République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Farhat AbbasSétif-1 – (Algérie)

THESE

Présentée à la Faculté des Sciences

Département d'Informatique
Pour l'Obtention du Diplômede

DOCTORAT EN SCIENCES

Option : Informatique

Thème

Approche Green Computing agile guidée par les ontologies d'accès automatique à des environnements intelligents

Présentée par

Kamel MANSOURI

Soutenu le :16/07/2019

Devant le jury composé de:

Pr. Kamel Nadjet	**Prof Université Ferhat Abbas Setif-1**	**Président**
Pr. Gasmi Abdelkader	**Prof Université Mohamed Boudiaf M'Sila**	**Examinateur**
Pr. Boubetra Abdelhak	**Prof. Université M^{ed} El Bachir El Ibrahimi B.B.A**	**Examinateur**
Dr. Salim Bouamama	**MCA. Université Ferhat Abbas Setif-1**	**Examinateur**
Dr. Adel Alti	**MCA Université Ferhat Abbas Setif-1**	**Rapporteur**
Dr. Philippe Roose	**HDR. Université de Pau France**	**Co-Rapporteur**

Remerciement

Louange à Dieu Tout Puissant de m'avoir donné courage et patience pour mener à bien cette thèse.

Mes tous premiers remerciements vont à Dr. Adel Alti, maitre de conférences à l'UFA-Sétif-1 pour m'avoir encadrée, conseillée et orientée le long de ce travail. Je lui exprime ici toute ma gratitude et ma reconnaissance.

Je remercie également monsieur Philippe Roose pour, maitre de conférences HDR, Université de PAU - France m'avoir fait l'honneur d'être co-encadreur de ce travail. Je le remercie pour leurs précieux commentaires et recommandations.

Je remercie également les membres du jury Pr. Kamel Nadjet pour l'honneur d'avoir accepté de présider le jury. Que le Pr. Gasmi Abdelkader, Pr. Boubetra Abdelhak, trouvent également toute ma gratitude en acceptant de faire partie de ce jury. Je tiens toute mes remerciements à Dr. Salim Bouamama pour l'honneur d'avoir accepté d'examiner ma thèse.

Mes remerciements vont également à L'équipe du Laboratoire et L'équipe du Département, pour leur très bon accueil et leur fructueuse collaboration.

Je tiens à exprimer à tous mes enseignants du primaire à l'université, toute ma reconnaissance, vénération et respect pour leur contribution inestimable à ma formation.

Je termine par un chaleureux remerciement à ma famille avec qui j'ai partagé toutes mes peines et joies.

RESUME

Au cours des dernières années, l'utilisation des smartphones et tablettes a augmenté de façon significative. En effet, les utilisateurs, accèdent à tout instant, et en tout lieu à des services (intelligents) accessibles sur le Cloud. L'utilisation massive de nouveaux services intelligents et technologies mobiles hétérogènes pour découvrir et déployer des services Cloud a conduit à un problème de compromis entre le coût d'accès et le déploiement des services et l'amélioration de la qualité des services (temps de réponse rapide, prix bas, sécurité améliorée, réduction de la consommation d'énergie, etc.). Dans ce contexte, cette thèse propose une approche dynamique inspirée de Green Computing et guidée par une ontologie pour laréduction de la consommation d'énergie et des émissions en carbone des centres de données. Cette méthodologie inclut la définition d'un cadre de travail impliquant la phase de conception sémantique ainsi qu'une technique bio-inspirée pour l'optimisation de la qualité des services intelligents et d'énergie dans le Cloud.

Dans une première partie, le cadre de modélisation fournit un niveau sémantique qui permet l'enrichissement automatique des requêtes clients et le filtrage sémantique des services de qualité. Elle facilite ainsi l'interopérabilité sémantique des services intelligents Cloud hétérogènes et gère les qualités des services. Cette hétérogénéité résulte de la représentation des différents langages et des fournisseurs Cloud de services. Pour uvoir satisfaire les différentes exigences de qualité de service, le modèle sémantique intègre les exigences et les préférences des clients. Le modèle sémantique (ontologie)se concentre sur l'optimisation d'énergie des centres de données qui stockent tous les services Cloud qui supportent les activités des utilisateurs et intégration des nouveaux services intelligents selon ses domaines d'applications (e-santé, e-communication, e-tourisme, e-éducation, etc.) via une classe générique catégorie de service et ses classes spécifiquessur le formalisme des langages sémantiques SPARQL. Ce formalisme est utilisé pour le filtrage sémantique des services de domaine et classifiée par préférences de client et consommation énergétiques.

Dans une deuxième partie, la problématique d'optimisation de l'énergie de sélection et de composition des services sélectionnés est traitée. Une nouvelle approche qui utilisant le principe de l'algorithme des colonies de fourmis(ACO) associé à des principes de filtrage sémantiques des services guidée par une ontologie est proposée pour améliorer les performances de l'essaim en termes de temps de réponse et de la quantité d'énergie consommée dans le Cloud.

Keywords : Environnement intelligent, Cloud, Energie, Optimisation, Ontologie, Qualité de Service, Sensibilité aux contextes, Algorithme de colonies de fourmis

ABSTRACT

Currently, everybody can access to existing services on the Cloud from a wide variety of mobile devices at any time and from anywhere (at home, working, on the car, etc.). The massive use of new heterogeneous mobile devices and technologies for discovering and deploying Cloud services has led a trade-off between costs and improved quality of services (e.g. fast response time, low price, improved security, the reduction of energy consumption and considerable emissions of Carbon, etc.). This trade-off has led most Cloud service providers to call for new intelligent, faster and energy-saving solutions.

In a first part, the semantic modeling framework provides a semantic level that allows the automatic enrichment of client requests and the semantic filtering of quality services. It facilitates the semantic interoperability of heterogeneous Cloud services and manages the quality of services. This heterogeneity results from the representation of different languages and Cloud service providers. To meet the different quality of service requirements, the semantic model integrates the requirements and contextual preferences of customers. The semantic model (ontology) focuses on the energy optimization of data centers that store all Cloud services that support user activities and integrate new smart services according to its application domains (e-health, e-communication, e-tourism, e-education, etc..) via a generic class of service class and its specific classes on the formalism of SPARQL semantic languages. This formalism is used for semantic filtering of domain services and classified by customer preferences and energy consumption.

In a second part, the issue of optimizing the energy of services composite is handled. A new approach based on Semantic Web technologies and Ant Colony Optimization (ACO) algorithm which intends to reduce the energy consumption of a wide variety of Cloud services.

Keywords: Intelligent Environment, Cloud, Energy, Optimization, Ontology, Context-aware, Quality of Service, Smart service, Ant Colony Optimization Algorithm

ملخص

حـــالياازداداستخداماالهواتفالذكيةوالأجهزةاللوحيةبشكلمكثف.
فيالواقع،المستخدمينيمكنهمالوصولفيأيوقت،وفيأيمكانإلىالخدماتالمتوفرة فيالسحابة. أدىالاستخداماالمكثف للهواتفالذكيةالمتنقلةغيرالمتجانسةوالتقنياتالمبتكرةلاكتشافونشرالخدماتالسحابيةإلىمقايضةبينتكاليفالوصولإلىالخدماتوتحسينجودةال خدمات (سرعةالاستجابة،انخفاضالأسعار،تحسينالأمن،انخفاضاستهلاكالطاقة،وماإلىذلك).
فيهذاالسياق،تقترحهذهالأطروحةنهجًادينامي كيًّامستوحىمنالحوسبةالخضراء(*Green Computing*) للحدمناستهلاكالطاقةوانبعاثاتالكربونفيمراكزالبيانات.
تتضمنهذهالمنهجيةمرحلةتصميمالهيكلالدلالي(أنطولوجيا)بالإضافةإلىتقنيةمستوحاةمنالتقنياتالحيويةلتحسينجودةالخدماتالذكيةوال طاقةفيالسحابة.

فيالجزءالأول من العمل المقدم،يوفرإطارالتصميممستوًدلاليًّايسمحبالإثراءالتلقائيلمتطلباتالزبونويسهلإمكانيةالتشغيلالمتبادلالدلاليللخدماتالسحابيةغيرالمتجانسةوإدارةجودةالخدمات. ينتجهذاالاختلافعناستعمالاللغاتالمختلفةومقدميالخدماتالسحابية المتنوعة.لتلبيةهذهالطلبات،يدمجالهيكلالدلالي (أنطولوجيا)متطلباتالعملاءوتفضيلاتهمالسياقية. يركزالهيكلالدلالي (أنطولوجيا) علىتحسينالطاقةلمراكزالبياناتالتيتخزنجميعالخدماتالسحابيةالتيتدعمأنشطةالمستخدموتكاملالخدماتالذكيةالجديدةوفقًالمجالاتالتطبيق (الصحة،الاتصالات،السياحة، التعليم،وماإلىذلك) عبرفئةشموليةمنفئاتالخدمةوفئاتهاالمتخصصةوفقا للغة. *SPARQL* يتماستخدامهذه المفاهيمللوصف الأنطولوجيللخدماتالسحابيةوتصنيفهاحسبتفضيلاتالعملاءواستهلاكالطاقة.

فيالجزءالثاني،يتمالتعاملمعإشكاليةتحسينطاقةوالحصول علىالخدماتالمختارة. منهججديديعتمدعلىتقنياتالويبالدلاليوخوارزمية *Ant Colony Optimization (ACO)* التيتهدفإلىتقليلاستهلاكالطاقةلمجموعةواسعةمنالخدماتالسحابية.

المفتـــاح: بيئة ذكية،السحابة،مركز البيانات، الطاقة، التحسين إلى الأمثل،الهيكل الدلالي(أنطولوجيا)، الخدمة، خوارزمية مستعمرة النمل، جودة الخدمة.

Table des matières

Table des figures

Liste des tableaux

Abréviations

ACO	Ant Colony Optimization
DSGreenACO	Dynamic Semantic Green Ant Colony Optimization
DSCxGreenSCloud	Dynamic Semantic ConteXt-aware Green Service Cloud Ontology
DVFS	Dynamic Voltage and Frequency Scaling
GA	Genetic Algorithm
E3-MOGA	Multiple Objective Genetic Algorithm
Green-IT	Informatique verte
HBCI	Human-Building-Computer Interaction System
HPC	High Performance Computing
IaaS	Infrastructure as a Service
ICA	Imprerialist Competitive Algorithm
IT	Information technology
Mesos-DNS	DNS-based Service Discovery for Mesos
MMKP	Multidimension Multichoice Knapsack Problem
MOGA	Multi-Objectif Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm
OGC	Open Green Cloud
OWL	Web Ontology Language
OWL-S	Semantic Markup for Web Services
PaaS	Platform as a Service
PSO	Particle Swarm Optimization
QoSSSP	QoS based Service Selection Problem
QoC	Quality of Context
QoS	Qualité of Service
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SaaS	Software as a Service
SDN	Software Defined Network
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
URI	Uniform Resource Identifier
UDDI	Universal Description Discovery and Integration
VDC	Virtual Data Centers
VM	Virtual Machine
WFlow_EU	Work Flow Effective Utility
WS_HEU	Web Service HEUristic for MMKP
WFlow_HP	Heuristic for HP Problem.
WSN	Wireless Sensor Network
W3C	World Wide Web Consortium
XML	Extensible Markup Language

INTRODUCTION GENERALE

1. Contexte et problématique

Au cours des dernières années, les organisations internationales telles que Google, Amazon, ou IBM se sont mises à offrir des services intelligents, distribués relatifs au stockage et au traitement de données à distance avec une garantie de disponibilité. Actuellement, tout le monde peut accéder aux services existants sur le Cloud à partir d'une grande variété d'appareils mobiles (Tablette, PC, Smartphone, SmartTV, Smart-Objets, etc.) à tout moment et de n'importe quel endroit (maison, poste de travail, voiture, etc.). L'utilisation massive d'appareils mobiles hétérogènes avec de multiples technologies de découvertes et d'accès aux services telles que Mesos-DNS, Consul, etc. Nous oblige à faire face à de nombreux nouveaux problèmes. L'un d'eux consiste à proposer des solutions simples garantissant un compromis entre les coûts (prix, consommation des ressources CPU, RAM et bande passante), le besoin de qualité de services (temps de réponse rapide, haute sécurité, fiabilité améliorée, etc.) et la consommation d'énergie parfois exorbitante. Les clients sont confrontés à des situations d'usages dans un contexte évoluant tel que l'évolution de la connectivité réseau, qui perturbe les différentes transactions et le choix des services à invoquer selon le contexte (par exemple le choix d'un service plutôt qu'un autre pour des raisons de bande passante insuffisante ou la disparition d'un matériel). Chaque fournisseur peut adresser un grand nombre de clients (requêtes) auxquels il est nécessaire de répondre dans un temps restreint (et souvent garanti contractuellement). Ceci peut impliquer le choix d'un service en lieu et place d'un autre auprès d'un autre fournisseur de Cloud parmi plusieurs existants. Cela implique qu'un fournisseur de Cloud doive implanter un nouveau mécanisme intelligent et dynamique pour gérer simultanément les demandes de plusieurs clients et ainsi répondre à la qualité de service requise en termes de sécurité, de prix, de fiabilité et de temps de réponse.

Des méthodes d'allocation des ressources, de sélection et de composition des services intelligents optimaux sensibles à la QoS ont été récemment proposées afin de réduire l'énergie dans les centres de données tout en respectant les paramètres QoS (sécurité, prix et temps de réponse, etc.) et les attributs de contexte (i.e. énergie)[1][2][3][4][5][6][7][8][9][10][11][12] . Cependant, ces mécanismes ne tiennent pas compte des changements des préférences de l'utilisateur (coûts minimaux, temps de réponse rapide, haute sécurité, etc.) pendant la découverte et le processus de sélection de services de qualité dans le Cloud. De plus, ces techniques n'ont pas pris

en compte la nature dynamique des performances du système Cloud. Les mécanismes actuels sont caractérisés par certaines insuffisances. Ils sont agnostiques au domaine applicatif qui, selon le domaine (tourisme, santé, transport, etc.), possède des besoins différents. Un nombre important des services Cloud hétérogènes issus de différents fournisseurs Cloud avec différentes contraintes d'exécution (énergie, RAM, CPU, bande passante) rend la gestion de la qualité de service très difficile. Cette hétérogénéité implique une dégradation de la qualité lors de sélection ou d'exécution de service. Un filtrage sémantique permettrait d'améliorer durant l'étape de recherche des services sémantiquement incompatibles avec ceux qui ont été déjà sélectionnés. Cependant, la qualité de service d'un fournisseur de Cloud n'est pas statique car elle évolue au cours du temps en raisonde nombreux facteurs tels que le nombre des accès clients entrants et sortants du Cloud d'une manière dynamique. Le développement d'une nouvelle technique capable de découvrir et de sélectionner facilement, rapidement et économiquement des services de qualité est de première importance face à l'accroissement des besoins. Notre objectif est la découverte des fournisseurs Cloud pertinents et la gestion des services Cloud en fonction de la QoS et du domaine via l'utilisation de techniques liées au web sémantique.

L'informatique verte ou bien « Green Computing » [1] se présente aujourd'hui comme un enjeu stratégique qui permet une optimisation des ressources des ordinateurs via l'utilisation de technologies d'économie d'énergie appliquées aux techniques et dispositifs. Nous nous intéressons dans cette thèse, principalement à fournir des services adaptés aux utilisateurs en fonction des coûts d'accès et de déploiement des services, profil de l'utilisateur, ses besoins fonctionnels et ses préférences de QoS et son contexte pour optimiser le coût énergétique de ces derniers.

Face aux besoins sans cesse grandissants et exponentiels en énergie utilisées par nos dispositifs et applications informatiques faisant usage du Cloud, il devient nécessaire de proposer des solutions réduisant cette consommation. Les difficultés inhérentes à ce type d'approches (coupures réseaux, bande passante fluctuante, charge des serveurs) qui nécessite de trouver des solutions afin de fournir les services de manière plus économe, en particulier en jouant sur leur sélection. Nous exploitons pour ce faire les approches sémantiques permettant l'enrichissement sémantique des requêtes utilisateur. Ces requêtes sont basées sur des modèles ontologiques génériques sémantiques et spécifiques et les approches stochastiques (ACO: Ant Colony Optimization) fournissent un moyen intelligent d'obtenir un temps de découverte rapide des services avec un minimum de coûtsqui respecte les besoins et les préférences de client. Le comportement social à base de fourmis guidée par les ontologies est

considéré pour l'amélioration du temps de découverte et la sélection des services pertinents et le taux de consommation de l'énergie.Deux modifications de l'algorithme ACO, portant sur le positionnement de la base et sur le modèle du taux d'évaporation de phéromone, ont été évaluées. L'adoption du choix adéquat du nombre de fourmis minimum par rapport au nombre de services Cloud est basée sur une fonction d'utilité de qualité. La stratégie de la fonction d'utilité de qualité a contribué en partie à l'amélioration des performances de l'essaim dans la découverte des services Cloud et l'optimisation au maximum la quantité d'énergie consommée dans le Cloud.

2. Objectifs et Contributions de la thèse

Ce document présent une approche basée sur deux concepts pour former une méthode hybride dans le but est l'optimisation de l'énergie dans la découverte et l'invocation des services

- Ontologie de description des services Cloud de qualité sensibles aux contextes. Nous avons proposé, une ontologie pour gérer un grand nombre de services Cloud hétérogènes groupés sémantiquement selon leur catégorie de service, leurs descriptions fonctionnelles, QoS et contextuelles. Cette ontologie facilite le processus de découverte des services Cloud pertinents, l'interopérabilité des services Cloud qui fournissent des qualités de service client, et met en évidence les contraintes sémantiques et les préférences des clients.

- Algorithmes de composition et de sélection des services avec QoS baptisé DSGreenACO (Dynamic Semantic Green Ant Colony Optimization). Ils s'exécutent afin de découvrir de manière intelligente et rapide les services Cloud pertinents sur les centres de données et répondant aux contraintes fonctionnelles des clients tout en permettant une optimisation énergétique. Le système filtre sémantiquement les services trouvés selon les préférences et le contexte courant du client et retourne les meilleurs services de qualité au client classés par leurs consommations énergétiques (CPU, bande passante et énergie).

3. Organisation de la thèse

Cette thèse est composée de cinq chapitres, plus une introductionet une conclusion.

Chapitre 1 : Convergence entre le Green Computing et les environnements intelligents

Nous présentons les concepts de base des environnements intelligents et ses domaines d'application. Nous abordons ensuite ainsi que une introduction sur le Green Computing et ses caractéristiques. Ce chapitre présente également les concepts de base des services et la sensibilité au contexte.

Chapitre 2 : Techniques d'optimisation de QoS et de l'énergie dans les environnements intelligents : Définitions et Etat de l'art

L'objectif de ce chapitre est de présenter un aperçu sur les différents domaines de recherches et les techniques d'optimisation énergétique dans le Cloud et nous montrons ses limites et ses avantages.Il existe deux approches d'optimisation de l'énergie dans le réseau Cloud. Des approches qui se focalisent à réduire les coûts d'énergie au niveau des centres de données, et d'autres approches qui se focalisent sur la réduction des coûts énergétiques dans des centres de données distribuées.

Chapitre 3 : Approche dynamique et sémantique Bio-inspirée d'optimisation de QoS et de l'énergie dédiée aux environnements intelligents.

Ce chapitre est consacré à la présentation de notre proposition pour optimiser l'énergie et la qualité de service liées à la découverte, la sélection et la composition des services hébergés dans les centres de données. En se basant sur une approche hybrideentre l'utilisation de l'ontologie(Web Sémantique) pour le filtrage sémantiquedes services et des métaheuristiques d'optimisation par colonie de fourmis(Bio-inspirée) pour l'optimisation de l'énergie pendant la découverte des services (Green Computing). Notre objectif est d'avoir une méthode dynamique guidée par les ontologies qui implémente la notion du Green Computing afin d'obtenir le chemin optimal (liste des centres de données et ses services) qui minimise la consommation de l'énergie et satisfait les besoins du client tout en permettant la découverte des services souhaités hébergés dans les centres de données.

Chapitre 4 : Description des services Cloud sensible aux contextes à base d'ontologie

Nous abordons dans ce chapitre le modèle ontologique d'optimisation d'énergie des centres de données qui stockent tous les services Cloud qui supportant les activités des utilisateurs selon ses domaines d'applications (e-santé, e-communication, e-tourisme, e-éducation, etc.) via une ontologie générique et ses ontologies de domaine. En outre, cette ontologie accélère le processus de découverte des services pertinents dans le Cloud, et améliore l'efficacité et la pertinence du processus de sélection et de composition des services Cloud. La classification sémantique est utile pour une prise de décision intelligente afin d'optimiser le processus de sélection du service cloud

selon les besoins des clients. Ces besoins diffèrent sensiblement en termes de contraintes et de préférences QoS.

Chapitre 5 : Composition économiques des services avec QoS basée sur lesalgorithmes bio-inspirés dynamiques et sémantiques

Nous proposons une nouvelle variante de l'algorithme d'optimisation par colonie de fourmis (ACO - Ant Colony Optimization) appelé DSGreenACO pour la résolution des problèmes d'optimisation de QoS et d'énergie. Nous proposons des informations heuristiques (débit inter-centre de données, charge appliquée sur les centres de données, nombre de requêtes clients sur les services hébergés) pour guider les fourmis dans leur découverte des services souhaités localisés dans les centres de données de l'espace de recherche précisés par le client puis filtrer à l'aide de règles appliquées à l'ontologie afin d'obtenir le chemin optimal qui minimise la consommation de l'énergie et satisfera les besoins du client.

Enfin, la conclusion générale parcourt les éléments réalisés dans la thèse, et propose quelques issues de la recherche, comme perspectives du travail réalisé.Dans annexe de ce document nous présentant les publications réalisées dans le cadre de ce travail.

Chapitre 1

1 Vers la convergence entre le Green Computing et les environnements intelligents

Sommaire

1.1 Introduction

De nos jours, les humains assistent à des sauts scientifiques et technologiques importants sous la forme d'une variété de dispositifs qui commencent à être utilisés par des personnes de tous âges et dans le cadre de leur routine quotidienne. L'informatiqueambiante est de plus en plus présente dans notre quotidien et consomme différents types de ressources : énergie (fabrication, transport, etc.), matériaux (papier, encre, etc.). Actuellement l'informatique représente 5% de la consommation mondiale d'énergie. C'est pour cela, il faut favoriser les conceptsqui consommentmoins d'énergie et réduire l'émission de CO_2.

L'informatique verte ou bien « Green Computing » [13]se présente aujourd'hui comme un enjeu stratégique qui permet aux entreprises à travers le monde de produire ou migrer leurs systèmes d'information dans un contexte de développementdurable.

Dans ce chapitre nous allons présenter les principaux axes de notre thème de recherche. Nous commençons d'abord par la description générale de l'informatique ambiante, ses différents concepts et ses domaines d'applications. Nous allons focaliser sur la notion de service et la notion de sensibilité au contexte pour décrire les applications ubiquitaires de tels environnements. Ensuite, nous introduisons le concept de Green Computing dans ce chapitre, en commençant d'abord par ses principales définitions,son historique et ses approches de base. Avant de conclure ce chapitre, nous présentons les problèmesde consommation d'énergie et de QoS liés à l'utilisation massive des services d'applications ubiquitaires dans le Cloud.

1.2 Les environnements intelligents ubiquitaire

1.2.1 Présentation de l'informatique ubiquitaire

La notion d'informatique ubiquitaire «Ubiquitous Computing» a été développé par Mark Weiser[14] qui est le responsable de recherche dans Xerox PARC (Palto Alto Research Center)au cours des années 80. Ila présenté sa vision de l'ordinateur du 21^{ième}siècle comme un terminal plus intelligent intégré dans les objets de la vie quotidien. Il a constaté que : « *les technologies les plus profondément enracinées sont les technologies invisibles. Elles s'intègrent dans la trame de la vie quotidienne jusqu'à ne plus pouvoir en être distinguées ».*

L'émergence d'un nouveau paradigme dit ubiquitaire se focalise sur les interactions des ressources de calculs distribués, pour assurer une assistance transparente d'un utilisateur dans les tâches qu'il accomplit au quotidien [2]. Avec l'augmentation de la puissance de traitement, des performances et fiabilité de stockage. Et également

concernée une large gamme d'infrastructures réseaux fiables, sécurisés avec une grande bande passante.La fusion de ces environnements avec les ordinateurs, le concept que les ordinateurs peuvent apparaître sous diverses formes à tout moment, est résumée sous le terme informatique omniprésent (Ubiquitous Computing).

Aujourd'hui, l'informatique pervasive fournit la base pour l'adaptation de comportement des objets de quotidiens communicantspour répondre aux nouvelles exigences des utilisateurs.Les objets communicants (capteurs et actionneurs) réagir de manière autonome sans l'intervention de l'être-humain. En raison de l'utilisation généralisée de sources d'information (images provenant de caméras embarquées, données de localisation GPS, identifiants et profils utilisateur stockés dans des applications d'informatique : sociale, commerciale, santé etc.).Tous ces dispositifs permettent l'échange de grande quantité d'informations disponibles.La présence de cette quantité d'information a poussé l'émergence de plusieurs types de services tels que le stockage de l'information, extraction de connaissances, la reconnaissance de l'activité ou la prédiction de l'intention des utilisateurs.

1.2.2 Définition d'un environnement intelligent

Les environnements intelligents sont un petit monde où toutes sortes d'appareils intelligents et différents objets de la vie quotidiennetravaillent en permanence pour rendre la vie des habitants plus confortable[15]. La figure 1.1 représente l'architecture générale de l'environnement intelligent.Donc, un environnement intelligent est une région largement équipée de capteurs, d'actionneurs et de composants informatiques hétérogènes qui coopèrent entre eux via des canaux de communication d'une façon transparente dans le but de faciliter la vie humaine.

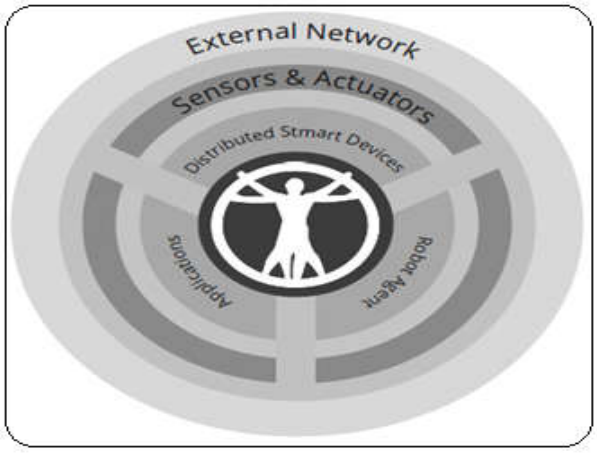


Figure 1-1Architecture d'un environnement intelligent [15].

Un environnement intelligent ubiquitaire est lié à l'informatique ubiquitaire (ou Ubiquitous Computing) qui est un concept en génie logiciel où l'informatique est faite pour apparaître partout et à tout moment. Ce concept est défini aussi comme un environnement riche de systèmes informatiques visibles ou invisibles tels que des capteurs, actionneurs, afficheurs et éléments de calcul, intégrés de façon transparente dans les objets quotidiens de notre vie et connectés par un réseau. Tous cesdispositifs communiquent et coopèrent pour atteindre le seul but de satisfaire nos besoins humains[16].Ce domaine est lié à l'informatique centrée sur l'utilisateur et met en évidence la sensibilité au contexte, qui définit la capacité d'un système à comprendre la situation actuelle dans l'environnement, à garder les traces de son évolution dans le système qui produit des réactions proactives. Pour éviter tout risque de confusion entre les termes utilisés dans le domaine de l'informatique ubiquitaire, nous présentons les définitions suivantes qui le caractérisent [5] :

- **Ubiquitaire** : accessible de n'importe où ;
- **Sensible au contexte** : qui prend en compte le contexte d'exécution ;
- **Pervasif** : qui associe ubiquité, mobilité et sensibilité au contexte ;
- **Transparent** : invisible pour l'utilisateur ;
- **Ambiante** : qui intègre dans les objets quotidiens.

1.2.3 Domaines d'application de l'environnement intelligent

Lesdifférents domaines d'application de l'environnement intelligent sontdiversetchacund'entre eux a ses caractéristiques et répond à certains critères spécifiques.

1.2.3.1 Domaine médical

L'environnement intelligenta un effet très important pour installer une culture médicale sur la santé des personnes. Elle permetd'accélérer l'intervention de l'équipe médicale, grâce aux notifications qui seront transmises aux personnes concernées.Dans ce contexte, on peut mentionner les montres Samsung Gear s4, LGWatch urbaine 2, etc. qui sont capable de mesurer la fréquence cardiaque, la température. Le patch de mesure et injection d'insuline, qui est capable de mesurer le taux d'insuline et d'injecter des doses selon le besoin [17]. En cas des situations critiques (attaques cardiaques), le smart-watch déclenche une réponse immédiate à la passerelle la plus proche et

demande une intervention médicale urgente où les coordonnées de la victime sont déterminées et envoyées.

1.2.3.2 Domotique

L'environnement intelligent dans la domotique permet d'assurer l'assistance des personnes dépendantes, la détection d'intrusions, etc. dans différents lieux (maisons, hôtels, entreprises, etc.). En citant par exemple Control Home Easily[18] qui est une application mobile sous Android qui contrôle les appareils domestiques à l'aide de widgets. Il découvre automatiquement les dispositifs intelligents de la pièce et les enregistrer avec différentes technologies matérielles et permet aux utilisateurs d'administrer et donner les ordres aux dispositifs. CAMP [15] est un programme fournit aux utilisateurs une interface virtuelle métaphore et magnétique pour traduire leurs actions sur un dispositif pour gérer et contrôler les objets intelligents de la maison. HBCI (Human-Building-Computer Interaction System) [19] est une application mobile Android qui donne à l'utilisateur l'économisassions de l'énergie et le confort en même temps. Il permet aux utilisateurs d'activer des actionneurs basés sur des capteurs connectés en définissant un seuil pour chaque capteur. De plus le système peut "apprendre" en temps réel les décisions préférées par l'utilisateur et, par conséquent, lui faciliter les ordres dont il a besoin

1.2.3.3 Domaine de voyage

Dans le domaine de l'information pour voyageur, un service peut être qualifié d'ambiant, s'il renseigne le voyageur en fonction de son environnement direct. Il s'agit donc de services capables d'adapter l'information en fonction des préférences spécifiques de l'utilisateur, de sa localisation selon le besoin afin de l'aider à réaliser son voyage.

1.2.3.4 Domaine de transport

Lors d'une inscription dans un réseau social ou professionnel (Linked, Google +, etc.), un remplissage du profil est nécessaire.En fonction des informations introduites, le système propose à tous les utilisateurs ayant un profil similaire, ou dans le cas des réseaux professionnels, toutes les sociétés intéressées par ce genre de profil. L'utilisateur peut être assimilé à une catégorie (débutant, professionnelou expert), dans ce contexte le système lui suggère des formations professionnelles pour accroitre ses compétences.

1.2.3.5 Domaine de l'énergie

Les maisons intelligentes ont été les premiers exemples de produits annoncés comme des environnements intelligents qui peuvent aider les occupants à gérer automatiquement l'éclairage et la température au nom de l'utilisateur. Le marketingintelligent permet une gestion plus efficace de ces ressources. Les gens du monde entier prennent ces questions beaucoup plus au sérieux aujourd'hui et les environnements intelligents peuvent être un instrument important pour atteindre ces objectifs.

Dans la section suivante, nous allons décrire les principaux défis posés dans le domaine de l'environnement intelligent.

1.2.4 Défis et caractéristiques de l'environnement intelligent

Les critères de nombre, mais surtout de variété et de dynamicité des services informatiques intelligents impliqués dans une application, décident du caractère pervasif de celle-ci. L'ouverture de l'environnement intelligent ubiquitaireà des services hétérogènes et variés dans un contexte évoluant rapidement pose les défis suivants :

1.2.4.1 Invisibilité et transparence

Un environnement intelligent ambiant doit satisfaire les besoins de l'utilisateur d'une manière transparente. Les environnements intelligents sont intrinsèquement dynamiques et hétérogènes, ce qui rend la technologie sous-jacente invisible est une tâche complexe vis à vis du lieu de l'utilisateur et son contexte actuel. Les défis de rendrel'accès à l'information plus transparente dans le domaine de la sélection et de composition des services intelligents sont nombreux et certains ont déjà été réalisés [18].Nous allons présenter dans les chapitres suivants l'approche que nous avons proposée pour la modélisation des services sémantiques sensibles au contexte et la façon de les optimiser.

1.2.4.2 Hétérogénéité

Des millions de dispositifs sont connectés, tous s'exécutant sur différentes plateformes d'exécution et systèmes d'exploitation, et communiquant via différents protocoles et supports de communication. Ces équipements communiquent via diverses technologies de communication filaires ou non filaires (ADSL, Wifi, Bluetooth, etc.) et consomment beaucoup d'énergie.

1.2.4.3 Dynamicité

Dans un environnement intelligent ubiquitaire, sa nature dynamique est due aux changements fréquents qui se produisent dans les environnements intelligents [18] et aux changementsdes besoins fonctionnels l'utilisateur, ses préférences et son contexte. Leprocessus d'adaptation doit répondre aux attentes des utilisateurs en réagissant à tous les changements attribués dans le contexte de l'utilisateur et le contexte de l'environnement. Le contexte utilisateur se présente comme besoins fonctionnels et QoSévolutives de l'utilisateur à travers le temps. Le contexte de l'environnement contient les informations sur l'état du système, tel que l'emplacement de l'utilisateur,son activité et de sa localisation, le débit de la connexion entre l'utilisateur et l'application,la disponibilité des ressources variées des objets communicants[20]. Pour cela, en déplaçant/remplaçant des services, on peut optimiser la consommation. Mais on peut remplacer certains services par d'autres moins qualité avec des fonctionnalités équivalentes mais moins gourmandes pour alléger la charge.

1.2.4.4 Economie d'énergie

L'utilisation des Smartphones et tablettes a augmenté de façon significative. En effet, les utilisateurs, accèdent à tout instant, et en tout lieu à des services intelligents accessibles sur le Cloud. L'utilisation massive de nouveaux services intelligents et technologies mobiles hétérogènes pour découvrir et déployer des services Cloud a conduit à un problème de compromis entre les coûts d'accès et de déploiement des services et l'amélioration de la qualité des services (temps de réponse rapide, prix bas, sécurité améliorée, réduction de la consommation d'énergie, etc.).

1.2.4.5 Flexibilité

La meilleure façon de répondre aux besoins et préférences des utilisateursest de déployer un système logiciel suffisamment flexible pour évoluer et s'adapter à tous changement des services et fluctuations des qualités de service. Par conséquent, la solution fournie à l'utilisateur doit être assez flexible pour réagir convenablement aux changements du contexte.

1.2.4.6 Fiabilité

La fiabilité des environnements intelligents ubiquitaires est essentielle, car un mauvais fonctionnement pourrait entrainer des conséquences sur les activités de l'utilisateur qui est au centre de ces systèmes. Par exemple, dans le cas d'assistance à une personne

malade, si un capteur (ex. de chute) tombe en panne, la sécurité de la personne pourrait être compromise.

1.2.4.7 Précision

La précision est d'importance cruciale lorsqu'il s'agit de services intelligents de précision qui peuvent échouer si le temps est retardé d'une milliseconde. Donc, le respect d'exigences strictes devient essentiel pour la santé et la sécurité des personnes.

1.3 Service et sensibilité au contexte

1.3.1 Architecture Orienté Service (SOA)

Plusieurs définitions sont utilisées pour définir et expliquer l'architecture SOA. La plupart de ces définitions insistent sur les aspects techniques de l'architecture, alors que, les autres s'intéressent aux caractéristiques métiers. Les définitions suivantes illustrent différentes vues de la SOA. Cependant, elles convergent toutes vers un seul sens :

« Une architecture SOA est une structure d'intégration de processus métier qui supporte une infrastructure des technologies d'information comme étant des composants et services sécurisés, standardisés et qui peuvent être combinés pour s'adresser aux priorités de changements métiers » [21].

Le modèle de conception et de développement de services Web, également appelé SOA (Service Oriented Architecture), explique comment archiver, réutiliser et intégrer des modules de services Web en tant que composant dans un autre système ou service [3]. SOA est de plus en plus utilisée pour un large domaine d'application et de nombreux leaders de l'industrie technologique l'ont adopté tel que Google, Amazon, IBM s'utilisent comme base de leurs produits, qu'ils exposent à des développeurs externes pour l'intégration avec de nouveaux produits, services et plates-formes. Cette tendance a facilité la création des modules réutilisables, des services de tous types consommés à la demande.

Du point de vue des applications, l'architecture orientée services permet le développement d'une nouvelle génération d'applications dynamiques ou composites. Ces applications permettent aux utilisateurs d'accéder à des informations et à des processus hétérogènes, et de les utiliser de différentes manières, notamment via le Web. Du point de vue de l'infrastructure, l'architecture orientée services permet au service Web de simplifier l'intégration des applications et des systèmes, de recombiner et de réutiliser les fonctionnalités des applications, et d'organiser les différentes phases

du processus de développement, dans un cadre cohérent et unifié. En réalité, la philosophie des SOA décompose une application monolithique en une suite de services assurant la modularité dans leurs fonctionnalités [2].

L'objectif de la notion de service est de promouvoir un accès simple et rapide aux fonctionnalités mises à disposition par les organisations. Aussi, la vision des services en tant que composants logiciels indépendants met en exergue les possibilités de coordination, ou composition de plusieurs services pour fournir des fonctionnalités avancées.Il existe une architecture pour les services dite de référence, elle contient trois couches principales, cette architecture vise trois objectifs importants [4] :

- identification des composants fonctionnels ;
- définition des relations entre ces composants ;
- établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

L'architecture de référence s'articule autour des trois rôles suivants (voirFigure 1-2)

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

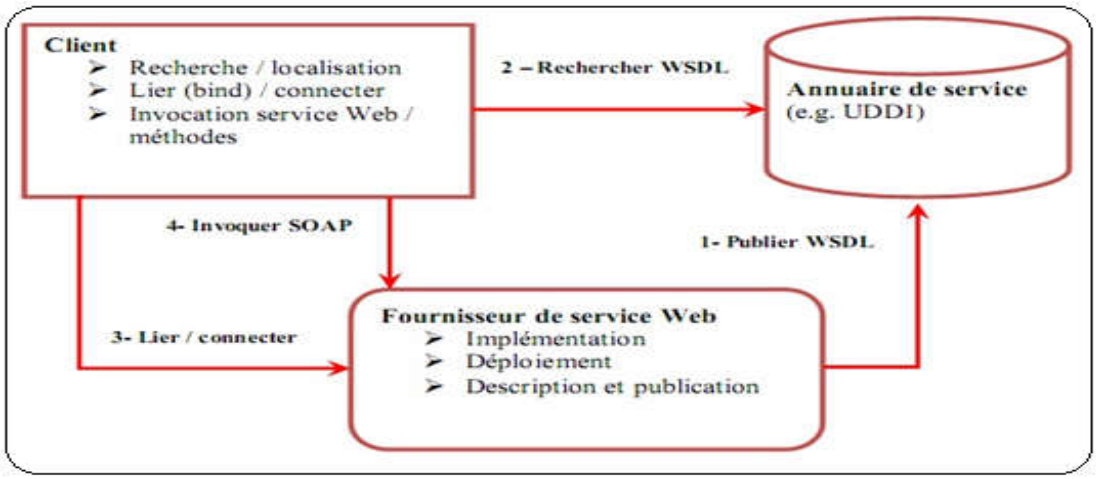


Figure 1-2Architecture de référence des services [15].

