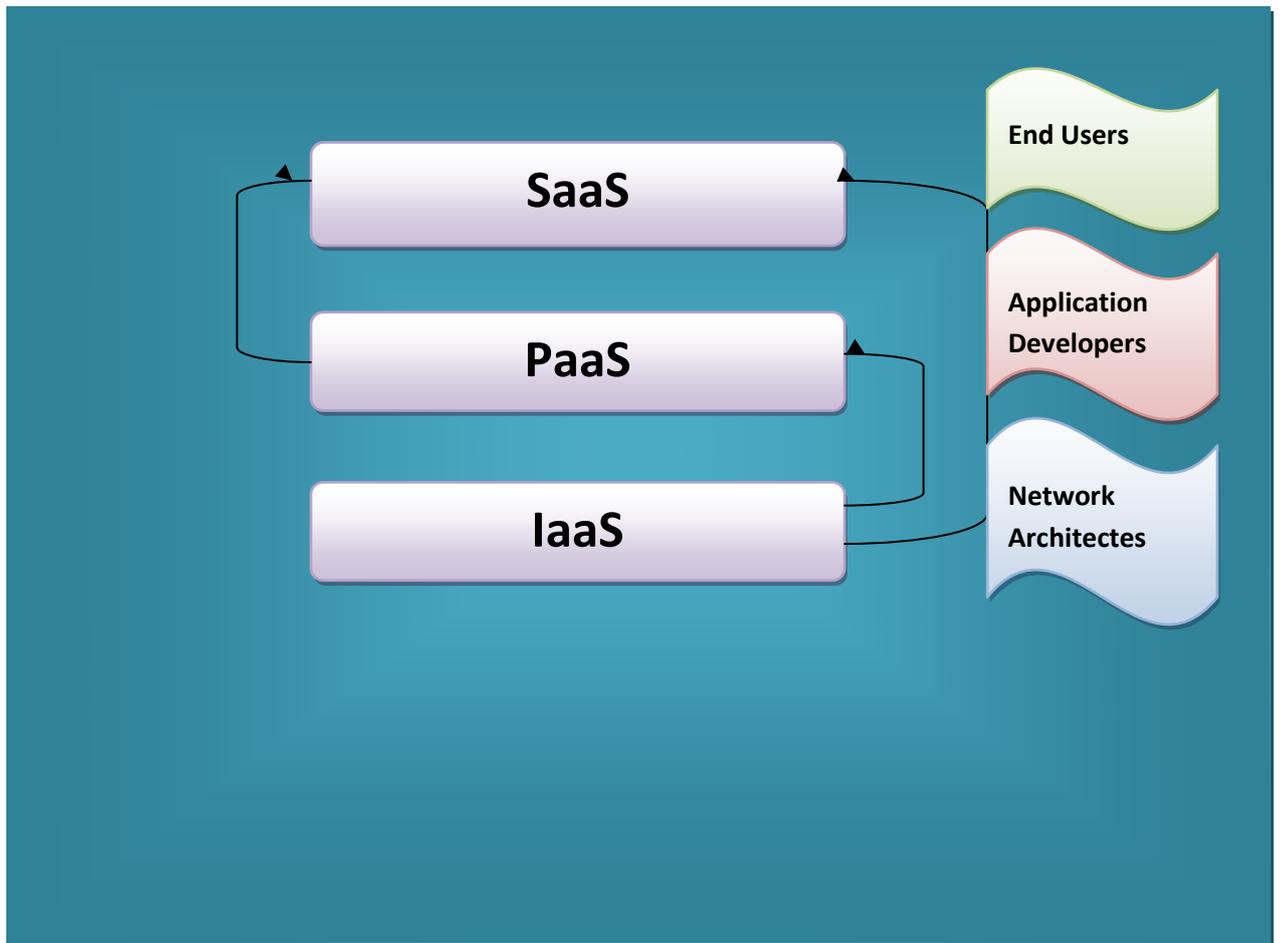


Figure I.3 : Les couches du Cloud computing

## II.7 Caractéristiques du Cloud

Les caractéristiques essentielles d'un nuage sont :

- ❖ Un service à la demande
- ❖ Un accès aux ressources par le réseau
- ❖ Mise en commun des ressources
- ❖ Flexibilité des ressources
- ❖ Un service mesuré

## II.8 Avantages et inconvénients des services Cloud

	SaaS	PaaS	IaaS
<b>Avantages</b>	<ul style="list-style-type: none"> <li>*Pas d'installation</li> <li>*Migration</li> <li>*Accessible via un abonnement</li> <li>Plus de licence</li> </ul>	<ul style="list-style-type: none"> <li>*Pas d'infrastructure nécessaire</li> <li>*Pas d'installation</li> <li>Environnement hétérogène</li> </ul>	<ul style="list-style-type: none"> <li>*Administration</li> <li>*Flexibilité d'utilisation</li> <li>*Capacité de stockage infini</li> </ul>
<b>Inconvénients</b>	<ul style="list-style-type: none"> <li>*Logiciel limité</li> <li>*Sécurité</li> <li>*Dépendance des prestataires</li> </ul>	<ul style="list-style-type: none"> <li>*Limitation des langages</li> <li>*Pas de personnalisation dans la configuration des machines virtuelles</li> </ul>	<ul style="list-style-type: none"> <li>*Sécurité</li> <li>*Besoin d'un administrateur système</li> <li>*Demande pour les acteurs du Cloud des investissements très élevés</li> </ul>

Tableau I.2 : Avantages et inconvénients des services [15]

## II.9 Avantages et inconvénients du Cloud Computing

Avantages	Inconvénients
<ul style="list-style-type: none"> <li>-Disponibilité et extensibilité</li> <li>-Dynamicité</li> <li>-Tolérance aux pannes</li> <li>-Mutualisation des ressources</li> </ul>	<ul style="list-style-type: none"> <li>-Hétérogénéité</li> <li>-Absence de localité</li> <li>-Portage des applications</li> <li>-Sécurité</li> </ul>

Tableau I.3 : Avantages et inconvénients du Cloud Computing

### **III L'Architecture Orienté Service (SOA)**

#### **III.1 Définition de SOA**

Le SOA est un paradigme fondé sur la description et l'interaction de services, autrement dit L'AOS est une approche architecturale permettant la création des systèmes basés sur une collection de services développés dans différents langages de programmation, hébergés sur différentes plates-formes avec divers modèles de sécurité et processus métier. [6]

L'idée maîtresse de l'architecture orientée service est que tout élément du système d'information doit devenir un service identifiable, documenté, fiable, indépendant des autres services, accessible, et réalisant un ensemble de tâches parfaitement définies. [3]

#### **III.2 Concepts de SOA**

Une architecture orientée service se conforme à divers principes de gestion des services influençant directement le comportement intrinsèque d'une solution logicielle et le style de sa conception.

- L'encapsulation des services.
- Le faible couplage des services avec la maintenance d'une relation réduisant les dépendances.
- Le contrat de service adhérent à un accord de communication, collectivement défini avec un ou plusieurs documents de description.
- L'abstraction des services dissimulant la logique du service à l'extérieur.
- La réutilisation des services partageant la logique entre plusieurs services avec l'intention de promouvoir la réutilisation.
- La composition des services.
- L'autonomie des services.
- L'optimisation des services.
- La découverte des services depuis leur description extérieure.

