



Intégration de services cloud computing pour la notification dans un réseau Ad-hoc

NGINDU MULUMBA Pascal¹, KALONJI SHIKAYI Sylvain², NTUMBA BADIBANGA Simon³

*Chef de Travaux à Institut Supérieur des Techniques Médicales de Kananga (ISTM)*¹

*Assistant à l'Académie Militaire de Kananga (ACAMIL)*²

*Professeur Ordinaire à Université de Kinshasa (UNIKIN)*³

En république démocratique du Congo (RDC)

Abstract : The rapid emergence of information technologies has increased the need for connectivity and optimal data processing. The flexibility and decentralized architecture of ad hoc networks, while offering solutions adapted to mobile and dynamic environments, also present challenges in terms of load management and balancing. The synergy of web services, cloud computing, and ad hoc networks transcends these limitations by offering distributed storage and processing capacities, while facilitating load balancing in a multi-queue environment.

This study develops the integration mechanisms of these technologies to streamline task distribution and load balancing, reducing wait times and optimizing resource allocation. The study demonstrates that combining ad hoc networks and cloud computing via web services is an effective approach for developing resilient, high-performance distributed systems capable of meeting the growing needs of users.

Keywords: *Cloud computing, notification, ad-hoc network, ad-hoc routing, service integration.*

Résumé: L'émergence rapide des technologies de l'information à augmenter les besoins en connectivité et en traitement optimal des données ; la flexibilité et l'architecture décentralisée caractérise les réseaux ad hoc, tout en proposant des solutions adaptées aux environnements mobiles et dynamiques, mais relève des défis en termes de gestion et de l'équilibrage de charge. La synergie des services web, du cloud computing et de réseau ad-hoc permet de transcender ces limites en offrant des capacités de stockage et de traitement réparties, tout en facilitant l'équilibrage de charge via une notification dans un environnement à multi file d'attente.

Cette étude développe les mécanismes d'intégration de ces technologies pour rationaliser la distribution des requêtes et l'équilibrage de charge, en réduisant les délais d'attente et optimisation d'allocation des ressources. L'étude prouve que la jonction des réseaux ad hoc et du cloud computing via les services web constitue une démarche efficace pour développer des systèmes répartis résilients, performants et capables de répondre aux besoins croissants des utilisateurs.

Mots-clés: *Cloud computing, notification, Réseau ad-hoc, routage ad-hoc, intégration de service.*

Digital Object Identifier (DOI): <https://doi.org/10.5281/zenodo.18086043>



1. Introduction

1.1. Contexte de la recherche

Avec l'émergence des technologies de l'information et de la communication, nous assistons à la montée constante des besoins en connectivité et en traitement efficace des données. Les réseaux ad hoc, caractérisés par la capacité à fonctionner sans infrastructure préétabli, et par leur structure décentralisée qui offrent une flexibilité importante pour la communication entre équipements mobiles ou stations de travail provisoire. Cependant, le caractère dynamique et parfois instable de ces réseaux est un défi majeur en termes de répartition de charge et de gestion des ressources.

Le cloud computing assure le stockage, le traitement et la gestion de grandes quantités de données sur des infrastructures distantes, donnant ainsi des possibilités d'équilibrage de charge et de gestion multi-file d'attente pour optimiser les performances des applications. Les services web et le cloud computing sont devenus des piliers essentiels pour la fourniture de services distribués et extensible.

La synergie des **réseaux ad hoc** avec le **cloud computing** via des **services web** ouvre de nouvelles perspectives pour améliorer la répartition des tâches et la gestion des ressources dans des environnements complexes. L'intégration de la puissance et l'élasticité du cloud, avec la flexibilité et l'autonomie des réseaux ad hoc à travers de service web, il devient possible de mettre en place des systèmes capables de distribuer efficacement la charge sur plusieurs files d'attente, tout en garantissant une qualité de service optimale.

Cette article s'aligne dans la perspective de déployer des solutions d'équilibrage de charge adaptées aux systèmes dynamiques, en exploitant la synergie entre réseaux ad hoc, services web et cloud computing. Le but poursuivi est de proposer un cadre pouvant gérer efficacement les requêtes des utilisateurs, de réduire le délai et d'optimiser l'usage des ressources disponibles tout en garantissant la notification en temps réel.

1.2. Problématique et hypothèse

L'absence d'infrastructure fixe, la mobilité des nœuds, la variabilité des connexions et les contraintes énergétiques sont de caractéristiques de réseaux Ad-hoc. Ces faiblesses compliquent la notification plus rapide, fiable et continue entre les nœuds. Le cloud computing, grâce à sa puissance de traitement, sa capacité de stockage et ses services reparties, pourrait remédier ces insuffisances.