1.3.2 La notion de contexte

Le concept de contexte joue un rôle clé dans les environnements intelligents ubiquitaires. Plusieurs chercheurs ont essayé de cerner ce concept au moyen de définitions et de classifications, dont nous citons les plus importantes dans cette section. Nous nous intéressons également à la sensibilité au contexte, pour présenter quelques solutions qui prennent ces concepts en considération dans la sélection et la composition de services.

1.3.2.1 Définition

La définition standard de contexte selon [4]: ”Le contexte décrit toutes les informations qui peuvent être utilisées pour caractériser la situation d’une entité. Une entité est une personne, un lieu ou un objet ”.

Dans [20], Le contexte est l’environnement d’exécution incluent l’environnement des ressources matérielles, l’environnement de l’utilisateur et l’environnement physique.

1.3.2.2 Catégories du contexte

Plusieurs catégories ont été proposées (ex. en deux classes : le contexte primaire qui contient les informations sur la localisation, l’identité, le temps et l’activité (statut) ; le contexte secondaire qui peut être déduit de ce dernier).Nous classons les informations contextuelles utilisées dans la majorité des travaux existant en cinqparties de contexte [20].

- **Contexte utilisateur :** permet d’obtenir toutes les informations qui concernent le profil de l’utilisateur son identification, ses relations avec les autres usagers, la liste de ses tâches, etc.).et ce qu’il préfère avoir comme qualité de service.

- **Contexte physique :** offre la possibilité d’intégrer des informations relatives à l’environnement physique, telles que la localisation, l’humidité, la température, etc.

- **Contexte de l’environnement :**représente les informations disponibles caractérisant les connexions réseau, heure, la date et les contraintes de localisation. Les connexions réseau sont les propriétés liées au réseau type, protocole et bande passante. L'emplacement, la date et l'heure représentent respectivement la situation géographique exactement dans la période où le service doit être accompli.

- **Contexte d'activité :** répertorie les événements qui se sont déroulés dans l'environnement ainsi que leur estampille temporelle (entrée d'une personne et le moment de son entrée, tempête de neige et l'estampille temporelle de la tempête, etc.).

- **Contexte matériel :** permet d'identifier les appareils de l'environnement comme les tablettes ou smartphones, sont limités dans leurs ressources (taille de la mémoire, la vitesse du processeur, la durée de la batterie, etc.) et agissent comme environnement d'exécution pour le service web. Une sensibilisation de ressource sur l'utilisation actuelle de la puissance de traitement, mémoire, etc. est une condition pour garantir une qualité de service minimum.

1.3.2.3 Sensibilité au contexte (context-awareness)

La notion de sensibilité au contexte (context-awareness) a donc été produite dans le cadre des recherches sur l'informatique ubiquitaire. La sensibilité au contexte s'agit de la capacité d'un système à découvrir et à réagir à des changements dans l'environnement où il se trouve. Ils signalent également l'importance de l'adaptation du système à ces changements [22].

Pour Abowd et al. [23], un système est sensible au contexte s'il utilise celui-ci pour fournir à l'utilisateur des informations et des services pertinents. Cette pertinence dépend de l'activité que l'utilisateur est en train de réaliser. Ces auteurs proposent également une classification des systèmes sensibles au contexte selon leur capacité de réponse aux changements de celui-ci. Cette réponse peut soit présenter des informations ou des services à l'utilisateur, exécuter automatiquement un service, ou stocker des données contextuelles.

1.3.2.4 L'utilisation de contexte pour améliorer les applications

La sensibilité au contexte peut améliorer les interactions entre l'utilisateur, son environnement et l'application. La conception des applications sensibles au contexte soulève de nouveaux défis. En effet, la manipulation du contexte est difficile pour différentes raisons : (1) - Les méthodes classiques de développement des logiciels sont difficiles à appliquer sur les applications sensibles au contexte, (2) - Les solutions proposées pour ce genre d'applications sont soit spécifiques à un besoin précis soit elles manquent d'abstraction lors de leur conception et (3) - La capture du contexte est souvent distribuée et mène à des conceptions complexes.

Dey [24] (Cf.Figure 1-3) a été le premier à proposer une séparation entre l'acquisition de contexte et son utilisation dans les applications. Il s'appuie sur une étude de nouvelles méthodes de conception des systèmes interactifs mobiles où la séparation entre l'application d'une part et le nouveau monded'interfaces, d'icônes, de menus, de pointeurs et de moyens d'interaction sur les terminaux mobiles d'autre part.

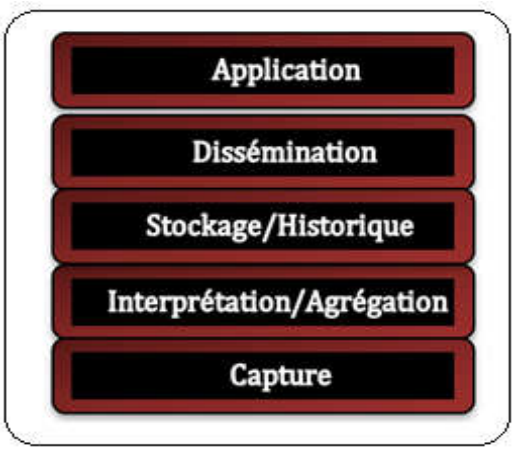


Figure 1-3Architecture générale d'uneapplication sensible au contexte.

1.4 L'informatique verte (Green Computing)

1.4.1 Introduction

La Terre est déjà confrontée au problème du réchauffement et des émissions de gaz à effet de serre en raison du développement de l'industrie, le transport et la consommation des ressources. On a vu déjà dans la partie précédente, comment les systèmes informatiques ont pénétré dans notre vie quotidienne, d'un serveur a un ordinateur personnel puis un portable, tablette, Smartphone jusqu'à l'arrivé aux objets intelligents. Cette croissance d'utilisation des systèmes informatiques a obligé les acteurs informatiques àfocaliserleurs attentions sur la consommation d'énergie de ces ressources informatiques et de proposer des idées et méthodes innovantes pour réduire la consommation d'énergie en utilisant des approches matérielles ou logicielles.

Les recherches sur la consommation d'énergie par les ressources informatiques sont poussées par le taux de consommation qui dépasse 3% et l'augmentation du cout de l'énergie mondiale. Une nouvelle branche de recherche appelée "Green Computing" a été inventé. Dans son article [5]« "Harnessing Green IT : Principles and Practices" », San Murungesan définit le domaine Green Computing comme suit « l'étude et la mise en place des méthodes pour la conception, la fabrication et l'élimination des ordinateurs, serveurs et sous-systèmes associés tels que moniteurs, imprimantes,

ainsi que les infrastructures de communication avec leurs logiciels pour minimiser leurs impact l'environnement ».

1.4.2 Définition

Le Green Computing peut être défini comme une approche innovante pour fabriquer des dispositifs matériels écologiques, des techniques pour développer des logiciels qui économise l'énergie et un ensemble de pratiques pour réduire la consommation de l'énergie[24].Les termes pour désigner le Green Computing sont nombreux : green-IT, informatique écologique, verte ou éco-responsable, éco-TIC...

Les principaux objectifs de l'informatique verte sont : (1) de réduire l'utilisation de matières dangereuses et (2) de maximiser l'efficacité de l'énergie au cours de la durée de vie des produits.

1.4.3 Pourquoi le Green Computing?

Avant le 18èmesiècle, les produits et les activités humaines étaient biodégradables, l'impact humain sur les ressources de la Terre était presque nul. Dans ces deux derniers siècles l'homme a consommé plus de ressources et a produit des déchets n'a absorbéen 10 siècles. Plusieurs raisons importantes pour adopter Green Computing : (1) Il favorise l'utilisation de produits dedéchets, (2) adopter le green peut également réduire les coûts et, par conséquent, bénéfique pour une entreprise, (3) il réduit également le risque d'utilisation de produits chimiques pouvant présenter un danger pour l'être humain, (4) il réduit également la pollution sonore (5) économise l'énergie pour une utilisation future. Les déchets informatiquesont une grande part à cause de la vie courte des produits informatiques et la vitesse des technologies informatiques.

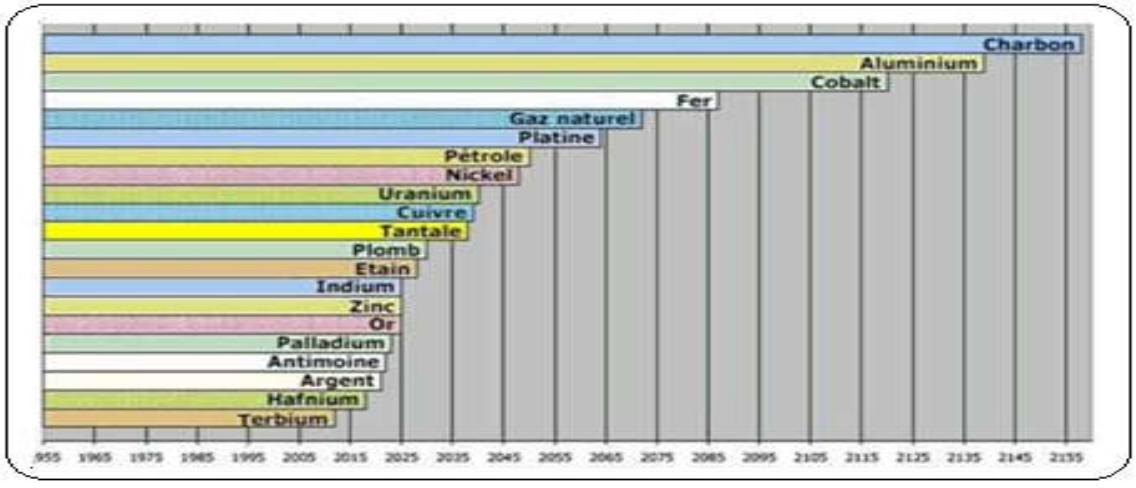


Figure 1-4Date prévisionnel d'épuisement des richesses de la terre[25]

Green Computing nous motive à repenser les tendances actuelles du matériel, des logiciels et des personnes pour minimiser l'impact sur la terre. Cela peut être réalisé par la fabrication de matériel respectueux de l'environnement, la conception et le développement d'un logiciel optimisé la consommation d'énergie et d'un système d'information pour économiser l'énergie, des processus et des pratiques écologiques adaptés pour les informaticiens. Il y a besoin de techniques et de méthodes pour concevoir et fabriquer des dispositifs respectueux de l'environnement, optimiser le codage de façon à réduire la consommation d'énergie et réduire la dissipation de la chaleur, éducation pour créer la meilleure mission de support des IT (Information Technology) et respecter les normes nationales et internationales. Le calcul distribué intelligent pour le partage d'informations sur le réseau peut également réduire l'énergie pour le flux de données. Le flux de données indésirable à travers l'application web est également l'un des grands problèmes responsables de la consommation d'énergie.

1.4.4 Approche globale de Green Computing

Le Green Computing représente uneéco-responsable de l'environnement informatique pour réduire la consommation électrique et les déchets électroniques de l'environnement. Virtualisation, Green Data Center, Cloud Computing, Grid Computing, l'optimisation de l'alimentation sont les technologies de l'informatique verte [2].

L'informatique verte traite les concepts de réduction de la consommation d'énergie, le recyclage des éléments dangereux est traité aussi la réduction et l'optimisation dans les entreprises en partageant les ressources (Cloud Computing). Il y a beaucoup d'étapes fondamentales qui peuvent être prises pour réduire de manière significative la consommation d'énergie et l'impact sur l'environnement [2].Il y a beaucoup d'étapes fondamentales qui peuvent être prises pour réduire de manière significative la consommation d'énergie et l'impact sur l'environnement, parmi les quelle on trouve :

- **Matériel électrique Basse** :les systèmes informatiques sont constitués de matérielsex : processeur, carte graphique, disque, ventilateur etc. Ces matériels doivent consommer moins d'énergie.

- **Réseau de capteurs sans fils** : capteur utilisé dans des zones différentes dans un centre de données pour déterminer la température de chaque zone. Ceci indiquera quelle région doit être plus froide et où réduire le refroidissement.

- **Virtualisation** : est l'utilisation de logiciels pour simuler un matériel. Dans le centre de données le serveur de système autonome est remplacé par un

serveur virtuel qui fonctionne en tant que logiciel sur un petit nombre d'ordinateurs, via un serveur virtualités.

- **Green Data Center :** par la production des composants recyclable, des déchets informatiques seront facilement réutilisée, ce qui réduisait l'impact sur l'environnement.

- **Cloud Computing** : il permet à quiconque d'obtenir des avantages environnementaux de la virtualisation. Il supprime également la nécessité pour l'utilisateur d'acquérir plusieurs ordinateurs puissants, car il fournit l'infrastructure en tant que service.

Dans notre travail on s'intéresse aux trois derniers points, Green Datacenter, la virtualisation et le Cloud Computing.

1.4.4.1 Green Data Center

Les centres de données ou les centres informatiques possèdent des systèmes informatiques et des composants associés, tels que le système de stockage de données et le système de télécommunication. Il a besoin d'une alimentation de secours, un système de refroidissement et un système de sécurité. Un centre de données vert est un centre de données qui dispose d'une gestion efficace du système etdes systèmes associés à la moindre consommation de l'énergie de l'environnement.

1.4.4.2 La virtualisation

La virtualisation est un ensemble de technologies permettant de faire fonctionner sur un seul serveur physique plusieurs machines virtuelles mettant en œuvre plusieurs systèmes d'exploitation et/ou plusieurs applications, indépendamment les uns des autres. Pour économiser du matérielle et de l'énergie, la virtualisation est écologique à plusieurs niveaux : (1) elle permet de réduire l'énergie consommée en réduisant le nombre de serveurs, (2) elle diminue également les besoins en matière d'air conditionné ou de système de refroidissement des salles de serveurs, (3) en minimisant l'achat de serveurs, et (3) l'entreprise minimise également l'injection de matériaux polluants dans l'environnement.LaFigure 1-5 représente la virtualisation.

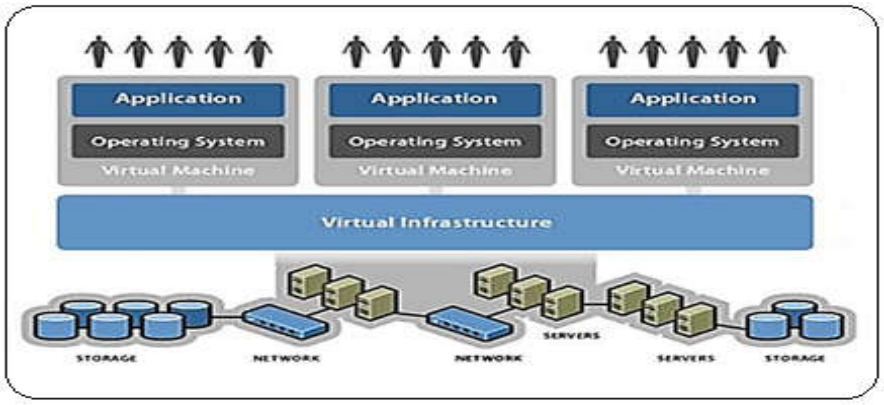


Figure 1-5La virtualisation[26]

1.4.4.3 Cloud Computing

Aujourd'hui, ilexiste plusieurs tentatives pour définir le "Cloud Computing". Par exemple, Vaquero et al.[27]ont comparé 22 définitions différentes dans une tentative de fournir une définition unifiée. Certaines de ces définitions sont générales : "les applications livrées en tant que services sur Internet et le matériel et les logiciels du système dans les centres de données qui fournissent ces services "[28]. D'autres sont plus spécifiques : "Un Cloud est un type de système parallèle et distribué constitué d'une collection des ordinateurs interconnectés et virtualisés dynamiquement provisionnés et présentés comme une ou plusieurs ressources informatiques unifiées baséessur les ententes de niveau de service établis par voie de négociation entre le fournisseurdes servicesetlesconsommateurs". LaFigure 1-6 représente l'architecture générale du Cloud Computing.

L'utilisateur deCloud peut utiliser les services ou l'application du Cloud Computing via un navigateur web ou une application tandis que le logiciel et les données de l'utilisateur sont stockées sur le serveur de données distant. Ainsi que le Cloud Computing permet aux entreprises d'obtenir leur application en cours d'exécution plus rapide avec une moindre intervention de l'homme et moins de maintenance, aussi il a l'avantage d'ajuster plus rapidement les ressources pour répondre à la demande d'activité imprévisibles.

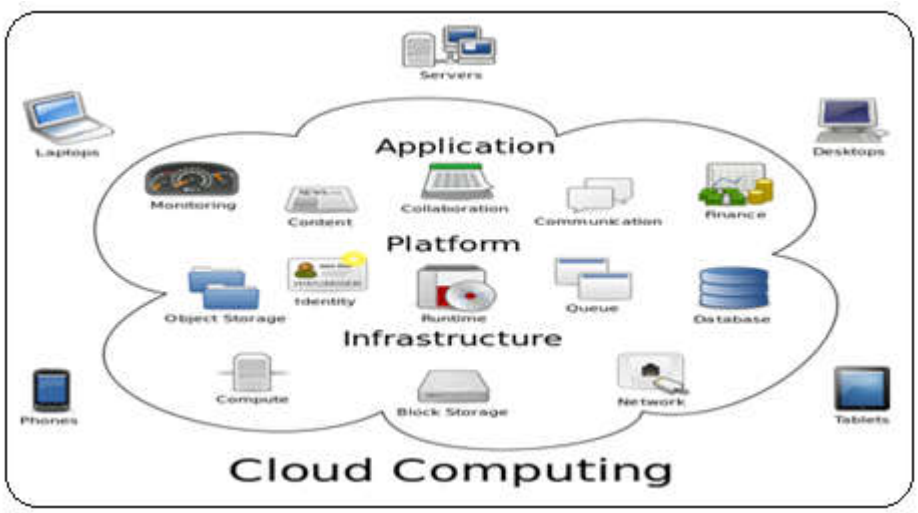


Figure 1-6Architecture du Cloud Computing[26]

1.5 Green Computing et les environnements intelligents

Les environnements intelligents sont des environnements à base de capteurs, actionneurs, de systèmes de stockage et de composants informatiques pour l'analyse et la décision. Ces systèmes travail en permanence pour récupérer, stocker et analyser

l'information. Par exemple dans le cas duSmart-Health, l'absence de la personne avec une activité permanente des capteurs des paramètres vitaux fixés dans la pièce est un gaspillage d'énergie. De cette problématique un nouveau paradigme tire son origine de la fusion des deux domaines Green Computing et les environnements intelligents dans le but de minimiser la consommation de l'énergie dans les environnements intelligents.

1.5.1 Capacité d'accès massive aux services sensible aux contextes

Aujourd'hui, les modes d'accès aux services sensibles au contexte (environnement, utilisateur)des systèmes intelligents sont variéset évoluenten nous proposant, de rester à chaque instant connecté à notre maison intelligent, à notre parking intelligent ou à notre environnement intelligent de travail.

De manière concrète, de nombreux terminaux légers (Tablette ou Smartphone) sont aujourd'hui utilisables pour communiquer et naviguer sur Internet en situation de mobilité ou de nomadisme. Les contextes d'exécution dans lesquels opèrent ces objets sont par nature hétérogènes. Les ressources offertes par les réseaux sans fil varient en effet avec le nombre et la position des utilisateurs. La disponibilité de la mémoire ou du potentiel de calcul des terminaux fluctue également dynamiquement. Enfin les besoins et les attentes des utilisateurs peuvent changer d'un instant au suivant. Ainsi, les recherches visant à doter les systèmes d'information modernes de capacités d'adaptation aux variations du contexte sont nombreuses.

Par ailleurs, grâce aux centres de données disponibles et grâce aux infrastructures performantes des réseaux accessibles au grand public, nous assistons également à un accès à tout instant, et en tout lieu à des services accessibles sur le Cloud. L'utilisation massive d'appareils mobiles hétérogènes avec de multiples technologies de découvertes et d'accès aux services telles que Mesos-DNS, Consul, etc. nous oblige à faire face à de nombreux nouveaux problèmes. L'un d'eux de proposer des solutions simples garantissant un compromis entre les coûts (au sens financier) et le besoin de qualité de service et la consommation parfois élevée d'énergie. Les clients sont confrontés à des situations d'usages dans un contexte évoluant tel que l'évolution de la connectivité réseau, qui perturbe les différentes transactions et le choix des services à invoquer selon le contexte (par exemple le choix d'un service plutôt qu'un autre pour des raisons de bande passante insuffisante ou la disparition d'un matériel). Chaque fournisseur peut adresser un grand nombre de clients (requêtes) auxquels il est nécessaire de répondre dans un temps restreint (et souvent garanti contractuellement). Ceci peut impliquer le choix d'un service en lieu et place d'un autre auprès d'un autre fournisseur de Cloud parmi plusieurs existants. Cela implique qu'un fournisseur de

Cloud doive implanter un nouveau mécanisme intelligent et dynamique pour gérer simultanément les demandes de plusieurs clients et ainsi répondre à la qualité de service requise en termes de sécurité, de prix, de fiabilité et de temps de réponse.

1.5.2 La dégradation de la QoS

Un service qui satisfait les contraintes de QoS est sélectionné pour participer à une composition peut ne pas offrir les mêmes QoS lors de l'invocation.Cependant, la qualité de service d'un fournisseur de Cloud n'est pas statique car elle évoluera au cours du temps en raison de nombreux facteurs tels que le nombre des accès clientsentrants et sortantsdu Cloud d'une manière dynamique.

1.5.3 La consommation d'énergie lors de découverte et déploiementdes services

Les problèmes d'optimisation d'énergie des centre de donnéeslors de la découverte et le déploiement des services sont aujourd'hui non seulement d'actualitémais aussi très importants, tant pour la compétitivité des serveursCloud de classe internationale (Google, Amazon, IBM, etc.). Malheureusement, surla question du compromis nécessaire pour avoir des temps de réponse accélérées à moindre coût, lors de déploiements des calculs intensifs, sans affecter négativement la qualité des applications. En raison du coût élevé de l'énergie et l'épuisement des ressources tel que l'énergie électrique qui est un facteur important dans notre cas pour la mise en marche des différents équipements intervenant dans la recherche des services Cloud.

1.6 Conclusion

Les environnements intelligents nécessitent des services de qualité et économiques pour rendre la vie des habitants plus confortable. Les fournisseurs Cloud offrir l'heure actuelle les meilleurs services distribués relatifs au stockage et au traitement de données à distance avec une disponibilité de tels environnements. Chacun de ces services ayant les aspects fonctionnels et les aspects non-fonctionnels (sécurité, fiabilité, prix, etc.) pour la performance vue par les utilisateurs.

Dans ce chapitre, nous avons réalisé une étude approfondie sur le concept qui regroupe l'ensemble des techniques et pratiques qui permettent de réduire l'empreinte environnementale et les coûts du système d'information appelé Green Computing. Nous nous sommes focalisés tout particulièrement sur les problèmes d'optimisation d'énergie et de Qualité de services des centres de données pour l'accès massiveaux services sensibles aux contextes.

Les différentes méthodes d'optimisation de qualité de service et de l'énergie qui sert à résoudre les problèmes de découverte et de la sélection des services pertinents font l'objet de chapitre suivant.

Chapitre 2

2 Sélection des services et optimisation de QoS et de l'énergie : Définitions et état de l'art

Sommaire

2.1 Introduction

Actuellement, tout le monde souhaite accéder à des services informatiques à tout instant, et en tout lieu à des services accessibles sur le Cloud à partir d'une grande variété d'appareils mobiles (Tablette, PC, Smartphone, SmartTV, Smart-Objets, etc.) avec différents modes d'interactions (souris, écran tactile, voix, détection vidéo, etc.). Les services hébergés ou auxquels ils accèdent à distance sont de plus en plus riches et complexes, et toutparticulièrementdans le domaine des environnements smart-*(Home, Health, Cities, Véhicule, etc.).

Dans ce domaine de qui nous intéresse plus particulièrement ici, l'utilisation massive d'appareils mobiles hétérogènes pour accéder aux services différents à partir de Cloud selon des scénarios mouvants, affecte à la fois la présentation de l'information mais également les services accédés et consomment une quantité d'énergie importante. Il existe un nombre important des services distribués relatifs au stockage et au traitement de données à distance.La sélection des servicesest parfois difficile etdoit être répondre aux besoins de moments des clients, ses contextes d'usages et ses préférences. Plusieurs approches ont été proposées : les approches d'optimisation de l'énergie et les approches de sélection des services avec l'optimisation de QoS. Ces approches permettant la modélisation des services d'une part, en se basent sur les standards W3C (profil utilisateur et services), d'inférer des contraintes et d'identifier les services et d'autre part sur la sélection de serviceset d'optimisation de QoS et d'énergie.

Dans ce chapitre nous présentons des définitions préliminaires, ensuite nous détaillons les différentesapproches de sélection de serviceset d'optimisation de QoS et d'énergieprésents dans la littérature. Enfin, un état de l'art des travaux qui proposent des approches de sélection de service Cloud est décrit suivi d'une étude comparative des travaux existants.

2.2 Présentation de la sélection de services Cloud

2.2.1 Définitions préliminaires

2.2.1.1 Définition de service

Selon Papazoglou: «Services are self-describing, open components that support rapid, low-cost composition of distributed applications» [29] . Cette définition considère un service comme une entité logicielle qui peut être utilisée par sa description.Il est caractérisé par une forte indépendance, le consommateur de service ne connait n'est le langage de programmation, n'est la plateforme d'exécution du service.

Selon W3C : un service est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML[29].Avant d'utiliser un service, il est nécessaire de connaitre ses capacités fonctionnelles (quelle est exactement l'action que le service peut réaliser) et ses caractéristiques non fonctionnelles (la disponibilité, le coût de son utilisation, etc.).

2.2.1.2 Description de services Cloud

Les services Cloud sont implantés par les fournisseurs, qui mettent à disposition les descriptions de services sous forme de fichiers. Ces descriptions sont centralisées et stockées dans des annuaires. Cette notion d'annuaire est comparable aux annuaires téléphoniques quenous utilisons pour accéder à des personnes ou des services. Les applications clientes envoient des requêtes aux annuaires pour sélectionner les services, de la même manière que nous recherchons un numéro dans un annuaire téléphonique. Elles téléchargent ensuite les descriptions des services sélectionnées, et les invoquent directement.

L'objectif de la notion de service est de promouvoir un accès simple et rapide aux fonctionnalités mises à disposition par les organisations. Aussi, la vision des services en tant que composants logiciels indépendants met en exergue les possibilités de coordination, ou composition, de plusieurs services pour fournir des fonctionnalités avancées.En plus des propriétés fonctionnelles, un service Cloud possède d'autres types de propriétés, tels que la qualité de service (QoS) qui font de ce composant logiciel le service le plus approprié à un client donné qu'à un autre. Une multitude de services Cloud sont fournispar différents fournisseurs et qui proposent des fonctionnalités similaires mais répondent différemment aux besoins et préférences spécifiques des clients.

2.2.2 Propriétés des services Cloud

On peut diviser les propriétés d'un service en deux types : **les propriétés fonctionnelles** et **les propriétés non fonctionnelles**.

2.2.2.1 Propriétés fonctionnelles

Unepropriétéfonctionnelleestunepropriétérelativeàlafonctionnalitéduservicequ'elle décrit [30]. Elles sont décrites généralement par une liste de condition à remplir avant et après l'invocation de service :

- Nom du service Cloud,
- Catégorie métier du service Cloud

- Paramètres d'entrés, leurs types et leurs tailles,
- Paramètres de sorties, leurs types et leurs tailles,
- Préconditions et post-conditions.

2.2.2.2 Propriétés non fonctionnelles

Les services sont construits pour satisfaire certains besoins fonctionnels des utilisateurs. Pourtant, ces utilisateurs ont aussi des besoins additionnels qui caractérisent leur aptitude à exister et à s'adapter aux variations de l'environnement dans lequel ils sont situés. Ces besoins sont dans la plupart du temps qualifiés de non fonctionnels. Dans cette catégorie on trouve par exemple : le temps de réponse, la stabilité à la charge, le coût de service, la consommation en énergie. On peut les classer en trois catégories :

2.2.2.1 Qualité de Service (QoS – Quality of Service)

La QoS est la mesure de la performance globale d'un service,notamment la performance vue par les utilisateurs. Pour mesurer quantitativement la qualité du service, plusieurs fonctions sont proposées dans la littérature souvent sous forme mathématique ou logique, mais elle prend toujours en considération le temps de réponse, la disponibilité, la stabilité a la charge, la sécurité et la tolérance.Chaque propriété est appelée attribut de QoS. Les attributs de QoS dépendent de domaine d'application, par exemple dans le domaine de télécommunication on focalise sur la QoS sécurité, dans le domaine de temps réel on focalise sur la QoS temps de réponse. Le tableau suivant décrit la liste des attributs de QoS couramment utilisés pour évaluer la qualité d'un service selon [31].

Les attributs de QoS	Description
Temps de réponse	Désigne la durée prise par un service pour répondre à une requête. Cette durée est susceptible d'inclure le temps de transmission engendré par le réseau lors de l'invocation du service.
Débit	Représente le nombre de demandes que le service est en mesure de traiter dans un intervalle de temps donné.
Disponibilité	Indique l'aptitude d'un service à assurer sa fonctionnalité pendant un temps donné. La valeur de la disponibilité s'exprime le plus souvent sous la forme d'un pourcentage.
Fiabilité	Cet attribut représente la capacité du service à réaliser correctement les tâches spécifiées dans sa description fonctionnelle.
Coût monétaire	Cet attribut désigne le prix qu'un client paie à chaque invocation d'un service.
Sécurité	Comporte différents aspects, tels que la confidentialité, l'intégrité, etc. caractérisant la sécurisation des échanges de messages entre le client et le service.

Tableau 2-1 Les attributs de QoS standards.

2.2.2.2 Qualité d'Experience de service (QoE – Quality of Experience) :

La qualité d'expérience permet une évaluation subjective d'un service, c'est-à-dire comment le service est perçu par le client. Dans cette catégorie on trouve par exemple, la qualité d'expérience sociale et la qualité d'expérience interactive qui mesure de la satisfaction d'utilisationde service.

Les attributs de QoE	Description
Expérience Sociale	Représente les différentes expériences utilisateurs des points de vue d'utilisation des services Cloud.
Expérience interactives	Reflète la satisfaction interactivedes utilisateurs.

Tableau 2-2 Les attributs de QoE standards.

2.2.2.3 Qualité de Context (QoC – Quality of Context)

La qualité de contexte dépend des spécifications des utilisateurs et de contexte. Dans cette catégorie on trouve par exemple, l'énergie consommée, la bande passante ou le cout d'invocation de service.

Les attributs de QoC	Description
Energie	Représente le niveau d'énergie consommé par un datacenter qui hébergé un service. Ce paramètre est mesuré en pourcentage. L'énergie est un paramètre positif, il est donc à maximiser.
Bande passante	La localisation de l'utilisateur en s'adaptant à son contexte d'une manière transparente.

Tableau 2-3 Les attributs de QoC standards.

2.2.3 Représentation sémantique des services

Un service Cloud sémantique est une unité d'application interopérable qui transcode les langages de programmation, les systèmes d'exploitation et les protocoles de communication réseaux. OWL-S [18]est un langage qui définit une ontologie des services Cloud. Il est basé sur le langage OWL. Il permet la découverte automatique et l'invocation de services Cloud basée sur leurs capacités et l'interopérabilité lors de l'exécution. Il fournit des modèles conceptuels aux services pour interpréter les informations fournies et rendre cette information accessible aux agents et aux programmes informatiques.

2.2.4 Découverte et sélection des servicesCloud

La découverte des services consiste à prendre comme entrée la requête de l'utilisateur, et renvoie la liste des services à partir de l'annuaire des services. Ceci est réalisé grâce à la correspondance syntaxique et sémantique des exigences fonctionnelles et les paramètres de QoS des services avec la requête utilisateur. La sélection des services permet de choisir des services en particulier parmi un ensemble de services selon des besoins fonctionnels ou des paramètres de QoS. Le code suivant décrit l'algorithme de sélection de service.

```
Input Goal = {g1,g2, g3,….gn} // Liste des besoins de l'utilisateur

      Service = {s1,s2,….sn} // Liste des services

Output MatchedServiceList ←ϕ // Liste des services qui répondant aux besoins d
l'utilisateur

Begin // correspondre les besoins de l'utilisateur avec les services

for each s in Service {

for each g in Goal {

        if match_service_category(g, s) AND match_service_city(g, s) AND

       match_service_QoS_parameters (g, s) then { MatchedService_List1.add(s); }

}

End;
```

Tableau 2-4 Algorithme de sélection des services.

Un touriste aventureux planifie un voyage en portant un sac à dos et à la recherche de nouveaux endroits à explorer. Sa Tablette avec GPS intégré est devenue indispensable pour trouver des moyens de transport à proximités qui correspondent à ses intérêts. Le tableau ci-dessous illustre un exemple représentant un extrait de la table de Matching syntaxique et sémantique des exigences fonctionnelles et des paramètres de QoS correspond au transport du domaine d'application smart-transport (scope-cible) et la requête client *a besoin* ***de services liés aux transports avec le prix est moins cher***(scope-source).

Requête Utilisateur	Service Transport	Fonction de Correspondance
Service.Transport	Service.Maritime Service. Bateau Service.Routier Service. TGV Service. Train Service.Tourism Service. Bus Service. Voiture Service.Aérien Service. Avion	Service. Transport ≡ Service Maritime Service.Transport ≡ Service. Bateau Service. Transport ≡ Service.Routier Service. Transport ≡ Service. TGV Service.Transport ≡ Service. Train Service. Transport ≡ Service Tourism Service. Transport ≡ Service. Bus Service. Transport ≡ Service. Voiture Service. Transport ≡ Service. Aérien Service. Transport ≡ Service. Avion
Transport.Prix = *moins cher*	Service.Prix	**Si** Service.Prix == Bas **Alors** Service est sélectionné Sinon Service n'est pas sélectionné Fin-Si

Tableau 2-4Correspondance requête utilisateur – service

2.2.4.1 Spécification de la requête utilisateur

La spécification de la requête utilisateur est définie par les besoins fonctionnels et non fonctionnels (QoS). Ils sont décrits sous forme des expressions en langage naturel ou sous formes des contraintes logiques du premier ordre ou la logique floue. Le langage du premier ordre est un langage plus expressif qui facilite la définition des exigences des utilisateurs.

Par exemple, si un visiteur a besoin trouver des restaurants, située à Sétif avec un temps de réponse rapide et prix moins cher. Cette contrainte peut être vue comme suit : **Type_Service = Restaurant AND Location = SETIF AND Prix = Bas AND Temps= Haut.**

2.2.4.2 Découverte des services Cloud

Dans le modèle des services, la découverte des services est l'activité de localisation d'une description de service sémantique traitable par la machine et qui répond à certains critères fonctionnels. On peut distinguer deux approches de localisation des services : les approches basées sur la recherche centralisée et les approches basées sur la recherche décentralisée. Dans la première approche, un annuaire universel des services UDDI (Universal Description Discovery and Integration) fournit un ensemble des services qui permettent la description et la découverte des organisations et d'autres fournisseurs des services. Elle offre aussi une API aux applications clientes, pour consulter et extraire des données concernant un service et/ou fournisseur. Dans la deuxième approche, un utilisateur soumet des requêtes via des brokers pour découvrir de manière distribuée les services Cloud sur les centres de données et répondant aux contraintes fonctionnelles des clients. Cependant, la gestion des informations dans un environnement distribué est une tâche complexe à cause de la description hétérogène des services Cloud par les différents fournisseurs et leurs consommations énergétiques importantes (RAM, CPU, bande passante et énergie).

Par exemple, si un client a besoin trouver des services santé a sa maladie pour planifier une intervention efficace. Si l'utilisateur ne reçoit pas de résultats convenables d'un fournisseur Cloud local, il peut envoyer ses besoins aux autres fournisseurs Cloud pour assurer une réponse rapide et efficace.

2.2.4.3 Matching des requêtes et des services Cloud

Les travaux d'appariement des requêtes et des services peuvent être répartis en deux principales catégories : les approches syntaxiques et les approches sémantiques. Les

techniques syntaxiques analysent une requête utilisateur comme un ensemble de chaines ou des définitions de paramètres et effectuent des Matching syntaxiques des chaines basées sur les méthodes de recherche d'informations [20]. Les techniques sémantiques exploitent les relations ontologiques pour réaliser les mappages entre les termes employés dans les descriptions des services et les requêtes utilisateurs. Quatre composants sont primordiaux dans le processus de Matching :

- **La sémantique des services :** la sémantique des services est obtenue via leurs descriptions. Ces descriptions forment les spécifications des services.

- **La sémantique des utilisateurs :**les types des spécifications des exigences par l'utilisateur doivent être compatibles avec les types des spécifications des descriptions des services.

- **La sémantique de domaine d'application :** Les concepts de domaines d'applications doivent être identifiés et représentés dans modèle interprétable par la machine. Exemple : Pour éviter des conflits de nommage on utilise des concepts communs tels que le concept vol et avion, produit et article, prix et coût, etc.

- **Le schéma de Matching :** décrit la correspondance entre les exigences utilisateurs et les descriptions des services selon la sémantique des services, des utilisateurs et des domaines d'applications.

2.2.5 Défis de la sélection et de la composition de servicesCloud

2.2.5.1 Respect des contraintes de QoS globale de l'utilisateur

L'objectif est de répondre aux besoins des utilisateurs et aux changements de l'environnement lorsqu'un grand nombre de services est disponible dans le Cloud.Le problème est la sélection du meilleur ensemble de services (à savoir en termes de qualité de service) pour répondant aux exigences de qualité de service globale de l'utilisateur.Nous nous attachons particulièrement à la gestion de la qualité de service des services/applications intelligents face aux contraintes matérielles, aux exigences de l'utilisateur et aux contextes d'usagells sont agnostiques au domaine applicatif et que selon le domaine (*tourisme, santé, transport, etc.*), les besoins diffèrent. Un nombre important des services Cloud hétérogènes issus de différents fournisseurs Cloud avec différentes contraintes d'exécution (*énergie, RAM, CPU, bande passante*) rend la gestion de la qualité de service très difficile. Cette hétérogénéité implique une dégradation de qualité de contrôle et de sélection des services.En effet, bien que les

caractéristiques proposées par ces fournisseurs Cloud tendent à améliorer la qualité du service rendu à l'utilisateur, leur intégration dans les systèmes traditionnels implique de relever plusieurs défis :

- **Mobilité :** la mobilité est un atout pour les systèmes intelligents ubiquitaires mais elle soulève néanmoins des problèmes. Le principe des systèmes ubiquitaires réside sur la collaboration des services et donc des serveurs Cloud qui les hébergent. La mobilité des périphériques entraine des variations de connexion qui peuvent avoir de lourdes conséquences sur le fonctionnement des services : l'indisponibilité des services et par conséquent dégrade la qualité de service.

- **Variation du contexte :** l'utilisateur et son périphérique évoluent dans un environnementsans cesse évoluant. Les besoins des utilisateurs évoluent, les ressources des périphériquesaussi tous comme l'environnement physique. Le fonctionnement desapplications peut êtreperturbé par ces modifications de contexte.

2.2.5.2 La dynamicité

Les environnements dynamiques exigent de répondre à la volée aux requêtes utilisateurs (c.à.d. au moment de l'exécution) bien que la disponibilité de services ne précise pas être connue a priori.La fluctuation de qualité de service en raison de la dynamicité de l'environnement. Ce problème se pose par exemple quand un ou plusieurs services faisant partie d'une composition ne sont plus disponibles ou de l'exécution ou que leur QoS baisse (par exemple, en raison de la déconnexion du réseau ou la connectivité réseau faible) pendant l'exécution de la découverte, de la sélection.