## IV Service web

### IV.1 Définition

Les services Web 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 services Web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service Web est déployé, d'autres applications (y compris des services Web) peuvent le découvrir et l'invoquer. [11]

Une architecture orientée services consiste essentiellement en une collection de services qui interagissent et communiquent entre eux. Cette communication peut consister en un simple retour de données ou en une activité (coordination de plusieurs services).

Selon W3C (World Wide Web Consortium)<sup>4</sup>, un Web service(ou service Web) est une application callable via Internet par une autre application d'un autre site Internet permettant l'échange de données (de manière textuelle) afin que l'application appelante puisse intégrer le résultat de l'échange à ses propres analyses. Les requêtes et les réponses sont soumises à des standards et normalisées à chacun de leurs échanges.

### IV.2 Caractéristiques

La technologie des services Web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web.

Les caractéristiques sont les suivantes :

-  -Large granularité (coarse-grained).
-  -Interface
-  -Localisable
-  -Instance unique
-  -Les services de présentations ou de référencement

---

<sup>4</sup> [www.w3c.org](http://www.w3c.org)

- ✚ -Les processus métiers
- ✚ -Les services de gestion et d'accès
- ✚ -Les services d'intégration

## V Problème de la recherche de composition des services Cloud

Le Cloud computing devient de plus en plus la technologie de choix en tant que plateforme de prochaine génération pour la conduite d'affaires. [10]

Un avantage significatif du Cloud computing est ses avantages économiques pour les utilisateurs et les fournisseurs de services.

La composition des services à forte intensité de données est devenue les défis les plus importants des applications SOA et le sujet de la recherche dans les quelques dernières années. [6]

La composition des services traditionnels est examinée du point de vue des utilisateurs [4] [17]. Comme lors du processus de composition, il existe de nombreux services candidats avec la même fonctionnalité, mais des attributs de qualité de service (QOS) différents, l'objectif est de choisir les bons services à composer afin d'obtenir une solution optimale pour répondre aux exigences QOS demandées par Un utilisateur de service (comme la disponibilité, le débit, le temps de réponse, la sécurité, etc.).

Par rapport à la composition traditionnelle des services, la composition des services cloud est habituellement basée sur la QOS à long terme et axée sur l'économie. [2]

Les techniques de composition traditionnelles basées sur la qualité considèrent habituellement les qualités au moment de la composition. [18]

Par exemple, quel service composite a les meilleures performances à l'heure actuelle? Il est fondamentalement différent dans les environnements de Cloud dans lesquels la fonction de service Cloud devrait durer une longue période.

Par exemple, quel service Cloud composite fonctionne-t-il le mieux au cours des prochaines années, même s'il n'est peut-être pas le meilleur? Ce travail présente une nouvelle approche de composition des services Cloud basée sur des séries temporelles. Les bases de données de séries temporelles sont répandues dans de multiples domaines de recherche, par exemple, multimédia, statistiques, etc. De nombreuses techniques [8],

[1] ont été proposées de manière efficace et analysent efficacement les modèles économiques dans le Cloud computing.

Nous identifions 3 acteurs dans l'environnement Cloud (SaaS, PaaS, IaaS) plus un autre acteur : Les utilisateurs finaux qui sont généralement de grandes entreprises et des organisations, par exemple, les universités, les gouvernements...

À l'instar de la composition des services traditionnels [9], la composition des services Cloud se déroule en deux étapes. Tout d'abord, un schéma de composition est construit pour une demande de composition. Deuxièmement, le plan de composition optimal est sélectionné. Un plan de composition est formé en choisissant des fournisseurs concrets de services cloud pour chaque SaaS et IaaS abstraites dans le schéma de composition.

Notre recherche se concentre sur la sélection des plans de composition basés uniquement sur des attributs non fonctionnels (Quality of-Service ou QoS) [16]. D'une part, Nous modélisons les exigences des utilisateurs finaux en tant que ensemble de séries temporelles. D'autre part, les fournisseurs des services cloud commercialisent leurs services (SaaS) à l'aide d'un ensemble de séries temporelles.

Chaque série temporelle représente les valeurs d'un attribut QoS correspondant sur une longue période. Par conséquent, le problème de composition des services Cloud devient un problème de recherche de similarité dont la requête est un ensemble de séries temporelles désirées.

Les techniques traditionnelles traitent rarement des requêtes de séries temporelles complexes qui nécessitent une corrélation entre les séries temporelles à utiliser lors de la correspondance de similarité. Cependant, la corrélation est fréquente lors de la composition du service où chaque attribut QoS est corrélé avec plusieurs autres attributs QoS.

Par exemple, en considérant que nous allons choisir le meilleur plan de composition en utilisant deux attributs de QoS, le temps de réponse et le coût. Le coût d'un service Cloud peut diminuer pendant une période, tandis que le temps de réponse du service Cloud diminue également. On peut facilement observer les corrélations existant entre les séries temporelles. Si nous traitons chaque série temporelle de manière indépendante, nous ne pourrions pas utiliser les corrélations inhérentes.

Nous nous référons à des groupes de séries chronologiques avec des corrélations en tant que groupes de séries temporelles (TSG). Étant donné un objet TSG de requête  $Q = \{q_1, q_2, \dots, q_l\}$ , où  $l$  est le nombre de séries temporelles dans  $Q$  et chaque série temporelle  $q_i = \{(x_m, t_m) \mid m = 1, 2, \dots, L_i\}$ , Où  $x_m$  est la valeur de la série temporelle à l'instant  $t_m$ , et un ensemble TSG  $D = \{TSG_1, TSG_2, \dots, TSG_N\}$ . La recherche de similarité sur TSG consiste à trouver les TSG les plus semblables à partir de  $D$  via, une fonction pré définie  $\text{dist}(Q, TSG_i)$ , où  $\text{result} = \text{argmin}_{i = 1, \dots, N} (\text{dist}(Q, TSG_i))$ .