La principale préoccupation consiste à savoir : comment concevoir une architecture de mutualisation services cloud-computing , réseaux ad-hoc et service web, pour éviter la surcharge du système et assurer l'équilibre dans l'affectation de requêtes et la diffusion fiable des notifications (messages, événements, alertes,) malgré le caractère instable de nœuds de connectivité dans le ad-hoc.

C'est ainsi, que nous nous sommes posé la question de savoir : quelle architecture intégrante faut-il pour assurer l'équilibrage des systèmes à travers la notification dans ces systèmes temps réel ?

Cependant, l'intégration de ces paradigmes (réseau ad-hoc, service web, et cloud computing), reste complexe, notamment à cause de faiblesse et limite ci-haut.

Nous formulons les hypothèses ci-après, vu le problème soulever ci-haut : la combinaison de réseau ad-hoc cloud et service web dans une architecture intégrante et une application d'affectation et équilibrage de charge dans ce système par notification sera une contribution importante pour éviter de congestion, la surcharge, le délai d'attente et améliore la fiabilité des notifications et le rendement global du système.

2. Le Réseau Ad Hoc

Un réseau ad hoc est un réseau sans fil sans infrastructures composé de nœuds mobiles capables de communiquer entre eux sans point d'accès central. Chaque nœud agit à la fois comme routeur, récepteur et, émetteur permettant le routage multi-saut.

Les caractéristiques principales des réseaux ad hoc sont :

- Auto-configuration et auto-organisation,
- Absence d'infrastructure fixe,
- Topologie dynamique due à la mobilité des nœuds,
- Routage adaptatif, souvent basé sur des protocoles tels que, OLSR, DSR ou AODV.

Les domaines d'applications des réseaux ad hoc couvrent des réseaux de secours et militaires, aux réseaux de capteurs, en passant par les **réseaux véhiculaires** (VANETs) et l'Internet des objets (IoT). (Mounir Frikha, 2010).

2.1. Le routage dans le réseau mobile

Les réseaux mobiles ad-hoc (MANETs) est caractérisent par une mobilité fréquente des nœuds, une

topologie dynamique, et l'absence d'infrastructure centralisée. Dans ce cadre, le rôle fondamental de protocole de routage est d'assurer une communication efficace et fiable entre les nœuds. (Sondi Obwang, P. 2023). L'objectif visé est d'identifier les protocoles de routage les plus efficaces en fonction des conditions spécifiques de fonctionnement d'un MANET.

2.2. Typologie des protocoles de routage

Dans une MANETs, on classe les principaux protocoles de routage en trois catégories, chacune ayant ses propres champs d'application et mécanismes:

Dans les réseaux ad-hoc, chaque élément peut effectivement émettre et recevoir des messages, et joué un rôle de relais des informations afin que les messages circulent dans le réseau. Chaque nœud du réseau doit donc posséder des capacités de routage dit ad hoc. Ainsi grâce à ce routage, la portée radio d'un nœud peut être virtuellement étendue en utilisant ses voisins comme relais de l'information. Exemple de la figure suivante le nœud A peut envoyer des messages au nœud G même si celui-ci est hors la portée radio (cercle en pointillé) du nœud A, il a utilisé la route BCFG

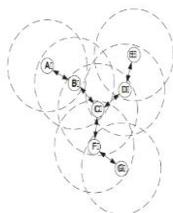


Figure 2. 1 : Protocoles de routage dans les réseaux

Le routage de l'information dans ce type de réseau est un problème complexe. Sachant que ces réseaux sont souvent peu stables : les nœuds peuvent être mobiles; les nœuds peuvent apparaître et disparaître du réseau à tout moment, soit parce qu'ils s'éteignent, soit parce qu'ils sortent de la portée radio de nœuds du réseau, les ressources des nœuds tel que : (capacité de calcul, énergie...) sont souvent limitées car ce sont des équipements légers, embarqués et mobiles. (Ghandour, F. 2013).

2.2.1. Protocoles proactifs (table-driven)

Les protocoles proactifs sont des protocoles de routage qui maintiennent constamment des informations à jour sur les itinéraires disponibles dans le réseau. (RISSON, F., et GAONA, N. 2004) Ils se différencient principalement par leur mode de mise à jour des tables de routage.

Ils présentent une faible latence mais entraîne une charge de trafic de contrôle plus considérable.

Exemples : DSDV (Destination-Sequenced Distance Vector), OLSR (Optimized Link State Routing),

2.2.1.1. Protocole OLSR

Le protocole OLSR (Optimized Link State Routing) utilise une version optimisée de l'algorithme Link State classique. Il est particulièrement conçu pour les réseaux ad hoc mobiles (MANETs). L'optimisation principale fournie par OLSR repose sur l'utilisation de Multipoint Relays (MPRs) afin de diminuer la surcharge liée à la diffusion des messages de contrôle.