2.2.5.3 La consommation de l'énergie

L'utilisation des smartphones et tablettes a augmenté de façon significative. En effet, les utilisateurs, accèdent à tout instant, et en tout lieu à des services intelligents accessibles sur le Cloud. L'utilisation massive de nouveaux services intelligents et technologies mobiles hétérogènes pour découvrir et déployer des services Cloud a conduit à un problème de compromis entre les coûts d'accès et de déploiement des services et l'amélioration de la qualité des services (temps de réponse rapide, prix bas, sécurité améliorée, réduction de la consommation d'énergie, etc.).

2.2.5.4 Limite de temps de sélection de services

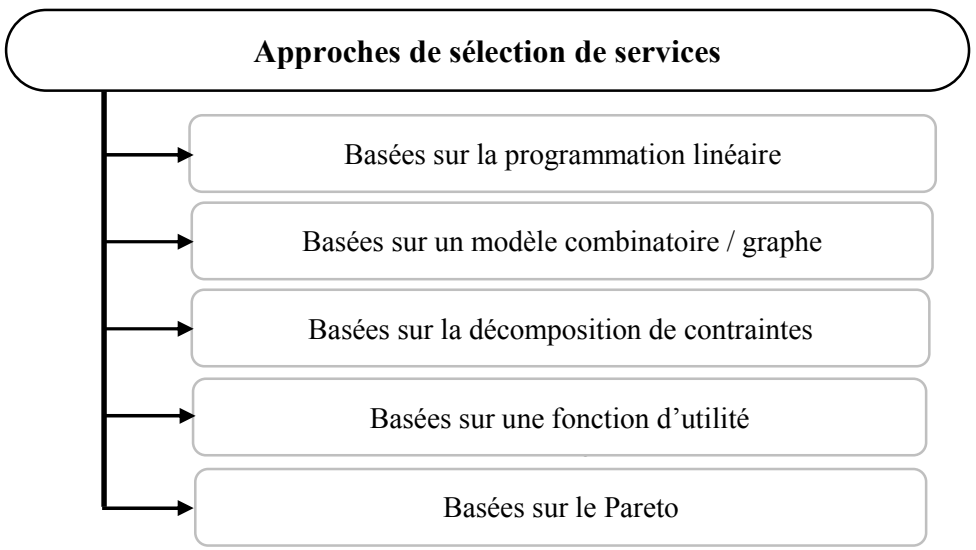
La sélection de services doit être exécutée lors de l'exécution de l'application. Par conséquent, le temps d'exécution des algorithmes de sélection de services est fortement limité alors que la complexité de calcul du problème est NP-difficile[32].

2.3 Etude de quelques travaux de sélection de services et d'optimisation de QoS et d'énergie dans le Cloud

2.3.1 Travaux de sélection des services Cloud sensibles aux QoS

Plusieurs travaux de recherche sur la sélection de service focalisenttout le domaine de l'optimisation. Aujourd'hui la recherche se focalise sur l'aspect non fonctionnel. En effet les auteurs cherchent à sélectionner un service parmi un ensemble de services équivalant, prenant en considération les propriétés fonctionnelles. L'objectif est de fournir un service au client garantissant la qualité requise.Le problème de sélection de service avec les critères de QoS (QoS based Service Selection Problem : QoSSSP) est un problème NP-difficile [32], car multi-objectif. Un modèle multi-objectif de QoSSSP peut être résolu en le convertissant en problème mono-objectif en utilisant des techniques de scalarisation.L'objectif de ces techniques est d'associer une valeur à chaque solution (service sélectionné) en fonction de certaines fonctions d'utilité, afin de sélectionner les meilleures solutions possibles. Différentes études incluant des méthodes exactes et heuristiques / métaheuristiques basées sur des techniques de scalarisation ont été proposées pour résoudre comme la programmation linéaire, les algorithmes génétiques (GA), l'optimisation des essaims de particules (PSO) et l'optimisation des colonies de fourmis (ACO). Le principal inconvénient des méthodes de scalarisation est que la pertinence de la solution dépend fortement de la conception des fonctions d'utilité (fitness). Des approches multi-objectives sont souvent utilisées lorsque deux ou plusieurs objectifs en conflits et doivent être prise en compte simultanément.

Les approches de sélection de services peuvent être classées selon la méthode de résolution utilisée. La figure ci-dessous (Figure 2-1) résume les méthodes selon



l'algorithme utilisé.

Figure 2-1Les différentes approches de sélection de service.

2.3.1.1 Approches basées sur la programmation linéaire

La programmation linéaire [33] a été utilisée dans plusieurs recherches pour résoudre le problème de la sélection optimale de services, parmi lesquelles on trouve les travaux de [34]. Ils proposent deux approches de sélection de services tenant compte des contraintes de QoS. La première effectue une sélection locale de services. Ensuite avec les technique d'aide à la décision multicritère, il cherche la solution global optimal.

La deuxième approche proposée par les auteurs tient compte des contraintes globales de QoS imposées au service composite, et formule la recherche de la composition optimale comme un problème de programmation linéaire en nombres entiers avec un ensemble de variables, une fonction objective à maximiser, et un ensemble de contraintes à satisfaire.

2.3.1.2 Approches basées sur un modèle combinatoire / graphe

La programmation linéaire en nombres entiers engendre une complexité exponentielle. Pour pallier cette limite, d'autres travaux de recherche utilisent le modèle combinatoire ou graphique, parmi lesquelles on trouve les travaux de [35]. Ils proposent des heuristiques pour le modèle combinatoire et d'autres pour le modèle de graphe.

Concernant le modèle combinatoire, il cherche d'une solution réalisable initiale qui est ensuite optimisée à travers un processus de remplacement des services ayant la valeur d'utilité la plus faible, appelés services nonréalisables.

Pour les structures de composition complexes, consiste à optimiser la composition concrète ayant la probabilité d'exécution la plus élevée, et à trouver une solution réalisable pour toutes les autres compositions alternatives [36].

2.3.1.3 Approches basées sur la décomposition de contraintes

La description de services et des requêtes utilisateurs sont spécifiées via des ensembles de contraintes fonctionnelles et de QoS. Par conséquent, le processus de Matching doit déterminer l'état de satisfaction entre deux ensembles de contraintes concernant le service et la requête. Nous pouvons citer les travaux d'Elgedawy et al. [37] et Goonerante [20]. Les inconvénients des méthodes de décomposition des contraintes sont l'incohérence des ensembles de contraintes et la complexité élevée des algorithmes de Matching.

2.3.1.4 Approches basées sur une fonction d'utilité

Le problème de sélection de services peut être résolu parune fonctionmono-objective (fonction d'utilité)via une technique de scalarisation. Nous pouvons distinguer les méthodes suivantes : recherche taboue [1], algorithme génétique (GA) [38], optimisation des essaims de particules (PSO) [39], algorithme de concurrence impérialiste (ICA) [28] et optimisation des colonies de fourmis (ACO) [24].Le principal inconvénient commun des méthodes de scalarisation est que la découverte du service composite avec le meilleur fitness peut pénaliser quelques QoS (i.e. si la fonction de fitness est la somme pondérée des critères QoS de sécurité et de temps de réponse, un service composite peut avoir le meilleur fitness avec une mauvaise sécurité).

2.3.1.5 Approches basées sur la dominance de Pareto

Les approches d'optimisation multi-objectifs sont souvent utilisées lorsque deux objectifs conflictuels ou plus doivent être considérés simultanément afin de négocier un ensemble de dominance au sens de Pareto (NSGA-II et E3-MOGA)[38].Les approches Pareto-optimales génèrent un ensemble de solutions, mais dans de nombreuses situations, les clients n'ont besoin que d'une solution qui doit être sélectionnée automatiquement parmi les solutions résultantes. L'algorithme NSGAII est le plus populaire des algorithmes d'optimisation multi-objectif. Il permet d'atteindre le front Pareto-optimal par un minimum d'itérations [38].

2.3.2 Travaux d'optimisation de la consommation d'énergie dans le Cloud

Plusieurs travaux récents portent sur les techniques d'optimisation énergétique des centres de données. Parmi ces travaux, une famille d'algorithmes d'optimisations vise à résoudre les problèmes d'optimisation d'énergie consommée dans le Cloud liée à l'utilisation massive des services. Il existe deux approches d'optimisation de l'énergie dans le Cloud. Des approches qui se focalisent à réduire les coûts d'énergie au niveau d'un centre de données, et d'autres approches qui se focalisent sur la réduction des coûts énergétiques dans des centres de données distribuées.

2.3.2.1 Green et efficacité au niveau des services

Dans [6], Amokrane et al. ont étudié la satisfaction des contrats Green SLA (i.e. Service Level Agreement) avec des exigences orientées green qui limite des émissions en carbone) des fournisseurs de services Cloud. En particulier, ils ont considéré les Green SLA qui spécifient une limite sur les émissions en carbone générées par le VDC (Virtual Data Centers) de chaque fournisseur de services. Pour cela, ils ont proposé

une plateforme GreensLater, qui orchestre l'approvisionnement et l'optimisation des ressources pour les multiples VDC hébergées par le fournisseur Cloud dans une infrastructure distribuée. Du point de vue du fournisseur Cloud, GreensLater profite de la variabilité dans l'espace et dans le temps des énergies renouvelables disponibles et les prix de l'électricité dans les différents centres de données. De plus, GreensLater utilise la reconfiguration dynamique pour optimiser dynamiquement l'allocation des ressources au fil du temps, tout en satisfaisant les contraintes des Green SLA (Service Level Agreement).

2.3.2.2 GreenHead : Placement Dynamique des Centres de Données Virtuels

Dans un autre travail, Amokrane et al. [7], ont proposé une plateforme de gestion des ressources pour le placement des VDCs) dans une infrastructure Cloud distribuée appelée GreenHead. Ce dernier vise à maximiser les revenus du fournisseur Cloud en minimisant les coûts d'énergie. GreenHead opère en deux étapes. Ils ont partitionné d'abord une requête VDC en plusieurs partitions telles que la bande passante intra-partition est maximisée et que sa demande est minimisée. L'objectif de ce partitionnement est de mettre les machines virtuelles qui s'échangent de gros volumes de données dans le même data center. Cela réduit considérablement le trafic acheminé par le réseau backbone d'interconnexion entre les centres de données, et améliore ainsi le taux d'acceptation des requêtes. Un fournisseur de services envoie sa requête de VDC au fournisseur Cloud, qui a la responsabilité d'allouer les ressources nécessaires. Naturellement, le fournisseur Cloud fera usage de son infrastructure distribuée avec l'objectif de maximiser son chiffre d'affaire et réduire les coûts d'énergie ainsi que l'émission en carbone. GreenHead est composé de deux types d'entités de gestion : (1) un contrôleur central qui gère l'ensemble de l'infrastructure et (2) un contrôleur local employé dans chaque centre de données pour gérer ses ressources.

2.3.2.3 Reconfiguration des flux de donnéesvia d'un contrôleur SDN

Amokrane et al.[8]proposent une plateforme qui permet de réduire la consommation d'énergie dans les réseaux de campus. Elle effectue une reconfiguration dynamique des flux de données et elle s'adapte dynamiquement au débit, tout en tenant compte des exigences et de la mobilité des utilisateurs. Cette proposition peut être intégrée en tant qu'application qui peut se déployer sur un contrôleur SDN (Software Defined Network). Les auteurs ont formulé le problème de réduction de la consommation totale d'énergie comme un programme linéaire. De plus, la fonction objective de programme linéaire prend en compte les couts de passage de l'état de veille à l'état actif pour un

nœud du réseau (points d'accès, Switch et routeurs Gateway), ainsi que le cout de consolidation de flux existants.D'autres travaux abordent le problème de la réduction de la consommation d'énergie à l'intérieur d'un centre de données unique en œuvrant sur l'ordonnancement des tâches dans le domaine du calcul intensif (High Performance Computing (HPC)) [40].

Diouri et al. [9] proposent également une plateforme de réduction de la consommation d'énergie dans les infrastructures HPC avec/sans connaissance préalable des applications et services à hébergé.

2.3.2.4 Allocation dynamique et économique des clusters

Da Costa et al. [41] proposent l'évaluation de la consommation de l'énergie et la prise de décision automatique pour l'allocation des ressources en clusters HPC. La plateforme proposée contient trois niveaux : (1) - journaux de consommation d'énergie dans un cluster d'un utilisateur en temps réel, (2) - l'intégration des utilisateurs dans les décisions d'économie d'énergie et (3) - la conception des techniques d'allocations automatiques des ressources pour avoir un compromis entre les performances des applications et la consommation d'énergie de l'infrastructure.

2.3.2.5 Allocation dynamique et économique des clusters

Lefèvre et al. [10] ont montré la capacité de réduire la consommation de l'énergie du centre de données jusqu'à 25% par la virtualisation et la fusion des machines virtuelles. La virtualisation permet d'améliorer l'efficacité de systèmes distribués à grande échelle. Cette technologie nécessite des mécanismes puissants pour la gestion des ressources. Ils ont utilisé la migration qui offre une nouvelle étape de la virtualisation en donnant la possibilité de déplacer une machine virtuelle déployée sur un nœud vers un autre nœud, ce qui supprime totalement le concept de localité dans les environnements virtualisés et qui par conséquent facilite la gestion des pannes, l'équilibrage de la charge et la maintenance système de bas niveau. Les opérations de migration impliquent une gestion plus flexible des ressources. Par opposition à l'affectation dynamique des machines virtuelles indépendantes aux tâches ou bandes passantes, les machines virtuelles différentes du VDC nécessitent une communication cohérente. Les auteurs ont proposé un modèle appelé OGC (Open Green Cloud). L'architecture d'OGC permet de mettre en état "off" les ressources de stockage, de calcul et de connexion.

Pour valider cette infrastructure, deux ordonnancements ont été utilisés : "Round-Robin" et "déséquilibré". Dans le premier ordonnancement, le premier job est mis dans

le premier nœud, le deuxième job dans le deuxième nœud, et ainsi de suite. Dans le deuxième ordonnancement, tous les jobs sont mis dans le premier nœud et s'il y a d'autres tâches dans le Cloud, ils utilisent le second nœud. Les expériences montrent que le scénario "Green" est le scénario qui consomme moins d'énergie dans les deux types d'ordonnancement. Le scénario "Green" avec l'ordonnancement "déséquilibré" consomme 25% moins d'énergie que le scénario de base avec l'ordonnancement "Round-Robin".

2.3.2.6 Planificateur dynamique des centres de données virtuel

Zhani et al. [11] ont présenté un Planner VDC, qui est un Framework d'intégration de centre de données virtuel dynamique prenant en compte les migrations, qui vise à générer des revenus élevés tout en minimisant le coût total de l'énergie au fil du temps. Et augmente ainsi le bénéfice des fournisseurs Cloud. Ils ont également proposé un algorithme de fusion VDC réduisant le nombre de serveurs physiques actifs pendant les périodes de demande faible. D'autres techniques ont été également proposées permettant d'allouer les ressources entre des centres de données distribués. Ils visent à réduire les coûts énergétiques [42] - [43], à minimiser l'émission de carbones dans les infrastructures [44] - [45] ou à la fois le coût de l'énergie et l'émission de carbone [46] - [47].

2.3.2.7 Contrôleur prédictif pour le déploiement dynamique des services

Zhang et al. [48] ont utilisé une plateforme de contrôle prédictif pour le déploiement de services dans les centres de données distribués. Les services sont dynamiquement placés dans des centres de données et ils sont migrés en fonction de la demande, du prix et de la fluctuation de l'électricité au fil du temps, tout en tenant compte du coût de la migration et la latence entre services et utilisateurs finaux. Pour ce faire, les auteurs formulent le problème en un programme linéaire dont l'objectif est de réduire le coût de l'énergie (par exemple, le coût de l'électricité) et le coût de reconfiguration (démarrer/arrêter des serveurs). Les auteurs ont utilisés des méthodes issues de la théorie des jeux dans le cas d'un seul ou de plusieurs fournisseurs.

2.3.2.8 Contrôleur prédictif pour le déploiement dynamique des services

Qureshi et al. [49] ont développé une politique de routage de sensibilité au coût d'énergie qui achemine les demandes selon l'endroit où l'énergie est la moins chère, et identifie la pertinence des écarts de prix de l'électricité dans de grands systèmes distribués afin d'estimer les économies qui pourraient résulter dans la pratique si le système a été déployé.

2.3.2.9 Contrôleur prédictif pour le déploiement dynamique des services

Dans [12], les auteurs ont étudié le problème de la réduction de la consommation d'énergie et les émissions de CO_2. La maximisation du profit et la garantie de la qualité de service (QoS) sont les objectifs de ce travail. Ils ont proposé un méta-ordonnanceur qui utilise un algorithme génétique multi-objectif (MOGA : Multiple Objective Genetic Algorithm) pour trouver le meilleur ordonnancement pour les applications au fil du temps. Les fonctions objectives de l'approche sont conçues pour minimiser la consommation d'énergie et les émissions de CO_2 avec une maximisation du profit de fournisseur. Différents paramètres ont été utilisés dans les expérimentations : la variation du coût des arrivées, le service client et le prix du cycle d'ordonnancement. Quatre types de taux d'arrivés ont été utilisés (vitesse basse, moyenne, haute et très haute). Les expériences ont été menées sur des traces réalistes tirées des archives du Feitelson Parallel Workload Archive (PWA). Une comparaison a été faite entre les algorithmes génétiques multi-objectifs. Les auteurs ont proposé une approche heuristique qui maximise le nombre d'application. Les expériences montrent que cette heuristique réduit la consommation d'énergie de 4,66%, les émissions de CO2 de 10,85%, maximiser les profits de 1,62% et les ordonnancements sont améliorés en moyenne de 2,67%.

2.3.2.10 Contrôleur prédictif pour le déploiement dynamique des services

Dans [50], les auteurs ont proposé une architecture décentralisée pour la gestion des ressources dans le Cloud Computing. La technique de migration à chaud des machines virtuelles a été utilisée. Pour économiser l'énergie, les nœuds sont mis dans un état activé/désactivé en utilisant la technique DVFS(Dynamic Voltage and Frequency Scaling). Cet algorithme est appliqué à une infrastructure de Cloud constituée de plusieurs nœuds physiques hétérogènes. Chaque nœud possède un CPU avec des processeurs performants et dispose de plusieurs machines virtuelles. Le nœud dans le Data Center est caractérisé par la quantité de RAM et de la bande passante du réseau. L'architecture du système proposé se compose de deux gestionnaires situés dans chaque machine physique et un dispatcheur. Dès qu'un utilisateur émet une requête, le dispatcheur distribue la demande au gestionnaire global. Le gestionnaire global échange les informations sur l'utilisation de la ressource et les machines virtuelles qui doivent être allouées et l'émission des ordres de migration VM. Les gestionnaires locaux envoient aux gestionnaires globaux des informations sur l'utilisation des ressources et les machines virtuelles sélectionnées pour la migration. Ils permettent également d'émettre des commandes pour

redimensionner les VMs. L'hyperviseur effectue les redimensionnements, les commandes et la migration des machines virtuelles selon les ordres reçus. La migration est effectuée lorsque : l'utilisation des ressources est trop faible ou atteinte 100%, machine virtuelle (VM) est en communication avec un autre réseau VM intensif situé dans un autre endroit où la température d'une ressource est trop forte.

Approches			QoS	Profil utilisateur	Dynamique	Contexte	Energie	Sémantique
Basé sur la programmation linéaire		[34]	Oui	Non	Dynamique	Oui	Non	Non
Basé sur les graphes		[35]	Oui	Non	Dynamique	Oui	Non	Non
Basé sur les contraintes		[4]	Oui	Non	Dynamique	Mobilité	Non	Non
		[5]	Oui	Non	Dynamique	Oui	Non	Non
Métaheuristiques	**Multi-objectives**	[51]	Oui		Statique	Non	Non	Non
	Mono-objectives	[1]	Oui	Non	Statique	Non	Non	Non
		[2]	Oui	Non	Statique	Non	Non	Non
		[3]	Oui	Oui	Dynamique	Non	Non	Non
Green		[6]	Non	Non	Statique	Non	Oui	Non
		[7]	Non	Non	Statique	Non	Oui	Non
		[8]	Non	Non	Statique	Non	Oui	Non
		[9]	Non	Non	Statique	Non	Oui	Non
		[10]	Non	Non	Dynamique	Oui	Oui	Non
		[12]	Non	Non	Dynamique	Oui	Oui	Non
		[11]	Non	Oui	Dynamique	Oui	Oui	Non

Tableau 2-5Comparaison des approches de sélection des services

2.3.3 Bilan et Synthèse

Le Tableau 2-1 illustre une étude comparative entre les différentes approches de sélection des services Cloud selon les critères d'évaluation. Les travaux basés sur les contraintes [4][5] utilisent un processus de filtrage selon des contraintes pour réduire l'espace d'exploration des services Cloud. Cette sélection introduit des erreurs aussi bien syntaxiques que sémantiques; ce qui implique une dégradation de qualité de sélection et de composition des services.

Les travaux basés sur les métas-heuristiques[51] se focalisent sur l'optimisation des QoS du service composite engendré. Cependant, l'appariement est basé sur la syntaxe et n'utilisent pas l'aspect sémantique. Un filtrage sémantique permettrait d'éliminer durant l'étape de la conception et de la sélection des services sémantiquement incompatibles avec ceux qui ont été déjà sélectionnés.

Les méthodes de scalarisation [1][2][3]transforment le problème d'optimisation multiobjectif en un problème monoobjectif en utilisant une fonction d'utilité (fitness). La gestion de la QoS statique est assurée par un processus de choix entre plusieurs services fournissant différentes qualités.En plus n'utilisent pas de mécanismes sémantiques au niveau du profil utilisateur pour satisfaire explicitement les préférences de l'utilisateur.

Les travaux basés sur l'optimisation de l'énergie[6][7][8][9][10][11] permettent de réduire l'énergie dans les Cloud. Les mécanismes proposés se concentrent principalement sur l'allocation optimale des ressources avec l'objectif de maximisation des revenus. Parmi les points qui non pas étaitabordés est la satisfaction des préférences clients. Le travail de[12]nous a permis de soulever ce problème lié à la satisfaction des besoins des clients tout en minimisant l'énergie dans le Cloud Computing.

La plupart des travaux existants n'utilisent pas de mécanismes sémantiques au niveau du profil utilisateur pour satisfaire explicitement les préférences de l'utilisateur, et donc ne visent pas à mieux sélectionner les services afin de fournir une meilleure réponse aux préférences des utilisateurs.De plus, ces travaux n'ont pas pris en compte la nature dynamique de la performance du systèmeCloud, tout en exploitent la flexibilité lors d'augmentation des charges et la consommation d'énergie sur un serveur (i.e. centre de données). Avec l'accroissement de la variété des fournisseurs de services Cloud hétérogènes, les clients ont besoin de services efficaces et toujours de haute qualité.

Le but de ce travail est la description sémantique durant la sélection, la composition des services Cloud afin de faciliter son adaptation au contexte d'utilisation. Afin de choisir le chemin de découverte et de sélection des services Cloud, les préférences utilisateur sont prises en compte (exemple, minimisation de temps de réponse), lors de l'optimisation de l'énergiede la découverte et exécution des processus de sélection de composition.

2.4 Conclusion

Dans ce chapitre, nous avons présenté les principales méthodes d'optimisation par métaheuristiques, de plus, nous avons cité les principaux travaux qui ont un rapport avec l'optimisation de l'énergie et de QoS dans leCloud Computing pour les applications sensibles au contexte.

Compte tenu du nombre croissant de ces travaux, l'utilisation des métaheuristiques pour résoudre les problèmes d'optimisation a fait ses preuves et s'est construit une solide base. Dans le chapitre suivant, nous présentons notre contribution dans le domaine du Green Computing par l'utilisation d'une nouvelle ontologie. Cette dernière qui permet d'utiliser l'aspect sémantique dans le processus de la sélection et de la composition des services Cloud afin d'assurer son efficacité qui est définit la base de nos propositions et contributions.

Chapitre 3

3 DSGreenACO : Approche dynamique et sémantique bio-inspirée d'optimisation de QoS et de l'énergie dédiée aux environnements intelligents

Sommaire

3.1 Introduction

Le Green Computingtendà devenir un domaine de recherche à part entière qui suscite beaucoup d'intérêts de la part des chercheurs de communautés très variées. Il est devenu un point de convergence technologique de nombreux acteurs du marché de l'informatique ambiante.L'optimisation de l'énergie est l'objectif primordial duGreen Computing.

Pour faire face aux besoins grandissants et exponentiels en énergie utilisées par nos dispositifs, services et applications ambiantes faisant usage du Cloud, il devient nécessaire de proposer des solutions réduisant cette consommation.Face aux difficultés inhérentes à ce type d'approches (*coupures réseaux*, *bande passante fluctuante, charge des serveurs*), nous devons trouver des solutions afin de fournir les services de manière plus économe, en particulier en jouant sur leur sélection. Nous exploitons pour ce faire les approches sémantiques permettant l'enrichissement sémantique des requêtes utilisateur.

De plus, comme les besoins des clients sont exprimés par différents langages et différentes expressions selon leurs contextes, des ontologies basées sur la sémantique sont nécessaires pour mieux répondre à ces besoins.L'ontologie a acquis une signification plus large dans l'interopérabilité sémantique entre les domaines d'intelligences ambiantes hétérogènes. Elles jouent un rôle important pour fournir des informations sémantiques de la qualité et du contexte, essentielles à la découverte et le filtrage des services de qualité. Dans ce cadre, les ontologies peuvent fournir un modèle conceptuel partagé pour enrichir la sémantique des requêtes clients et permettre au client de localiser un service de qualité indépendamment du contexte du service évolué et du fournisseur de Cloud sélectionné. Par ailleurs, les approches bio-inspiréesfournissent un moyen intelligent d'obtenir un temps de découverte rapide des services avec un minimum de coûts.

Dans ce chapitre on va étudier, à travers une revue de la littérature, l'apport potentiel des méthodes bio-inspirées, et les ontologies dans la réalisation de notre proposition. Plus précisément : la première section donne un aperçu sur les méthodes bio-inspirées, tandis que la deuxième section définit les ontologies en présentant leurs types, ainsi que leurs langages de manipulation. Nous présentons ensuite également une nouvelle approche dynamique sémantique bio-inspiré Green Computing guidée parune ontologie afin d'assurer une optimisation de la qualité de services et d'énergie dans le Cloud.

3.2 L'intelligence en essaimet les méthodes bio-inspirées

3.2.1 Présentation

L'intelligence en essaim est une sous-catégorie de l'intelligence artificiellequi est motivée par le comportement intelligent des groupes dans les systèmes écologiques naturels des insectes sociaux comme les abeilles, les fourmis, les guêpes et les termites.

Le domaine de l'informatique bio-inspirée devient de plus en plus important. De plus en plus d'applications et de méthodes intelligentes sont explorées par les informaticiens pour différents contextes. Alors que certaines études exploitent l'application de ces algorithmes, d'autres études mettent en évidence l'amélioration des algorithmes.

Dans cette partie, nous décrivonsbrièvement ces méthodes et leurs portées. Nous soulignons dans quel contexte ces méthodes sont utilisées. Cela ouvre la voie ultérieurement à des études menées à l'avenir pour choisir un algorithme approprié.

3.2.2 Les méthodes bio-inspirées

3.2.2.1 Les réseaux de neurones

Les réseaux neuronaux sont souvent définis comme des algorithmes de traitement de données non linéaires adaptatifs combinant plusieurs unités de traitement connectées sur un réseau en différentes couches. Ces réseaux se caractérisent par une auto-adaptation, une auto-organisation et un potentiel d'apprentissage basé sur l'entrainement et les retours d'expérience de l'environnement dans lequel ils opèrent [52].

Ces réseaux de neurones tentent de reproduire la façon dont les neurones de tout organisme sont codés pour prendre en compte les entrées et les opérations dans une boîte noire. La différence entre les résultats obtenus et les résultats souhaités est renvoyée en retour pour améliorer les systèmes de traitement [53]. En outre, un tel réseau pourrait également avoir plusieurs entrées et plusieurs couches. Dans [54]les auteurs montrent comment former les réseaux en fonction de la disponibilité des données. Cependant, les réseaux de neurones peuvent être combinés avec d'autres algorithmes et méthodes, en fonction des besoins du problème, afin de fournir de meilleures capacités de prédiction au système.

3.2.2.2 Les algorithmesgénétiques (AG)

Les algorithmes génétiques ont été introduits pour s'adapter aux techniques puissantes de calcul pour obtenir une solution hautement appropriée. C'est une

métaheuristique de recherche évolutive qui imite le processus de sélection naturelle. Il est utilisé pour résoudre une variété de problèmes simples et multi-objectifs combinatoires.

Cet algorithme est basé sur une fonction de maximisation ou de minimisation et un mécanisme de codage. Initialement, une population des solutions du problème est générée aléatoirement. Une fois que les solutions sont triées sur la base de fonction objective, deux opérateurs génétiques (croisement et mutation) sont appliqués pour produire des nouvelles solutions. L'opérateur de croisement consiste à combiner deux solutions parents fournis par la sélection pour reproduire une solution enfant qui partagent de nombreuses caractéristiques de ses parents. Un opérateur de mutation est effectué sur certaines solutions (individus) en fonctions de la probabilité de mutation pour améliorer la solution candidate et éviter l'optimum local [55].

3.2.2.3 Ant Colony Optimization (ACO)

L'optimisation par colonies de fourmis est un algorithme de recherche heuristique permettant de résoudre des problèmes d'optimisation combinatoireNP-complet. Il est basée sur la communication indirecte d'agents simples (appelés ici fourmis) qui recherchent des informations, médiées par des traces de phéromone artificielles. Un groupe de fourmis est sélectionné et un pool de variables de décision est défini dans le problème. Les fourmis sélectionnent les variables de conception pour créer les solutions candidates. Au fur et à mesure que les fourmis explorent les solutions candidates, une mise à jour locale de la solution est effectuée en fonction de son optimalité. Ces solutions de type candidate sont utilisées pour modifier la valeur des traces en fonction de la mise à jour locale de manière à sélectionner les solutions de meilleure qualité dans la génération suivante. Le cycle est répété jusqu'à atteigne le nombre d'itérations spécifié par l'utilisateur. On peut noter que les solutions candidates créées dans la phase initiale ouvrent donc la voie à l'optimalité [56].

3.2.2.4 Optimisation par essaims particulaires

L'optimisation par essaims particulaires est un algorithme heuristique inspiré du comportement collectif des animaux tels que la formation des poissons, les essaims d'insectes ou la volée d'oiseaux. Le groupe tente d'atteindre l'objectif collectif du groupe en se basant sur les commentaires des autres membres. Il est utilisé pour les problèmes où la fonction à optimiser est discontinue, non différentiable avec trop de paramètres non linéairement liés[19].Cet algorithme opère dans une séquence de quelques étapes itératives. Chaque particule ou membre de l'essaim (par exemple un

oiseau ou un poisson) essaie de détecter une solution potentielle à tout moment. Il communique un signal proportionnel à l'adéquation de la solution candidate aux autres particules de l'essaim. Chaque particule ou membre de l'essaim peut donc détecter la force du signal communiqué par les autres membres, et donc l'adéquation de la solution candidate à une fonction de remise en forme. Lorsqu'une particule ou un membre tente de se concentrer sur une solution candidate plus appropriée parmi les solutions candidates disponibles localement, sur la base de différents mécanismes d'apprentissage [30], une nouvelle direction de mouvement et une influence inertielle les particules vers une solution optimale partout où cela est possible [57].

3.2.2.5 Algorithme de colonie d'abeilles

L'algorithme de colonie d'abeilles est un algorithme d'optimisation basé sur une métaheuristique qui recherche une solution numérique optimale parmi un grand nombre d'alternatives tout en essayant de résoudre des problèmes NP-difficileDans cet algorithme, les vecteurs de la population sont d'abord initialisés comme source potentielle de nourriture. Les abeilles employées recherchent initialement de nouvelles sources de nourriture avec un stimulus aléatoire. Une fois qu'une source de nourriture est identifiée (une solution candidate), son aptitude est identifiée et calculée. Dans la phase suivante, si une nouvelle source de nourriture est découverte par la suite (une nouvelle solution candidate) par des "abeilles employées" ayant une meilleure aptitude, la nouvelle source est adoptée, sinon la nouvelle est rejetée [58].Le cycle est répété jusqu'à ce que le nombre de génération atteigne le maximum.

3.2.3 Classification des méthodes bio-inspirées

D'après la littérature et les travaux de recherche déjà développé par ces méthodes on peut classer les méthodes par les domaines où ils ont utilisé dans le tableau suivant :

Algorithmes / Domaine	Réseaux neuronaux	AG	ACO	Essaims particulaires	Colonie d'abeilles
Normalisation des données	Oui				
Problèmes de détection	Oui	Oui	Oui	Oui	Oui
Problèmes de sélection		Oui	Oui		Oui
Problèmes d'optimisation		Oui	Oui	Oui	Oui
Problèmes de tri		Oui			
Problèmes d'optimisation de valeur numérique à un seul objectif			Oui		Oui
Problèmes d'optimisation de variables mixtes			Oui		
Problèmes de prise de décision multi-objectifs		Oui	Oui	Oui	Oui
Problèmes de classification	Oui	Oui			
Problèmes d'optimisation continue			Oui		
Problème de recherche et d'optimisation (continue)			Oui		
Problèmes d'optimisation déterministes				Oui	
Problème d'affectation				Oui	Oui
Problème de routage déterministe					Oui

Tableau 3-1travaux de recherche développé par les méthodes bio-inspirées

Nous concluons, qu'à part la méthode des réseaux de neurones, les autres méthodes sont utilisables dans différents problèmes d'optimisation. Mais au vu de ce tableau et des critères imposé par notre problème, notre choix se porte sur l'algorithme de colonies de fourmis, pour ce qu'il apporte en matière :

- L'architecture des réseaux n'est pas stable.

- Approches multiobjectifs est indispensable, puisque souvent deux ou plusieurs objectifs contradictoires doivent être considérés simultanément pour la sélection de service.

- Enfin l'évolution du système n'influe pas sur la méthode.

3.2.4 La métaheuristique ACO (Ant Colony Optimization)

3.2.4.1 Présentation de la métaheuristique ACO

La métaheuristique ACO (Ant Colony Optimization) est une famille de métaheuristiques d'optimisation a été inspirée, essentiellement, par le comportement social des fourmis réelles. Initialement développé par Dorigo *et al.*[59] pour la recherche de chemins optimaux dans un graphe. Les algorithmes de colonie de fourmis sont appliqués pour résoudre collectivement des problèmes complexes[60]. On peut citer :

1. les problèmes de recherche de plus court chemin entre deux points dans lesquels les propriétés des problèmes sont dynamiques (réseaux de capteurs sans-fil),

2. les problèmes NP-complet, les algorithmes ACO sont hybridés avec un autre optimiseur local qui prend les solutions trouvées par les fourmis à un optimum local,

3. lesproblèmes d'optimisation combinatoires. Un problème d'optimisation combinatoire est défini par plusieurs variables, une fonction objectif et un ensemble des contraintes, dont la résolution se heurte à une explosion du nombre de combinaisons à explorer.

3.2.4.2 Principe de fonctionnement

Dans la vie réelle, il existe plusieurs chemins, avec certains plus courts que d'autres. Imaginons une colonie des fourmis vit dans un terrain ou il y a unnid et une source de nourriture. Les fourmis exploreront ce terrain à la recherche de la nourriture. Le but est de déterminer le plus court chemin à partir de nid jusqu'à la source de nourriture via l'interaction des fourmis. Initialement, les fourmis choisissent les différents chemins aléatoirement avec la même probabilité (Figure 3-1(a)). Les fourmis sont en train de suivre une piste de phéromones, à un moment donné, on a un obstacle qui barre la route des fourmis (Figure 3-1(b)). Les fourmis qui arrivent à côté de l'obstacle doivent choisir soit d'aller en haut soit d'aller à bas (Figure 3-1(c)).

Puisque aucune phéromone n'est déposée le long de l'obstacle, il y a autant de fourmis qui partent sur le chemin en haut que le chemin en bas. Néanmoins, puisque le chemin haut est plus court que celui de bas, les fourmis du haut arrivent plus vite à la source. Elles commencent ses retours avant les fourmis en bas. Le nombre de fourmis qui passe en haut va augmenter, ce qui augmentera encore la concentration de phéromones. De plus, l'évaporation des phéromones sera plus forte sur la piste de haut du fait que sa longueur est supérieure. La piste en bas sera dont rapidement abandonnée, parce qu'elle en est beaucoup moins imprégnée : les fourmis passeront toutes très rapidement par la piste la plus courte (Figure 3-1)[61].

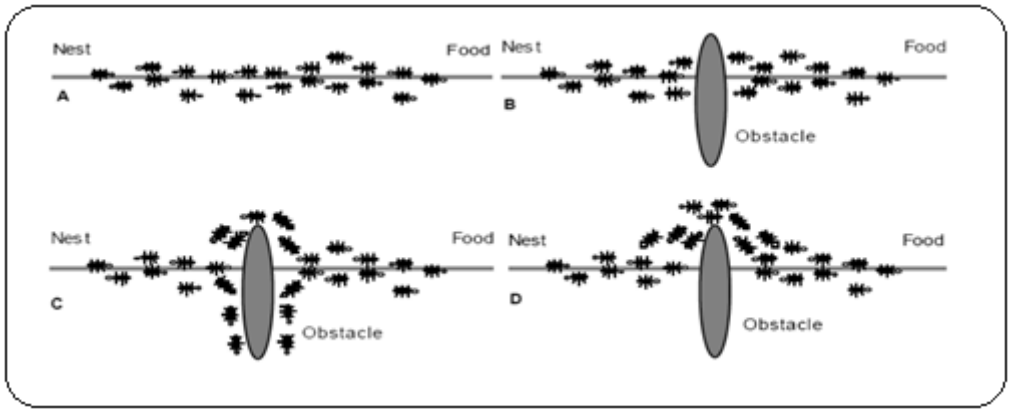


Figure 3-1Effet de la coupure d'une piste de phéromone.

3.2.4.3 Les variantes de l'algorithme ACO

Les variantes de l'algorithme original ACO sont :

- algorithme ANT system (Système formique) (1991) - Dorigo et al.[62].
- algorithme Elitist Ant System (1992) - Dorigo et al. [63].
- algorithme ANT-Q (1995) - Gambardella et Dorigo.[64].
- algorithme Ant Colony System (1997) - Stützle et Hose [65].
- algorithme Rank Based Ant System (1999) - Bullnheimer et al.[66].
- algorithme Max-Min Ant System (2000) - Stützle et Hose[67].
- algorithme Approximate Nondeterministic Tree Search (2000) – Maniezzo et Carbonaro[68].
- algorithme Hyper-Cube Framework (2001) - Blum et al.[69].

Dans ce qui suit on va présenter le premier algorithme de colonies de fourmis : ANT System.

3.2.4.4 Algorithme ANT System pour le voyageur de commerce

Le premier algorithme de colonies de fourmis est appelé le Ant System (Système formique)[62]. Il vise notamment à résoudre le problème du voyageur de commerce (Traveling Salesman Problem (TSP)), où le but estde trouver le plus court cycle hamiltonien permettent de relier un ensemble de villes.

L'algorithme repose sur un ensemble de fourmis, chacune parcourant un trajet parmi ceux possibles. À chaque étape, la fourmi choisit de passer d'une ville à une autre en fonction de quelques règles : (1) - elle ne peut visiter qu'une fois chaque ville, (2) - plus une ville est loin, elle a moins de chances d'être choisie, (3) - plus l'intensité de phéromone disposée sur l'arête entre deux villes est grande, plus le trajet aura de chance d'être choisi, (4) - une fois son trajet terminé, la fourmi dépose, sur l'ensemble des arêtes parcourues, plus de phéromones si le trajet est court et (5) - les pistes de phéromones s'évaporent à chaque itération.