### V.1 Exemple

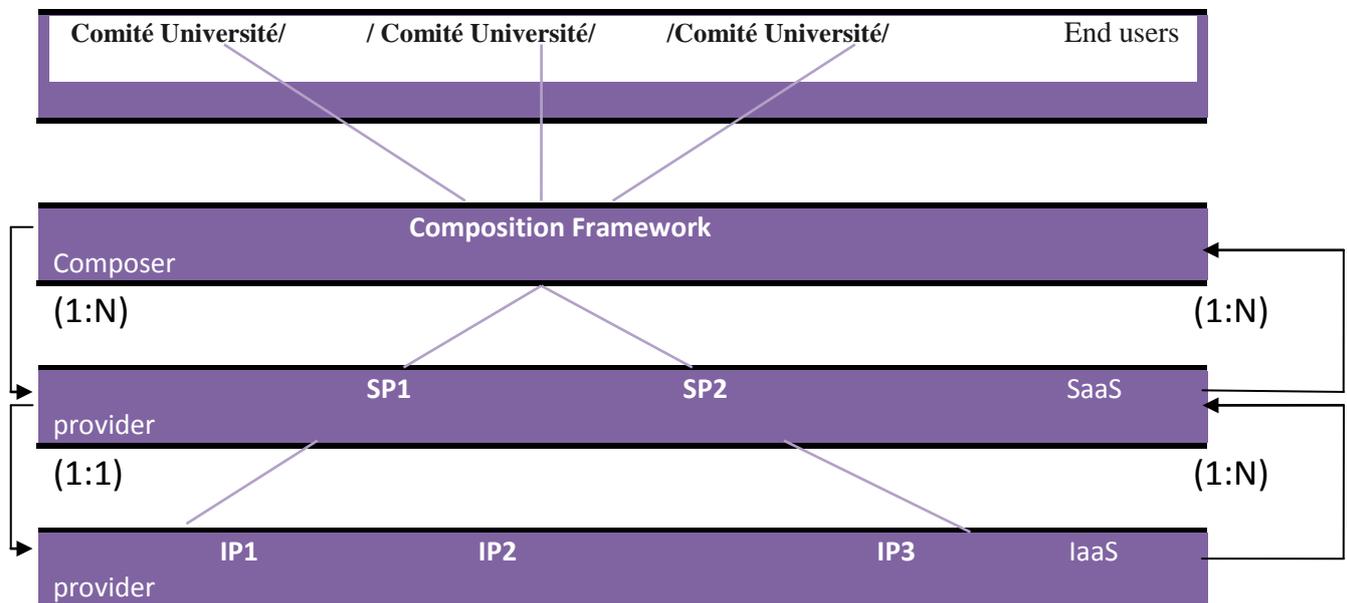


Figure I.4 : Composition des Services Cloud [9]

### V.2 Conclusion

Par rapport au cadre de composition des services traditionnels dans le calcul orienté service (SOC), on considère la composition des services dans une perspective à long terme. Les modèles économiques Cloud pour les utilisateurs finaux et les fournisseurs des services cloud sont exploités pendant la composition. En particulier, nous proposons d'utiliser des séries temporelles pour représenter les modèles économiques. La composition des services cloud est ensuite modélisée comme un problème de recherche de similarité dans la base de données à plusieurs séries temporelles.

Chapitre II :  
**Chapitre II :**

Conception & Réalisation

## I. Introduction

Dans le cadre de cette partie, notre recherche se concentre sur la sélection des plans de composition basés uniquement sur des attributs non fonctionnels (Quality of-Service ou QoS) Plus particulièrement nous modélisons les exigences des utilisateurs finaux en tant que ensemble de séries temporelles d'une part, les fournisseurs des services Cloud commercialisent leurs services (SaaS) à l'aide d'un ensemble de séries temporelles, D'autre part. Le plan de ce chapitre est décrit comme suit : Tout d'abord nous décrivons la collection de test, ensuite nous présentons la conception (les algorithmes QA, QR, distance QR, distance finale entre QA et QR, et génération), après nous montrons le prototype ainsi que les résultats expérimentaux, et enfin nous terminons par une conclusion.

## II. Présentation de la collection de test

Dans cette section nous présentons la collection de test utilisée dans notre approche. Il existe deux exigences lors de la conception de notre approche de composition. Tout d'abord, une représentation en séries temporelles est nécessaire pour décrire et mesurer l'information générale extraite d'un TSG. Deuxièmement, l'algorithme de recherche de similarité doit être plus évolutif par rapport à d'autres approches, puisque l'environnement cloud est plus évolutif que les autres plates-formes existantes.

Pour générer les ensembles de données synthétiques, on génère d'abord les séries temporelles, chaque série temporelle est désignée par  $\{(Qos_i, t_i) / i = 1,2,3..l\}$  où  $l$  la longueur est égale à 150. Les séries temporelles dans l'ensemble de données sont générées au hasard, Ces séries temporelles forment le TSG. Nous créons d'autres  $L$  TSGs ou  $L$ =nombre de TSGs créer. Deux structures de donnée, désignées **QA** et **QR**, sont introduites pour représenter des séries temporelles lors de la composition des services Cloud. Nous générons 3 classes, chaque classe contiens un certain nombre de services(TSGs) Le problème de la composition du service Cloud devient un Problème de recherche de similarité dont la requête est un ensemble de séries temporelles souhaitées, nous utilisons 2 mesures de similarités basées sur la distance euclidienne entre les valeurs de QOS et la corrélation entre les critères de QOS. Pendant la composition du service la corrélation est prévalente, où chaque attribut QOS est corrélé avec plusieurs autres attributs QOS, nous allons sélectionner les meilleurs plans de compositions en utilisant une fonction objective.

### III. Conception

Dans cette partie, nous avons présenté les algorithmes utilisés dans notre approche:

#### III.1 Approche QA (QoS Attribute of TSG)

//Calculer QA

Entrée :

$Tsg_1, Tsg_2$

//Structure

$Tsg[ q_1, q_2, q_3 ]$

$q[ 150.\text{TimeSérie} ] .\text{Taille} \leftarrow 150$

TimeSérie [  $x_i, T_i$ ]

TimeSérie  $\leftarrow TS$

//Algorithme

**Début**

1.  $Q_{\text{global}} Tsg_1 \leftarrow Tsg_1.q_1 + Tsg_1.q_2 + Tsg_1.q_3 ;$
2.  $Q_{\text{global}} Tsg_2 \leftarrow Tsg_2.q_1 + Tsg_2.q_2 + Tsg_2.q_3 ;$
3.  $\text{TailleGlobal} \leftarrow Tsg_1.q_1.\text{Taille} + Tsg_1.q_2.\text{Taille} + Tsg_1.q_3.\text{Taille} ;$
4.  $\text{Somme} \leftarrow 0$
5. **Pour**  $i \leftarrow 0$  à  $\text{TailleGlobal}$  **Faire**
6.      $\text{Somme} \leftarrow \text{Somme} + \text{Puissance} ( Q_{\text{global}} Tsg_1.TS[i].x - Q_{\text{global}} Tsg_2.TS[i].x )$
7.  $\text{Distance} \leftarrow \text{Racine} (\text{Somme}) ;$

**Fin.**

**III.2 Approche QR (QoS Relation for TSG)**

//Calculer QoS R

**Entrée :** *Tsg***Sortie :** QoSRelation

//structure :

QoSRelation  $\leftarrow$  Matrice de QrelationQrelation  $\leftarrow$  [ *TimeVariance*, *QoSVariance* ]

//Algorithme

**Début**

1. Taille  $\leftarrow$  3 ; //nbr de requete ( *d.r.c* )
2. **Pour** *i*  $\leftarrow$  0 à Taille **Faire**
3.     **Pour** *j*  $\leftarrow$  0 à Taille **Faire**

**Début**

4. Requete *i*  $\leftarrow$  *Tsg*.requete[*i*] ; *Tr<sub>i</sub>*  $\leftarrow$  requete *i*.Taille ;
5. Requete *j*  $\leftarrow$  *Tsg*.requete[*j*] ; *Tr<sub>j</sub>*  $\leftarrow$  requete *j*.Taille ;
6. Qrelation.TimeVariance.**Tn<sub>i</sub>**  $\leftarrow$  requete *i*[0].**T** ; .....(1)
7. Qrelation.TimeVariance.**Tn<sub>j</sub>**  $\leftarrow$  requete *j*[0].**T** ; .....(2)
8. Qrelation.TimeVariance.**Tn<sub>i</sub>**  $\leftarrow$  requete *i*[*Tr<sub>i</sub>*].**T** ; .....(3)
9. Qrelation.TimeVariance.**Tn<sub>j</sub>**  $\leftarrow$  requete *j*[*Tr<sub>j</sub>*].**T** ; .....(4)
10. Qrelation.TimeVariance.**V[1]**  $\leftarrow$  (1)-(2) ;
11. Qrelation.TimeVariance.**V[2]**  $\leftarrow$  (3)-(4) ;
12.     **Si** (*i*==*j*) Qrelation.TimeVariance.**V[3]**  $\leftarrow$  0
13.     **Sinon si** ( (1) >= (2) ) Qrelation.TimeVariance.**V[3]**  $\leftarrow$  (3)-(2)
14.     **Sinon** Qrelation.TimeVariance.**V[3]**  $\leftarrow$  -((2)-(3))

**fin Si**

15. **Si** (i==j) Qrelation.TimeVariance.V[4] ← 0
16. **Sinon si** ((1) >= (2)) Qrelation.TimeVariance.V[4] ← (3)-(2)
17. **Sinon** Qrelation.TimeVariance.V[4] ← -((3)-(1))

**fin Si**

18. **Pour** k=0 à Tr **Faire**

**Début**

19. QosVariance[k] ← |requete i[k].x – requete j[k].x|;

**Fin**

20. QosRrelation[i][j] ← Qrelation ;

**Fin**

**Fin.**

### III.2.1 Approche de distance QR

Entrée :  $QosR_1, QosR_2$

//Algorithme

**Début**

1. Float [] Somme ;
2. Int *index* ← 0 ;
3. **Pour** i ← 0 à  $QosR_1$ .Qrelation.Taille **Faire**
4. **Pour** j ← 0 à  $QosR_1$ .Qrelation [0].Taille **Faire**

**Début**

5. Float[]  $QosV_1$  ←  $QosR_1$ .Qrelation [i][j] ;

```

6.      Float[] QosV2 ← QosR2.Qrelation [i][j] ;

7.      Pour k ← 0 à QosV1.Taille Faire

8.      Somme ← Somme + (QosV1[k] - QosV2[k]) * (QosV1[k] - QosV2[k]);

      Fin pour

9.      Somme[ index ] ← Somme ;

10.     Index++

      Fin

Fin

11.    Float Total ← 0

12.    Pour i ← 0 à index Faire

13.     Total ← Total + Somme[i] ;

      Fin pour

14.    Float QR ← Racine ( Total )
    
```

La relation entre  $\mathbf{T}_m$  et  $\mathbf{T}_n$  se réfère aux différences entre les valeurs de QoS dans les séries temporelles et les intervalles de temps assignés. Par conséquent, nous introduisons deux relations de base définies par [5], i.e., la variance dans les valeurs *QosVariance* et la variance dans le temps *TimeVariance*

$$QosVariance = (|x_{ms1} - x_{ns1}|, |x_{ms2} - x_{ns2}|, \dots, |x_{msl} - x_{nsl}|) .$$

Où  $x_{msi}$  est la valeur QOS échantillonnée de  $\mathbf{T}_m$  et  $x_{nsj}$  est la valeur QOS échantillonnée de  $\mathbf{T}_n$ . *VIT* est un descripteur de la relation entre les intervalles de  $\mathbf{T}_m$  et  $\mathbf{T}_n$  dénotés comme vecteur 4D *TimeVariance* ( $\mathbf{T}_m, \mathbf{T}_n$ ) [5] :

$$TimeVariance = (\mathbf{T}_m, \mathbf{T}_n) = (t_{m1} - t_{n1}, t_{ml} - t_{nl}, t_{m,n}^3, t_{m,n}^4),$$

Où  $t_{m1}$  et  $t_{n1}$  sont les premières valeurs de  $T_m$  et  $T_n$  et  $t_{m1}$  sont les dernières valeurs

$t^3_{m,n}$  et  $t^4_{m,n}$  sont utilisés pour décrire l'état de chevauchement entre deux intervalles de séries temporelles, ils sont définis comme suit:

$$t^3_{m,n} = \begin{cases} 0, m = n, \\ t_{m1} - t_{n1}, t_{m1} \geq t_{n1}, \\ -(t_{n1} - t_{m1}), t_{m1} < t_{n1}. \end{cases}$$

$$t^4_{m,n} = \begin{cases} 0, m = n, \\ t_{m1} - t_{n1}, t_{m1} \geq t_{n1}, \\ -(t_{n1} - t_{m1}), t_{m1} < t_{n1}. \end{cases}$$

Donné un objet TSG qui a  $n$  séries temporelles  $T_1, T_2 \dots T_n$  on peut générer  $n^2$  descripteurs de relation  $R_{ij}$ (QosRelation) de chaque paire de séries temporelles  $T_i$  et  $T_j$  de l'objet TSG où :

$$R_{ij} = \{ QosVariance (T_i, T_j), TimeVariance (T_i, T_j) \}$$

**III.3 Approche de distance finale entre QA et QR**

//Algorithme

*Entrée QA, QR***début**

Float DistanceFinal

1. DistanceFinal  $\leftarrow$  Racine ( $QA * QR$ )**Fin****III.4 Approche de génération**Structure : *Qsi* [float *débit ,rép,cout*]

// Algorithme

**Début**1. *Qsi* [ ] generation<sub>1</sub> ;2. **Pour** *i*  $\leftarrow$  0 à 150 **Faire****Début**3. generation<sub>1</sub> [*i*].*débit*  $\leftarrow$  Random () ;4. generation<sub>1</sub> [*i*].*rép*  $\leftarrow$  Random () ;5. generation<sub>1</sub> [*i*].*cout*  $\leftarrow$  Random () ;**fin pour**6. *Qsi* [ ] generation<sub>2</sub> ;7. **Pour** *i*  $\leftarrow$  0 à 150 **Faire****Début**8. generation<sub>2</sub> [*i*].*débit*  $\leftarrow$  Random () ;

```

9.          generation2 [i].rép ← Random () ;
10.         generation2 [i].cout ← Random () ;

          fin pour

11.  Qsi [ ] generation3 ;

12.  Pour i ← 0 à 150 Faire

      Début

13.          generation3 [i].débit ← Random () ;
14.          generation3 [i].rép ← Random () ;
15.          generation3 [i].cout ← Random () ;

          fin pour

//TimeSérieGroupe Tsg

16.  Pour i ← 0 à 150 Faire

      Début

17.          Tsg.debit[i].x ← Min(generation1 [i].débit, generation2 [i].débit,
generation3 [i].débit) ;

18.          Tsg.debit[i].T ← i ;

      Fin pour

19.  Pour i ← 0 à 150 Faire

      Début

20.          Tsg.rép[i].x ← generation1 [i].rép* generation2 [i].rép*
generation3 [i].rép;

21.          Tsg.rép [i].T ← i ;

      Fin pour

22.  Pour i ← 0 à 150 Faire

```

**Début**

23.  $Tsg.cout[i].x \leftarrow generation_1 [i].cout + generation_2 [i].cout + generation_3 [i].cout ;$

24.  $Tsg.cout[i].T \leftarrow i ;$

**Fin pour**

**III.5 Recherche de similarité des groupes de séries temporelles**

Étant donné deux objets TSG :