Le fonctionnement de l'algorithme d'optimisation ANT System est décrit comme suit :

Etape 1 - Initialisation

Dans cette étape, la population initiale est générée aléatoirement. Elle est constituée de K fourmis. La quantité de la phéromone est initialisée avec une valeur constante positive non nulle(τ_{ij} =c) si l'arête e(i,j) ϵ E.

Etape 2 – Cycle (Itération)

Etape 2.1 – Déplacement des fourmis

Une fois que les fourmis sont créées, une répartition des k fourmis sur les n villes (nœuds) est effectuée. L'affectation de chaque fourmi sa ville de départ se définit via la liste taboue. Notant que la liste taboue a pour but de sauvegarder les villes visitées par une fourmi. Les fourmis se déplaceront ensuite en choisissant leur destination suivant la probabilité définie par l'équation suivante :

$$\begin{cases} P_{ij}^{K}(t) = \frac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{(j \epsilon N_i^K)} [\tau_{ij}]^{\alpha} * [\eta_{ij}]^{\beta}} \ / j \epsilon N_j^k \\ 0 \ ailleurs \end{cases} \quad (1)$$

- N_i^K : L'ensemble des voisins de i non encore visités par la fourmi k ;
- $\tau_{ij}(t)$: La quantité de phéromone déposée sur l'arête (i, j).
- η_{ij}: L'information heuristique, appelée visibilité, représentant l'inverse de la distance entre deux villes i et j, elle est utilisée pour diriger le choix des fourmis vers des villes proches et éviter les villes trop lointaines.

- α, β : deux paramètres qui contrôlent respectivement l'importance de l'intensité de la phéromone et la distance entre les sommets i et j. avec :
    - α=0, seule la distance entre les nœuds est prise en compte ; la ville la plus proche donc choisie à chaque pas.
    - Au contraire β=0, seule l'intensité de la phéromone est considérée. D'où, une convergence rapide vers une solution qui n'est peut-être pas optimale.

De ce fait, un compromis entre ces deux paramètres α et β, jouant sur les comportements de l'exploitation et l'exploration, s'avère nécessaire.

Etape 2.2 –Mise à jour de la phéromone

Dans cette étape, chaque fourmi après n déplacements, calcule la longueur **L** de son tour et refait son tour en sens inverse tout en déposant des phéromones selon l'équation suivante :

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\,\tau_{ij}^{k}(t) \quad (2)$$

Avec $\Delta\,\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^{K}(t)} \; si \; (i,j) \in T^{K}(t) \\ 0 \; sinon \end{cases}$

- $T^{K}(t)$: Le chemin parcouru par la fourmi k à l'itération t
- $L^{K}(t)$: La longueur du chemin parcouru par la fourmi k à l'itération t
- Q : Un paramètre fixe relatif à la mise à jour de phéromone

Etape 3 – Fin cycle

Etape 3.1 – L'évaporation de la phéromone

Afin d'empêcher une augmentation continue d'une seule solution et de favoriser l'exploration d'autres trajets peut être meilleurs que ceux déjà explorés, après chaque itération, l'évaporation de la phéromone est effectuée selon l'équation suivante :

$$(t+1) = (1-\rho)\tau_{ij}(t) \quad (3)$$

ρ : Le coefficient d'altération de la phéromone.

Etape 3.2 – Sélection de la meilleure solution

Les solutions sont triée sur la base du principe distance (plus court chemin) et le meilleur tour de ce cycle est mémoriser s'il est le meilleur que celle de cycle précédent.

Etape 4 – Le critère d'arrêt

Le cycle est répété jusqu'à ce que le nombre de visites atteigne un certain nombre d'itérations défini par l'utilisateur.

3.2.4.5 Avantages et limites de l'algorithme ACO

D'après l'étude en peut dire que l'algorithme de colonie de fourmis est efficace pour les problèmes d'optimisation, surtout ceux de type voyageur de commerce ou problèmes similaires. La résolution des problèmes peut s'effectuer en temps réel, donc en cas d'imprévu il n'y a besoin de refaire tout. De plus l'indépendance entre les fourmis fournit un aspect d'implémentation parallèle très forte. La rétroaction positive permet de découvrir rapidement de bonnes solutions, enfin il est adéquat aux applications dynamiques (il s'adapte aux changements de contexte tels que les nouvelles distances, nouvelle ville, etc.). Par contre l'analyse et la projection des problèmes réels sur la méthode est difficile, ainsi que le principe de décisions aléatoire de trajet des fourmis difficile à implémenter. De plus la découverte de nouveau chemin en cas de changement de contexte peut prendre un nombre important d'itérations. Enfin puisque la recherche est expérimentale plutôt que théorique, le temps de convergence est incertain mais la convergence est garantie.

3.3 Ontologies et web service sémantique

Àcause de l'émergence d'une quantité d'informations très importantes sur le net, aujourd'hui l'un des défis informatiques les plus répandus consiste à fournir *«la bonne information à la bonne personne au bon moment »*. Pour atteindre cet objectif, il faut des associations transparentes entre les personnes, les agents logiciels et différents types de systèmes informatiques. Ces connexions sont nécessaires pour soutenir l'émergence de communautés dynamiques qui peuvent échanger et utiliser efficacement la gamme complète des données, des informations, et des connaissances. L'apparition du Web sémantique pour compléter le manque et aidé à réaliser ce défi.

Le Web sémantique est une extension évolutive du World Wide Web (W3) actuel dans lequel les contenus Web peuvent être exprimés dans un format lisible, compris et utilisé par les agents logiciels, afin que les agents logiciels puissent facilement trouver, partager et intégrer plus d'informations d'une manière très efficace[38].

Le Web actuel est une source énorme et croissante d'informations et de services. Ces informations et services doivent être partagés non seulement par des personnes mais aussi par des applications. Dans la pile de l'architecture Web en couches sémantiques, l'ontologie est une structure hiérarchique des concepts les plus importants liés à un domaine particulier.

L'ontologie joue un rôle important dans l'échange d'informations et le partage en élargissant interopérabilité syntaxique du Web à l'interopérabilité sémantique.

Dans la partie qui suit, on présente ces deux notions d'interopérabilité et leurs impacts sur notre problématique

3.3.1 Présentation

La plupart du contenu Web actuel est conçu uniquement pour les êtres humains. Les informations sur le Web sont semi-structurées et ne définissent que la manière dont le contenu doit être affiché dans un navigateur Web.

Le langage HTML (Hyper Text Markup Language) est le langage dominant pour la rédaction des pages Web. HTML ne donne pas d'information sur le sujet et la nature du contenu Web. C'est pourquoi il est difficile pour les programmes informatiques de manipuler ces informations de manière automatique.Pour cette raison, il devrait exister des moyens standards d'exprimer ces informations afin que les agents logiciels puissent mieux interagir et travailler sans aucune intervention humaine.

Le Web sémantique englobe un ensemble de principes de conception, de groupes de travail collaboratifs et de diverses technologies. Certains éléments du Web sémantique doivent encore être mis en œuvre, mais d'autres éléments du Web sémantique sont exprimés dans des spécifications formelles. Le terme a été inventé par Tim Berners-Lee, inventeur du Web et directeur du World Wide Web Consortium (W3C) qui initialement exprimait sa vision du Web sémantique comme suit :

«I have a dream for the web, in which computers, become capable of analyzing all the data on the web – the contents, links and transactions between people and computers»

Cette idée a poussé le Web basé sur le langage de balisage simple (HTML) vers un langage sémantique basé sur des langages de balisage plus expressifs (RDF, OWL), un Web significatif. Le tableau suivant montre l'évolution du contenu du web dans le temps.

Temps	1995----------->2005			
Type	Statique	Dynamique	Syntaxe	Sémantique
Codage	HTML	RDBMS	XML	RDF/OWL
Utilisateur	Homme	Homme	Homme et application	Homme et application
Applicatior	Browsers	Browsers	Process Integration, EAI, Workflows	Intelligent agents, Semantic Engines

Tableau 3-2L'évolution du contenu du web dans le temps.

3.3.2 Définition

L'ontologie est un vocabulaire agréer qui fournit un ensemble de concepts bien fondés pour construire des connaissances significatives de haut niveau permettant de définir la terminologie sémantique du système de manière bien définie et sans ambiguïté. Il possède un ensemble de concepts bien définis pouvant être utilisés pour construire des connaissances structurées. Bien que une taxonomie améliore la sémantique des termes dans un* vocabulaire, l'ontologie inclut une relation plus riche entre les termes [70].Une caractéristique importante de l'ontologie est la sémantique formelle du monde réel et l'interopérabilité de la machine. En raison de cette propriété importante, l'ontologie permet de partager et de réutiliser les ontologies chez les humains et les machines[19]. Selon Sean Luke et Jeff Heflin, «la représentation explicite de la

sémantique sous-jacente aux données, programmes, pages et autres ressources Web permettra un Web basé sur la connaissance qui fournira un niveau de service qualitativement nouveau » [71].

En résumé, l'ontologie présente une représentation explicite de la sémantique. Grâce à la combinaison des ontologies et du Web, il est possible de gérer de nombreuses questions liées au partage des connaissances, à la réutilisation et à l'intégration de l'information [72].Une autre caractéristique importante desontologies est qu'elles peuvent soutenir le raisonnement. Par exemple, en utilisant l'ontologie, on peut affirmer que si A est un ancêtre de B et que B est un ancêtre de C, on peut en déduire que A est un ancêtre de C.En utilisant cette capacité de raisonnement de l'ontologie, nous pouvons rendre un certain nombre d'applications capables de gérer des informations complexes. Qui donne une vraie vision sur le Web sémantique.

3.3.3 Types d'ontologies

On identifie trois types d'ontologies selon un niveau décroissant d'abstraction :

- **ontologies génériques** (Top-Level Ontology) : présentent un plus haut niveau d'abstraction et de généralité. Les ontologies sont un formalisme formel, car elles issues d'un développement systématique, rigoureux et axiomatique de la logique de toutes les formes et modes d'existence. L'adoption de principes rigoureux dans la conception de l'ontologie formelle répond au besoin de disposer de connaissances pouvant être partagées et transférées d'un contexte à l'autre. Elles sont dédiées à des utilisations générales (ex : WordNet). Une ontologie formelle est donc une théorie des distinctions formelles entre les éléments d'un domaine, indépendamment de leur réalité[73].

- **ontologies de domaine :** ou dédiées à une tâche plus spécifique : limitée à la représentation de concepts dans des domaines donnés (géographie, médecine, écologie, etc.) et qui spécialise les concepts de l'ontologie globale.Elle permet d'exprimer des conceptualisations spécifiques à un domaine. Le domaine est décrit au travers de concepts spécifiques au domaine et des relations entre eux qui modélisent les principales activités de base du domaine en question.

- **ontologies d'applications** : offrent le plus fin niveau de spécificité, c-à-d qu'elles sont dédiées à un champ d'application précis à l'intérieur d'un domaine et décrivent le rôle particulier des entités de l'ontologie de domaine

dans ce champ. Par exemple, l'ensemble des spécifications sur la forêt de Montmorency constitue une ontologie d'application qui spécifie les concepts généraux pouvant provenir d'une ontologie de domaine [73].

3.3.4 Utilisation des ontologies dans les services Cloud

Actuellement, dans le domaine des services Cloud, les ontologies sont utilisées pour représenter la sémantique des données et former un agrément sur un vocabulaire commun. Pour permettre l'automatisation des diverses tâches liées aux services Cloud, l'idée consiste à enrichir les descriptions des services Cloud (*limitées qu'aux aspects fonctionnels*) par d'autres informations supplémentaires compréhensibles par machines. Ces informations permettent d'interpréter les descriptions des services Cloud. En d'autres termes, c'est la sémantique des descriptions des services Cloud, puisque deux services Cloud peuvent avoir la même description syntaxique mais avoir deux sens différents. Les services Cloud seront annotés à base des ontologies de domaine du travail pour qu'un agent logiciel puisse avoir connaissance de la sémantique d'une description d'un service Cloud, il lui suffit juste d'accéder à une ontologie du domaine. La combinaison des technologies du Web sémantique à ceux Services Cloud permetd'automatiser la découverte de services Cloud, c'est à dire localisation automatique des services Cloud qui fournissent une fonctionnalité particulière et qui répondent aux propriétés demandées par l'utilisateur. Pour pouvoir effectuer une découverte automatique, le procédé de découverte doit être basé sur la similitude sémantique entre la description déclarative, faite par l'utilisateur, du service demandé et celle du service offert.

3.3.5 Les langages de description des ontologies

3.3.5.1 RDF et RDF Schéma

Selon la définition du W3C, RDF est un modèle de données formel spécialement conçu pour les métadonnées compréhensibles par la machine; utilisé pour offrir des descriptions standard des ressources Web. En d'autres termes, RDF est un langage de métadonnées à usage général pour représenter des informations sur le Web. Ce modèle est utilisé pour décrire et créer des relations entre les ressources.Une ressource peut être une personne, un objet ou une page Web.La structure RDF utilise des triplets RDF sous la forme de sujet (sujet identifié par une URL), un prédicat (le type de métadonnées, également identifié par une URL, également appelée Property), Object (la valeur de ce type de métadonnées).

3.3.5.2 OWL (Web Ontology Language)

OWL est basé sur le langage d'intégration ontologique (Darpa Agent Markup Language). OWL décrit les propriétés et les classes en utilisant plus de vocabulaires que le langage RDF, car il s'agit d'une extension de RDF. Il est spécialement conçu pour le Web sémantique, où le contenu Web doit être traité par des agents logiciels. L'ontologie OWL inclut des descriptions de classes avec leurs différents types de propriétés et d'instances. Il décrit également la structure hiérarchique des classes. OWL est conçu pour les applications puisqu'ilestnécessaire pour le traitement du contenu des informations par les agents logiciels. Il prend une meilleure interopérabilité entre la machine et le contenu Web, par rapport à XML, RDF et RDFS, car il offre un vocabulaire supplémentaire de type sémantiqueet formel[74].

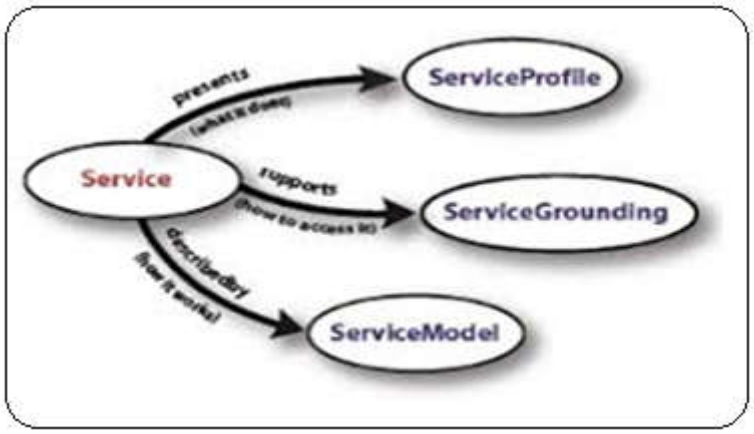


Figure 3-2Description de l'ontologie OWL-S

3.3.6 Bilan

Après cette étude sur les ontologies, nous pouvons conclure que la notion d'ontologie représente une approche très efficace dans le domaine de la représentation des connaissances. Elles apparaissent désormais comme une clé pour la représentation des services dans notre étude. La manipulation automatique des informations sur les services au niveau sémantique, donne un avantage à la sélection du bon service

3.4 DSGreenACO : Approche Dynamique Sémantique Bio-inspiré Green Computing guidé par une ontologie

Notre approche consiste à utiliserles principes de l'algorithme de colonie de fourmis (ACO) et les principes de filtrage sémantiques des services guidé par une ontologie baptisée DSGreenACO (Dynamic Semantic Green Ant Colony Optimization).Ils sont agnostiques au domaine applicatif et que selon le domaine (*tourisme, santé, transport, etc.*), les besoins diffèrent. Un nombre important des services Cloud hétérogènes issus

de différents fournisseurs Cloud avec différentes contraintes d'exécution (énergie, RAM, CPU, bande passante) rend la gestion de la qualité de service très difficile. Cette hétérogénéité qui implique une dégradation de qualité de contrôle et de sélection des services. Un filtrage sémantique permettrait d'améliorer durant l'étape de recherche des services sémantiquement incompatibles avec ceux qui ont été déjà sélectionnés.

En parallèle, nous proposons des informations heuristiques (débit inter-centre de donnée, charge appliquée sur les centres de données, nombre de requêtes clients sur les services hébergés) pour guider les fourmis dans leur découverte des services souhaités localisés dans les centres de données de l'espace de recherche précisés par le client puis filtrer à l'aide de règles appliquées à l'ontologie afin d'obtenir le chemin optimal qui minimise la consommation de l'énergie et satisfera les besoins du client. Cet algorithme est ainsi utilisé pour guider les futurs choix et indiquer aux autres fourmis des directions à suivre ou au contraire à éviter en partageant le savoir commun des fourmis. Notre solution consiste ainsi à proposer une nouvelle approche pour l'optimisation d'énergie dans les centres de données tout en tenant compte des besoins et les préférences des clients.Cette approche (voir Figure 3-3) est une hybridation entre l'utilisation de l'ontologie pour le filtrage sémantique des services et des métaheuristiques d'optimisation par colonie de fourmis pour l'optimisation de l'énergie pendant la découverte des services (Green Computing).

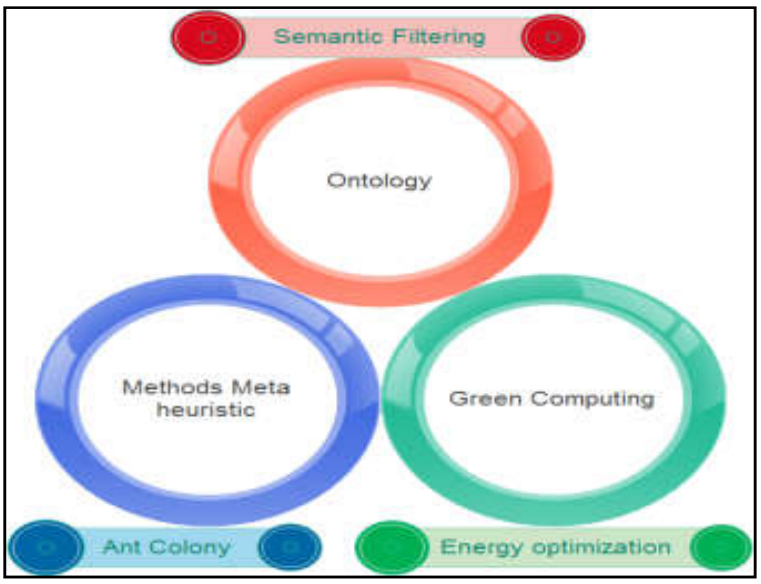


Figure 3-3 Approche proposée.

3.4.1 Motivations

La motivation principale de ce travail est de proposer une approche intelligente pour optimiser les qualités énergétiques et les services Cloud. L'approche proposée doit

prendre en charge tous les langages requis par les clients pour identifier le bon fournisseur de services qui pourrait répondre à leurs exigences fonctionnelles, qualitatives et contextuelles. En utilisant notre approche, nous sommes en mesure de mesurer la qualité de service des fournisseurs de services dans le but d'identifier les services de Cloud Computing les plus appropriés pour le client tout en minimisant l'effort énergétique. L'approche proposée est implémentée par une ontologie contextuelle de service de Cloud Computing générique qui implémente le concept de Green Computing et ACO.

3.4.1.1 La nécessité et l'importance de la méthode ACO

Pour notre travail, nous avons retenu la méthode des colonies de fourmis qui requiert de plus en plus l'attention de la communauté scientifique. Parce qu'elle est extrêmement performante dans de nombreux domaines.L'algorithme ACO est généralement utilisé dans WSN (Wireless Sensor Network) pouraméliorer la découverte des services, l'efficacité énergétique et la réduction de la consommation d'énergie. Ceci est le résultat du comportement de l'ACO qui est capablepour construire un chemin de découverte optimal du nœud source versnœud de destination qui réduire la consommation d'énergie et les délais de réponse. L'algorithme ACO utile pour :

- **réduction de la consommation d'énergie :** Après quelques recherches sur la localisation des services Cloud qui réponds aux besoins des utilisateurs, on a trouvé que la découverte des services dans tous les chemins possibles consomme plus d'énergie qu'une découverte via des stratégies de sélection des chemins pertinents, ainsi, plus le graphe des centres de données est important plus d'énergie peut être consommé à cause des problèmes intervenant de déplacement des fourmis. Pour cela, on a exploité la concurrence des fourmis avec des politiques de sélection des meilleurs chemins pour réduire la consommation d'énergie. Cet algorithme sera enrichi dans notre travail avec l'heuristique du débit inter-donné et l'heuristique de la charge du centre de données.

- **efficacité d'exploration totale de l'espace de recherche :** ACO est très efficace lorsqu'il s'agit d'exploiter une zone de l'espace de recherche. Puisque, la concurrence des fourmis donne plus de chance de découvrir des meilleurs servicesqui répondent aux préférences de l'utilisateur et aux contextes d'utilisation. En effet, elles permettent : (1) - la réduction de trafic dans le

réseau, (2) la conservation d'énergie, (3) la répartition de la consommation d'énergie et (4) l'exploitation totale de l'espace de recherche.

- **auto-adaptation aux problèmes statiques et dynamiques :** ACO est facilement adapté pour résoudre les problèmes d'optimisation combinatoire des réseaux ubiquitaires dynamiques [75] et les environnements mobiles. Les problèmes qui peuvent être résolus en utilisant ACO sont la réduction de l'énergie et l'équilibrage de charge. L'algorithme ACO est souvent utilisé pour déterminer les itinéraires optimaux de la source à la destination à travers une approche métaheuristiques. En plus de déterminer les chemins optimaux, le problème de découverte des services Cloud sont affectés au même centre de données qui conduisent au nœud ayant une charge de travail élevée peut éventuellement être résolu en utilisant la technique d'équilibrage de charge. C'est parce que les chemins optimaux peuvent changer éventuellement en utilisant des mises à jour globales et / ou locales de phéromones. Cependant, la performance de l'algorithme ACO peut encore être plus loin étendu afin d'obtenir un débit maximal, délai minimum, consommation d'énergie minimale du centre de données pour équilibrer la charge de l'ensemble des centres des données et en même temps d'étendre la durée de vie du réseau Cloud.

- **qualité d'expérience** : les historiques des colonies pour ses déplacements, pour ses découvertes des services Cloud et sélectionne (le ou) les services pertinents.

3.4.1.2 La nécessité et l'importance des ontologies

L'utilisation de l'ontologie du point de vue technique permet d'assurer :

- **Homogénéité :**différentes informations contextuelles de différentes domaines smart-* (tourisme, transport, santé, etc.) et différentes qualitésde services utilisantla même ontologie,confèreà l'ontologie un rôled'homogénéité au moins pour les informations de qualitéde serviceetles informations de contextequ'elledécrit.

- **Interopérabilité :** les ontologiesfournissentun bon moyenpourtraiter l'interopérabilité sémantiqueentre les représentationshétérogènesdeserviceCloud comme les paramètres de QoS et les paramètres de contexte par une représentationsémantique commune.

- **Expressivité :** la plupart des langages d'ontologie tels que RDFetOWL-S sont efficacesavec une expressivitéélevéeetbeaucoup mieuxpourexprimerla sémantiquedesdiverses dispositifs portables (Smart phone, laptop, Tablet et divers paramètres de QoS.

- **Facilité de découverte et de sélection :** les ontologiesrendent la découverteet lasélectionde la qualité des servicesCloud plus facile, plus rapideetautomatique. Le principal mécanisme pour ladécouvertede serviceest le *service Registry* et la sémantiquepeut être utilisée pour la gestion des paramètres de QoS et les paramètres de contexte.

3.4.2 Objectives

Notre approche permet de fournir des services Cloud pertinents afin de répondre aux besoins des utilisateurs et aux préférences de qualité de service et de réduire la consommation énergétique d'un nombre large des services Cloud grâce à une sélection sémantique dynamique de services Cloud.L'objectif principal de notre approche est d'avoir une méthode dynamique guidée par les ontologies qui implémente la notion du Green Computing afin d'obtenir le chemin optimal (liste des centres de données et ses services) qui minimise la consommation de l'énergie et satisfait les besoins du client tout en permettant la découverte des services souhaités hébergés dans les centres de données.

Dans ce qui suit nous allons décrire l'architecture générale de notre système, le modèle fonctionnel de notre approche.

3.4.3 Architecture et fonctionnement de DSGreenACO

3.4.3.1 L'architecture générale de DSGreenACO

L'architecture générale de la découverte et de la sélection dynamique et sémantique des services Cloud nécessite trois composants (voirFigure 3-4) :

- **Client :** est l'appareil que l'utilisateur utilise tel qu'un ordinateur, un ordinateur portable, un smartphone ou une tablette par exemple.

- **Service SémantiqueProxy-Cloud** : se compose d'un composant analyseur sémantique du profil client enregistré en RDF/XML, d'un composant **DSGreenACO** (*Dynamique Semantic-based Green Computing ACO*) et d'un composant de gestion sémantique du contexte.

- **Composant d'analyseur sémantique du profil :**ildoit être indépendant de n'importe quelle technologie et de n'importe quel langage particulier afin de supporter toutes les contraintes utilisateurs et ses préférences (*format SPARQL,* String, etc.). Le composant analyseur sémantique du profil fournit une interface nécessaire pour traduire et enrichir les contraintes du profil par des services sémantiques utilisant l'ontologie spécifique au domaine.

- **Composant DSGreenACO** : permet de découvrir sémantiquement des services Cloud sur un graphe de centres de données (i.e. défini par des nœuds et des arcs, permettent de représenter le réseau des centres de données)pertinents basés sur les historiques des colonies pour ses mouvements compétitifs et sur notre ontologie qui est capable de garantir les besoins fonctionnels et non-fonctionnels des clients. Le composant DSGreenACO recherche le meilleur service en fonction du poids de chaque QoS. Une QoS définie comme un ensemble de paramètres de métadonnées. Elle peut s'exprimer de trois manières différentes : Valeur sémantique (*Faible, Moyen, Haut*), Valeur numérique/booléenne ou Intervalle [*Min, Max*]. Pour trouver les solutions de qualité, l'algorithme d'optimisation d'ACO, et les correspondances de violations de contraintes et des préférences sémantiques des clients doivent être introduits.

- **Composant de gestion sémantique du contexte** : le composant de gestion sémantique évalue dynamiquement le profil utilisateur (*préférences utilisateur et contexte d'utilisation*) et le contexte du service Cloud (*connexion réseau, taux de charge des serveurs Cloud, fournisseur de services Cloud disponible, etc.*) dynamiquement pour sélectionner de meilleurs services de Cloud à l'utilisateur.

- **Le registre des services Cloud and profils utilisateurs** : contient la description sémantique des services Cloud de tous les fournisseurs de services Cloud et la description des profils d'utilisateurs au sein de notre ontologie.

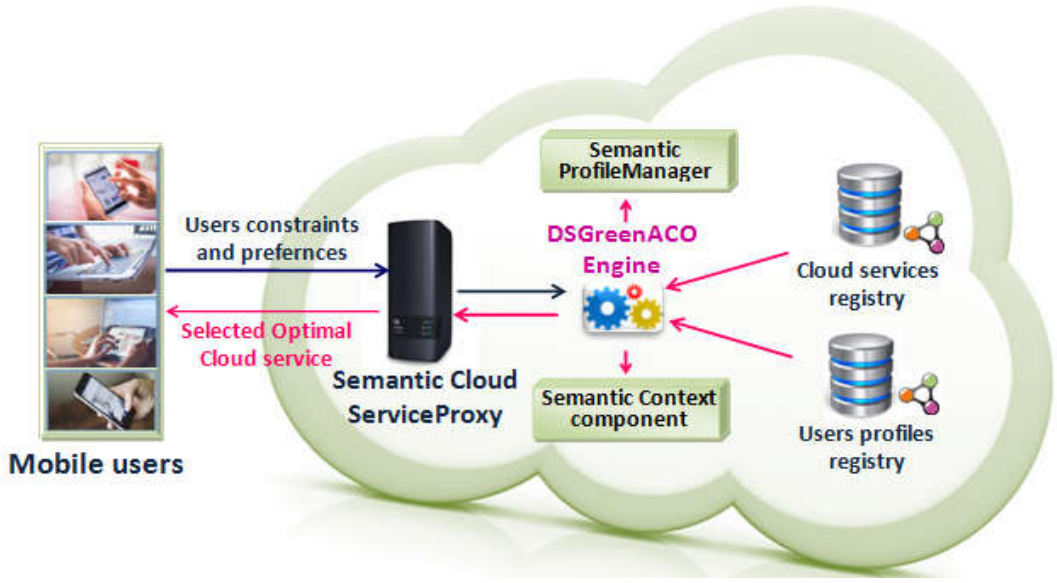


Figure 3-4L'architecture générale de notre approche.

3.4.3.2 Fonctionnement de DSGreenACO

Le modèle fonctionnel de notre approche est décrit dans laFigure 3-5. Lorsqu'un utilisateur configure sa session, il fournit plusieurs contraintes fonctionnelles (*réservation d'hôtel, localisation de transport et de destination sportive, bon service de santé, etc.*) et des contraintes qualité de service (*prix bas, temps de réponse rapide, bonne sécurité, etc.*). Un client exprime ses contraintes et ses préférences visuellement à travers une interface utilisateur, elle sera traduite vers une expression logique des attributs fonctionnels et attributs de QoS basé sur notre modèle d'ontologie. Certaines d'autres informations sont automatiquement récupérées depuis son appareil, d'autres sont spécifiques à l'utilisateur telles que l'âge, la langue préférée, etc. Ces informations sont stockées dans notre ontologie. Le profil utilisateur qui contient les informations personnelles et contextuelles de l'utilisateur et ses préférences enregistré sous forme RDF/XML est utilisé (*ou mis à jour*) lors de la recherche des services Cloud.

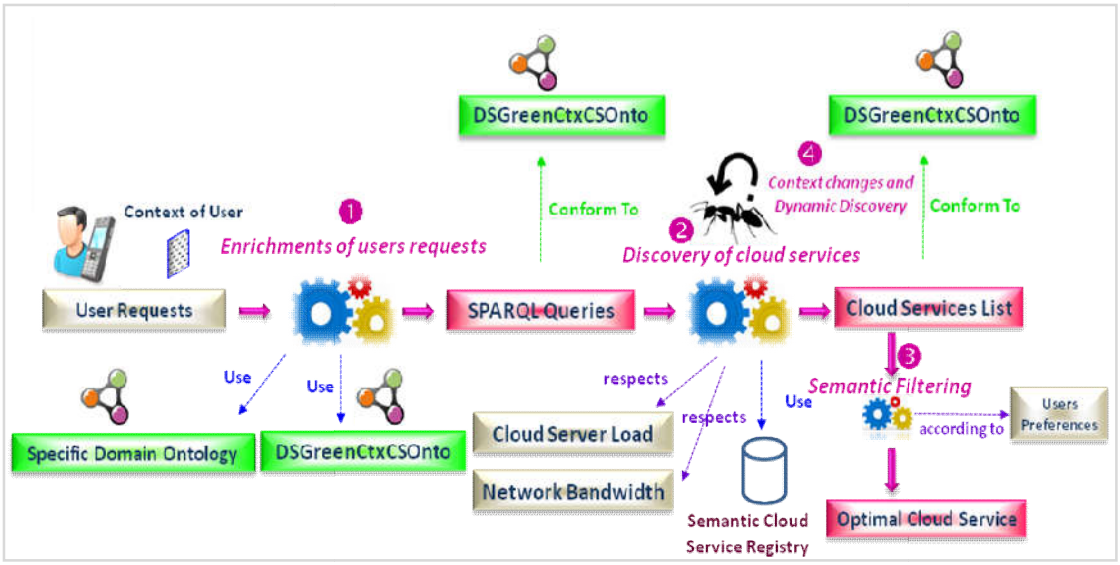


Figure 3-5 Modèle fonctionnel de l'approche DSGreenACO.

Le composant *Service SémantiqueProxy-Cloud* reçoit le profil, il fait appel au composant analyseur de profil sémantique pour analyser et enrichir sémantiquement le profil par d'autres services spécifiques en utilisant une ontologie générique et une ontologie de domaine spécifique sur la base des liens sémantiques explicite qui lient un concept générique (*i.e. catégorie de service*) de notre ontologie et les concepts spécifiques des ontologies de domaine. Ensuite, il lance le composant DSGreenACO pour découvrir d'éventuels services sémantiques disponibles sur un graphe de centres de données pertinents basés sur les historiques des colonies pour ses déplacements pour ses découvertes des services Cloud et sélectionne-le ou les services pertinents. Nous utilisons des critères incluant diverses politiques (*économie d'énergie, charge du processeur, bande passante faible*). Une fois que toute la liste de services de Cloud a été trouvée, elle est sémantiquement filtrée en fonction des préférences de l'utilisateur et les meilleurs services sont renvoyés àl'utilisateur.

3.4.4 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche "DSGreenACO" pour la gestion économique et efficace des services Cloud afin de sélectionner et composer des services Cloud pertinent de qualité et sensible au contexte selon les besoins et les préférences des clients. Cette approche est une hybridation entre l'utilisation de l'ontologie pour le filtrage sémantique des services et des métaheuristiques d'optimisation par colonie de fourmis pour l'optimisation de l'énergie pendant la découverte des services (Green Computing). Nous avons détaillé ensuite une nouvelle approche sémantique et dynamique d'optimisation de l'énergie et de QoS dans le Cloud pour les environnements intelligents.

Dans le chapitre suivant nous allons présenter une nouvelle ontologie qui permet d'utiliser les aspects sémantiques pour l'enrichissement des requêtes utilisateurs et pour le filtrage sémantique des services Cloud afin d'obtenir des services Cloud de qualité selon les attentes et les préférences des clients, optimal en termes de consommation d'énergie. Cette ontologie joue le rôle de noyau de notre approche proposée.

Chapitre 4

4 Description des services Cloud sensibles aux contextes à base d'ontologie

Sommaire

4.1 Introduction

Actuellement, les services distribués de différents fournisseurs augmentent de plus en plus en matière d'hétérogénéité. L'hétérogénéité des services est liée à la taille des données (*les flux vidéo sont difficilement gérables selon le type de connexion*), les plateformes d'exécutions, de la variété des mécanismes de communication, de la vitesse de transmission ainsi que de la variété des médias (audio, vidéo, texte et image). Cette hétérogénéité implique une sélection des services à un niveau sémantique, afin d'éviter les solutions ad-hoc qui ne sont pas réutilisables et/ou généralisés.

Dans le cadre de l'informatique ubiquitaire, l'accès à des services existant sur le Cloud est favorisé par l'internet des objets (Smart-Object) et les terminaux mobiles hétérogènes (Tablette, PC, Smartphone, SmartTV, etc.) avec de multiples technologies de découvertes et d'accès aux services appartenant à des utilisateurs avec des besoins différents. Cela pose des problèmes environnementaux importants au niveau de la consommation d'énergie comme le rejet des émissions de CO2. Le recours aux stratégies d'optimisation d'énergies fait partie des solutions privilégiées pour répondre à ces enjeux.

Ainsi, les besoins des utilisateurs sont exprimés par différents langages et différentes expressions selon leurs contextes. Des ontologies contextuelles basées sur la sémantique sont nécessaires pour mieux répondre à ces besoins. Les clients doivent localiser les fournisseurs disposant d'un service de meilleure qualité dans les limites de leur budget. La recherche de services pose plusieurs problèmes cruciaux dont celle de la gestion optimale de la QoS et de l'énergie. En termes de QoS, la recherche optimale reste encore un challenge ouvert en raison de la nature du problème (NP-difficile), ajoutons les changements aléatoires pouvant se produire dans de tels environnements, comme par exemple, l'apparition ou disparition de services, la dégradation de la qualité d'un service à cause de la mobilité des utilisateurs ou en raison de l'utilisation intensive de ce service.

Dans ce chapitre, nous présentons une nouvelle ontologie différente des ontologies proposées dans la littérature. En effet, nous proposons un système intelligent, économique en termes d'énergies pour la sélection et la composition flexible des services Cloud basés sur la sémantique des services Cloud. Le système facilite les annotations sur les qualités des services Cloud et les contraintes et préférences des clients en attribuant des métadonnées sémantiques aux propriétés de la qualité et du contexte. Dans ce système, un mécanisme d'interrogation est appliqué sur les services

Cloud annotés sémantiquement pour découvrir et sélectionner automatiquement des services Cloud pertinents et répondre aux besoins particuliers des différents clients.

4.2 Objectives

L'objectif principal est de construire un modèle de description sémantique des services Cloud sensibles aux contextes et les besoins fonctionnels et non fonctionnels des clients pour optimiser l'énergie et les critères de QoS lors de la découverte et de la sélection sémantique des services Cloud. Un modèle sémantique doit contenir l'état contextuel d'utilisation : environnement réseau, contexte des services et contexte des utilisateurs. Pas seulement les QoS des différents services doivent être compatibles, mais aussi les fournisseurs de services doivent respecter certaines propriétés et certaines contraintes. Les caractéristiques des ressources et la nature des contraintes rendent le partage etla gestion des QoS encore plus difficiles. Pour exploiter au mieux la qualité de service sensible au contexte, on a besoin d'une ontologie pour décrire leur sémantique.

L'implémentation de ce modèle d'ontologie utilise le langage OWL. Actuellement, les services Cloud sont décrits en utilisant le langage standard de service web sémantique (OWL-S) qui facilite la découverte et la sélection de services Cloud. Cependant, les informations contextuelles ne sont pas considérées par OWL-S. Ce qui implique les inconvénients suivants :

- une mauvaise qualité de service, conséquence de l'utilisation de la description syntaxique sans la description sémantique des informations du contexte et la sémantique des informations de la mobilité.
- difficulté d'aboutir à des services Cloud de qualité comme la fiabilité et l'efficacité.
- la gestion de la qualité des applications dans les domaines smart dépend de leurs matériels et plates-formes logicielles.

Ces inconvénients peuvent être remédiés, en étendant OWL-S pour correspondre aux exigences des utilisateurs avec les spécifications du fournisseur au niveau sémantique, et prendre en compte les informations de contexte avec les paramètres de la qualité de service (QoS) et l'énergie.

4.3 Ontologie de noyau

Dans cette section nous présentons notre conception de modèle de description sémantique des services Cloud et les besoins fonctionnels et non fonctionnels des

utilisateurs. L'utilisateur doit être capable de localiser une qualité de service à travers son sens, indépendant de comment la qualité de service est appelée, le langage utilisé, comment le contexte de service évolué, et qui est le fournisseur de services. Pas seulement les QoS des différents Cloud services doivent être compatibles, mais aussi les plateformes d'exécutions des services et les ressources Cloud. Les caractéristiques des ressources et la nature des contraintes rendent la gestion de ces QoS encore plus difficiles. Nous visons à mieux capter la sensibilisation des services aux ressources Cloud et à mieux contrôler la qualité des services au niveau sémantique, à fournir une meilleure sélection d'un service via ses propriétés sémantiques, à adapter et personnaliser un service aux préférences des utilisateurs, sans oublier la qualité énergétique du service.

Notre ontologie principale (core ontology) est décrite par la figure 5.1. Cette ontologie se concentre sur l'optimisation d'énergie des centres de données qui stockent tous les services Cloud qui supportant les activités des utilisateurs selon ses domaines d'applications (smart-* : Home, Health, Cities, Vehicles, etc.) via une classe générique « catégorie de service » et ses classes spécifiques.

Notre ontologie des services Cloud en informatique verte baptisée DSCxGreenSCloud (Dynamic Semantic ConteXt-aware GREEN Service Cloud Ontology). Elle est utilisée pour :

- permettre aux fournisseurs de services la publication des métadonnées au sujet de leurs services Cloud avec leurs contraintes de contexte dans le registre UDDI.

- décrire sémantiquement de tous types de service Cloud de tout fournisseur de services en se basant sur la compatibilité des QoS.

- séparer les services Cloud selon leur catégorie, leur QoS et leur contexte.

- séparer les requêtes fonctionnelles et non-fonctionnelles et contrôler et maintenir la qualité des services Cloud.

- enrichir des requêtes clients par l'application de notre ontologie générique et du modèle d'ontologie de domaine spécifique.

- filtrer sémantiquement des services Cloud selon les préférences QoS et le contexte courant du client et retourne une liste des meilleurs services au client

classés selon leur consommation énergétique et des émissions en carbone des centres de données (RAM, CPU, bande passante et énergie).

- assurer de la compatibilité technique et sémantique des services Cloud hétérogènes sensibles aux contextes.

Notre ontologie décrit les informations contextuelles des services Cloud et la prise en compte de la qualité des services et du contexte (*centres de données, préférences de l'utilisateur et ses contraintes ; environnement*). Elle implémente la notion de Green Computing afin d'obtenir le chemin optimal (*liste des centres de données et ses services*) qui minimise la consommation de l'énergie et satisfait les besoins du client tout en permettant la découverte des services souhaités hébergés dans les centres de données.

La Figure 4-1présente la structure générale de notre ontologie. Les fournisseurs vont instancier les classes de l'ontologie et publier les instances résultant sous forme de fichiers OWL. L'ontologie adoptée vise à capturer la sensibilité du contexte et à contrôler la qualité des services au niveau sémantique. En raisonnant les besoins fonctionnels et non fonctionnels des clients mobiles, les meilleures services correspondants sont déduits afin d'optimiser la qualité et les exigences contextuelles des clients et l'énergie des centres de données. Elle contient 6 concepts clés : le profil **Utilisateur**, le **Cloud Provider**, le **DataCenter**, le **Contexte**, le **Service Cloud** et les **QoS**.

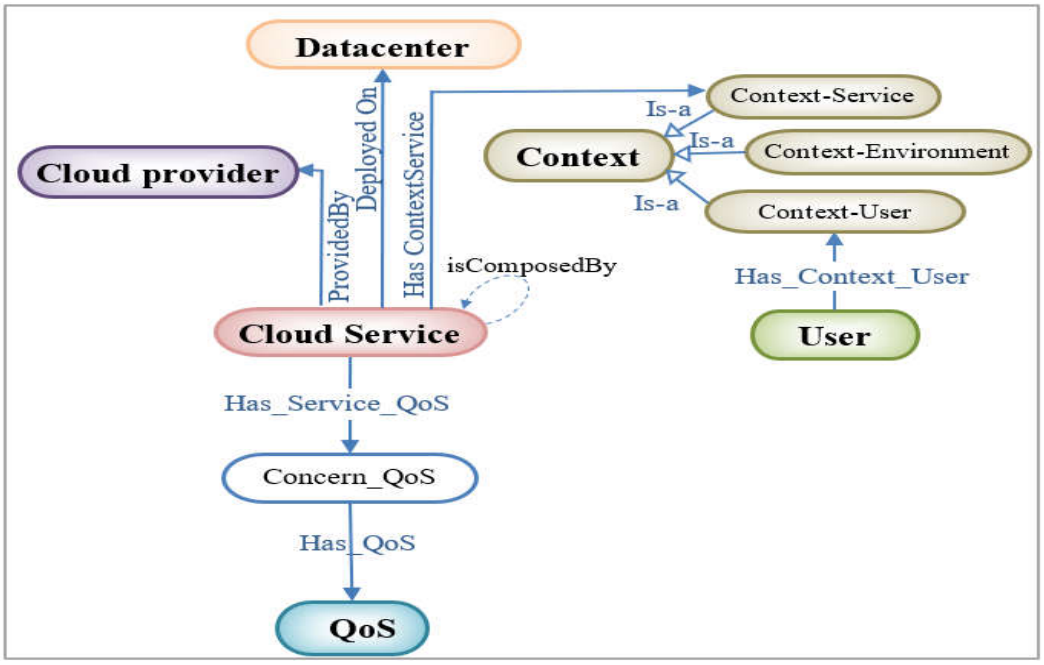


Figure 4-1 Structure général de l'ontologieDSCxGreenOnto.

4.3.1 Modélisation des services Cloud

Le « *Cloud service* » est le concept clé de l'ontologie proposée. Il définit des concepts communs ainsi que les relations entre ces concepts pour représenter les paramètres liés au service. Il est caractérisé par un identifiant unique, un nom, une URI, son contexte et son modèle du service (*SaaS, IaaS, PaaS*), le domaine de service (transport, hôtel,..), le fournisseur de services (Microsoft Azure, Amazon, Google,…), ainsi que les contraintes de contexte.

L'ontologie présentée sélectionne un service dans un environnement dynamique en confrontant un contexte requis (client) avec un contexte fourni (QoS). Chaque service Cloud est décrit par un ensemble de propriétés de QoS et de contexte. L'ontologie couvre des services de différents domaines tels que la santé des patients, le transport, le tourisme, télécommunication, etc. En raison de l'hétérogénéité des services ainsi que leurs protocoles de communication (GSM, 3G, 4G, etc.), ilspeuvent être spécifiés plus facilement en utilisant une représentation sémantique. Chaque service est enrichi par un ensemble de paramètres (*disponibilité, localisation et temps d'exécution, exigences matérielles et logicielles, préférences d'exécution de servic*e) et de paramètres liés à la qualité de service. Un service Cloud est une suite d'actions qui peuvent être utilisées pour caractériser une activité utilisateur. Les services sont organisés en hiérarchies de classes (*services médicaux, liés au tourisme, multimédias, de stockage et de commerce*) et liées par des services qui fournissent des données ou récupèrent des modifications. La Modèle sémantique du profil utilisateur.montre le modèle sémantique de service et toutes ses caractéristiques.

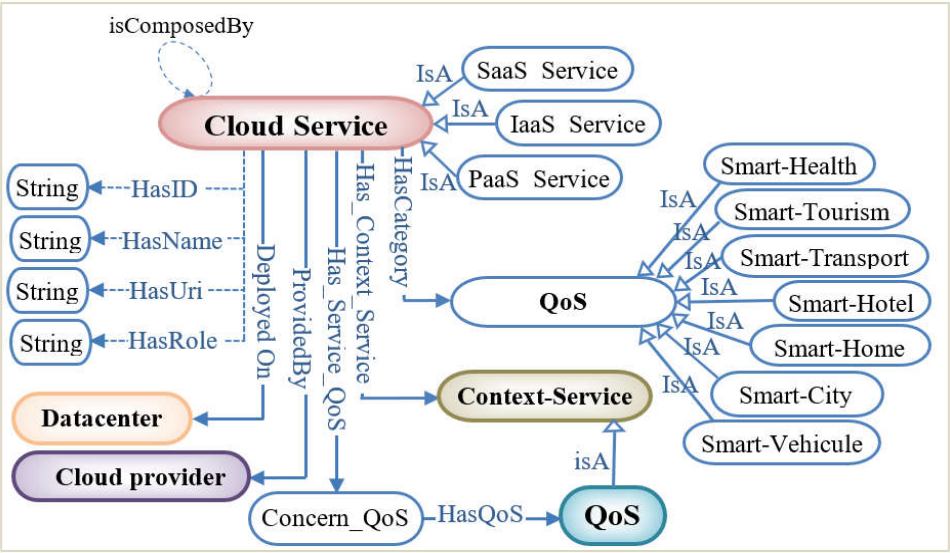


Figure 4.2 Modèle sémantique du service.

Dans la Figure 4-2, nous présentons un exemple de description du service sémantique de réservation de transport L'exemple, représenté en OWL, correspond à l'ontologie du service Cloud, appelée espace de noms DSCxGreenSCOnto. Comme l'illustre laFigure 4-2, l'attribut ID (ligne 1) représente un identificateur de service unique généré par UDDI pour faire référence à un service Cloud. Le service Cloud SaaS_reservation consiste en une opération de réservation de transport (ligne 5) enrichie par des informations sémantiques. Ces informations correspondent au rôle de service (réservation transport) auquel la sémantique du paramètre sera associée (format in = détail client, format out = boolean) (lignes 5 à 15). Ce service est associé à une contrainte de contexte (lignes 17 à 19) et possède deux QoS (lignes 20 à 23). La première qualité de service (lignes 33 à 36) représente le prix du service Cloud (prix 100$). La deuxième qualité de service représente le temps de réponse du service Cloud (letemps de réponse est élevé).

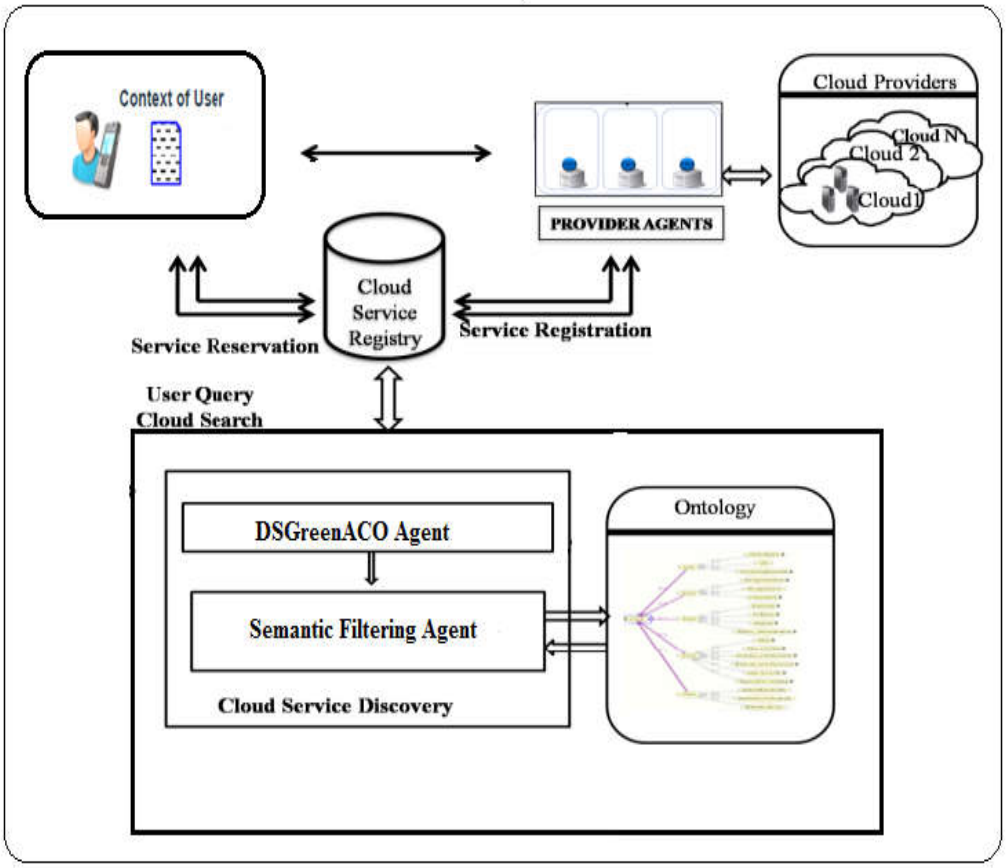


Figure 4.3 Service de réservation représenté par DSCxGreenSCloud.

```
< DSCxGreenSCloud:CloudService rdf:ID="SaaS_01" >
  <DSCxGreenSCloud:hasName> "SaaS_Reservation_Vol" </DSCxGreenSCloud:hasName>
  <DSCxGreenSCloud:hasCategory>"SmartTransport"</DSCxGreenSCloud:hasCategory >
  < DSCxGreenSCloud: hasUri> " " < /DSCxGreenSCloud: hasUri >
  < DSCxGreenSCloud: hasRole> " " < /DSCxGreenSCloud:hasRole>
  < DSCxGreenSCloud: Operation> rdf:ID="OP_1" rdf: Name="OP_1">
      < DSCxGreenSCloud:hasParameters>
          < DSCxGreenSCloud:Parameter rdf:ID="SaaS_Reservation" >
          < DSCxGreenSCloud:Parameter rdf:ID="SaaS_Reservation" >
          < DSCxGreenSCloud:Parameter rdf:ID="SaaS_Reservation" >
          < DSCxGreenSCloud:Parameter rdf:ID="SaaS_Reservation" >
      < /DSCxGreenSCloud: hasParameters >
< /DSCxGreenSCloud:Operation>
< CxQSCloudSERP:hasQoS rdf:resource ="#QoS_1_1" >
< CxQSCloudSERP:hasQoS rdf:resource ="#QoS_1_2" >
.....
</DSCxGreenSCloud:CloudService>
< DSCxGreenSCloud:QoS rdf:ID="QoS_1_1">
   <DSCxGreenSCloud:hasName>Price</DSCxGreenSCloud:hasName>
   <DSCxGreenSCloud:hasQoSValue >100</ DSCxGreenSCloud:hasQoSValue>
   <DSCxGreenSCloud:hasQoSLogicValue >High</DSCxGreenSCloud: hasQoSLogicValue >
   < DSCxGreenSCloud:hasQoSNormValue >0.68</DSCxGreenSCloud:hasQoSNormValue >
< DSCxGreenSCloud:QoS rdf:ID="QoS_1_2">
< DSCxGreenSCloud:QoS rdf:ID="QoS_1_2">
   <DSCxGreenSCloud:hasName>Time</DSCxGreenSCloud:hasName>
   <DSCxGreenSCloud:hasQoSValue >10</ DSCxGreenSCloud:hasQoSValue>
   <DSCxGreenSCloud:hasQoSLogicValue>High</DSCxGreenSCloud:hasQoSLogicValue>
< DSCxGreenSCloud:hasQoSNormaValue >0.79</DSCxGreenSCloud:hasQoSNormValue >
< DSCxGreenSCloud:QoS rdf:ID="QoS_1_2">
.....
```

Figure 4-2 Description de service de transporte

4.3.2 Modélisation de QoS

Des services Cloud peuvent avoir les mêmes fonctionnalités avec des qualités de service différentes. La qualité d'un service (QoS) dans notre ontologie est décrite comme un ensemble de paramètres de métadonnées QoS (voirFigure 4-4). Les paramètres de QoS sont:

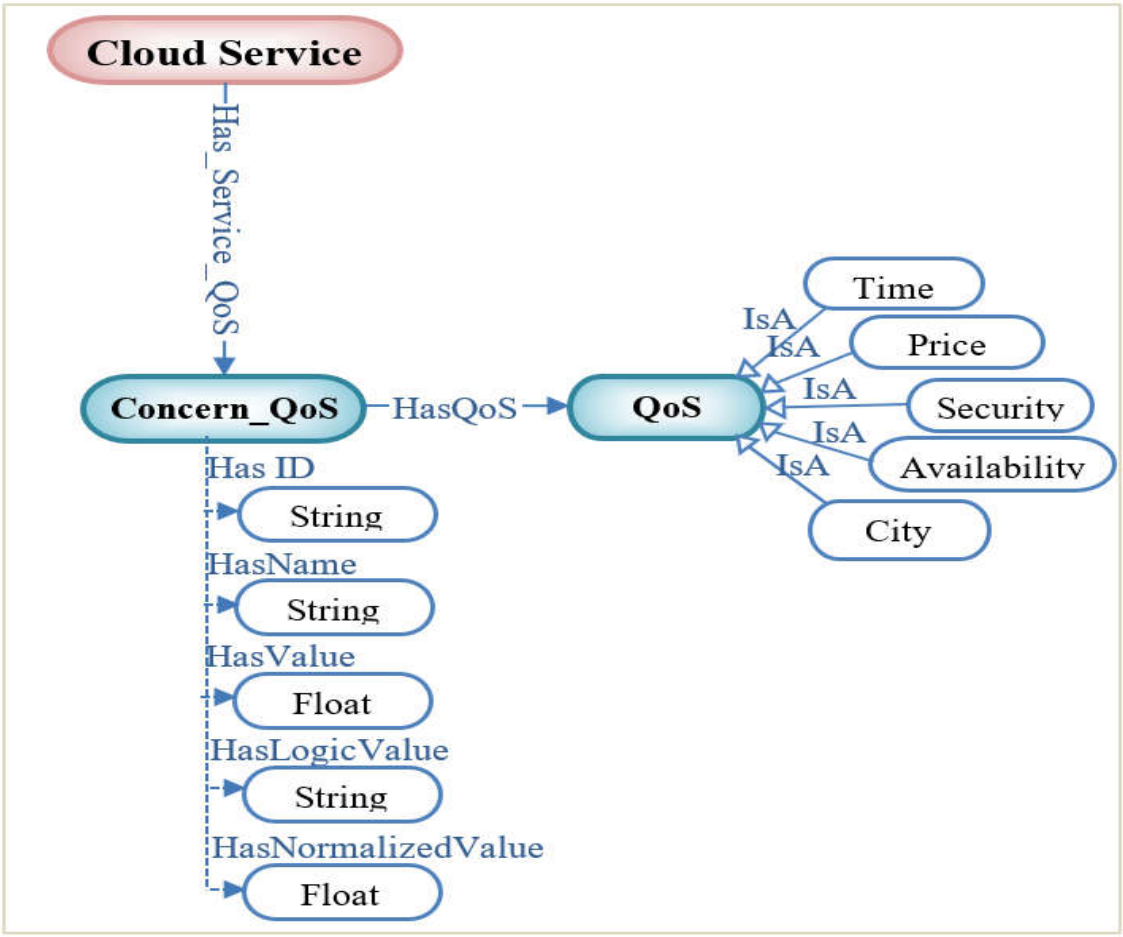


Figure 4.4 Modèle sémantique de QoS.

- **Temps de réponse :** est le temps mis entre l'envoi de la requête par l'utilisateur et la réception du résultat envoyé par le serveur Cloud. Le temps de réponse, appelé aussi latence, représente le temps requis par un service pour répondre à une demande.

- **Prix du service:**cet attribut représente le coût monétaire que l'utilisateur doit payer pour utiliser une ressource, il est obtenu en compensant le coût du matériel et des logiciels. La plupart des fournisseurs fixent des prix différents pour leurs services.

- **Sécurité du service :**ce paramètre représente le degré de sécurité du service. La sécurité vise à réaliser et à garantir les objectifs suivants : la confidentialité, l'authentification et l'intégrité.

- **Disponibilité :**ce paramètre prend une valeur entre 0 et 1. En effet, à un instant donné, un service peut être disponible (c.à.d. disponibilité > 0) ou indisponible (c.à.d. disponibilité = 0).

- **Fiabilité :** représente l'aptitude d'un service à s'exécuter sans détériorer l'information traitée tout en respectant les demandes et les conditions contractuelles. La fiabilité représente le taux du succès d'un service pour répondre à une invocation.

Laqualité d'un service $S_i$est représentéesous forme d'un vecteur QoS (S_i)=< $q_{i,1}$, $qi_{,2}$...,$q_{i,n}$ >, ou n représente le nombre d'attributs de QoS requis par l'utilisateur et $q_{i,j}$ représente la valeur de l'attribut j ($1 \leq j \leq n$) de QoS du service Si. Afin de normaliser les valeurs des attributs de QoS, on utilise les formules suivantes :

- Normalisation des attributs positifs :

$$q'(i,j) = \frac{q(i,j)_j^{max} - q(i,j)}{q(i,j)_j^{max} - q(i,j)_j^{min}} (1)$$

- Normalisation des attributs négatifs

$$q'(i,j) = \frac{q(i,j) - q(i,j)_j^{min}}{q(i,j)_j^{max} - q(i,j)_j^{min}} (2)$$

$$w_k = \sum_{i=k}^{n} (1/i)/n (3)$$

Pour mieux guider le processus de sélection et de composition des services Cloud, nous proposons donc dans la section suivante des descriptions des profils, des descriptions des contextes d'exécutions ainsi que la description des datacenters.

4.3.3 Modélisation du profil utilisateur

Un profil utilisateur décrit un ensemble des informations sur le contexte de l'utilisateur, ses besoins fonctionnels, ses contraintes et ses préférences de QoS et du contexte. Pour qu'un Cloud service puisse être sélectionné, il est primordial que ce service satisfasse l'ensemble des contraintes imposées par ce profil. Les propriétés importantes comprennent également les contraintes de l'utilisateur : âge, langue et localisation. LaModèle sémantique du profil utilisateur.montre le modèle sémantique du profil utilisateur avec toutes ses métadonnées.

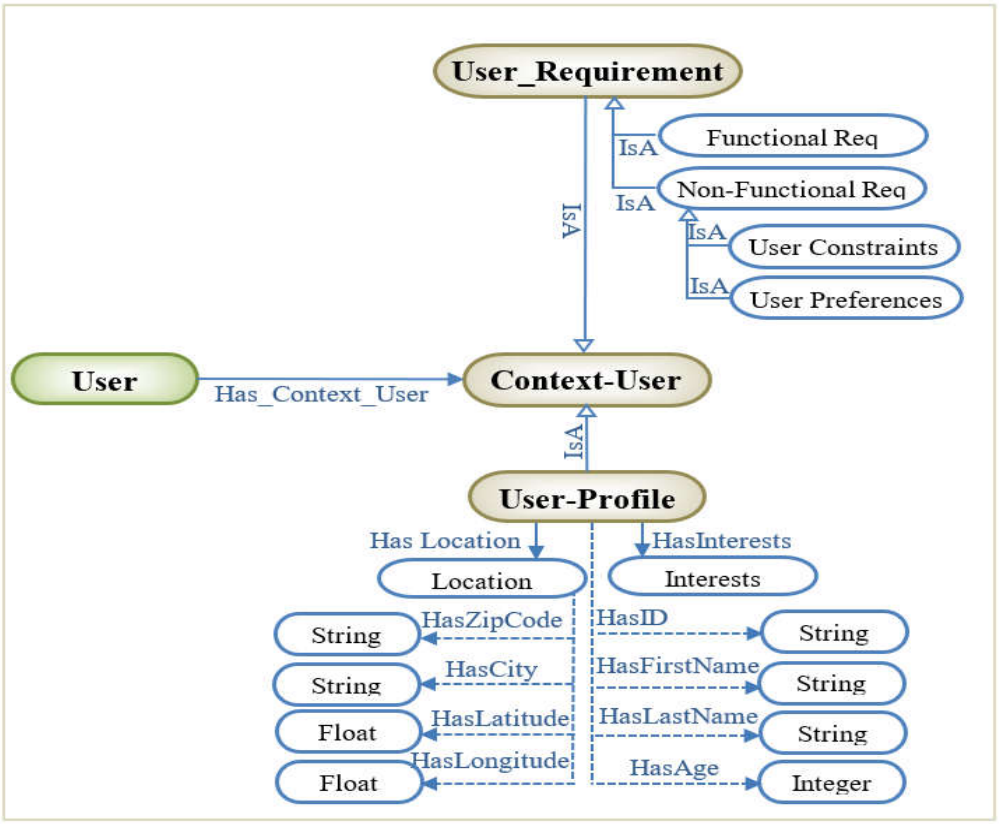


Figure 4-3Modèle sémantique du profil utilisateur.

- **La contrainte utilisateur** décrit les limites et les restrictions exigées par le client concernant les propriétés fonctionnelles et non fonctionnelles (QoS et contexte). Elle se compose des expressions sémantiques logiques simples ou complexes. Par exemple, un client veut obtenir un service de type transport des personnes avec une sécurité haute et un prix moins cher. Cette contrainte de contexte peut être vue comme suit : **Type_Service = Transport AND Prix = Pas chère AND Sécurité= Haut.** Les contraintes de QoS de l'utilisateur doivent être explicites pour pouvoir les satisfaire. Ces contraintes peuvent être satisfaites par chaque service ou par le service composite. Un utilisateur spécifie ses préférences et ses contraintes visuellement via des appareils mobiles (tablette, smartphone, etc.) à travers une interface graphique traduisent vers des règles logiques attributs de QoS et ses priorités basées sur notre modèle d'ontologie. L'ontologie stocke ces informations dans le profil utilisateur sous forme RDF/XML. Les contraintes sont traduites en valeurs de contraintes sémantiques (*faible, moyenne, haute*). Ensuite, ces valeurs sémantiques sont mappées sur trois intervalles:

- QoS Haut si la valeur normalisée dans l'intervalle [0.7*QoS_Max, QoS_Max]

- QoS Moyen si la valeur normalisée dans l'intervalle [0.3* QoS_Max, 0.7* QoS_M].
- QoS Faible si la valeur normalisée dans l'intervalle [0, 0.3* QoS_Max].

- **Les préférences de l'utilisateur** peuvent être utiles pour sélectionner des services Cloud sensibles au contexte avec des QoS optimisées. Pour cela, l'utilisateur doit indiquer un niveau de préférence pour chaque attribut de QoS à l'aide d'une interface graphique. Chaque utilisateur à ses préférences personnelles. Par exemple, un utilisateur considère plus d'importance à la sécurité d'un service Cloud par rapport à son temps d'exécution, tandis qu'un autre accorde une plus grande importance au temps d'exécution par rapport à la sécurité. Les préférences de QoS sont ordonnées par le client sous forme d'une liste, de la préférence la plus importante jusqu'à la moins importante. Cet ordre affecte un poids à chaque attribut qi de QoS, appelé « weight ». Les centrioles de classement constituent un moyen de convertir des rangs (1er, 2ème, 3ème) en poids qui sont des valeurs numériques. Si n est le nombre des attributs de QoS, le poids de chaque attribut QoS_k est calculé par l'équation 4 en utilisant le concept de « Rank Order Centroids »[76].

$$w_k = \sum_{i=k}^{n} (1/i)/n \quad (4)$$

Exemple

Soit S un service Cloud. La sécurité et le prix sont les attributs de QoS de ce service. Un utilisateur considère que le prix est le critère le plus important, et la sécurité, l'attribut le moins important. Dans ce cas les poids sont calculés comme suit :

- poids de l'attribut prix : w1 = (1+1/2) /2 = 0.75.
- poids de l'attribut sécurité : w2 = (1+1/2) /2 = 0.25.

4.3.4 Modélisation de contexte

Le contexteest tout type d'informations qui peut être collecté à partir du service (nom, version, besoins en ressources), de l'environnement (heure, lieu) et de l'utilisateur (contraintes et préférences). Il identifie également où la personne, le service et la ressource sont situés. Notre ontologie utilise ces informations pour sélectionner un service dans un environnement dynamique, en confrontant un contexte requis

(utilisateur, service) pour un contexte fourni (fournisseur Cloud).La Figure 4-6 montre les sous-classes de base du contexte de classe de notre ontologie qui se compose de :

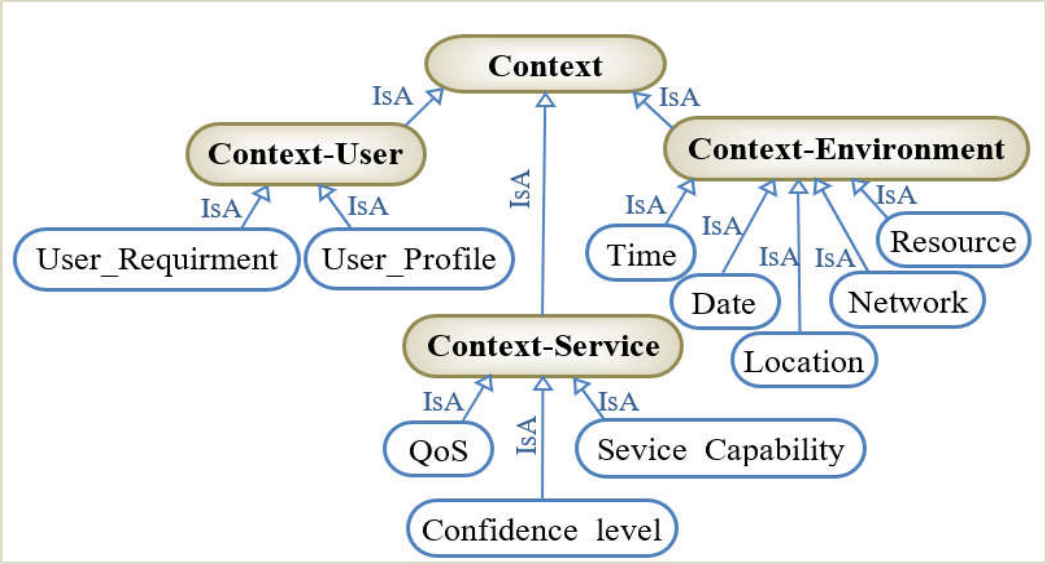


Figure 4.6 Modélisationsémantique de contexte.

Contexte = {Utilisateur, Service, Environnement}.

- ***Le contexte utilisateur :*** un profil décrit un ensemble de contraintes et préférences contextuelles. Pour qu'un Cloud service puisse être sélectionné, il est primordial que ce service satisfasse l'ensemble des contraintes et les préférences de l'utilisateur imposées par ce profil. Les propriétés importantes comprennent également : *âge, langue et localisation.*

Context-utilisateur= {profil utilisateur, contraintes et préférences utilisateur}.
Profile utilisateur = {nom, rôle, ID, âge, adresse, langue, localisation}
Contraintes d'utilisateur = {qualité de l'environnement, QoS}.
Préférence d'utilisateur = {qualité de l'environnement, QoS}.

- ***Le contexte lié au service :*** Le contexte du service Cloud est défini par les capacités et les QoS du service. Les capacités du service Cloud sont consistées par des attributs sémantiques comme préconditions, les effets et post-conditions. Le contexte de service contient les capacités de service, sa version et un niveau de confiance. Le niveau de confiance de service est très intéressant dans le cas d'avoir deux services Cloud qui fournissent les mêmes fonctionnalités. Ils peuvent avoir différentes valeurs pour les attributs de performance. Le contexte de service se définit par un niveau de confiance. Le niveau de confiance de service est très intéressant dans le cas d'avoir deux

services Cloud qui fournissent les mêmes fonctionnalités. Ils peuvent avoir différentes valeurs pour les attributs de performance.

Contexte lié au service : {version, capacité de service, niveau de confiance}

- ***Le contexte d'environnement :*** L'environnement décrit des informations contextuelles telles que : les connexions des réseaux disponibles (WIFI, GSM, etc.), heure, la date et les contraintes de localisation géographiques. Les connexions entre centres de données ayant une bande passante limitée. L'emplacement, la date et l'heure représentent respectivement une situation géographique bien précise ou le service doit être accompli. Les centres de données agissent comme environnement d'exécution pour le service Cloud.

Contexte d'environnement : {Temps, location, date, réseau, ressource}

4.3.5 Modélisation des centres de données

Un data center ou centre de données est une infrastructure composée d'un réseau d'ordinateurs et d'espaces de stockage. Cette infrastructure peut être utilisée par les clients pour stocker et distribuer de grandes quantités de données à travers un réseau interne ou via un accès Internet.En règle générale, les centres de donnéessont très énergivores d'énergie. Notre ontologie repose fortement sur stratégies intelligentes d'optimisation énergétiques et de QoS afin d'offrir des services Cloudselon les désirs des clients tout en en limitant les émissions de CO_2 et la consommation d'électricité.

La classe Datacenter est réduite à l'ensemble des centres de données standards comme Google, Amazon. Il s'agit des centres de données publiques, privé, hybride et les ressources matérielles ainsi que les services disponibles. Il regroupe des serveurs, des commutateurs de réseau, des routeurs, des firewalls, et bien entendu des câbles et des racks physiques permettant d'organiser et d'interconnecter tout cet équipement informatique.

Un centre de données est défini par un fichier RDF/XML qui regroupe des serveurs "Host" et ses caractéristiques (*système, CPU, RAM, énergie et débit*), et les équipements réseau. LaFigure 4-4 présente les principaux composants d'un centre de données. Un centre de données représente un environnement physiquequi regroupe des ressources informatiques (serveurs, baies de stockage…) permettant le stockage, le traitement et la protection de données.

Nous avons identifié quatre types de ressources informatiques intéressantes dans les centres de données : "Host", "Software", "Power", et "Network" (cf.Figure 4-4). Chaque centre de données est caractérisé par la charge courant de réseau "Networkload" et la charge courant de centre de données "Workload".

- La charge de travail "workload" appliquée à un centre de donné à un instant donnée est exprimée comme un seuil entre 0 et 1. Le seuil est calculé par rapport à la charge nécessaire au traitement des demandes de services à un instant et les ressources disponibles (i.e. CPU, ressources de stockage disponible, RAM disponible). Un centre de données est chargé si le rapport atteint un seuil de 1 (inspiré et adapté de [77]).

- La charge de réseau " Networkload" sur un centre de données est le débit courant inter-centre de donnée entre un centre donné et les autres centres de données. Cette information est exprimée comme un intervalle correspondant à une valeur logique :

  - Networkload = Haut si disponibilité est dans l'intervalle]0.7*BwMax, B BwMax].
  - Networkload = Moyen si disponibilité est dans l'intervalle [0.3*BwMax, 0.7* BwMax].
  - Networkload =Faible si disponibilité est dans l'intervalle]0,0.3* BwMax].

  Tel que BwMax est la bande passante maximale entre les centres de données.

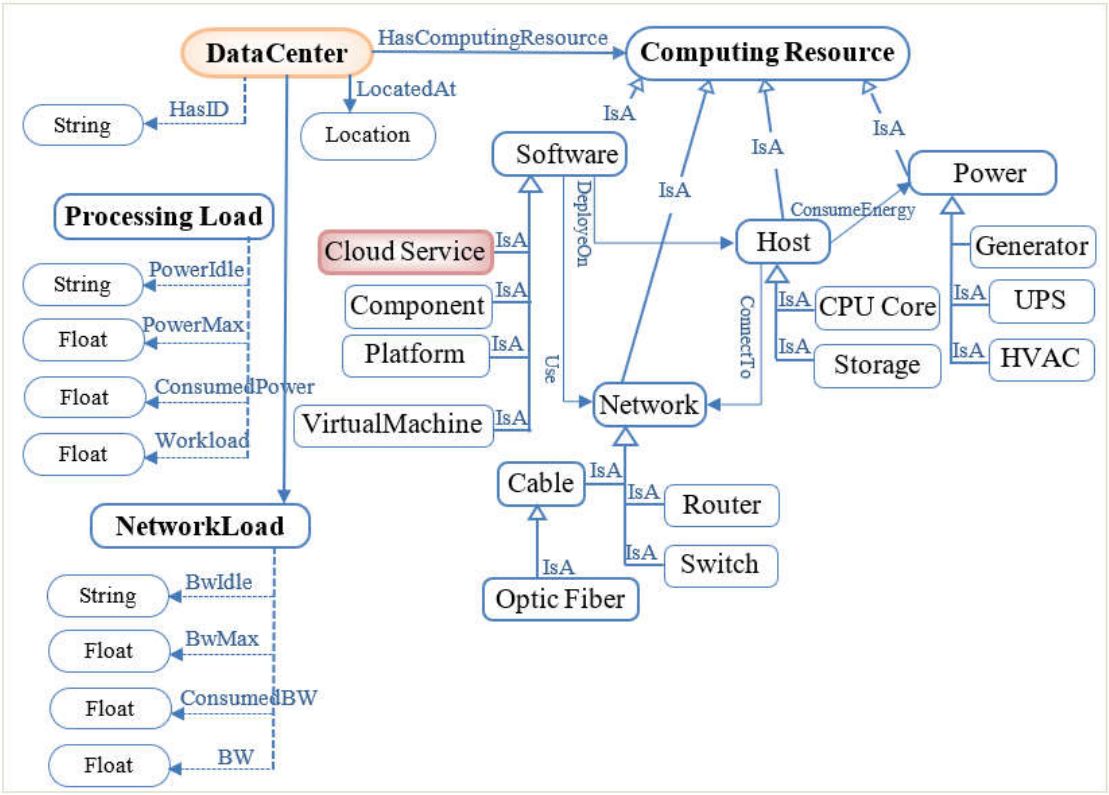


Figure 4-4 Modélisationsémantique des centres de données.

4.3.5.1 Serveur (Host)

Cette classe décrit une machine plus puissante capable de traiter un nombre important des travaux et d'exécuter des services. La classe "Host" a besoin d'énergie pour exécuter des services Cloud et être connectée à un réseau "Network".

4.3.5.2 Réseau (Network)

Cette classe décrit toutes les ressources réseaux d'un centre de données et leurs consommations en bande passante afin de garantir le fonctionnement d'un service. Notre choix s'est fixé sur quelques équipements réseau, tels que : les commutateurs et les câbles de fibre optiques. Le commutateur représente une entité réseau qui peut être configuré en tant que routeur ou commutateur. Il peut modéliser les passerelles dans la transmission des données à un hôte ou un autre commutateur basé sur l'endroit où les données s'appartiennent.

4.3.5.3 Logiciels (Software)

Cette classe décrit toutes les ressources logicielles d'un centre de données et leurs consommations. Une taxonomie logicielle, les services disponibles pour l'utilisateur, les

descriptions de logiciel, et leur état de fonctionnement (Figure 4-4). La classe "Software" contient les sous classes : "Plateforme", "Virtual Machine", "Service", "Component", etc.

4.3.5.4 Energie (Power)

Cette classe décrit toutes les ressources énergétiques d'un centre de données et leurs consommations pour fonctionner correctement. La classe "Power" contient les sous classes : commutateur électrique, des générateurs dédiés au backup et un système de ventilation et de refroidissement.

4.3.6 Modélisation des fournisseurs de service

Le fournisseur de services fournit des services auxutilisateurs. L'utilisateur est libre de choisir le fournisseur de service le plus approprié mais n'a aucun droit de changer le prix de service.Chaque fournisseurpubliéun fichier de description des services locaux sous une forme (décrite à l'aide d'OWL/XML).De nombreux fournisseurs tels queMicrosoft Azure et Google App Engineont créé dans leurs data centres des serveurs de Cloud à la demande. Mais chaque fournisseur Cloud a ses spécificités en termes de vitesse, d'élasticité, de latence, etc. Il existe trois principaux types de fournisseursde servicesCloud : public, privé et hybride (voirFigure 4-5).

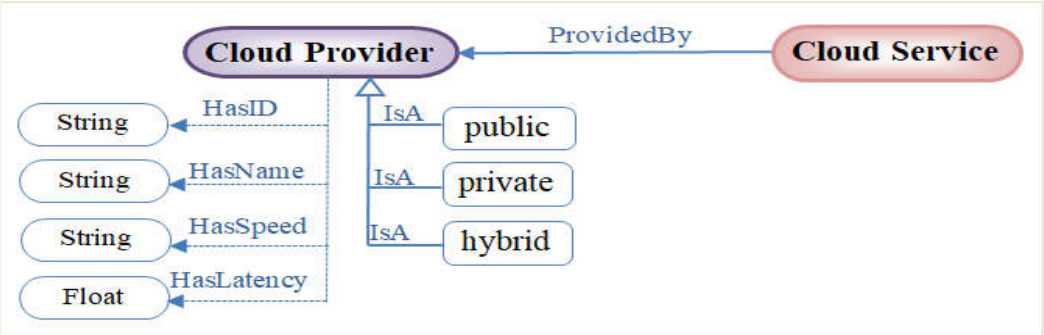


Figure 4-5 Modélisationsémantique des fournisseurs de service.

- **Fournisseur public :** est un fournisseur ouvert dans lequel les services sont fournis par un tiers. Les services d'infrastructure et de plateforme sont fournis au public selon le contrat de niveau de service (SLA) entre le fournisseur et le client. Le fournisseur public moins coûteux pour l'hébergement des services.

- **Fournisseur privé :** est un fournisseur de services dans lequel l'infrastructure et la plateforme sont exploitées entièrement par le fournisseur de service.Un

fournisseur privé offre de nombreux avantages qu'un fournisseur public tel que l'élasticité basée service.