$TSG_i = \{QA_i, QR_i\}$  a  $TSG_j = \{QA_j, QR_j\}$

nous définissons la distance entre deux TSG  $dist(TSG_i, TSG_j)$  : la distance entre 2 QAs

$$DIS(QA_i, QA_j) = \sqrt{\sum (qc^p_i - qc^p_j)^2}$$

Où  $qc^p_i$  et  $qc^p_j$  sont les valeurs p series temporelles de  $TSG_i$  et  $TSG_j$

la distance entre 2 QRs

$$DIS(QR_i, QR_j) = \sqrt{\sum (R^m_{ij} - R^n_{ij})^2}$$

Selon les fonctions de distance définies sur les deux caractéristiques, la distance globale entre deux objets TSG

$$dist(TSG_i, TSG_j) = DIS(QA_i, QA_j) \times DIS(QR_i, QR_j)$$

## IV. Présentation de prototype

### IV.1 Outils et environnement de développement

Avant de commencer l'implémentation de notre application, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblé être un bon choix vu les avantages qu'ils offrent, Notant que Java est un langage de programmation orienté objet, libre, simple et portable, nous avons utilisé le langage de programmation « Java » (java 1.8.0\_60), avec l'IDE « NetBeans 8.1 »

### IV.2 Présentation

#### ❖ Fenêtre d'accueil



Figure II.5 : Interface d'accueil

❖ Fenêtres de simulation

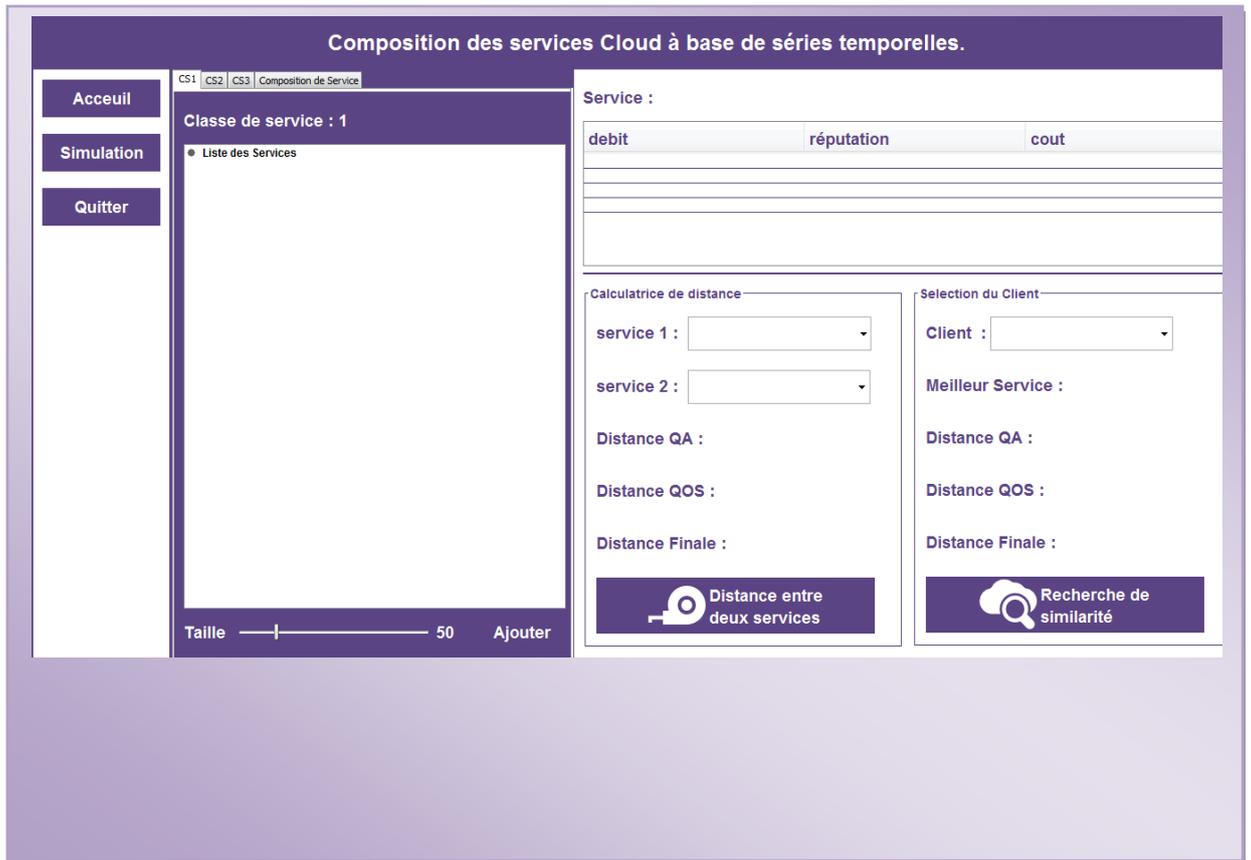


Figure II.6 Interface de simulation

L'interface suivante se compose de : Classes des services, bouton de composition des services tableau de (débit, cout, réputation) d'un service, bouton de calcul de distance entre deux services et bouton de recherche de similarité un bouton pour l'accueil et un bouton pour quitter.



Figure II.7 Liste des services de classe CS1

Nous pouvons indiquer la taille d'une liste (classe1 contient 15 services)