- **Fournisseur hybride :** est une combinaison du fournisseur public et du fournisseur privé. Cependant, avoir des fournisseurs publics et privés fonctionnant ensemble nécessite l'interopérabilité et la portabilité des services et les données pour permettre la communication entre les modèles.

4.4 Implémentation de l'ontologieDSCxGreenSCloud

Après avoir conçu le modèle d'ontologie dans la partieprécédente, nous passons maintenant à sa formalisation et son implémentation. Ceci est fait à travers l'éditeur deprotégé 3.4.8. Protégé est un éditeur visuel d'ontologie open source gratuit. Il possède une interface modulaire, ce qui permet l'édition et la visualisation des ontologies. Il permet aussi le contrôle et la cohérence de l'ontologie par des vérifications de contraintes.

4.4.1 Implémentation des classes

On a suivi une approche Top-Down qui a été choisi pour définir la hiérarchie des classes, puis la définition des concepts plus généraux, par la suite on les a spécialisés.LaFigure 4-6montre une vue globale de l'ontologie avec ses classes et sous classes.

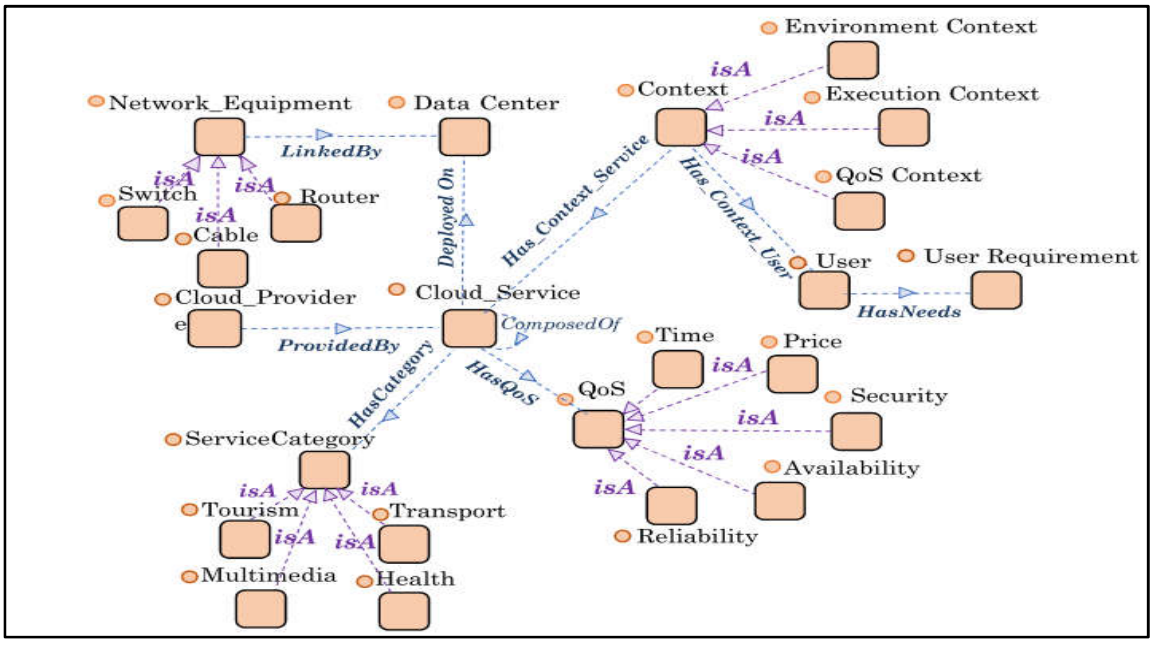


Figure 4-6 Les classes de l'ontologie DSCxGreenSCloud

4.4.2 Implémentation des propriétés

Après la modélisation des classes et sous classes, on introduit les propriétés pour chaque classe de l'ontologie pour donner une valeur sémantique à chaque service publié. LaFigure 4-6montre l'implémentation des propriétés des classes et les relations entre elles.

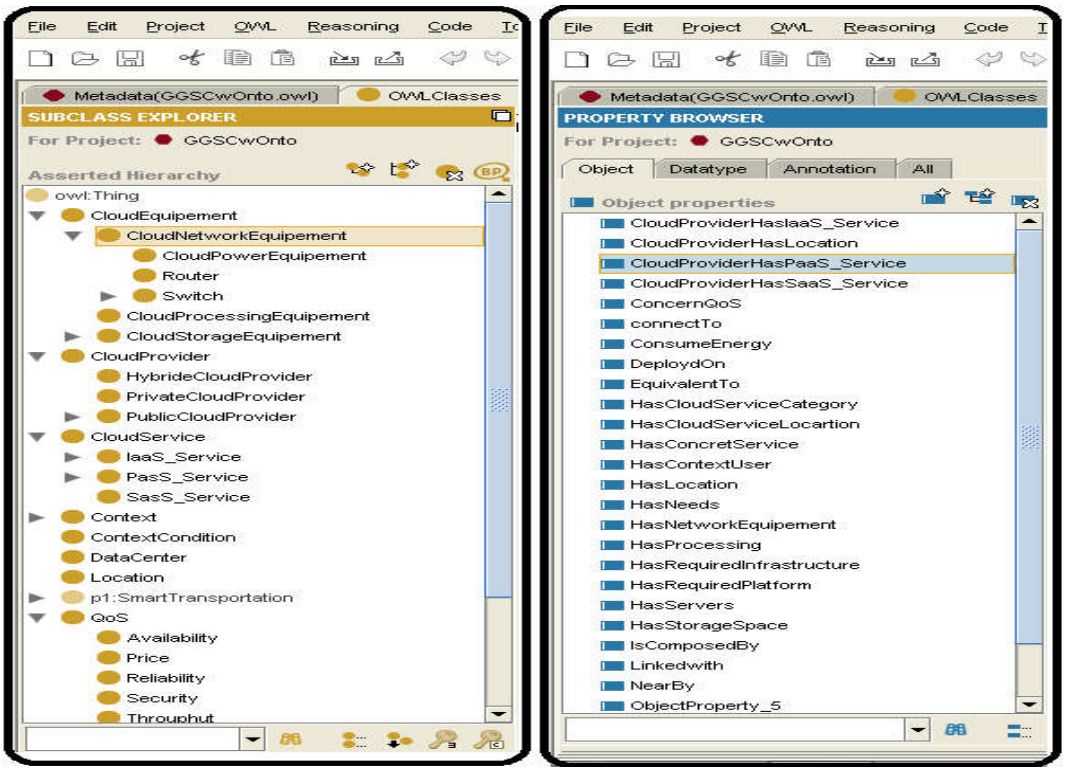


Figure 4-7 Les propriétés et les relations entre les classes.

4.4.3 Implémentation des individus

La création des instances (individus) pour les classes : **profilutilisateur**, le **Cloud Provider**, le **DataCenter**, le **Contexte**, le **Service Cloud** et les **QoS**se fait en deux manières. La première est une génération automatique et aléatoires des individus et la deuxième à partir d'un ensemble des données réels. LaFigure 4-8 montre quelques instances de la classe de l'ontologie DSCxGreenOnto.

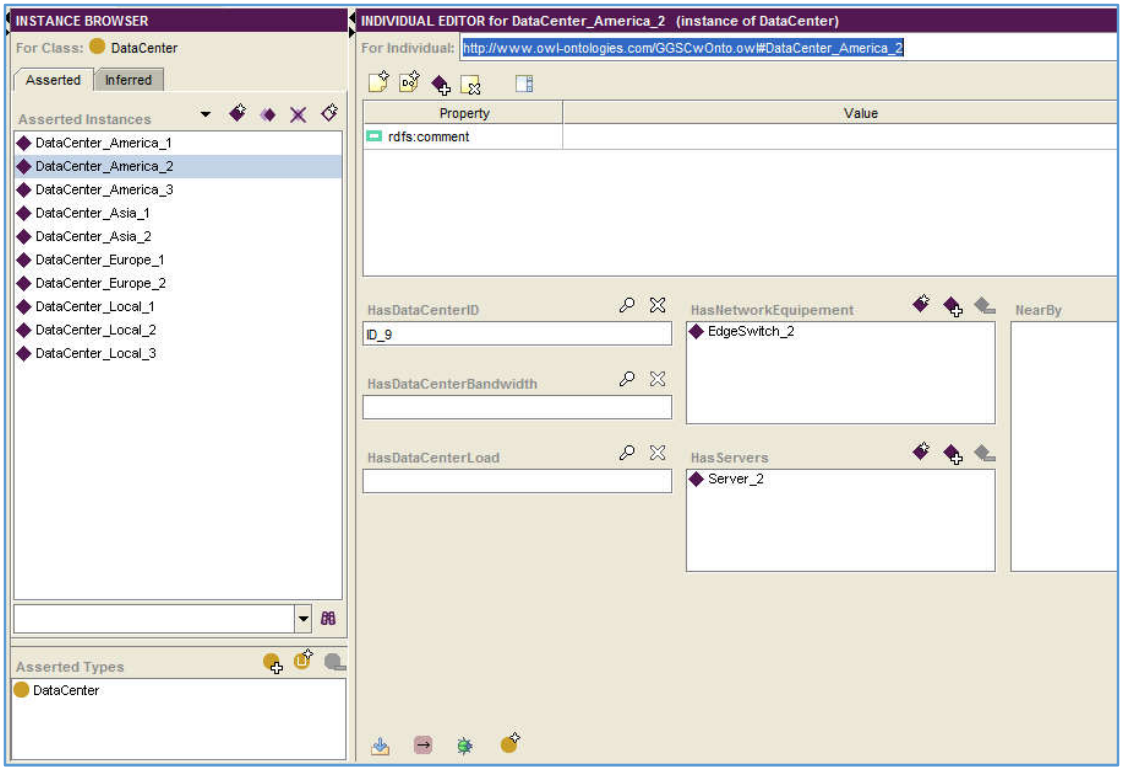


Figure 4-8 Les individus l'ontologie DSCxGreenSCloud

4.4.4 Lesrequêtes SPARQL

Nous allons également décrire quelques requêtes SPARQL. SPARQL est un acronyme récursif pour SPARQL Protocol and RDF Query Language. Comme son nom l'indique, SPARQL est à la fois un protocole et un langage de requête. Le langage de requête SPARQL est un langage dont la syntaxe ressemble à SQL pour l'interrogation des graphes RDF par filtrage. Les caractéristiques du langage comprennent des modèles de base conjonctives et des filtres de valeurs, notamment. Le protocole SPARQL est une méthode pour l'invocation distante des requêtes SPARQL. Elle spécifie une interface simple qui peut être prise en charge via HTTP ou SOAP et qu'un client peut utiliser pour émettre des requêtes SPARQL sur des points d'accès SPARQL. SPARQL est le langage de requête développé par le W3C pour interroger les graphes RDF. Il est adapté à la structure spécifique de triplets qui les constituent. Il permet d'exprimer des requêtes interrogatives ou constructives.SPARQL est un langage de requêtes sur des descriptions OWL.

Sur le modèle d'ontologie DSCxGreenSCloud, nous présentons quelques exemples de requêtes SPARQL qui peut être spécifiées dans le but de récupérer des informations importantes contenues dans un modèle d'ontologie. Notez que ces requêtes peuvent être utilisées par les processus de sélection et de composition afin de répondre aux contraintes et préférences des utilisateurs. LaFigure 4-9 illustre quelques requêtes

SPARQL. Une requête peut renvoyer une liste contenant tous les services du DSCxGreenSCloud sémantiquement équivalents avec différents QoS (R01). Une requête peut retourner une liste contenant toutes les informations de QoS d'un service Cloud (R02). Une requête peut renvoyer tous les *services de transport*, enrichi par des services sémantiquement proches de service générique de l'ontologieDSCxGreenSCloudet de l'ontologiedomaine de "*transport*" (R03).Une requête peut retourner la liste des services filtrés selon les préférences du client (*leprix est moins cher*) pour identifier les centres de données qui offrent le service souhaité par le client (R04).

Requête 1: Liste des services équivalents.

```
PREFIX cloudSGreenOnto:
<http://www.owl-ontologies.com/GGSCwOnto.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?sid ?sname ?slocation ?srole ?scategory
WHERE {
?sc cloudSGreenOnto:HasCloudServiceID  ?sid .
   ?sc cloudSGreenOnto:HasCloudServiceLocation  ?slocation .
   ?sc cloudSGreenOnto:HasCloudServiceName  ?sname .
   ?sc cloudSGreenOnto:HasCloudServiceRole  ?srole .
?sc cloudSGreenOnto:HasCloudServiceCategory  ?scategory .
}
```

Requête 2: Liste des informations de QoS d'un service Cloud.

```
PREFIX cloudSGreenOnto:
<http://www.owl-ontologies.com/GGSCwOnto.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?sid ?qos_service ?qos_value ?qos_nv ?qos_logic_value
WHERE {
   ?sc cloudSGreenOnto:HasCloudServiceID  ?sid .
   ?qos_service cloudSGreenOnto:QoS_Concern_Service ?sc .
   ?qos_service cloudSGreenOnto:HasQoSValue ?qos_value .
   ?qos_servicecloudSGreenOnto:HasQoSNormalizedValue ?qos_nv.
   ?qos_service cloudSGreenOnto:HasQoSLogicValue ?qos_logic_value .}
```

Requête 3: Liste des informations de QoS d'un service Cloud.

```
PREFIX cloudSGreenOnto:
<http://www.owl-ontologies.com/GGSCwOnto.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX cloudSGreenOnto:
```

```
<http://www.owl-ontologies.com/GGSCwOnto.owl#>
SELECT ?service
WHERE {
?service cloudSGreenOnto:hasserviceCategory ?serviceCategory
?service cloudSGreenOnto:hastype ?servicetype
?service cloudSGreenOnto:hasCO2energy ?CO2Egy
?serviceCategory rdfs:subClassOf ?serviceCategoryClass
   FILTER (?servicetype =service:Transport)
}
```

Requête 4: Liste des informations de QoS d'un service Cloud.

```
PREFIX cloudSGreenOnto:
<http://www.owl-ontologies.com/GGSCwOnto.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX cloudSGreenOnto:
<http://www.owl-ontologies.com/GGSCwOnto.owl#>
SELECT ?service
WHERE {
?service cloudSGreenOnto:hasserviceCategory ?serviceCategory
?service cloudSGreenOnto:hastype ?servicetype
?service cloudSGreenOnto:hasCO2energy ?CO2Egy
?serviceCategory rdfs:subClassOf ?serviceCategoryClass
   FILTER (?servicetype =service:Transport)
}
```

Figure 4-9 forme de requêtes SPARQL

4.5 Conclusion

Dans ce chapitre, nous avons détaillé notreontologie, dédiée aux applications intelligentes sensibles au contexte. Cette ontologie sert à la description sémantique des services dans le Cloud et de guider le processus de sélection et de composition des services afin de produire uneapplication adaptée aux différentesexigences du client.L'objectif de l'ontologie proposée est de fournir au client des services personnalisés qui qui s'adaptent au client en fonction de ses préférences et son contexte.

En se basant sur cette ontologie, dans le chapitre suivant, nous détaillonsnotre approche bio-inspirée sémantique Green Computing pour la sélection et la composition des services Cloud. Nous présentons également quelques évaluations et comparaisons avec d'autres méthodes existantes dans la littérature.

Chapitre 5

5 Sélection économiques des services avecQoS basée sur unalgorithme bio-inspiré dynamiqueet sémantique

Sommaire

5.1 Introduction

Dans ce chapitre nous proposons une nouvelle variante de l'algorithme d'optimisation par colonie de fourmis (ACO - Ant Colony Optimization) appelé DSGreenACO (Dynamic Semantic Green Ant Colony Optimization) pour la résolution des problèmes d'optimisation d'énergie et des émissions en carbone des centres de données. Le système permet la sélection automatique, dynamique et sémantique des services Cloud et leur composition rapide afin d'obtenir un service composite personnalisé. Cette approche est une hybridation entre l'utilisation de l'ontologie proposée pour le filtrage sémantique des services et des métaheuristiques d'optimisation par colonie de fourmis pour l'optimisation de l'énergie pendant la découverte des services. Le système consiste en deux fonctionnalités principales. La première fonctionnalité consiste à une méthode dynamique guidée par les ontologies qui implémente la notion du Green Computing afin d'obtenir le chemin optimal (*liste des centres de données et ses services*) qui minimise la consommation de l'énergie et satisfait les besoins du client tout en permettant la découverte des services souhaités hébergés dans les centres de données. La deuxième offre au client un service optimisé qui respecte ses besoins et préférences.Nous présentons également un prototype pour la sélection et la composition des services Cloud. Le prototype filtre sémantiquement les services trouvés selon les préférences et le contexte du client et retourne une liste de meilleurs services de qualité au client classés par leurs consommations énergétiques (RAM, CPU, bande passante et énergie). Nous présentons des résultats de comparaisons entre les techniques utilisées dans notre approche avec d'autres techniques existantes dans la littérature. L'efficacité et les économies d'énergie de notre proposition ont été validées et évaluées au moyen de multiples expériences sur des ensembles de données aléatoires et réels.

5.1.1 Modélisation et formalisation du problème

Un algorithme hybride basé sur les méthodes métaheuristiques et les ontologies a été utilisé comme une stratégie de sélection et de composition sémantique et dynamique des services Cloud. En particulier, le comportement social à base de fourmis guidée par les ontologies est considéré pour l'amélioration du temps de découverte et la sélection des services pertinents et le taux de consommation de l'énergie. Deux modifications de l'algorithme ACO, portant sur le positionnement de la base et sur le modèle du taux d'évaporation de phéromone, ont été évaluées. L'adoption du choix adéquat du nombre de fourmis minimum par rapport au nombre de services Cloud est basée sur une fonction d'utilité de qualité. La stratégie de la fonction d'utilité de qualité proposée a contribué en partie à l'amélioration des performances de l'essaim dans la

découverte des services Cloud et l'optimisation au maximum de la quantité d'énergie consommée dans le Cloud. Un filtrage sémantique permettrait d'améliorer durant l'étape de recherche des services sémantiquement incompatibles avec ceux qui ont été déjà sélectionnés. Le client fournit en premier lieu un ensemble de qualité de service. Ensuite, en appliquant notre algorithme, nous sommes en mesure d'évaluer la QoS des fournisseurs de services dans le but d'identifier les services Cloud les plus adaptés au client avec une quantité optimale d'énergie consommée. Notre approche prend en considération les facteurs suivants :

- L'ensemble$ss = \{ss_1, ss_2, \dots, ss_n\}$ désigne un ensemble de n services sémantiques. Chaque service sémantique ss_i est lié à un ensemble de m services Cloud $\{cs_{i,1}, cs_{i,2}, \dots, cs_{i,m}\}$ ayant les mêmes fonctionnalités avec des valeurs de QoS différentes.
- Le service Cloud cs_i est défini par des paramètres de QoS. Le QoS du service de Cloud cs_i est défini comme suit :$QoS(cs_i) = \{q_{i,1}, q_{i,2}, \dots, q_{i,r}\}/0 \leq q_{i,j} \leq 1$, où m représente le nombre d'attributs QoS requis par le client. Il peut être divisé en 2 sous-ensembles: positif et négatif, où pour chaque QoS positive (QoS^+), des valeurs plus grandes indiquent une meilleure performance (par exemple la fiabilité, la sécurité et la disponibilité), alors que pour la QoS négative (QoS^-), des valeurs plus petites indiquent une performance plus élevée (par exemple le prix et le temps de réponse).
- Le système Cloud est modélisé sous forme de graphe$G(\mathrm{V}, \mathrm{E})$, où V représente l'ensemble des sommets et E est l'ensemble des arêtes. Chaque sommet représente un centre de données dc et chaque arête $e$représente un lien entre deux centres de données.
- Les Pf = <Pf_1, Pf_2,…, Pf_n> définissent les préférences qualitatives du client. Les préférences du client exprimées en valeurs sémantiques de l'ontologie sont transformées vers des valeurs quantitatives. Par exemple, le temps de réponse rapide est transformé vers une valeur logique « High » (temps de réponse normalisé dans l'intervalle [0.7, 1]), le temps de réponse moyen est transformé vers une valeur logique« Middle » (temps de réponse normalisé dans l'intervalle [0,3, 0,7[) et le temps de réponse lent est transformé vers une valeur logique « Low » (valeur normalisée dans l'intervalle [0, 0,3[).
- L'ensemble $DC = \{dc_1, dc_2, \dots, dc_N\}$ désigne un ensemble de N centres de données. Chaque centre de données représente un fournisseur de services. Les centres de données hébergent des serveurs et d'autres équipements

nécessaires pour fournir des services de qualité aux clients. Un hôte est composé de un ou plusieurs cœurs CPU et une quantité fixe de mémoire et de stockage qui peut être distribué aux services Cloud. Ainsi, l'hôte est défini par s_j , qui est la charge de travail appliquée à un centre de donnée j dans un temps donné à celle des ressources disponibles (CPU, mémoire disponible et stockage disponible). La bande passante entre un centre de données dc_iet les autres centres de données est définie comme un ensemble$BW_i = \{ bw_{i,1}, bw_{i,2}, \dots, bw_{i,N}\}$, où $BW_{i,j}$ représente le débit du chemin entre le centre de données i et le centre de données j. Nous modélisons les centres de données dans un Cloud avec un graphe complet ou les sommets représentent des centres de données et les arêtes représentent les bandes passantes.le coût E^{BW}entre deux centres de données est le poids de l'arête. on a obtenu le graphe représenté dans la figure suivante :

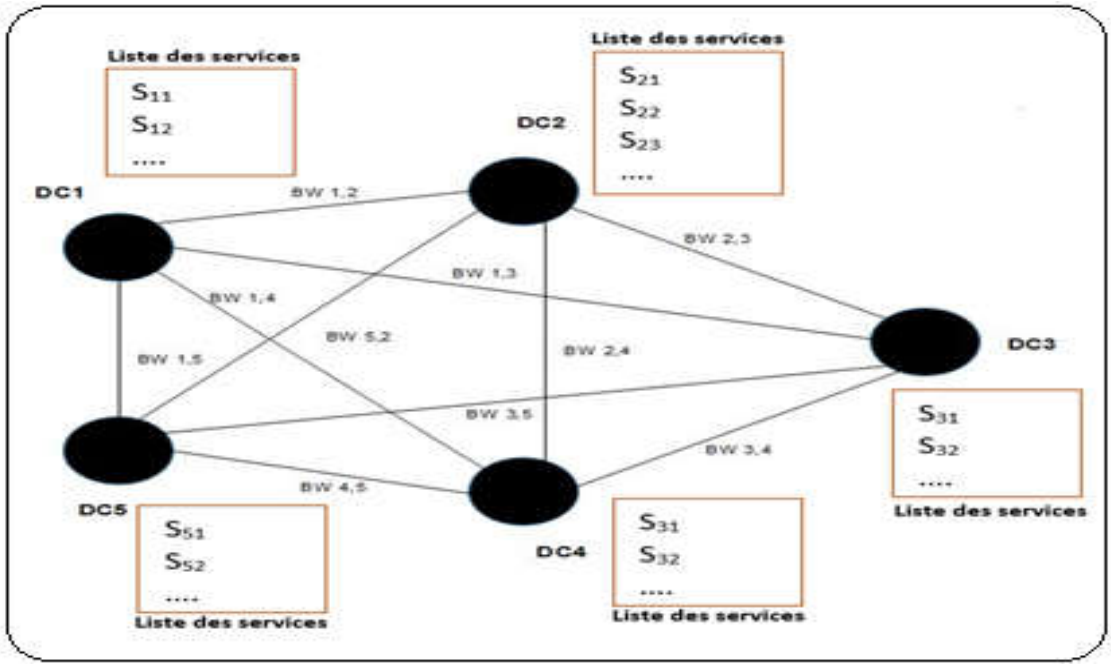


Figure 5-1Graphe de centres des données.

La bande passante entre un centre de données et un autre est illustrée par la matrice suivante :

	DC1	DC2	DC3	DC4	DC5
DC1	0	$bw_{1,2}$	$bw_{1,3}$	$bw_{1,4}$	$bw_{1,5}$
DC2	$bw_{2,1}$	0	$bw_{2,3}$	$bw_{2,4}$	$bw_{2,5}$
DC3	$bw_{3,1}$	$bw_{3,2}$	0	$bw_{3,4}$	$bw_{3,5}$
DC4	$bw_{4,1}$	$bw_{4,2}$	$bw_{4,3}$	0	$bw_{4,5}$
DC5	$bw_{5,1}$	$bw_{5,2}$	$bw_{5,3}$	$bw_{5,4}$	0

Tableau 5-1Modélisation de bande passante entre les centres de données

Lescharges de travaux appliquéessur les centres de donnéessont illustrées par la vecteur suivant :

Centres de données	DC1	DC2	DC3	DC4	DC5
Charge de travail	s_1	s_2	s_3	s_4	s_5

Tableau 5-2Charges destravaux appliqués aux centres de données.

5.1.2 La consommation d'énergie

L'énergie consommée pour les demandes de services sur un seul chemin p peut être notée comme $E_{cons}(\mathrm{p})$. Il est défini comme la consommation totale de bande passante et la consommation totale d'énergie du processeur appliquée aux centres de données pour les demandes de services sur le chemin p, on a :

$$f_{energy} = E_{cons}(p) = \sum_{e \in E(p)} E_{cons}(e_{i,j}) + \sum_{dc=V(p)} E_{cons}(dc_i) \quad (5.1)$$

Où$\mathrm{E(p)}$ et$\mathrm{V(p)}$ représentent l'ensemble de tous les liens et l'ensemble de tous les centres de données d'un chemin p (p ∈ Paths). Le niveau d'énergie actuel du centre de données dc_i ($dc_i \in DC$) d'un chemin p est indiqué par $E_{cons}(dc_i)$ et chaque centre de données est fourni avec la même énergie initialeE^0. La consommation électrique du data center est inspirée des travaux de Prosperi et al [77] et adapté comme suit :

$$E_{cons}(dc_i) = S_i E_a^{CPU}(dc_i) + (1 - S_i) E_s^{CPU}(dc_i) \quad (5.2)$$

Où

- E_a^{CPU}est la puissance consommée dans l'état actif d'un centre de données dc_i.
- E_s^{CPU}est la puissance consommée dans l'état de veille d'un centre de données dc_i.
- S_i est la charge de travail actuelle (utilisation) d'un centre de donnéesdc_i.

Le niveau de bande passante actuel du lien entre 2 centres de données i et j du chemin p est noté$E_{cons}(e_{i,j})$. La bande passante la consommation est inspirée aussi du travail deProsperi et al [77] et adaptée comme suit :

$$E_{cons}(e_{ij}) = \frac{bw_{ij}}{B} E_a^{BW}(e_{ji}) + (1 - \frac{bw_{ij}}{B}) E_s^{BW}(e_{ji}) \quad (5.3)$$

Où

- E_a^{BW} est la bande passante consommée entre deux centres de données i et j dans l'état actif.

- E_s^{BW} est la bande passante consommée entre deux centres de données i et j à la fois dans l'état de veille. bw_{ij} est la bande passante disponible dans une période de temps donnée.

- B est la bande passante maximale entre les centres de données.

5.1.3 La fonctionobjective QoS personnalisée

Les préférences de qualité du client sont des éléments clés pour évaluer les services Cloud disponibles. La qualité de chaque service Cloud est définie comme suit:

$$f_{QoS} = Score\ (cs_k) = \sum_{j=1}^{|QoS|} w_j * q_j / k \in [1, \ldots\ldots\ldots, n] \quad (5.4)$$

Où

- $|QoS|$désigne le nombre des attributs de QoS, $w_j$est le poids de j'ième attribut de QoS en fonction de son importance par le client calculé par Chouarfia et Yahlali[78].

- n est le nombre des services Cloud découverts.

Notre objectif de trouver un ensemble de services Cloud pertinents tels que (1) l'énergie totale est minimisée et (2) les préférences globales du client sont satisfaites dans un temps minimum (nombre minimum d'itérations). Cela peut être défini en utilisant la fonction objective suivante :

$$Fitness = \frac{w_{QoS} * f_{Qos}}{w_{energy} * f_{energy}} \quad (5.5)$$

Oùw_{energy} et w_{QoS} sont des poids associés respectivement à deux objectifs : énergie et QoS. Tous les deux sont définis entre [0, 1]. Plus la valeur de poids est élevée, plus c'est un critère important. La fonction objective est personnalisable et paramétrable en fonction des différents besoins des clients et du contexte d'exécution.

5.1.4 L'évolution dynamique des requêtes clients

La charge de travail normalisée S_i du centre de données dc_iest formellement définie comme suit:

$$S_i = \frac{\sum_{j=1}^{n} req_{ji} * s_{ji}}{S_{tot}} (5.6)$$

Où

- req_{ji} indique la requête de service d'utilisateur j présente ou pas sur un centre de données i.
- S_{ji} est l'utilisation du CPU d'une requête j dans un centre de données$i$.
- S_{tot} est l'utilisation totale du CPU de toutes les demandes.

5.1.5 L'évolution dynamique de la consommation de la bande passante

La consommation énergétique normalisée bw_{ik} entre deux centres de données dc_i et dc_k est formellement définie comme suit:

$$bw_{ik} = \frac{\sum_{j=1}^{n} req_{ji} * Q_j^{ik}}{B} (5.7)$$

Où

- req_{ji} indique la requête de service d'utilisateur j présente ou pas sur un centre de données$i$ par la bande k.
- Q_j^{ik} est l'utilisation du la bande d'un demande j sur un lien i, k
- Best l'utilisation totale de la bande passante de toutes les demandes.

5.1.6 Description détaillée de l'algorithme

L'objectif de l'algorithme proposé est d'obtenir le chemin optimal (liste des centres de données et ses services) qui minimise la consommation de l'énergie et satisfait les besoins du client tout en permettant la découverte des services souhaités hébergés dans les centres de données. Il permet aussi d'offrir au client un service optimisé qui respecte ses besoins et préférences. Nous avons choisi la méthode de colonie de fourmis pour la sélection de service selon les critères QoS et énergie et nous avons adopté le concept de l'ontologie comme structure de description sémantique des services Cloud pour le filtrage des services adéquats sémantiquement réponds aux critères QoS et énergie.Dans l'algorithme présenté, le processus de découverte de services dans le Cloud est modélisé en utilisant des colonies concurrentes afin d'exploiter la capacité des colonies à trouver des solutions optimales en assimilant la culture et les expériences sociales des colonies.

Au départ, une population initiale de colonies est générée et dispersée au hasard sur les fournisseurs de services, chaque fournisseur de services représente une solution pertinente au problème résolu et elle est évaluée selon une fonction de coût définie (*économie d'énergie, charge de traitement, faible bande passante, etc.*). Ensuite, les meilleurs fournisseurs de services Cloud à l'instant *t*sont sélectionnés. Le nombre de colonies assignées pour un meilleur fournisseur de Cloud est défini en fonction de ses qualités de service.La fonction d'utilité Uq de chaque fournisseur de Cloud CP_i (i = 1… N) est calculer comme suit:

$$U_q(CP_i) = \sum_{j=1}^{nb} \sum_{k=1}^{|QoS|} w_k * q_{j,k} \qquad (5.8)$$

Où

- $q_{j,k}$ sont les valeurs normalisées de la j^{ème} QoS pour un service *k* fourni par le fournisseur Cloud CPi.
- w_k est le poids du kth attribut QoS en fonction de son importance pour le client
- nb est le total des services Cloud fournis par le fournisseur de Cloud i.

La valeur d'utilité normalisée de chaque fournisseur Cloud est évaluée comme suit:

$$U'_q(CP_i) = \frac{U_q(CP_i) - min\,(U_q(CP_i))}{max\left(U_q(CP_i)\right) - min\,(U_q(CP_i))} \qquad (6.9)$$

Pour attribuer des fourmis à un fournisseur CloudCP_i, on va procéder comme suit:

- Etablir le vecteur W = [W_1, W_2…. W_N] où $W_i = \frac{U'_q(CP_n)}{\sum_{j=1}^{N} U'_q(CP_j)}$est le poids normalisé de chaque fournisseur Cloud n et and *U'q* sa valeur utilité normalisée.
- Le nombre de fourmis pour chaque fournisseur de Cloud correspond au nombre total de fourmis multiplié par son poids normalisé afin de mieux valoriser le meilleur fournisseur de Cloud.

Notre système donne tous les chemins possibles prouvant la qualité des services de Cloud. Le « courtier » de services Cloud sélectionne le chemin optimal. Nous utilisons différentes politiques (*économie d'énergie, profil d'utilisateur, faible bande passante, etc.*) et les historiques des colonies sur leurs précédents mouvements (*principe des phéromones laissées sur les piste*s). En ajoutant l'heuristique de la charge du centre de données, nous pouvons trouver une solution qui s'adapte aux changements des

besoins et préférences des utilisateurs. Nous exploitons la capacité des fourmis à réagir aux changements environnementaux et aux conditions inattendues. Ainsi, lorsqu'un chemin est obstrué par quelques obstacles, le processus de concours des colonies donne plus de chance de trouver d'autres solutions performantes parmi parcours possibles. Ainsi, utilisé cet algorithme guide les décisions futures et informe les autres fourmis du chemin à suivre ou à éviter en partageant les connaissances communes. L'utilisation des traces de phéromone permet d'exploiter l'expérience de recherche acquise par les fourmis et ainsi renforcer l'apprentissage pour la construction de solutions dans les itérations futures de l'algorithme. En même temps, nous utilisons des informations heuristiques (*débit intra-centre de données, charge CPU disponible appliquée sur les centres de données*) pour guider les fourmis dans leur découverte des services souhaités localisés dans les centres de données de l'espace de recherche par le client, puis ils sont filtrés par l'application de l'ontologie afin d'obtenir le chemin optimal qui minimise la consommation de l'énergie et tout en satisfaisant les besoins du client.

Nous proposons d'adapter l'algorithme de colonies de fourmis à notre problème pour la découverte des services souhaités par le client sur l'ensemble des centres de données en tenant compte du choix du chemin optimal qui consomme peu d'énergie (*Bande passante, CPU, RAM*) en se basant sur l'historique des fourmis lors de son déplacements (*les phéromones laissées sur les pistes*) puis nous ajoutons l'heuristique du débit intra-centre de donnée (*i.e. on peut étendre aussi aux émissions de CO2, la température, etc.***)** et l'heuristique de la charge appliquée sur le centre de donnée pour trouver une solution qui s'adapte aux différentes exigences du client tout en contournant les obstacles rencontrés sur les réseaux liés aux modifications de l'environnement. Ainsi, lorsqu'un ancien chemin est obstrué par un obstacle quelconque, les fourmis trouveront un autre chemin.

Nous présentons les principales étapes pour bien comprendre le fonctionnement de notre algorithme d'optimisation. Tout d'abord, un utilisateur soumet des requêtes avec ses contraintes via l'interface graphique du client mobile. Notre système prend cette requête en entrée, enrichi par des services spécifiques utilisant SPARQL basé sur notre ontologie générique et ontologie de domaine spécifique. Après cela, avec l'application de l'algorithme DSGreenACO qui prend en entrée la liste des services enrichis souhaités par le client, les fourmis sont lancés afin de découvrir les services Cloud pertinents sur des centres de données. Lorsque sa recherche est terminée, elle prend le chemin guidé par la phéromone; sinon, elle poursuivra sa découverte vers les centres de données non visités comme suit:

- Toutes les fourmis sont mises sur un centre de donné de manière aléatoire. Chaque fourmi a une mémoire qui stocke la solution partielle qu'elle a construit jusqu'ici (la mémoire contient seulement le centre de donné du début). À partir du centre de donné initial, une fourmi se déplace itérativement d'un centre de donné vers un autre avec une règle de probabilité. Étant donné un centre de donné "i" une fourmi "k" choisit d'aller au centre de donné "j" avec une probabilité donnée près :

$$P_{i,j} = \frac{[ph_{i,j}]^{\alpha} * [BP_{i,j}]^{\beta} * [S_j]^{\delta}}{\sum_{j=1}^{N} [ph_{i,j}]^{\alpha} * [BP_{i,j}]^{\beta} * [S_j]^{\delta}} \qquad (5.10)$$

Où :

- $Ph_{i,j}$: est la quantité de la phéromone sur le chemin entre le centre de donné i et le centre de donné j. Les fourmis estiment le temps lié au débit réseau et charge serveur qu'elles mettent pour aller d'un centre i à un autre j et marquent ensuite à l'aide de phéromones, les chemins qu'elles viennent d'emprunter. Plus le délai est court, plus l'intensité du marquage est importante. La densité d'abaissement de phéromone s'adapte en permanence et de manière totalement décentralisée au changement de la bande passante et la charge courant de datacenter

- $BP_{i,j}$: est l'information heuristique a priori disponible (le débit inter-centre de données sur le chemin entre le centre de donné i et le centre de donné j). Cette information est exprimée comme un intervalle correspondant à une valeur logique de QoS ;

  - Bp = Haut si disponibilité est dans l'intervalle [0.7*B, B].

  - Bp = Moyen si disponibilité est dans l'intervalle [0.3*B, 0.7*B [.

  - Bp = Faible si disponibilité est dans l'intervalle [0, 0.3*B [.

- Tel que B est la bande passante maximale entre les centres de données).

- S: est la charge de travail appliquée sur le centre de donné j à l'instant t exprimée comme un seuil entre 0 et 1. Le seuil est calculé par rapport à la charge nécessaire au traitement des demandes de services à l'instant t du Datacenter j et les ressources disponibles (i.e. CPU, ressources de stockage disponible, RAM disponible). Un datacenter est chargé si le rapport atteint un seuil de 1.

- N: est l'ensemble de centre de données que la fourmi k n'a pas encore visité. Cet ensemble forme le voisinage potentiel de la fourmi k.

- α, β et δ sont des paramètres qui déterminent le degré d'influence de la densité de phéromone, l'information heuristique et la disponibilité du service sur le choix de prochain centre de données. Les valeurs sont comprises entre 0 et 1. Si l'une de ces valeurs est égale à zéro, cela signifie que l'heuristique de cette valeur est négligée lors du choix du chemin lors de la découverte des services.

Lorsqu'une fourmi termine la construction d'un chemin complet (*de longueur n*), on passe à la phase de mise à jour de la phéromone. Cette dernière se décompose en deux parties, la première consiste à abaisser la densité de phéromone (*c'est l'évaporation de la phéromone*), la deuxième permet à chaque fourmi de déposer la phéromone sur les arcs qui appartiennent à son visite par :

$$ph_{i,j} = (1-\rho) * ph_{i,j} + \sum_{k=1}^{m} \Delta ph_{i,j}^{k} \quad \forall (i,j) \qquad (5.11)$$

Où :

- 0 < ρ < 1 est le taux d'évaporation de la phéromone. Le paramètre ρ est employé pour éviter l'accumulation illimitée de phéromone, et permet à l'algorithme d'oublier les mauvaises décisions faites précédemment. Avec ce mécanisme la force de phéromone associée aux arcs qui ne sont pas choisis par les fourmis, sera diminuée exponentiellement avec le nombre d'itérations.

- m : est le nombre de fourmis.

- $\Delta ph_{i,j}^{k}$: est la quantité de dépôts de la fourmi k de phéromone sur les arcs.

Elle est définie par :

$$\Delta ph_{i,j}^{k} = \begin{cases} 1/L_k & If \ \ phoromone \ \ k \ travels \ \ on \ path \ (i,j) \\ 0 & otherwise \end{cases} \qquad (5.12)$$

Où : L_k est la longueur de chemin visité par la fourmi k.

Nous remarquons que plus que la visite de la fourmi est courte, plus la phéromone reçue par les chemins de la visite est importante, favorisant les chemins courts (ou voisinage proches). En général, les chemins qui sont employés par beaucoup de fourmis et qui sont contenus dans des visites plus courtes recevront plus de

phéromone et auront plus de chance d'être choisis dans les futures itérations de l'algorithme. Ensuite la fourmi met à jour (meilleur chemin qui contient la liste des services trouvés et ses centres de données pertinents) la solution qu'elle a construite si elle est meilleure que la précédente. A la fin de toutes les itérations les résultats obtenus sont : (1) - le meilleur chemin, (2) - le temps consacré à trouver ce chemin. Le chemin trouvé par DSGreenACO est ensuite filtré par l'application de l'ontologie selon les exigences du client, affichage des centres de données qui satisfait le besoin du client et en cas de changement du contexte (mauvaise qualité de service, problème réseau) relancer DSGreenACO.

Les chemins découverts sont filtrés en interrogeant notre ontologie à l'aide de règles SPARQL (Simple Protocol and Resource Description Framework). Par exemple, un client est soumis à des préférences de qualité de service et de coût, dans lesquelles le prix du service doit être bon marché et répondre à un service rapide et sécurisé. De nombreuses approches [79][80] ont été présentées dans la littérature. Nous utilisons SPARQL comme langage de recommandation du W3C pour filtrer les services pertinents en fonction des préférences de l'utilisateur. Le filtre est implémenté sous forme de requête basée sur un graphe sur plusieurs opérateurs SPARQL prédéfinis (Filtre, Option) selon que les valeurs logiques QoS sont référencées dans leurs termes sémantiques. Ainsi, pour chaque terme et sa valeur logique QoS correspondante (par exemple, le temps «lent» correspond à la valeur logique «basse»).

En cas de changement du contexte (changement des préférences de l'utilisateur ou d'un problème de connectivité réseau), il sera nécessaire de redémarrer DSGreenACO.L'utilisation de traces de phéromones permet d'exploiter l'expérience acquise par les fourmis et renforce ainsi l'apprentissage pour la construction de bonnes solutions dans les futures itérations de l'algorithme. En même temps, nous avons proposé des informations heuristiques (flux entre centres de données, charge de travail du centre de données) pour guider les fourmis dans leur processus de découverte des services proposés situés dans le centre de données de l'espace de recherche par le client; et ensuite filtré en appliquant l'ontologie pour obtenir le chemin optimal minimisant la consommation d'énergie et les qualités de service afin de satisfaire les besoins du client.

Les étapes de l'algorithme sont présentées dans (Figure 5-2)

Inputs:
- Besoins fonctionnels et préférences contextuelles du client.
- Registre des services Cloud sémantiques.
- Ontologie générique (O_1) et ontologie de domaine (O_2).

Outputs :
- Services optimaux
- Chemin optimal avec temps de découverte de chemin.

1.Enrichissement sémantique des besoins du client (O_1, O_2)

2. Initialisation : Les fourmis sont placées aléatoirement dans des fournisseurs de services. Chaque fourmi a une mémoire qui stocke la solution partielle qu'elle a construit (la mémoire contient seulement le datacenter du début). Initialisation de nombre des itérations par Max_itérations et le coefficient représentant l'évaporation des traces de phéromones.

3. Dépôt des fourmis sur un fournisseur Cloud CP_i selon l'équation6.9

4. Itération

Pour itération =1 jusqu'à Max_itérations **faire**

Pour k =1 jusqu'à M**faire**

- Une fourmi se déplace itérativement d'un centre de donné vers un autre avec une règle de probabilité $p_{ij}(t)$
- Calculer le cout de chemin

FinPour

Mettre à jour la solution optimale si elle est mieux que la solution précédente.

Mettre à jour de la phéromone par les fourmis et les par l'évaporation

FinPour

5. Filtrage sémantique des services souhaités par le client

Figure 5-2L'algorithme DSGreenACO.

5.2 Algorithme de composition sémantique et dynamique des services Cloud

L'algorithme de composition de services est basé sur la sélection par domaine pour une optimisation globale. Ce type d'algorithmes présente l'inconvénient du non-respect des contraintes globales dans la sélection par domaine. Pour pallier cet inconvénient, nous sommes basés sur l'historique d'invocations des services. L'algorithme de composition des services est basé sur l'algorithme de sélection DSGreenACO des services par domaine. L'algorithme est très utile car les services sont distribués dans un environnement à grande échelle, ainsi les services peuvent être changéséventuellement. Le système a donc besoin d'une méthode qui peut réagir aux différentes contraintes. De plus,il offre au client un service composite combinant une confiance basée sur des services intelligents et des économies d'énergie propres à chaque domaine, qui répondent à ses préférences.

La Figure 5-3présente l'architecture générale du système de composition des services Cloud. Le client envoie sa requête via un terminal au système. Le broker décompose la

requête en plusieurs sous-requêtes par domaine et les envoie au proxy approprié. Chaque proxylance l'algorithme DSGreenACO pour sélectionner les services du domaine approprié afin de les renvoyer au Broker. Enfin, le broker lance l'algorithme d'assemblage des servicessélectionnésbasant sur le principe de confiance.

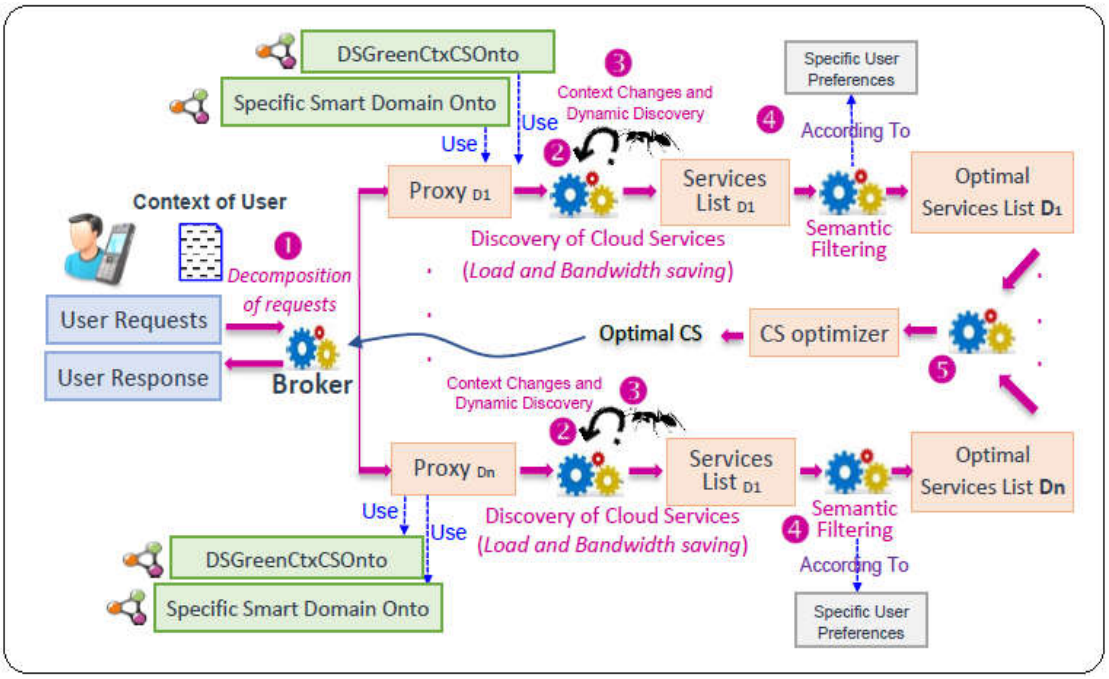


Figure 5-3Modèle fonctionnel de l'approche DSGreenACO.

Le composant Broker classifie sémantiquement les services Cloud par domaine selon leur spécialité et leur QoS. Par exemple, un service hôtel, un service de restauration, des services transports…etc. Chaque domaine est géré par un proxy qui privilège les préférences des clients en minimisant le coût de l'énergie.

L'algorithme de composition de services est basé sur la sélection par domaine pour une optimisation globale. Ce type d'algorithmes présente l'inconvénient du non-respect des contraintes globales dans la sélection locale. Pour pallier cet inconvénient, nous sommes basés sur l'historique d'invocations du service. L'algorithme de composition des services est basé sur l'algorithme de sélection DSGreenACO des services par domaine. L'algorithme est très utile car les services sont distribués dans un environnement à grande échelle, ainsi les services peuvent être changés éventuellement. Le système a donc besoin d'une méthode qui peut réagir aux différentes contraintes. De plus, il offre au client un service composite combinant une confiance basée sur des services intelligents et des économies d'énergie propres à chaque domaine, qui répondent à ses préférences. Le Broker reçoit la requête du client, avec l'application de notre ontologie générique et de l'ontologie du domaine pour

enrichir la requête. Ensuite il décompose la requête en paire de service (S_i, S_j) selon les domaines existants. Par exemple: une demande de services (smart-weather, smart-transport, smart-hotel) se décompose en (smart-weather, smart-transport), (smart-transport, smart-hotel), (smart-weather, smart-hotel). Donc il produit C_N^2 paires de requête. Ensuite, chaque paire est envoyée à deux proxys adéquats. Chaque proxy sélectionne la solution selon l'algorithme DSGreenACO. Ensuite il renvoie une réponse sous forme de tableau contenant tous les services S_j qui coopèrent avec le service S_i, avec leur facteur de confiance C_j. Les solutions ($2 * C_N^2$) sont cumulées dans un tableau dès leur réception par le broker, comme illustré dans leTableau 5-3.

Smart-domain	Coopération entre deux pairs des services		Poids de confiance
Proxy- Weather	$S_{Weather, 1}$	$S_{Hotel, 1}$	$Wweather_{11}$
	$S_{Weather, 2}$	$S_{flight, 2}$	$Wweather_{12}$
Proxy- Hotel	$S_{Hotel, 1}$	$S_{flight, 1}$	$WHotel_{11}$
	$S_{Hotel, 2}$	$S_{Weather, 2}$	$WHotel_{22}$
Proxy- flight	$S_{flight, 1}$	$S_{Weather, 1}$	$WFlight_{11}$
	$S_{flight, 2}$	$S_{Hotel, 2}$	$WFlight_{12}$

Tableau 5-3Scores de confiance entre les services coopérants

Le facteur de confiance est calculé selon l'historique du service. Pour chaque domaine I ∈ [1… nb], et pour tous les services des domaines cités dans le tableau, on calcule le facteur de confiance selon la formule suivante :

$$Score(s_{D_i}) = Nb\_occ(s_{D_i}) * Max\ (W_{S_{ij}})\ . \qquad (6.13)$$

Avec :

- I ∈ [1... nb] ou *nb* est le nombre de smart domaines.
- $Nb\_occ(S_{D_i})$ est le nombre des occurrences du service Di dans le tableau.
- $\mathrm{Max}\ (W_{S_{ij}})$ est le max de valeur de confidence assignée par les différents proxys pour le service "i".

Enfin, il sélectionne pour chaque domaine intelligent, le service qui a le score le plus élevé pour reconstruire la réponse. Une fois la réponse est renvoyée, il demande à chaque proxy d'augmenter le facteur de confiance pour les services sélectionnés. Si un nouveau service est inscrit pour la première fois, il sera ajouté à la liste des services coopérants avec une valeur moyenne de facteur de confiance. Au cours du temps, le

poids des services qui coopèrent aux requêtes des clients augmente. Pour qu'un nouveau serviceentre en concurrence avec les services ayant un facteur de confiance élevé, il doit minimiser le taux de consommation de l'énergie.

5.3 Réalisation de prototype et évaluation

Afin de justifier et de valider nos propositions présentées dans les sections précédentes, nousavonsdéveloppéunprototypequipermetde sélectionner des services sensiblesau contexte.Ilcontientplusieursinterfacesutilisateurafindepermettreauclientdespécifierses besoinsfonctionnelsetses préférences.Leprototypefournitdes services Cloud adéquatsvial'algorithme DSGreenACO.Ces servicessontoptimauxen matière de QoS et énergie consommée selon les besoinsfonctionnelset les préférences spécifiées par leclient. L'implémentation de prototype est faite un ordinateur avec d'un processeur Intel (R) I5 de 2.5Ghz, d'un RAM de 4Go et un disque dur de 0.5T, qui fonctionne sous Microsoft Windows 7. Nous avons choisi le langage JAVAcomme logiciel de développement sous l'environnement NetBeans IDE 8.2.

5.3.1 Présentation deprototype

Leprototyperéalisécontientlesmodulesduprocessusdesélectionsémantiqueet dynamique des servicesCloudafind'obtenir un servicepersonnalisésensibleaucontexte.Ilcontientégalement des interfaces utilisateur pour permettre au client de spécifier ses besoins fonctionnels et ses préférences. Les fonctionnalités principales de notre prototype sont:

- **Gestion des services Cloud :** la liste des services Cloud concrets est enregistrée dans un registre du proxy. Un service virtuel est associé à un ensemble de services concrets avec les mêmes fonctionnalités. Le gestionnaire de service met continuellement à jour le registre du proxy chaque fois qu'un service Cloud est modifié.
- **Sélection dynamique des services Cloud :** une fois que les besoins fonctionnels et les préférences du client sont spécifiés, le système sélectionne les services Cloud appropriés.
- **Optimisation des QoS et de l'énergie :** le système fournit un service concret qui respecte les besoins fonctionnels et les préférences du client via la sélection sémantique et dynamique des services Cloud concrets en optimisant les QoS et la consommation de l'énergie.

5.3.2 Principe de fonctionnement

Pour bien comprendre le fonctionnement de notre algorithme bio-inspiré DSGreenACO qui sert à optimiser l'énergie et le temps de la découverte des services dans les centres de données. Le prototype doitsélectionnerdesservices Cloud en optimisant les QoS selon la spécification des besoins fonctionnels et des préférences QoS du client. On va présenter les étapes de fonctionnement. À l'apparition de la fenêtre principale, le client spécifie ses besoins fonctionnels et les valeurs des préférences de QoS (temps, prix, fiabilité, disponibilité et sécurité) à l'aide d'une interface utilisateur comme illustré dans laFigure 5-4.puis validé.

Figure 5-4Interface pour la spécification des préférences de QoS.

Ensuite, les paramètres du problème sont spécifiés par l'utilisateur (nombre des centres de données, nombre des itérations et le taux de l'évaporation) et les paramètres heuristiques (l'heuristique de la phéromone, l'heuristique du débit intra-centre de données et l'heuristique de la charge appliquée sur le centre de données) comme le montre laFigure 5-5.

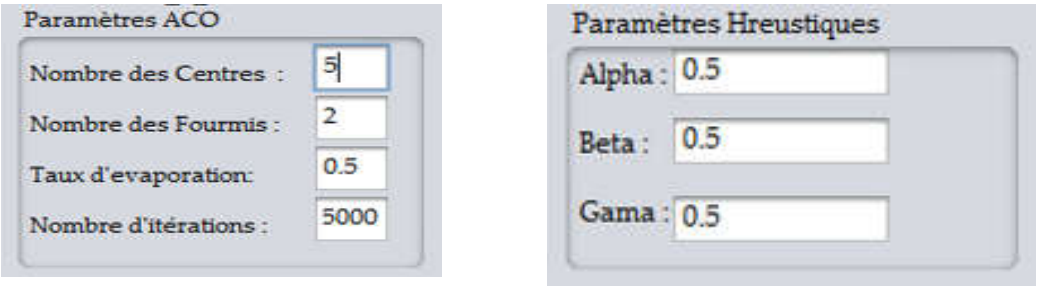


Figure 5-5Spécification des paramètres du problème et des paramètres heuristiques.

Pour afficher le graphe des centres de données, le client doit cliquer sur le bouton « afficher graphe des centres de données » comme indiqué sur laFigure 5-6. Le graphe s'affiche au milieu de la fenêtre principale sous forme d'un graphe complet ou les sommets représentent les centres de données et les arcs représentent les chemins

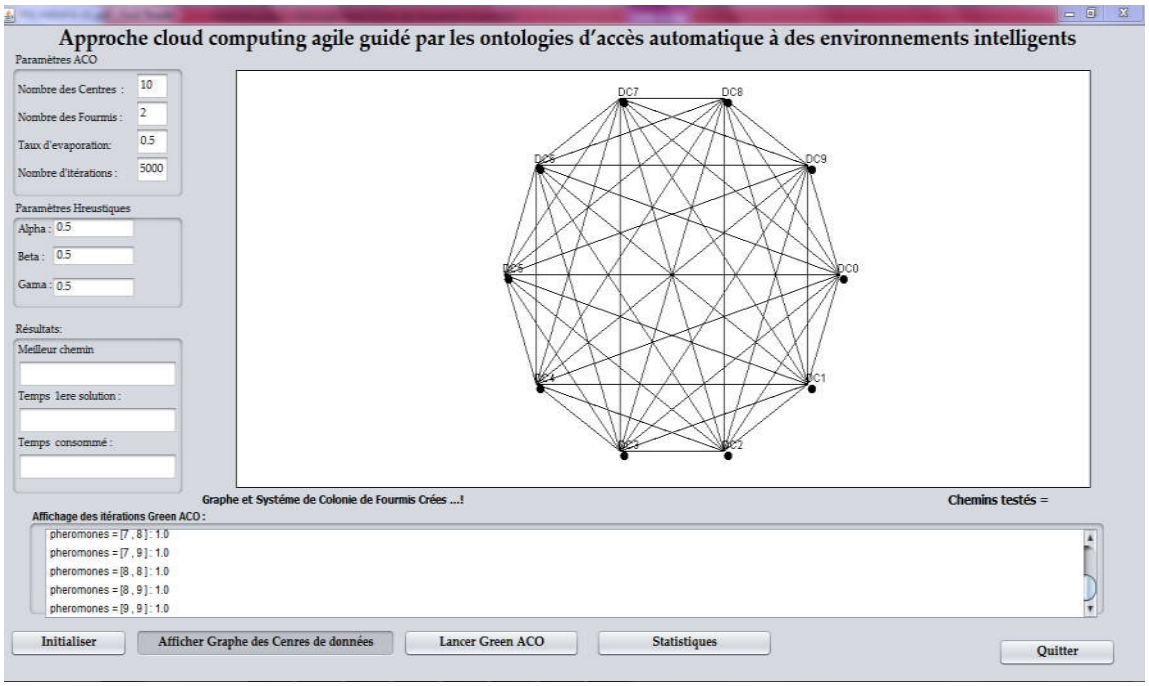


Figure 5-6Affichage de graphe des centres des données.

Enfin, pour obtenir la solution optimale finale, le client doit spécifier l'ordre de ses préférences comme indiqué sur laFigure 5-7. En effet, le système calcule les poids des attributs de QoS etsélectionnelasolutionoptimaleayantlemeilleurscoreparmi l'ensemble des solutions disponibles.

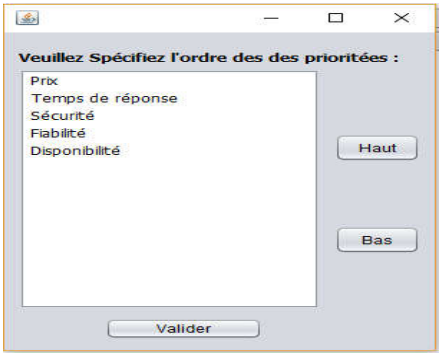


Figure 5-7Spécification de l'ordre des préférences.

Plusieurs scénarios de découverte et sélection des services liés aux différentes contraintes d'un client (se*rvices touristiques, services de santé, services de transport, etc*.) ont été testés. Pour sélectionner le meilleur chemin qui minimise la consommation d'énergie et maximise la QoS en fonction des préférences (*prix, sécurité, fiabilité, temps de réponse, etc*.), le client spécifie ses préférences visuellement à travers une interface utilisateur traduit vers des règles logiques des attributs de QoS et ses priorités via des appareils mobiles (tablette, smartphone, etc.). Dans le premier scénario, un client a besoin de services liés aux transports. L'objectif est de permettre d'offrir aux clients des services de transport avec un faible coût avec l'utilisation des nouveaux systèmes de transport distribués des diverses technologies renouvelables.

Étape 1: Description et enrichissement sémantique des besoins du client

Avec DSGreenACO, il est possible d'appliquer notre ontologie pour le filtrage sémantique des services de transports et classifiée par consommation énergétique à un certain moment de la journée. DSGreenACO prend la requête client de "*recherche des services de transport*", enrichi par des services sémantiquement proches de la requête client de l'ontologie de domaine de "*transport*". Avec l'application de notre ontologie générique et l'ontologie du domaine, notre système est en mesure d'enrichir la requête client avec des services de transport de diverses technologies renouvelables (*i.e. véhicules électriques*) et de filtrer les centres de données qui contiennent les services pertinents par une simple requête SPARQL.

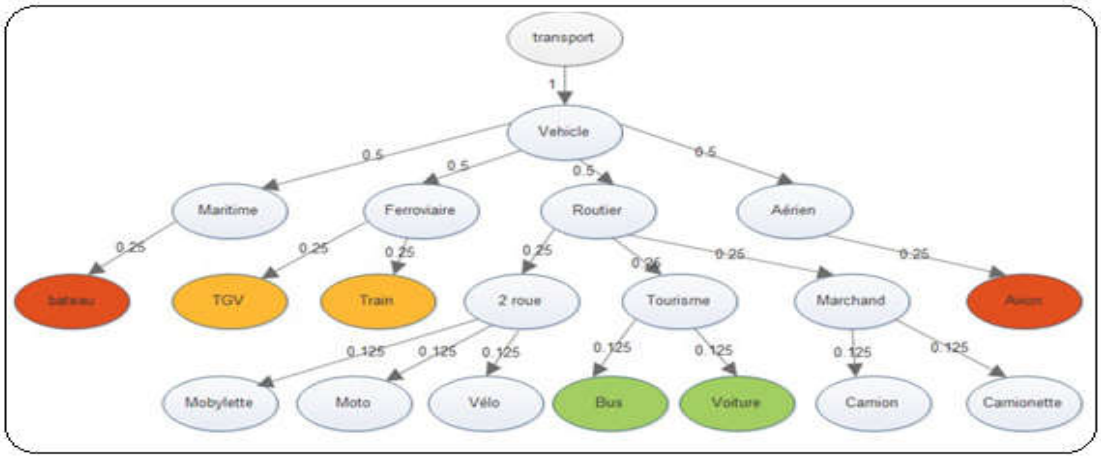


Figure 5-8Exemple d'ontologie de domaine de transport.

Étape 2 : Lancement de DSGreenACO

Après le filtrage des centres de données, notre système crée un graphe des centres de données trouvés hébergeant de tels services ainsi que la charge appliquée sur les

sommets et le débit intra-centre. Le processus DSGreenACO retourne le chemin optimal qui minimise l'énergie (*Bande passante, CPU, RAM*) et le coût (*prix*) pour parcourir les centres de données qui offrent le service souhaité par le client dans temps court.

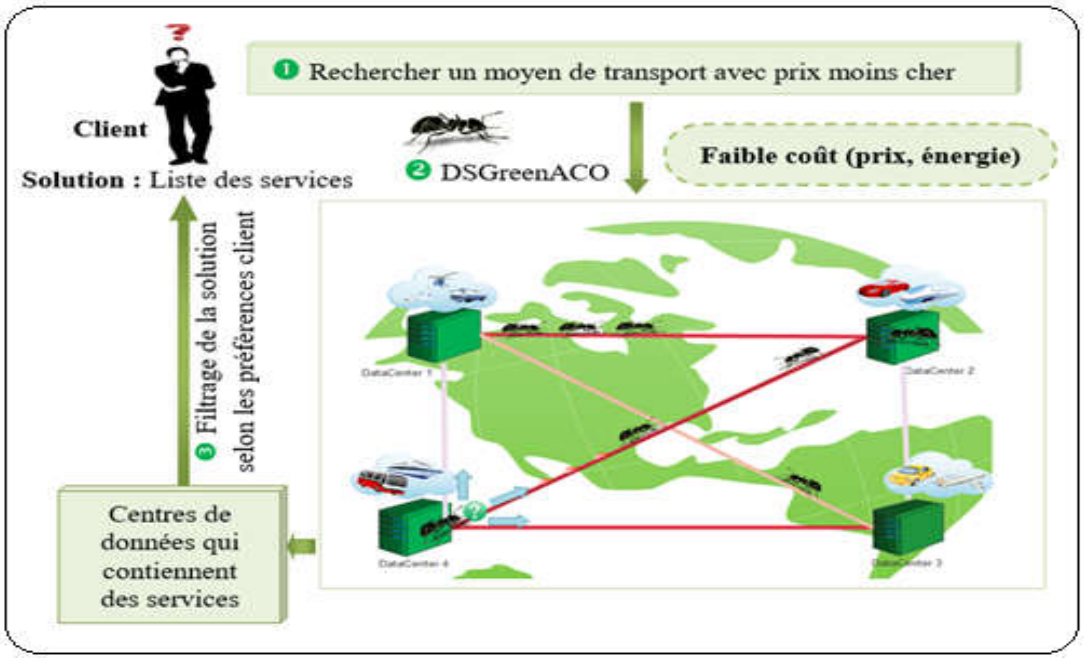


Figure 5-9Sélection des services de transport selon les préférences de client

Àla fin de l'itération l'algorithme se termine et affiche une fenêtre contenait le message suivant : GreenACO terminé avec succès ! Les résultats de la découverte sont affichés comme le montre laFigure 5-10.

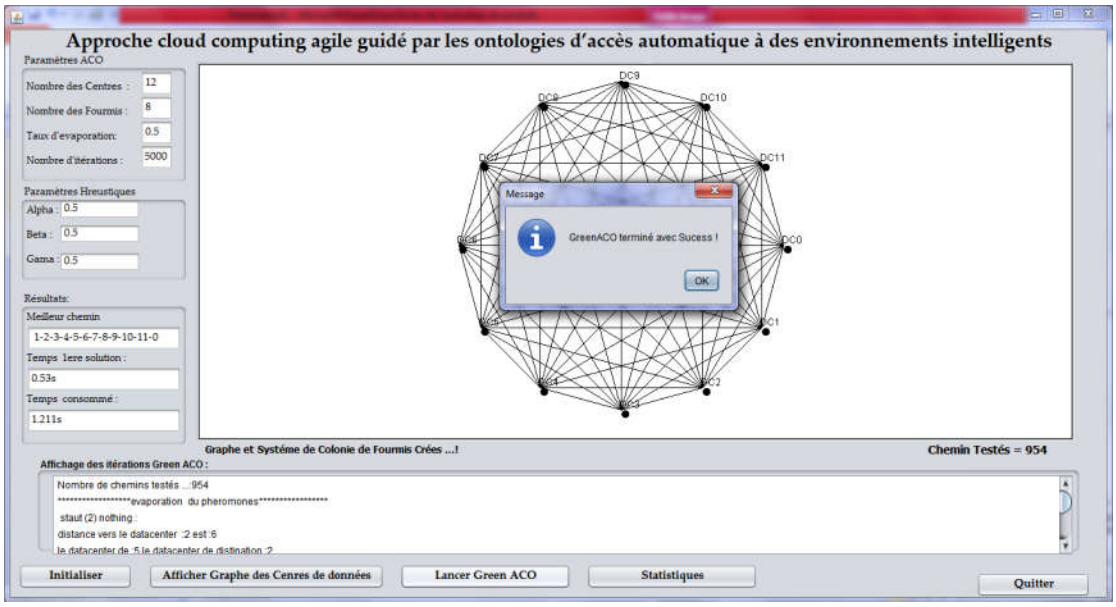


Figure 5-10Sélection des services de transport selon les préférences de client

Étape 3 : Filtrage sémantique des services souhaités par le client

En appliquant l'ontologie, les services trouvés sont filtrés selon les préférences du client (*leprix est moins cher*) pour identifier les centres de données qui offrent le service souhaité par le client.

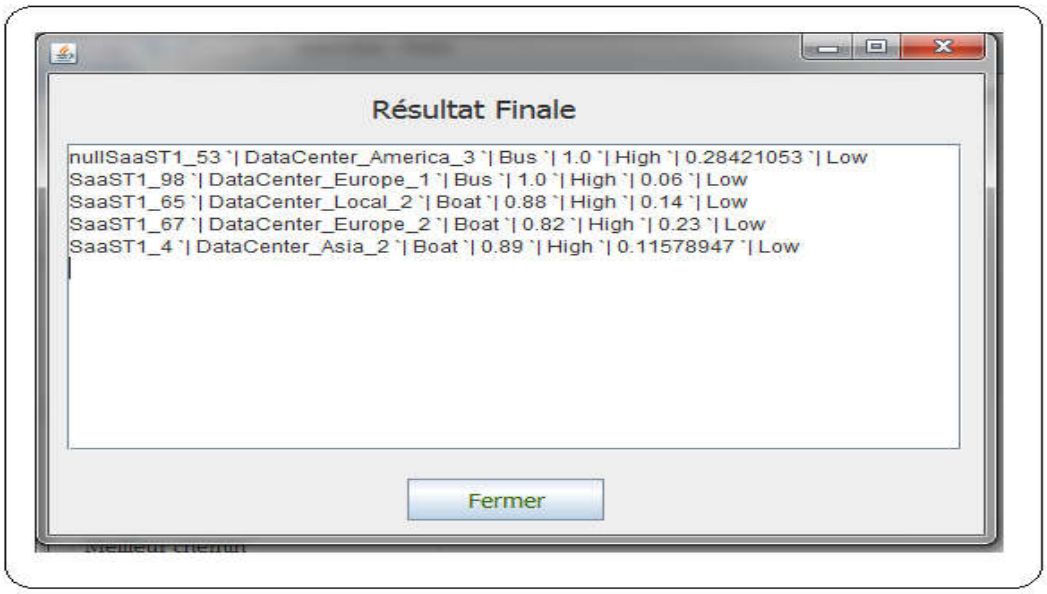


Figure 5-11La solution optimale trouvé par notre prototype.

Dans le premier scénario, Le client a la possibilité d'avoir des services optimaux basés sur des changements des préférences contextuels, en ajoutant d'autres préférences de service (*par exemple, les services de transport ordonnés par prix et sécurisés sont demandés*). Les fourmis vont découvrir les services de transport sur le meilleur chemin des centres de données qui minimise l'énergie (*bande passante, CPU, RAM*). DSGreenACO renvoie les services dont le prix est le plus bas tout en garantissant une certaine sécurité.

5.4 Evaluation des performances et comparaisons

Aprèsavoirprésentél'implémentationdenotresystèmeetsonapplication,nousallonsévaluer notreapprocheparrapportauxsolutionsexistantesdesdifférents auteurs,etnousallonsprésenterdescomparaisonsaveccessolutionsselondifférentscritères .

Le premier ensemble de données est un ensemble de données réelles collectées par Zheng et al. [81]. Dans lequel, environ 5825 services Cloud ont été fournis par différents centres de données. Actuellement, il aété appliqué dans de nombreuses études. Dans cet ensemble de données, et pourchaque service, les valeurs des temps de réponse et du débit nécessaire varient enfonction de ses appels par les différents

utilisateurs. Le second ensemble de données est généré aléatoirement par un algorithme implanté en JAVA.

5.4.1 Ensemble des données réel

Dans le tableau suivant une partie de l'ensemble recueilli par Zheng, en réalité c'est une matricede 339 × 5825 pour le temps de réponse. Chaque entrée dans une matrice représente le temps de réponse en seconde observé par un utilisateur pour un service Cloud. Cette expérience a lieu en août 2009. Ou 5, 1 974 675 invocations de services Cloud du monde réel sont exécutées par 339 utilisateurs de services de 30 pays sur 5 825 services Cloud du monde réel de 73 pays participant à cette expérience.Le tableau suivant montre les valeurs des attributs de QoS, pour chaque service concrets, qui sont introduits dans l'ontologie.

S. Sémantique	S.Concret	Temps	Prix	Fiabilité	Disponbilité	Sécurité	Energie	Débit
SV1	sc1, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc1, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc1, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc1, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc1, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV2	sc2, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc2, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc2, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc2, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc2, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV3	sc3, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc3, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc3, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc3, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc3, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV4	sc4, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc4, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc4, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc4, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc4, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV5	sc5, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc5, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc5, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc5, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc5, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV6	sc6, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc6, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc6, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc6, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc6, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV7	sc7, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc7, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc7, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc7, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc7, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV8	sc8, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc8, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc8, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc8, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc8, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV9	sc9, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc9, 2	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc9, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc9, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc9, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836
SV10	sc10, 1	5.982	0.228	0.237	0.221	0.222	0.527	0.453
	sc10, 1	2.13	0.262	0.273	0.251	0.254	0.427	0.652
	sc10, 3	0.854	0.366	0.376	0.357	0.358	0.115	0.649
	sc10, 4	0.693	0.226	0.233	0.22	0.219	0.344	0.765
	sc10, 5	0.866	0.227	0.233	0.22	0.221	0.366	0.836

Tableau 5-4 une partie de l'expérience de Zheng

5.4.2 Comparaison de temps d'exécution

Cette expérimentation vise à valider l'efficacité et la performance de notre approche pour résoudre les contraintes de l'utilisateur et ses préférences où le temps d'exécution pour trouver le chemin optimal et les métriques de consommation d'énergie sont pris en compte pour comparer efficacement notre algorithme avec d'autre travail similaire

[82]. Comme le montre laTemps de réponse dans l'ensemble de données réelavec le nombre croissant de services disponibles sur le Cloud, DSGreenACO offre de meilleurs résultats comparés au bénéfice du temps de réponse de l'algorithme génétique NSGAII. C'est la conséquence d'une approche guidée par l'ontologie pour la découverte intelligente de services de Cloud de qualité sur un graphique de centres de données pertinents, y compris diverses politiques d'optimisation (économie d'énergie, charge du processeur, faible bande passante).

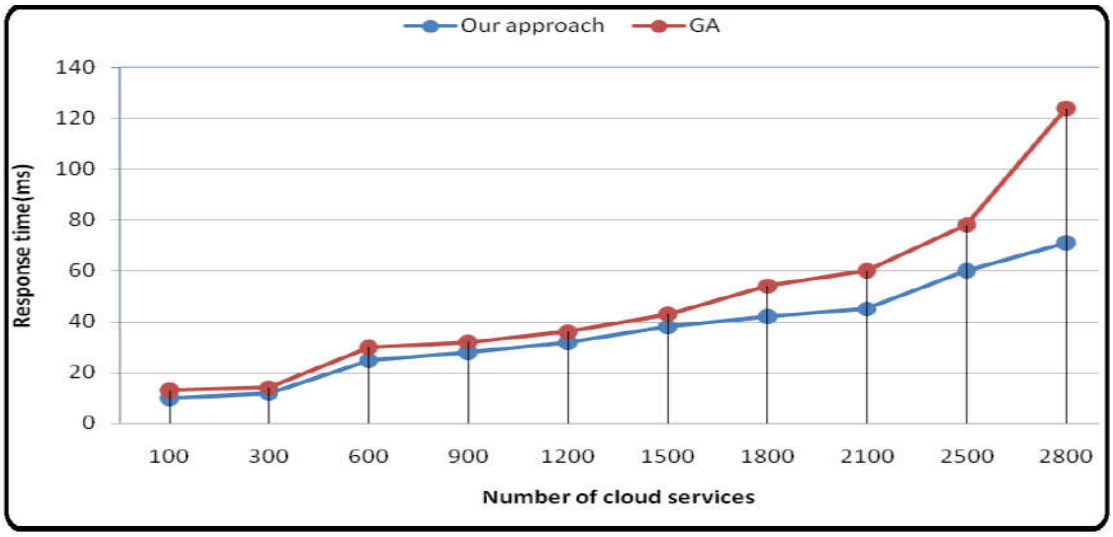


Figure 5-12Temps de réponse dans l'ensemble de données réel

5.4.3 Comparaison de la consommation de l'énergie

Dans la Consommation de l'énergie dans l'ensemble de données réel, lorsque le nombre d'utilisateurs augmente, la consommationd'énergie de DSGreenACO est meilleure que la consommation d'énergie del'algorithme génétique. On constate aussi que lorsque le nombre d'utilisateurs estinférieur à 25, les solutions trouvées par DSGreenACO sont un peu moins bonnesque l'algorithme génétique qui correspond à un nombre faible et limité d'utilisateurs,cependant, lorsque le nombre d'utilisateurs est supérieur à 25, les solutions trouvéespar notre algorithme sont meilleures que la solution trouvée par algorithmegénétique. Pour l'optimisation de l'énergie, avec le nombre variable d'utilisateurs,notre approche réalise un bon compromis entre le temps de réponse et les critèresénergétiques.

On peut conclure que DSGreenACO offre un bon compromis entre letemps de réponse et les critères énergétiques.

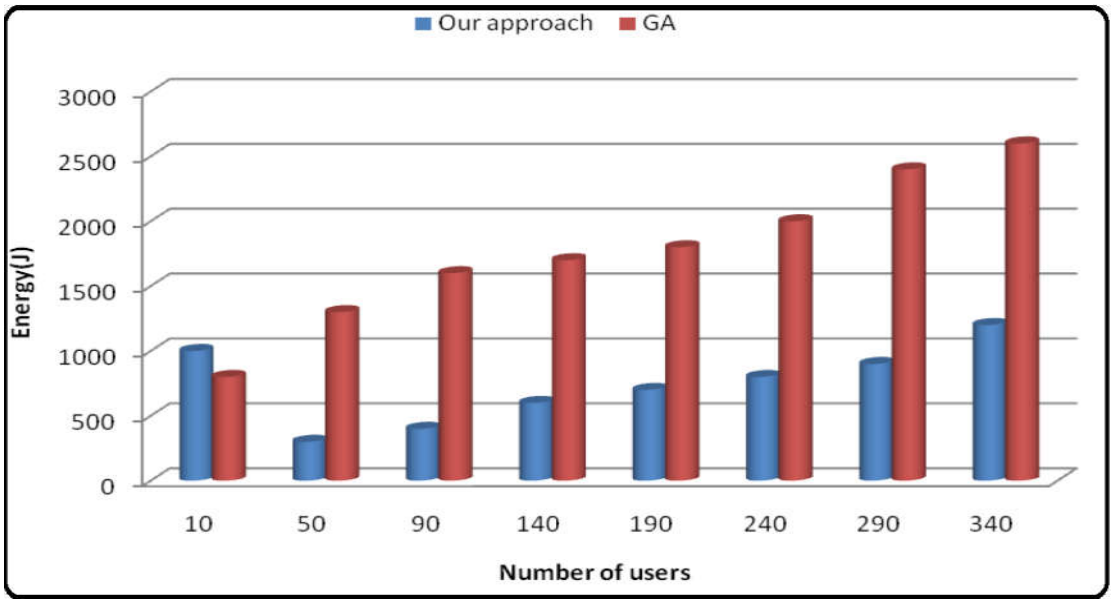


Figure 5-13Consommation de l'énergie dans l'ensemble de données réel

5.4.4 Ensemble des données aléatoire

Le second ensemble de données est aléatoirement généré. Elle consiste à simuler 100 000 services Cloud qui représentent des services de Cloud génériques, classé dans 50 domaines. Ils sont classés par domaine (santé, transport et tourisme) et pour chaque service générique, un ensemble de2000 services candidat associés des services spécifiques, avec différentes valeurs de QoS normalisées. Par souci de simplicité, chaque service prend en compte deux attributs QoS : le prix et le temps de réponse en tant qu'attribut de QoS. Il est facile d'ajouter ou de remplacer ces attributs afin d'étendre le modèle de QoS sans modifier l'algorithme d'optimisation de service proposé.

5.4.5 Comparaison de temps d'exécution

La Figure 5-12 montre la comparaison du temps de réponse pour les deux algorithmes comparés avec le nombre croissant de services concrets.Le temps de réponse de DSGreenACO est évidemment meilleur que le temps de réponse de l'algorithme génétique NSGA-II dans tous les cas. Ceci est dû au fait que l'algorithme stochastique NSGAII est contrôlé seulement par certaines politiques régissant le processus d'optimisation tandis que notre algorithme combine les politiques d'optimisation et les règles guidées par l'ontologie qui bénéficient des anciennes expériences.