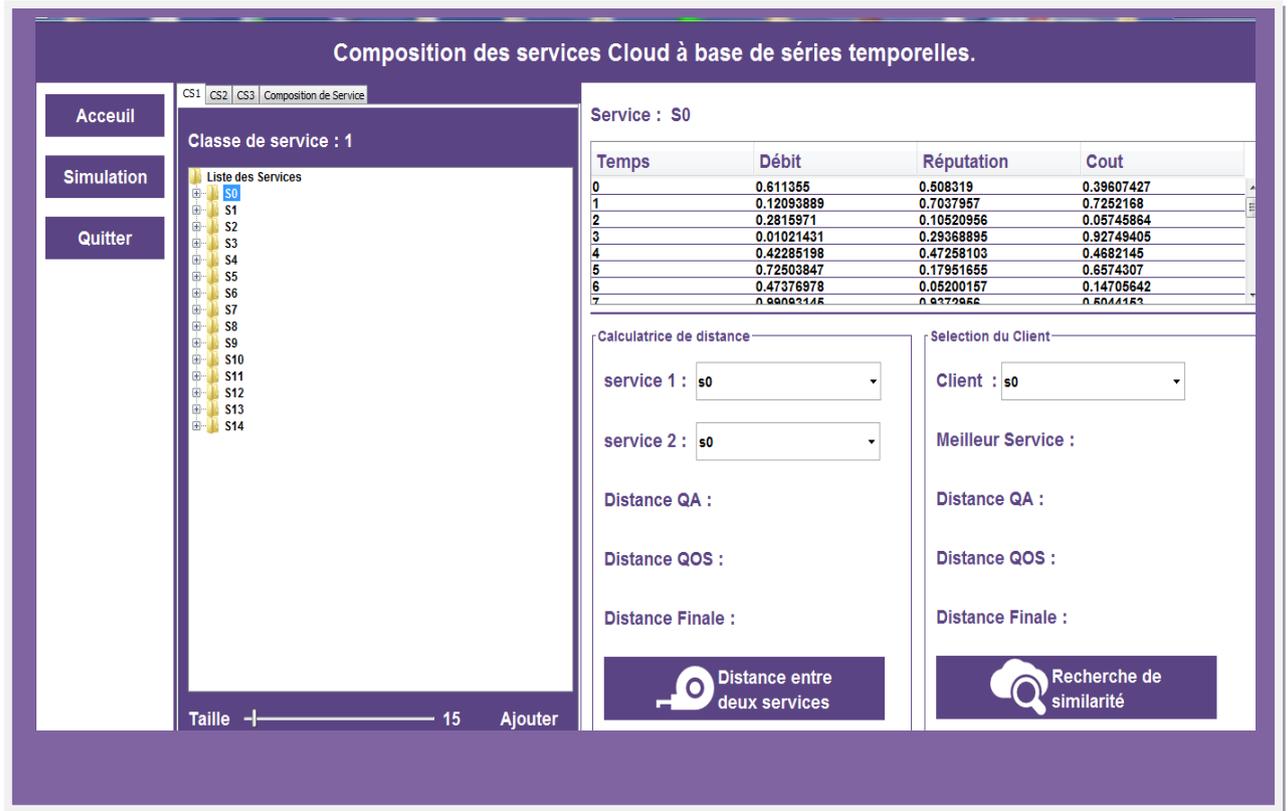


Figure II.8 Affichage de (débit, cout, réputation) d'un service

Chaque service possède un certain débit, cout, réputation pour chaque instant  $t$  avec  $t$  varie entre 0 et 149

Service : S0

Temps	Débit	Réputation	Cout
0	0.611355	0.508319	0.39607427
1	0.12093889	0.7037957	0.7252168
2	0.2815971	0.10520956	0.05745864
3	0.01021431	0.29368895	0.92749405
4	0.42285198	0.47258103	0.4682145
5	0.72503847	0.17951655	0.6574307
6	0.47376978	0.05200157	0.14705642
7	0.99093115	0.9379956	0.5044153

Service : S0

Temps	Débit	Réputation	Cout
142	0.22969808	0.9221991	0.9841242
143	0.6807232	0.23766743	0.50122905
144	0.7721579	0.6162132	0.2345614
145	0.3512805	0.21379028	0.09916523
146	0.1484607	0.2606668	0.16609897
147	0.2944412	0.7785799	0.14131445
148	0.43828017	0.200056	0.76314443
149	0.42038727	0.27002946	0.21962197

Figure II.9 Tableau de débit, cout, réputation de service S0

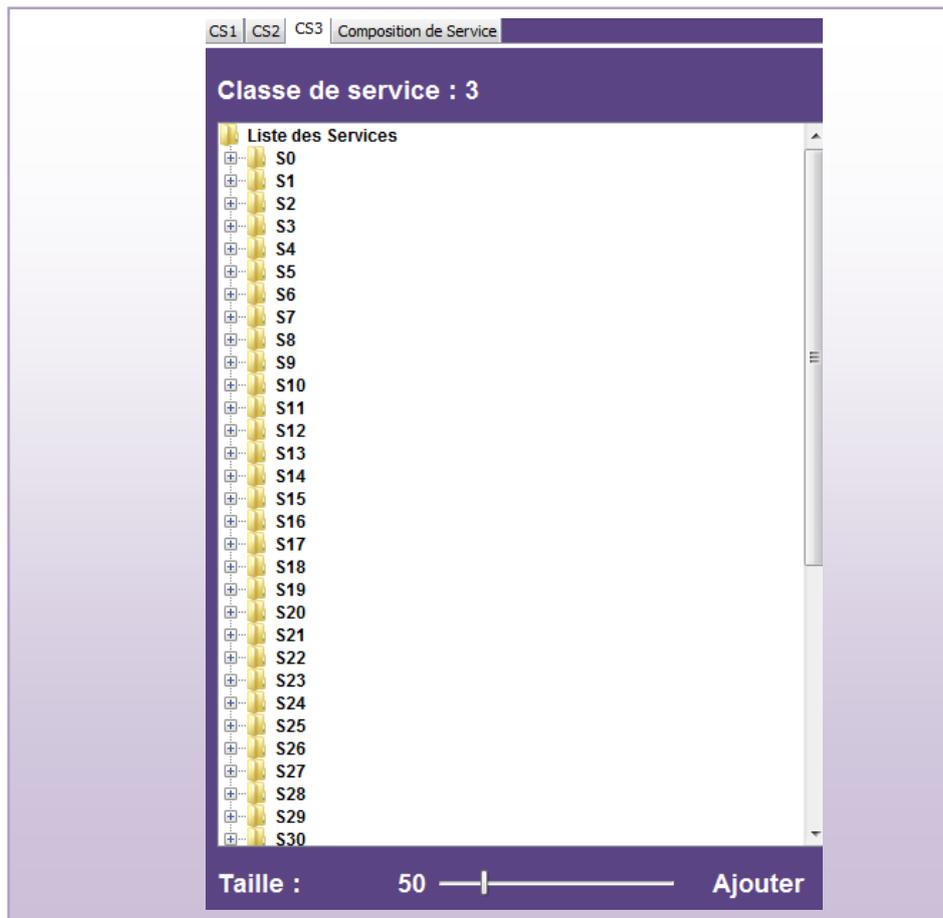
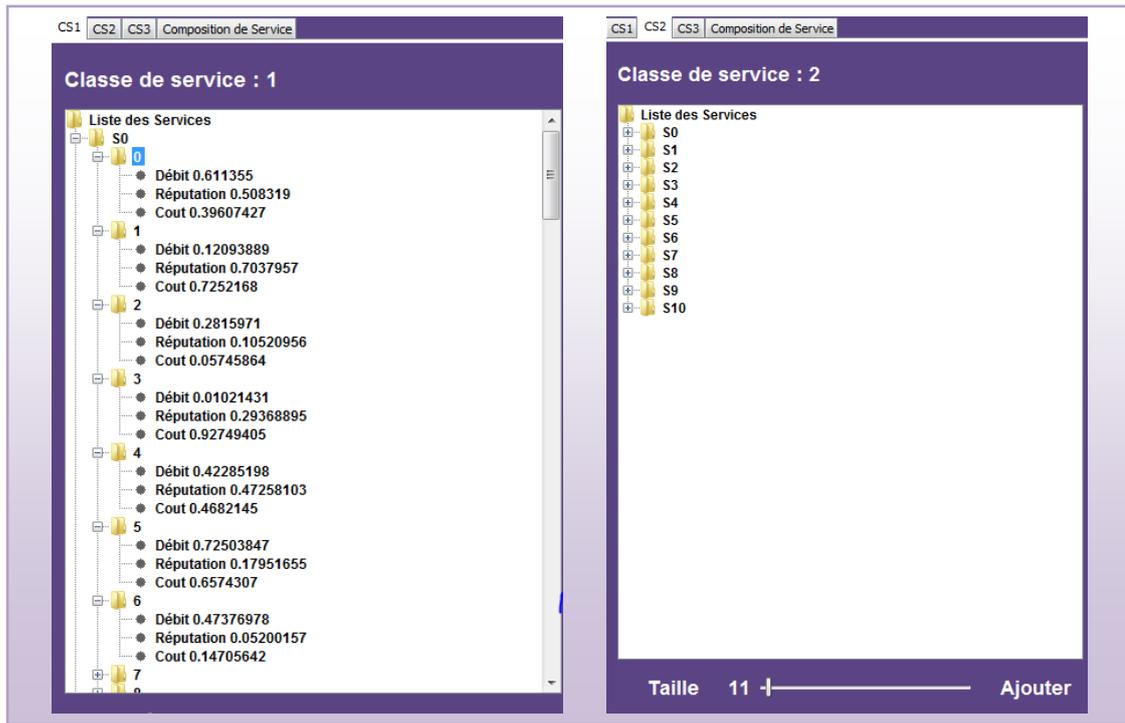