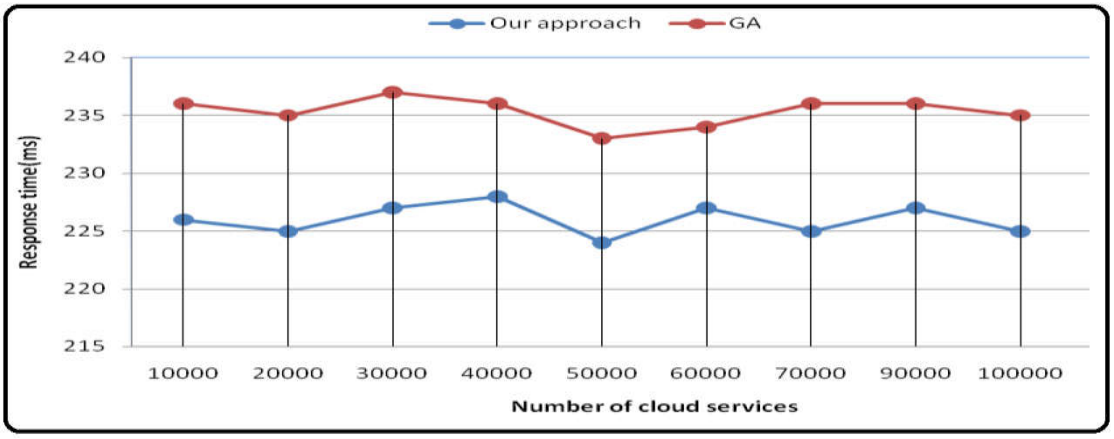


Figure 5-14Temps de réponse dans l'ensemble de donnéesgénéréaléatoirement.

5.4.6 Comparaison de la consommation de l'énergie

Dans la Consommation de l'énergie dans l'ensemble généréaléatoirement, nous pouvons également constater que la consommation d'énergie de DSGreenACO est minimale. Dans l'ensemble de données aléatoires, lorsque l'on augmente le nombre d'utilisateurs, la sélection de services obtenus par notre DSGreenACO a une optimalité (c'est-à-dire le gain de temps de réponse et la consommation d'énergie). De plus, on observe que la consommation d'énergie de DSGreenACO est toujours la plus faible dans tous les cas. Donc, on peut conclure que notre algorithme fonctionne mieux que l'algorithme génétique en termes d'optimalité. Cette optimalité résulte de l'application directe de politiques d'optimisation énergétiques et de la découverte intelligente de services de Cloud basés sur les centres de données pertinents.

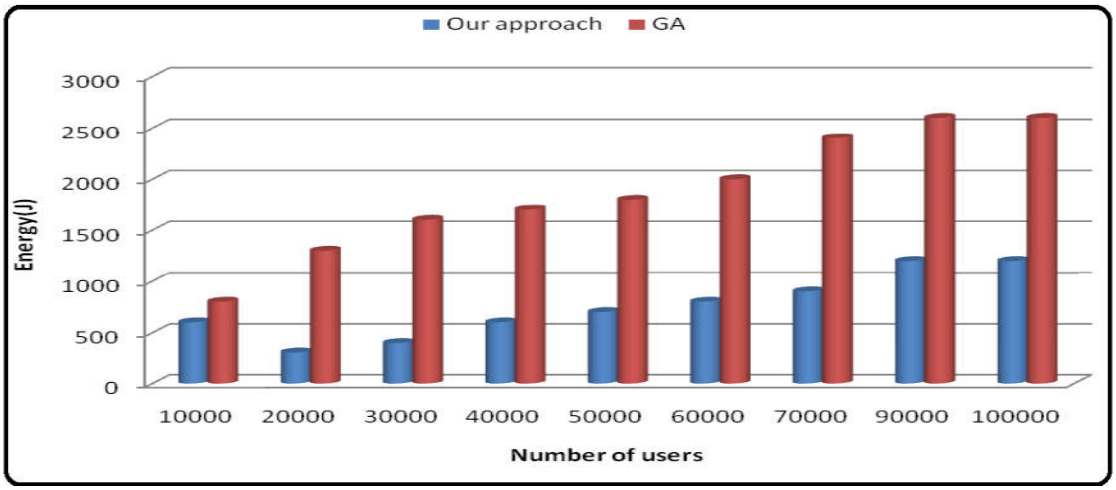


Figure 5-15Consommation de l'énergie dans l'ensemble généréaléatoirement

5.5 Conclusion

Nous avons présenté dans ce dernier chapitre, l'implémentation de notre simulateur de découverte et de sélection dynamique des services pour la gestion contextuelle des besoins fonctionnels et des préférences des entreprises au sein de Cloud. Les fonctionnalités développées sont illustrées par des captures d'écrans qui décrivent les étapes nécessaires pour l'utilisation de notre simulateur basé sur la personnalisation de l'algorithme DSGreenACO selon les préférences et le contexte utilisateur. Enfin nous avons analysé les résultats d'expérimentation et positionné nos résultats par rapport aux travaux existants en les comparants selon quelques métriques et critères d'évaluations. En comparant les résultats de l'algorithme proposé avec d'autres algorithmes montrent que notre algorithme est plus performant en matière de vitessed'exécution, répond mieux aux contraintes et préférences utilisateur et conduit à un temps de calcul raisonnable.

CONCLUSION GENERALE

Notre travail consiste à réaliser un système intelligent pour optimiser les consommations énergétiques(*RAM, CPU, bande passante et énergie*) et les qualités des services avec un réseau des centres de données Cloud pour les environnements intelligents. En effet, le système doit réagir convenablement aux différents changements des préférencesdes clients pour identifier le bon fournisseur de services qui pourrait répondre à leurs besoins fonctionnels et préférences QoS et contextuelles. Pour cette raison, nous avons proposé une approche qui en mesure de prendre en compte les préférences et le contexte courant du client et son environnementet retourne une liste de meilleurs services de qualité au client classés par leurs consommations énergétiques (*RAM, CPU, bande passante et énergie*).

Àcet objectif, nous avons proposé une nouvelle ontologiegénérique contextuelle des services intelligents qui implémente le concept de Green Computing. Elle est conçue pour le filtrage sémantique des servicespertinents à l'aide de règles appliquées à l'ontologie afin d'obtenir le chemin optimal quiréduit la consommation d'énergie et des émissions en carbone des centres de données. Elle est utilisée pour la description sémantique des services Cloud hétérogènes. Elle permet d'interpréter sémantiquement les préférences de QoS du client en tant que valeurs qualitatives et déduire leurs valeurs quantitatives grâce à des règles d'inférence. L'utilisation de raisonnement sémantique sur les deux ontologies combinées (haut niveau, domaine smart) enrichie par des règles SPARQL facilite la modélisation des profils utilisateurs, la recherche et personnaliser des services intelligents des domaines smart-city, smart-transport, etc. Notre objectif derrière cette ontologie, ne se limite pas à la représentation des services Cloud hétérogènes dans le Cloud, mais plutôt à l'exploitation de la sémantique de ces services, tout en tirant profit de tous les avantages apportés par la notion d'ontologie, y compris l'enrichissement de requêtes clients qui sont nécessaires pour prendre des décisions intelligentes comme l'interprétation des préférences qualitatives vers des valeurs quantitatives. Donc les ontologies dans ce domaine peuvent être utilisées pour fournir un soutien d'automatisation de sélection et de composition des services selon les besoins de client, ses préférences et son contexte et l'utilisation efficace des méthodes et techniques bio inspirées de découverte des services.

Afin de découvrir sémantiquement des services pertinents et optimaux en termes de QoS et consommation énergétiquesur un graphe de centres de données des fournisseurs Cloud, un algorithme Dynamique et Sémantique basé Green Ant Colony Optimization (DSGreenACO) est proposé pour la sélection de meilleurs chemins de découverte des services Cloud afin de répondre aux besoins fonctionnels et non-fonctionnels des clients. Deux modifications de l'algorithme ACO, portant sur l'adoption du choix adéquat du nombre de fourmis minimum par rapport au nombre de services Cloud est basée surune fonction d'utilité de qualité. Cette stratégie a contribué en partie à l'amélioration des performances de l'essaim dans la découverte et la sélection des services Cloud et l'optimisationau maximum la quantité d'énergie consommée dans le Cloud.Nous avons proposé aussi une extension multi – proxies distribués pour la composition des services Cloud basons sur la notion de confiance.

Afin de valider les résultats de notre approche nous avons implémenté un prototype et nous avons utilisé l'éditeur d'ontologie open source Protégé version Protégé-3.4.8 et NetBeans. Ce prototype permet d'optimiserl'énergie des centres de données pour la personnalisation des services des applications intelligents qui s'adapte aux différents changements du contexte client via la composition sémantique et dynamique des services Cloud. En outre,les résultats de l'expérimentation montrent bien qu'une approche hybride Green ANT et ontologie permet d'économiser l'énergie et de sélectionner efficacement un chemin de déploiement selon les préférences des clients, les résultats étant très satisfaisants. L'étude comparative de notre approche avec d'autres approches multi objectif a montré l'efficacité de l'algorithme DSGreenACO. Elle a montré également que les résultats de sélection et la composition des services sont pertinents selon les besoins fonctionnels et préférences des clients.

Dans des travaux de perspectives, nous envisageons d'aborder plusieurs perspectives de recherche qui nous semblent pertinentes à court et à long terme peuvent être abordées pour l'amélioration des propositions réalisées. Ces perspectives sont :

- **Gestion de la mobilité des utilisateurs :** ce point consiste à étendre l'approche proposée des services intelligents sensible aux contextes dans les domaines smart-* (smart-tourisme, smart-city, smart-transport, etc.) avec une meilleure gestion autonomique de qualité de service vis-à-vis de la mobilité des utilisateurs. Cela quand il s'agit de la gestion de la mobilité clients, on fait intervenir des contraintes de continuation de découverte de la liste des services selon la localisation de l'utilisateur et son situation. C'est pour cela nous recourons à l'utilisation des techniques de surveillance de contexte utilisateur en utilisant des méthodes dédiées.

- **L'intégration des aspects multi-agents :**actuellement, le domaine des smart- * (domicile, ville, santé, tourisme, etc.) est naturellement orienté multimédia et hétérogène. Dans un tel domaine, on assiste à une utilisation croissante de périphériques mobiles hétérogènes, ainsi que de capteurs transmettant des données (IoT). Nous envisageons des expérimentations réelles de l'approche proposée par l'utilisation des capteurs physiques et l'incorporation des systèmes des méthodes d'optimisation multi-agents distribuées permettent un traitement parallèle avec temps d'exécution meilleur.

- Vers une architecture distribuée à base des services Cloud pour la découverte sécurisée des services intelligents à la demande :la croissance quantitative du nombre de services hétérogènes sur le Cloud, ainsi que la faible structuration de cette immense masse des données, appelle au développement des techniques de découverte efficaces, robustes et adaptées aux besoins et attentes des utilisateurs. Notre approche adopte une technique bio-inspirée centralisée telle que, les fournisseurs de services peuvent avoir un contrôle total sur toutes les transactions du système (par exemple, l'historique surveillé, paiement des droits d'informations, etc.). Dans ce contexte, le composant proxy qui prend en charge un grand nombre d'utilisateurs simultanés peut être difficile et relativement coûteuse. Ce coût de maintenance peut être réduit en adoptant une architecture décentralisé basée sur une cryptographie pour la transaction de contenus privé des utilisateurs. L'introduction de nouvelles techniques émergeantes de l'IoT tel que les Middleware et le Blockchain semblent d'être efficaces. La première facilite le déploiement et la reconfiguration des services et la seconde renforce les aspects de confiance et de sécurité lors de çadécouverte des services.

REFERENCES BIBLIOGRAPHIQUES

[1] B. cogrel, B. Daachi, Y. Amirat et A. Chibani, «Selection de services basée sur l'impactenenvironnementubiquitaire,» *Proceedings of the 7th French-speaking Conference on Mobility and Ubiquity Computing,* n° 7, pp. 9-15, 2011.

[2] A. Ben Hamida, F. Le Mouël, S. Frénot et M. Ben Ahmed, «Une approche pour un chargement contextuel de services dans les environnements pervasifs.,» *Ingénierie des Systèmes d'Information (ISI),* vol. 3, n°13, pp. 59-82, 2008.

[3] A. Dan, R. Johnson et T. Carrato, «SOA service reuse by design,» *Proceedings of the 2nd International Workshop on System Development in SOA Environments, Leipzig, Germany,* pp. 25-28, 2008.

[4] D. G. Abowd, G. D. Atkeson, J. Hong, S. Long, R. Kooper et M. Pinkerton, «Cyberguide : A mobile contextaware tour guide,» *In ACM Wireless Networks,* 1997.

[5] S. Murugesan, «Harnessing Green IT: Principles and Practices,» *IEEE IT Professional,* pp. 24-33, 2/2008.

[6] A. Amokrane, F. Zhani et al, «On satisfying green SLAs in distributed clouds,» *Proceedings of IEEE/ACM 10th International Conference on Network and Service Management (CNSM),* Nov, 2004.

[7] A. Amokrane, G. Pujolle et al, «GreenHead: Virtual Data Center Embedding a cross Distributed Infrastructures, , vol. 1, no. 1, pp. 36–49, 2014.,» *IEEE Transactions on Cloud Computing,* vol. 1, pp. 36-49, 2014.

[8] A. Amokrane, R. Langar et al, «A Green framework for energy efficient Management in TDMA-based Wireless Mesh Networks,» *Proceedings of IEEE/ACM International Conference on Network and Service Management (CNSM),* 2012.

[9] M. Diouri, G. Chetsa, O. Gluck, L. Lefevre, J. Pierson, P. Stolf et G. Da Costa, «Energy efficiency in high-performance computing with and without knowledge of applications and services,» *International Journal of High Performance Computing Applications, SAGE Publications,* vol. 27, n°3, p. 232–243, 2013.

[10] L. Lefevre et A.-C. Orgerie, «Designing and evaluating an energy efficient Cloud,» *The Journal of Supercomputing,* vol. 51, n°3, p. 352–373, mar. 2010.

[11] M. F. Zhani, R. Boutaba et Al, «VDC Planner: Dynamic Migration-Aware Virtual Data Center Embedding for Clouds,» *proceedings of the IFIP/IEEE International Symposium on Integrated Network Management,* p. 18–25, 2013..

[12] Y. Kessaci, N. Melab et E.-. G. Talbi, «Optimisation multi-critére pour l'allocation de ressources sur clouds distribués avec prise en compte de l'énergie,» *Rencontres scientifiques France Grilles,* 2011.

[13] P. Pazowski, «GREEN COMPUTING: LATEST PRACTICES AND TECHNOLOGIES,» *Managing Intellectual Capital and*

Innovation for Sustainable and Inclusive Society: Managing Intellectual Capital and Innovation; Proceedings of the MakeLearn and TIIM Joint International Conference 2015, ToKnowPress ,bari,italy, pp. 1853-1861, mai 2015.

[14] M. Weiser, «The Computer for the Twenty-First Century,» *Scientific American,* vol. 3, n°265, pp. 94-104, septembre 1991.

[15] D. J. Cook et S. K. Das, Smart Environments: Technologies, Protocols, and Applications, John Wiley & Sons, Inc, 2005, pp. 3,4,9.

[16] M. Weiser, R. Gold et J. Seely Brown, «The origins of ubiquitous computing research at PARC in the late 1980s,» *IBM systems journal,* vol. 38, n°4, 1999.

[17] Y. Jicheng, Z. Yuqi, Y. Yanqi, D. Rocco, S. Wujin, R. Davis, S. Frances, J. Ligler, B. Buse et G. Zhen, «Microneedle-array patches loaded with hypoxia-sensitive vesicles provide fast glucose-responsive insulin delivery,» *Dean Ho, University of California, Los Angeles,* juil 2015.

[18] M. David, B. Mark, H. Jerry, L. Ora, M. Drew, M. Sheila, N. Srini, P. Massimo, P. Bijan et A. Al., «Owl-s: Semantic markup for web services,» W3C member submission, 2004.

[19] j. Cissoko, DuclosM. et al, «La santé personnalisée: les objets connectés pour adopter de nouveaux comportements,» *Pratiques en Nutrition: santé et alimentation,* vol. 13, n°50, pp. 30-36, 2007.

[20] S. Bill, A. Norman et W. Roy, «Context-aware computing applications,» chez *Workshop on IEEE Mobile Computing Systems and Applications*, 1994.

[21] J. Matjaz B., «SOA Approach to Integration: XML, Web services, ESB, and BPEL in real-world SOA projects,» Packt Publishing Ltd, 2007.

[22] S. William Noah, «A system architecture for context-aware molbil computing,» PhD thesis, Columbia University , New York, NY, 1995.

[23] G. D. Abowd et al., «Towards a better understanding of context and context-awareness,» *International symposium on handheld and ubiquitous computing. Springer, Berlin, Heidelberg,* n°22, pp. 304-307, 1999.

[24] F. Laforest et F. Le Mouél, *Systèmes d'information pervasifs, Cours de master,* Grenoble(France), 2006.

[25] Gigi, «ressources,» [En ligne]. Available: http://www.terresacree.org/ressources.htm. [Accès le 19 07 2018].

[26] M. Pushtikant et S. Shailendra, «A Study about Green Computing,» *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, n°6, pp. 790-794, June 2013.

[27] L. M. Vaquero, L. Rodero-Merino, J. Caceres et M. Lindner, «A break in the clouds: Towards a cloud definition,» *ACM SIGCOMM Computer Communication ,* vol. 1, n°39, pp. 50-55, 2008.

[28] J. Singh et S. Vatta, «Green computing: eco friendly technology. Int J Eng Res Gen Sci. 2016;4(1):,» *international journal of engineering research and general science,* vol. 1, n°4, pp. 280-283, 2016.

[29] M. P. Papazoglou et W.-J. v. d. Heuvel, «Service oriented

architectures: approaches, technologies and research issues,» *The VLDB Journal, Springer-Verlag,* pp. 389-416, 2007.

[30] G. Chetsa, L. Lefebvre et al, «Exploiting performance counters to predict and improve energy performance of HPC systems,» *Future Generation Computer Systems,* vol. 36, n°8, pp. 287-298, 2014.

[31] D. Geebelen, K. Geebelen et al, «QoS prediction for web service compositions using kernel-based quantile estimation with online adaptation of the constant offset,» *Information Science,* n°268, p. 397–424, 2014.

[32] V. Gabrel, M. Manouvrier et C. Murat, «Web services composition: Complexity and models,» *Discrete Applied Mathematics,* vol. 196, p. 100–114, Dec. 2015.

[33] H. Karloff, «Linear Programming,» *Modern Birkhäuser Classics,* 1991.

[34] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam et H. Chang, «QoS-aware middleware for Web services composition,» *IEEE Transactions on Software Engineering,* vol. 30, n°5, p. 311–327, May 2004.

[35] T. Yu, Y. Zhang et K.-J. Lin, «Efficient algorithms for Web services selection with end-to-end QoS constraints,» *ACM Trans. Web,* vol. 1, n°1, pp. 1-26, May 2007..

[36] M. E. KHANOUCHE, «Sélection et composition de services sensibles à l'énergie et à la qualité de service en environnements ubiquitaires et intelligents ambiants,» Thèse de Doctorat, Université de A.MIRA, BEJAIA, 2017.

[37] K. D. Anind et al., «An architecture to support context-aware applications,» Institute of Technology, Georgia, 1999.

[38] Y. Yujie et C. Haopeng, «Qos-aware service composition using nsga-ii 1,» *Proceedings of the 2nd International Conference on Interaction Sciences:Information Technology, Culture and Human, ACM,* p. 358–363, 2009.

[39] H. Assma, S. Yousra, A. Khadeejah et A. Eman, «An introduction to erp systems : Architecture, implementation and impacts,» *International Journal of Computer Applications,* vol. 9, n°167, pp. 1- 4, 2017.

[40] G. Chetsa, L. Lefebvre et al, «Exploiting performance counters to predict and improve energy performance of HPC systems,» *Future Generation Computer Systems,* vol. 36, n°8, p. 287 – 298, 2014.

[41] G. D. Costa, M. D. de Assuncao, J. Gelas, Y. Georgiou, L. Lefèvre, A. Orgerie, J. Pierson, O. Richard et A. Sayah, «Multi-facet approach to reduce energy consumption in clouds and grids: The green-net framework,» *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking,* p. 95–104, 2010.

[42] Q. Zhang, Q. Zhu, M. Zhani et R. Boutaba, «Dynamic Service Placement in Geographically Distributed Clouds,» *IEEE 32nd International Conference on Distributed Computing Systems (ICDCS),* p. 526 –535, june 2012.

[43] D. Hatzopoulos, I. Koutsopoulos, G. Koutitas et W. Van Heddeghem, «Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources,» *IEEE International Conference on Communications (ICC),* p. 4217–4221, 2013.

[44] Z. Liu, M. Lin, A. Wierman, S. H. Low et L. L. Andrew, « Geographical load balancing with renewables,» *SIGMETRICS Perform. Evaluation Review,* vol. 39, n°3, p. 62–66, Dec. 2011.

[45] Y. Guo, Y. Gong, Y. Fang, P. Khargonekar et X. Geng, «Energy and network aware workload management for sustainable data centers with thermal storage,» *IEEE Transactions on Parallel and Distributed Systems,* n°99, pp. 1-1, 2013.

[46] P. X. Gao, A. R. Curtis, B. Wong et S. Keshav, «It's not easy being green,» *Proceedings of the ACM SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communication, ser. SIGCOMM'12, New York, NY, USA,* p. 211–222, 2012.

[47] Z. Abbasi, M. Pore et S. Gupta, « Impact of workload and renewable prediction on the value of geographical workload management,» *the 2nd International Workshop on Energy Efficient Data Centers (E2DC), held as a part of ACM e-Energy,* 2013.

[48] Q. Zhang, Q. Zhu, M. Zhani et R. Boutaba, «Dynamic Service Placement in Geographically Distributed Clouds,» *IEEE 32nd International Conference on Distributed Computing Systems (ICDCS),* p. 526 –535, june 2012.

[49] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag et B. Maggs, «Cutting the electric bill for internet-scalesystems,» *SIGCOMM Computing Communication Review,* vol. 39, n°4, p. 123–134, August 2009.

[50] A. Beloglazov et R. Buyya, «Energy efficient management in virtualized cloud data centers," in Cluster, Cloud and Grid Computing (CCGrid),,» *10th IEEE/ACM International Conference, (Melbourne, Australia),* p. 826–831, 2010.

[51] J. Joy, M. Gerla et al, «Energy efficient, context- aware cache coding for mobile information-centric networks,» *ACM International Conference on Distributed and Event-based Systems, Irvine, CA,,* pp. 270-280, 2016.

[52] C. F. Touzet, LES RESEAUX DE NEURONES ARTIFICIELS INTRODUCTION AU CONNEXIONNISME COURS, EXERCICES ET TRAVAUX PRATIQUES, Marseille - France: Collection de l'EERIE, EC2, Juillet 1992.

[53] S. Grossberg, «Nonlinear neural networks: Principles, mechanisms, and architectures,» *Neural networks,* pp. 17-61, 1988.

[54] J. Schmidhuber, «Deep learning in neural networks,» *An overview. Neural Networks,* vol. 61, pp. 85-117, 2005.

[55] M. Mitchell, S. Forrest et J. H. Holland, «The royal road for genetic algorithms: Fit-ness landscapes and GA performance,» *European Conference on artificial life,* pp. 245-254, 1992.

[56] M. Dorigo et M. Birattari, «Ant colony optimization,» *Encyclopedia of machine learning, Springer US,* pp. 36-39, 2010.

[57] Z. H. Zhan, J. Zhang et al, «Orthogonal learning particle swarm opti-mization,» *IEEE Transactions on Evolutionary Computation,* vol. 8, n°15, pp. 832-847, 2011.

[58] D. Karaboga, B. Gorkemli et al, «A comprehensive survey: artificial bee colony (ABC) algorithm & applications,»

Artificial Intelligence Review, vol. 1, n°42, pp. 21-57., 2014.

[59] M. Dorigo et C. Blum, «Ant colony optimization theory: A survey.,» *Theoretical Computer Science,* vol. 2, n°344, pp. 243-278, 2005.

[60] D. Marco et S. Thomas, «The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances (Technical Report IRIDIA-2000-32),» International Series in Operations Research and Management Science, Kluwer, 2001.

[61] A. COSTANZO, T. LUONG et al, «Optimisation par colonies de fourmis,» 19 mai 2006.

[62] M. Dorigo, V. Maniezzo et A. Colorni, «Positive feedback as a search strategy,» rapport technique numéro 91-016, Dip. Elettronica, Politecnico di Milano, Milano, Italy, 1991.

[63] A. Colomi, M. Dorigo et V. Maniezzo, «Distributed optimization by ant colonies,» *Proceedings of the 1stEuropean Conference on Artificial Life,* n°1, pp. 134-142, 1991.

[64] Gambardella, L.M.; Dorigo, M.;, «Ant-Q: A Reinforcement Learning Approach to the Travelling Salesman Problem, In A. Prieditis and S. Russell (Ed.),» *Proceedings of the Eleventh International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann,* pp. 252-260, 1995.

[65] M. Dorigo et L. Gambardella, «Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,» *IEEE Transactions on Evolutionary Computation,* vol. 1, n°1, pp. 53-66, 1997.

[66] B. Bullnheimer, R. Hartl et C. Strauss, «An improved ant system algorithm for the vehicle routing problem,» *Annals of Operations Research,* n°89, pp. 319-328, 1999.

[67] T. Stützle et H. Hoos, «MAX-MIN Ant System,» *Future Generation Computer Systems,* vol. 16, n°8, p. 889–914, 2000.

[68] M. Maniezzo et A. Carbonaro, «An ANTS heuristic for the frequency assignment problem,» *Future Generation Computer Systems,* vol. 8, n°16, pp. 927-935, 2000.

[69] C. Blum, A. Roli et M. Dorigo, «HC–ACO: The hyper-cube framework for ant colony optimization,» *Proceedings of MIC'2001—Meta–heuristics International Conference. Porto, Portugal,* vol. 2, p. 399–403, 2001.

[70] S. Daniel et P. Aditya, «RDFGraph: New Data Modeling Tool for Semantic Web,» *World Academy of Science, Engineering and Technology,* n°53 , 2009 .

[71] E. Durfee, V. Lesser et D. Corkill, «Trends in Cooperative Distributed Problem Solving,» *IEEE Transactions on Knowledge and Data Engineering,* vol. 1, n°1, pp. 63-83, 1998.

[72] A. Gómez-Pérez, M. Fernández-López et O. Corcho, «Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web,» *Springer-Verlag, New York.,* 2003.

[73] wikipédia, «Ontologie,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Ontologie_(informatique)#Types. [Accès le 1 9 2018].

[74] S. Carolina, D. Oscar, G. Angélica, B. Marla et S. René, «Representation of an Electrical Engineering Curriculum

using an Ontology of Controlled Quality,» chez *Eleventh LACCEI Latin American and Caribbean Conference for Engineering and Technology*, Cancun, Mexico, August, 2013.

[75] H. Nasir, K. Ku-Mahamud et E. Kamioka, «Ant colony optimization approaches in wireless sensor network: Performance Evaluation,» *Journal of Computer Science,,* vol. 13, n°6, pp. 153-164, 2017.

[76] W. G. Stillwell, S. David A. et E. Ward, «A comparison of weight approximation techniques in multiattribute utility decision making,» *Organizational behavior and human performance,* vol. 1, n°28, pp. 62-77, 1991.

[77] P. Mattia CF, P. Luciano, A. Isabella et al, «Combinatorial analysis and algorithms for quasispecies reconstruction using next-generation sequencing,» *BMC Bioinformatics,* vol. 12, p. 5, 2001.

[78] M. YAHLALI et A. CHOUARFIA, «SCAE: Software Component Assembly Evaluation,» *International Journal of Software Engineering and Its Applications,* vol. 8, n°3, pp. 255-268, 2014.

[79] M. Chen, Y. Zhang et al, «AIWAC:affective interaction through wearable computing and cloud technology,» *IEEE Wirelless Communication,* vol. 1, n°22, pp. 20-27, 2015.

[80] L. Zhou, «QoE-driven delay announcement for cloud mobile media,» *IEEE Trans Circuits Syst Video Technol.2017;27(1):84-94.,* vol. 1, n°27, pp. 84-94, 2017.

[81] S. Huifeng, Z. Zibin et al, «Personalized Web Service Recommendationvia Normal Recovery Collaborative Filtering,» *IEEE TRANSACTIONS ON SERVICES COMPUTING,* vol. 6, n°4, pp. 573-579, 2013.

[82] J. J. Liu, C. J. Jiang et al, «A Transaction and QoS-Aware Service Selection Approach Based on Genetic Algorithm,» *IEEE Transactions on systems, man and cybernetics: systems,* vol. 7, n°45, pp. 1035-1046, 2015.

[83] S. Kartakis, M. Antona et C. Stephanidis, «Control smart homes easily with simple touch,» *Proceedings of the international ACM workshop on Ubiquitous meta user interfaces,* p. 1–6, 2011.

[84] J. Hsu, P. Mohan, X. Jiang, J. Ortiz, S. Shankar, S. Dawson-Haggerty et D. David Culler, «HBCI(Human-Building-Computer Interaction),» chez *BuildSys 2010*, Zurich, Switzerland, 2010.

[85] S. Khan, K. F. Li, E. G. Manning et M. Andakbar, «Solving the knapsack problem for adaptive multimedia systems,» *Studia Informatica Universalis 2,* vol. 1, p. 157–178, 2002.

[86] E. Bonabeau, M. Dorigo et G. Theraulaz, «Inspiration for optimization from social insect behavior,» *Nature ,* n°406, p. 39–42, 2000.

[87] M. Steve, «Definition of "Wearable Computer",» chez *International Conference on Wearable Computing ICWC-98*, Fairfax VA, US, May 1998.

[88] Samsung, «smart-home,» [En ligne]. Available: https://www.smartthings.com/smart-home. [Accès le 17 07 2018].

[89] R. Want, A. Hopper, V. Falcão et J. Gibbons, «The Active Badge Location System,» *ACM Transactions on Information*

Systems (TOIS), vol. 10, Jan. 1992.

[90] J. Hiranya, «Service-Driven Computing with APIs: Concepts, Frameworks, and Emerging Trends,» chez *Handbook of Research on Architectural Trends in Service-Driven Computing*, University of California – Santa Barbara, USA, IGI Global, 2014, p. 3.

[91] J. Humble, A. Crabtree, T. Hemmings, K. Åkesson, B. Koleva, T. Rodden et P. Hansson, «Playing with the Bits“ User-Configuration of Ubiquitous Domestic Environments,» *the series Lecture Notes in Computer Science,* vol. 2864, p. 256–263, 2003.

[92] K. F. Herwig et H. Volker, «Classification of Smart Environment Scenarios in Combination with a Human Wearable Environment Communication Using Wireless Connectivity,» chez *4th International Conference on Information Technology, Control, Chaos, Modeling and Applications (ITCCMA-2017)*, Dubai, UAE, Mai 2017.

[93] D. Nigel et G. Hans-Werner, «Beyond Prototypes: Challenges in Deploying Ubiquitous Systems,» *PERVASIVEcomputing (IEEE),* vol. 2, pp. 29-32, 2002.

[94] Y. C. Lee et A. Y. Zomaya, « “Energy efficient utilization of resources in cloud computing systems,”,» *Journal of Supercomputing,* vol. 60, n°2, p. 268–280, 2012.

[95] M. Rodriguez-Martínez, H. Valdivia, J. Seguel et M. Greer, «Estimating power/energy consumption in database servers,» *Procedia Computer Science,* vol. 6, p. 112–117, 2011.

[96] A. Beloglazov, J. Abawajy et R. Buyya, «Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,» *Future Generation Computer Systems,* vol. 28, n°5, p. 755–768, 2012.

[97] J. Sekhar et G. Jeba, «Energy efficient vm live migration in cloud data centers,» *International Journal of Computer Science and Network (IJCSN),* vol. 2, n°2, p. 71–75, 2013.

[98] J. Cao, Y. Wu et M. Li, «Energy efficient allocation of virtual machines in cloud computing environments based on demand forecast,» *Advances in Grid and Pervasive Computing (A. in Grid and P. Computing, eds.), Springer, Berlin Heidelberg,* vol. 7296, p. 137–151, 2012.

[99] deco du net, «Dictionnaire > Définitions Normes d'Internet > Service web,» [En ligne]. Available: http://www.dicodunet.com/definitions/normes/service-web.htm. [Accès le 19 07 2018].

[100] D. Johann, P. Alain, T. Éric et S. Patrick, Métaheuristiques pour l’optimisation difficile, Eyrolles; ISBN : 2-212-11368-4, 2003.

[101] Y. LAALAOUI, «Ordonnancement temps réel hors-ligne à l"aide de Système de Colonie de Fourmis en environnement monoprocesseur,» these magister en Informatique, Institut National d'Informatique, Alger, 2005.

[102] P. R. Swaminarayan, P. Sanskruti et V. Kamlesh, «An Interoperable Knowledge Representation in Distributed Environment,» *International journal of Computational Intelligence Research,* vol. 5, n°3, pp. 223-240, 2009.

[103] V. Fortineau, «Contribution à une modélisation ontologique desinformations tout au long du cycle de vie du produit,»

Ecole nationale supérieure d'arts et métiers - ENSAM, paris,France, 2014.

[104] O. M., «owl-dl-semantics,» 2013. [En ligne]. Available: http://www.obitko.com/tutorials/ontologies-semantic-web/owl-dl-semantics.html.

[105] R. Stefan, K. Holger, M. Pascal, B. Andrew et L. Miroslaw, «Understanding and quantifying the future of cloud computing,» Forrester"Technical report", 2011.

[106] B. J.S. Chee et C. Franklin J.r., «Cloud Computing,» *Technologies and Strategies of the Ubiquitous Data Center, CRC Press,* pp. 43-60, 2010.

[107] M. Pushtikant et S. Shailendra, «A Study about Green Computing,» *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, n°6, pp. 790-794, 2013.

[108] M. Kamel, A. Adel, R. Philippe et L. Sébastien, «Dynamic Semantic-Based Green Bio-inspired Approach for Optimizing Energy and Cloud Services Qualities,» *Transactions on Emerging Telecommunications Technologies – Wiley,* vol. 5, n°29, 2018.

[109] F. Barron, «Selecting a best multiattribute alternative with partial information about attribute weights,» *Acta Psychol,* n°80, p. 91–103, 1992.

[110] A. Amokrane, F. Zhani et al, «On satisfying green SLAs in distributed clouds,» *Proceedings of IEEE/ACM 1 Oth International Conference on Network and Service Management (CNSM),* 2014.

[111] K. E. H. Truong et G. Abowd, «Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home,» *Ubiquitous Computing, Computer Science,* vol. 3205, p. 143–160, 2004.

ANNEXE : LISTE DES PUBLICATIONSPERSONNELLES

Publication dans des revues internationales

Kamel Mansouri, Adel Alti, Philippe Roose, Sébastien Laborie – Dynamic Semantic-Based Green Bio-inspired Approach for Optimizing Energy and Cloud Services Qualities – Transactions on Emerging Telecommunications Technologies – Wiley – ISSN: 2161-3915 – DOI:10.1002/ett.3305. IF: 1.61.

Communications dans des conférencesInternationales

Kamel Mansouri, Adel Alti, Philippe Roose– Selection and Composition of Cloud Smart Services Using Trust Semantic-Based Green Quality Approach – The 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS'2018) – October, 2018 Tebessa, Algeria.

ملخص

في الوقت الحالي، يمكن للجميع الوصول إلى الخدمات الموجودة على السحابة من مجموعة متنوعة من الأجهزة المحمولة في أي وقت ومن أي مكان (في المنزل، العمل، في السيارة، وما إلى ذلك). أدى الاستخدام المكثف للأجهزة والتكنولوجيات الجديدة غير المتجانسة لاكتشاف ونشر الخدمات السحابية إلى مقايضة بين التكاليف وتحسين جودة الخدمات (مثل وقت الاستجابة السريع، وانخفاض الأسعار، وتحسين الأمن، والحد من استهلاك الطاقة والانبعاثات الكبيرة من الكربون، وما إلى ذلك). لقد قادت هذه المقايضة معظم مقدمي الخدمات السحابية إلى الدعوة إلى حلول ذكية وأسرع وأخرى موفرة للطاقة. تهدف الرسالة إلى اقتراح نهج جديد يعتمد على تقنيات الويب الدلالي وخوارزمية مستعمرة النمل (ACO)التي تهدف إلى تقليل استهلاك الطاقة لمجموعة متنوعة من الخدمات السحابية. نهجنا عام، وبالتالي، يقدم g بنية تحتية مرنة حيث يمكنهم بسهولة تحقيق تفضيلاتهم. من أجل التحقق من صحة اقتراحنا، استخدمنا الويب الدلالي وتقنياتCloud ، بالإضافة إلى خوارزمية ACO من خلال تجربة فعاليتها وتوفير الطاقة في عدة مجموعات بيانات عشوائية وحقيقية.

مفاتيح:بيئة ذكية،السحابة (Cloud)، مركز البيانات؛ الطاقة؛ التحسينإلى الأمثل،الهيكل الدلالي(أنطولوجيا)،الخدمة، خوارزمية ANT، جودة الخدمة.

Abstract

Currently, everybody can access to existing services on the Cloud from a wide variety of mobile devices at any time and from anywhere (at home, working, on the car, etc.). The massive use of new heterogeneous mobile devices and technologies for discovering and deploying Cloud services has led a trade-off between costs and improved quality of services (*e.g. fast response time, low price, improved security, the reduction of energy consumption and considerable emissions of Carbon, etc*.). This trade-off has led most Cloud service providers to call for new intelligent, faster and energy-saving solutions. The thesis aims to propose a new approach based on Semantic Web technologies and Ant Colony Optimization (ACO) algorithm which intends to reduce the energy consumption of a wide variety of Cloud services. Our approach is generic and, therefore, offers to customers a flexible infrastructure where they can easily achieve their preferences. In order to validate our proposal, we have used the Semantic Web and the Cloud technologies, as well as ACO algorithm by experiment their effectiveness and energy saving in several random and real□world data sets.

Keywords: Smart environment, Cloud; Data Center; energy; optimization, ontology, service, ANTalgorithm, QoS.

Résumé

Au cours des dernières années, l'utilisation des smartphones et tablettes a augmenté de façon significative. En effet, les utilisateurs, accèdent à tout instant, et en tout lieu à des services accessibles sur le Cloud. L'utilisation massive de nouveaux dispositifs et technologies mobiles hétérogènes pour découvrir et déployer des services Cloud a conduit à un problème de compromis entre les coûts d'accès et de déploiement des services et l'amélioration de la qualité des services (temps de réponse rapide, prix bas, sécurité améliorée, réduction de la consommation d'énergie, etc.). Dans ce contexte, nous tenons compte de l'ensemble des besoins et préférences des clients en nous focalisant tout particulièrement sur l'optimisation d'énergie. Pour ce faire, nous proposons une nouvelle approche utilisant les principes de l'algorithme des colonies de fourmis associé à des principes de filtrage sémantiques des services guidée par une ontologie pour la réduction de la consommation d'énergie et des émissions en carbone des centres de données. Afin de valider notre proposition, nous avons appliqué nos propositions liées aux domaines du web sémantique, du Cloud et l'algorithme ANT dans des expérimentations liées à la réduction de l'énergie et son efficacité sur plusieurs ensembles de données aléatoire et réels.

MOTS-CLÉS :environnement intelligent Cloud, énergie, optimisation, ontologie, service, algorithmecolonie de fourmis,QoS.