Figure II.10 Affichage des services des classes CS1, CS2, CS3

## ❖ Fenêtre des Approches

**Calculatrice de distance**

**service 1 :** s3

**service 2 :** s4

**Distance QA : 7.8631163**

**Distance QOS : 8.996467**

**Distance Finale : 8.410723**

 **Distance entre deux services**

**Figure II.11 Calculatrices de distance entre deux services**

Nous appliquons les algorithmes QA, QR pour calculer la distance entre deux services de nos choix



Figure II.12 Génération de la base

Si nous avons par exemple dans chaque classes 5 services alors nous obtenons  $5*5*5$  compositions

The screenshot displays two main panels. The left panel, titled 'Calculatrice de distance', contains two dropdown menus for 'service 1' (set to 'C4') and 'service 2' (set to 'C3'). Below these are three lines of text: 'Distance QA : 5.0707703', 'Distance QOS : 9.946625', and 'Distance Finale : 7.101905'. At the bottom of this panel is a dark blue button with a white icon of two overlapping circles and the text 'Distance entre deux services'. The right panel, titled 'Selection du Client', features a dropdown menu for 'Requete' (set to 'C3'). Below it are three lines of text: 'Meilleur Service : S2', 'Distance QA : 4.978804', and 'Distance Finale : 6.9208384'. At the bottom of this panel is a dark blue button with a white magnifying glass icon and the text 'Recherche de similarité'.

**Figure II.13** Calcule de distance et recherche de similarité entre les compositions

Nous appliquons les mêmes algorithmes QA, QR pour calculer la distance entre deux compositions de nos choix et nous pouvons aussi rechercher la similarité entre une requête parmi nos compositions, nous calculons la distance entre la requête et tous les autres services, la distance minimale est considérée comme un meilleur service.

CS1	CS2	CS3
C1-S0	C2-S0	C3-S0
C1-S0	C2-S0	C3-S1
C1-S0	C2-S0	C3-S2
C1-S0	C2-S0	C3-S3
C1-S0	C2-S0	C3-S4
C1-S0	C2-S0	C3-S5
C1-S0	C2-S1	C3-S0
C1-S0	C2-S1	C3-S1

Temps	Débit	Réputation	Cout
0	0.090073705	0.14674425	1.1875291
1	0.2184547	0.08266796	1.7473558
2	0.00431635	0.012766129	1.3228575
3	0.24807441	0.13745189	0.98328245
4	0.21820685	0.046075515	1.0962006
5	0.08550256	0.10313423	1.234123
6	0.008446072	0.024431707	1.5777987
7	0.1733467	0.002284566	1.4066133

**Figure II.14 Affichage de débit, cout, réputation des compositions**

Les compositions ont aussi un certain débit, cout, réputation.

Débit= le minimum des débits des services.

Réputation= la multiplication des réputations des services.

Cout= la somme des couts des services.

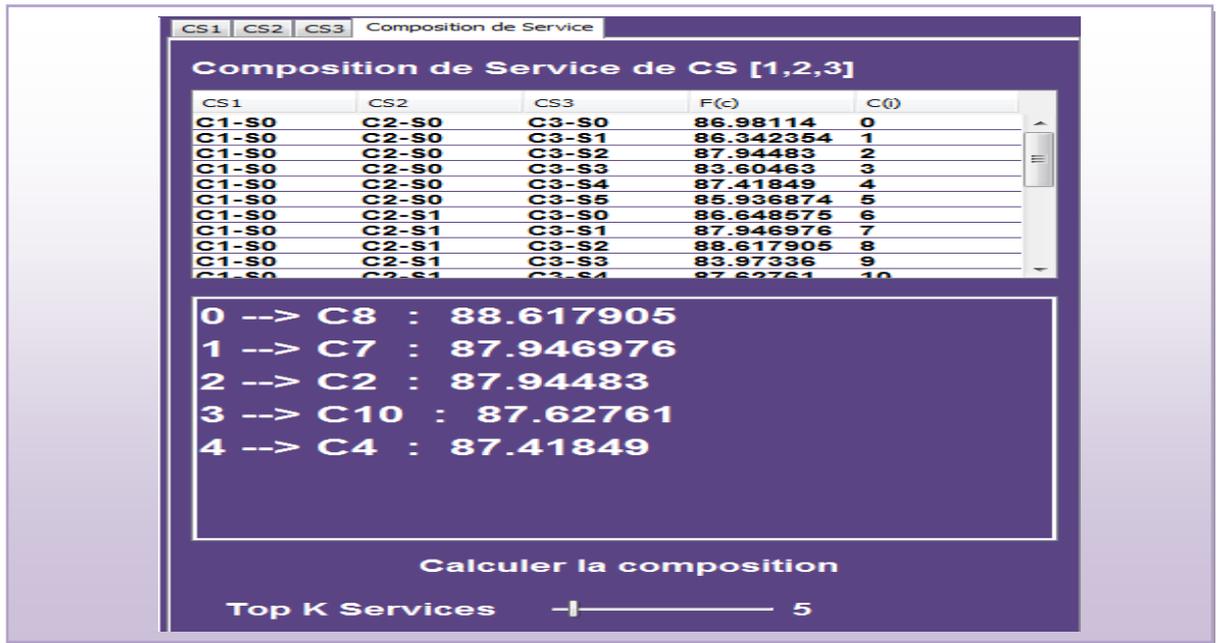


Figure II.15 Calcul de top K des compositions

Le top K est calculé selon une fonction objective F

$$F(c) = \sum_{i=1}^{nbre\ criteres} W_i \left( \sum_{t=0}^{|periode|} AggreQoS_i(t) \right)$$

### V. Spécification

Cette section montre nos expériences liées à la composition du services cloud . Ces expériences ont été menées sur une machine Intel(R) Core(TM) i3 CPU M 380 @2.53GHz avec 4.00 Go de la RAM, sous le système d’exploitation Windows 7.

Le temps d’exécution : c’est la période de temps qui s’écoule entre la validation de la requête de l’utilisateur et la réception des résultats finaux (pour le cas de découverte, composition ou sélection).

Les tops k des compositions sont générés selon la fonction objective F

$$F(c) = \sum_{i=1}^{nbre\ criteres} W_i \left( \sum_{t=0}^{|periode|} AggreQoS_i(t) \right)$$

On caractérise chaque composition c par le triplet (aggreQoS1(t), aggreQoS2(t), aggreQoS3(t)) pour chaque instant t (et t=0 jusqu'à 149)

Nous associons le même poids  $w=0.33$  aux trois critères (débit, cout, réputation)

## VI. Conclusion

Dans ce chapitre, nous avons décrit brièvement la collection de test utilisée dans notre application. Ensuite nous avons présenté la conception en spécifiant les algorithmes et les mesures de similarité.

Nous avons aussi montré dans la partie expérimentation les résultats obtenues avant et après la génération de la base.

# Conclusion générale

### *Conclusion générale*

Le cloud computing présente une technologie prometteuse qui facilite l'exécution des applications scientifiques et commerciales. Il fournit des services flexibles et évolutifs, à la demande des utilisateurs, via un modèle de paiement à l'usage.

Le terme Cloud Computing, est un nouveau modèle informatique qui consiste à proposer les services informatiques sous forme de services à la demande, Cette nouvelle technologie permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosse quantité d'informations.

La composition des services cloud est considéré dans une perspective à long terme. Les modèles économiques Cloud pour les utilisateurs finaux et les fournisseurs des services cloud sont exploités pendant la composition.

Dans notre travail nous avons traité le problème de la recherche de composition des services cloud. Ce dernier devient un Problème de recherche de similarité dont la requête est un ensemble de séries temporelles souhaitées, nous avons utilisé 2 mesures de similarités basées sur la distance euclidienne entre les valeurs de QOS et la corrélation entre les critères de QOS. Dans la partie expérimentation, nous avons amélioré notre approche les résultats obtenus confirment la supériorité de l'approche proposée.

Références bibliographiques  
**Références bibliographiques**

**Références bibliographiques**

- [1] Bashir, F.I., Khokhar, A.A., Schonfeld, D.: Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Transactions on Multimedia* 9(1), 58–65(2007).
- [2] Berkelaar, M., Eikland, K., Notebaert, P., et al.: *lpsolve: Open source (mixed-integer) linear programming system*. Eindhoven U. of Technology (2004).
- [3] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: *Web services architecture*. Technical report, W3C, Web Services Architecture Working Group, February (2004).
- [4] Canfora, G., Di Penta, M., Esposito, R., Villani, M.: An approach for QoS-aware service composition based on genetic algorithms. In: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pp. 1069–1075 (2005).
- [5] Cui, B., Zhao, Z., Tok, W.H.: A framework for similarity search of time series cliques with natural relations. *IEEE Transactions on Knowledge and Data Engineering* 24(3), 385–398 (2012).
- [6] Douglas, B .K.: *The Savvy Manager’s Guide to Web Services and ServiceOriented Architectures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, (2003).
- [7] Johnston, S.: Diagram showing overview of cloud computing. Retrieved from [http://en.wikipedia.org/wiki/File:Cloud\\_computing.svg](http://en.wikipedia.org/wiki/File:Cloud_computing.svg). (2009).
- [8] Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. *Data Mining in Time Series Databases* 57, 1–22 (2004).
- [9] Milanovic, N., Malek, M.: Current solutions for web service composition. *IEEE Internet Computing*, 51–59 (2004).
- [10] Motahari-Nezhad, H., Stephenson, B., Singhal, S.: Outsourcing business to cloud computing services: Opportunities and challenges. *IEEE Internet Computing* (2009).
- [11] Ponge, J.: *Model Based Analysis of Time- aware Web Services Interactions*. Thèse de Doctorat de l’Université Blaise Pascal – Clermont-Ferrand II, dans le cadre de l’Ecole Doctorale des Sciences pour l’Ingénieur, France, (2008).

- [12] Rahul, B., Nitin, C.: Cloud Computing: Service, models, Types, Database and issues, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March (2013).
- [13] Rajan, S., Jairath, A.: Cloud computing: The fifth generation of computing. Communication Systems and Network Technologies (CSNT), 2011 International Conference on. IEEE, (2011).
- [14] Sullivan, L.: Cloud Computing in Life Sciences R&D, report on InsightPharmaReports, Cambridge Healthtech Institute (CHI), 781-972-1353 (2010).
- [15] Vincent, K., et al.: Cloud Computing, IUT Nancy Charlemagne, (2009/2010).
- [16] Ye, Z., Bouguettaya, A., Zhou, X.: QoS-aware cloud service composition based on economic models. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) ICSOC 2012. LNCS, vol. 7636, pp. 111–126. Springer, Heidelberg (2012)
- [17] Ye, Z., Zhou, X., Bouguettaya, A.: Genetic algorithm based qoS-aware service compositions in cloud computing. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part II. LNCS, vol. 6588, pp. 321–334. Springer, Heidelberg (2011).
- [18] Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. IEEE Transactions on Software Engineering 30(5), 311–327 (2004).

**Liste des tables :**

**Tableau I.1:** Les grands acteurs mondiaux de Cloud.....14  
**Tableau I.2:** Avantages et inconvénients des services.....16  
**Tableau I.3:** Avantages et inconvénients du Cloud Computing.....16

**Table des figures :**

**Figure I.1:** Evolution de l'informatique.....10  
**Figure I.2:** Centre de données informatique en nuage.....11  
**Figure I.3:** Les couches du Cloud computing.....15  
**Figure I.4:** Composition des Services Cloud.....21  
**Figure II.5** Interface d'accueil.....32  
**Figure II.6** Interface de simulation.....33  
**Figure II.7** Liste des services de classe CS1.....34  
**Figure II.8** Affichage de (débit, cout, réputation) d'un service.....35  
**Figure II.9** Tableau de débit, cout, réputation de service  $S_0$ .....36  
**Figure II.10** Affichage des services des classes CS1, CS2, CS3.....37  
**Figure II.11** Calculatrices de distance entre deux services .....38  
**Figure II.12** Génération de la base.....39  
**Figure II.13** Calcule de distance et recherche de similarité entre les compositions.....41  
**Figure II.14** Affichage de débit, cout, réputation des compositions.....41  
**Figure II.15** Calcul de top K des compositions.....42

## Résumé:

Dans ce projet, nous adressons le problème de recherche de compositions de services Cloud, Notre solution proposée consiste à concevoir des mesures de similarités entre les besoins de l'utilisateur (exprimés sous forme de séries temporelles) et les services Cloud publiés Ces mesures doivent gérer la distance entre les séries temporelles de même types ainsi que les corrélations entre les critères hétérogènes.

**Mots clés:** l'informatique en nuage, composition des services Cloud, séries temporelles

## Abstract:

In this work, we address the problem of cloud service composition. This issue is tackled by implementing a set of similarity measures between the user's need (represented as temporal series) and the advertised cloud services (represented with the same model) The proposed similarity measures handle both the correlation between the homogeneous QoS criteria and the heterogeneous QoS criteria.

**Key words:** Cloud computing, cloud services composition, time series.

## ملخص:

في هذه المذكرة، نعالج مشكلة تركيب الخدمات السحابية، والحل المقترح لدينا هو تصميم تدابير التشابه بين احتياجات المستخدم (معبّر عنها بالسلاسل الزمنية)، والخدمات السحابية، يجب أن تكون هذه التدابير إدارة المسافة بين السلاسل الزمنية من نفس النوع وكذلك الارتباطات بين المعايير الغير المتجانسة

**كلمات مفتاحية:** الحوسبة السحابية، تكوين الخدمات السحابية، السلاسل الزمنية