1. Principe de base de l'algorithme Link State

Dans un protocole LS classique :

- Chaque nœud diffuse à tous les autres son état de lien (Link State Advertisements - LSAs).
- Chaque nœud découvre ses voisins directs.
- Chaque nœud construit ensuite une vue d'ensemble du réseau et calcule les chemins optimaux avec l'algorithme de Dijkstra.

2. Optimisation OLSR : Multipoint Relay (MPR)

OLSR introduit les principes de MPR pour réduire la redondance :

- Chaque nœud choisit un sous-ensemble de ses voisins, appelés MPRs.
- les MPRs Seuls sont autorisés à retransmettre les messages de contrôle (HELLO, TC - Topology Control).
- Cela réduit le volume de messages de contrôle dans le réseau.

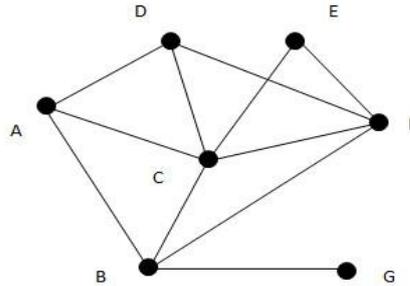


Figure 2.2. : Architecture MPR

Les nœuds choisis pour assurer le relais de l'information dans le réseau sont les MPR. Ce sont les nœuds MPR qui assurent le routage dans le réseau ad-hoc.

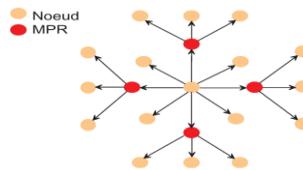


Figure 2.3 : le nœud MPR dans OLSR

OLSR étant un protocole proactif utilise périodiquement des messages de contrôle afin de mettre à jour les tables de routage. Les deux principaux messages utilisés sont les paquets "Hello" et les paquets TC (Topology Control).

3. Étapes principales de l'algorithme OLSR :

1. Chaque nœud détecte ces voisins directs ;
2. Sélection des MPRs ;
3. Diffusion des messages de contrôle via MPRs ;
4. Calcul des routes : Chaque nœud utilise l'algorithme de Dijkstra sur la topologie obtenue pour calculer la route optimale vers chaque destination. Voici les étapes de l'algorithme OLSR

Tableau n° 1 : Etapes de l'algorithme OLSR

Etape	Description
HELLO	Détection des voisins
Sélection MPR	Optimisation de la diffusion
TC via MPR	Mise à jour de la topologie
Dijkstra	Calcul des routes optimales

4. Complexité

Temps de calcul : similaire au Link State standard (Dijkstra : $O(N \log N)$ avec N nœuds) ; Réduction du trafic de contrôle : grâce aux MPRs ; Besoins en mémoire : chaque nœud garde une table de voisinage, MPRs, et topologie réduite.

2.2.1.2. Algorithme de Dijkstra

Cet algorithme est couramment employé pour déterminer le chemin le plus court dans les réseaux, en particulier dans les protocoles à état de lien tels que l'OLSR. Il conserve une collection de nœuds identifiés avec leur distance minimale par rapport à la source, et il les élargit progressivement en sondant les voisins. Voici les conditions que doit respecter l'algorithme de Dijkstra : Le graphe doit disposer de poids de liens non négatifs et est applicable aux graphes orientés comme non orientés.

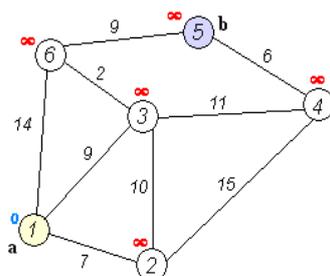


Figure 2.4. Graphe d'algorithme digikstra

L'algorithme de Dijkstra pour trouver le plus court chemin entre a et b.

Trouver le chemin optimal entre un nœud du départ $s \in V$ et tous les autres nœuds dans un graphe pondéré $G = (V, E)$, où :

- V représente l'ensemble des nœuds,
- E est un sous l'ensemble des arêtes, ($E \subseteq V \times V$)

Chaque arête $(u, v) \in E$ se voit attribuer un poids $w(u, v) \geq 0$, qui symbolise le coût du lien.

1. formulation mathématique du modèle de Dijkstra

❖ Données :

- Graphe : $G=(V, E)$
- Coût des liens (Poids): $w: E \rightarrow \mathbb{R} \geq 0$
- Nœud de départ: $s \in V$

❖ Variables :

- $d(v)$: évolue de s à v
- $\pi(v)$: prédécesseur de v sur l'itinéraire le plus court depuis s
- S : collection des nœuds dont la distance minimale a été trouvée (ensemble « exploré »)

a) Algorithme Dijkstra (formulation mathématique) :

1. Initialisation :

$\forall v \in V : d(v) \leftarrow \infty, d(s) \leftarrow 0, \pi(v) \leftarrow \text{null } S \leftarrow \emptyset$

Itération (tant que $S \neq V$) :

○ Sélectionner $u \in V \setminus S$ tel que :

$$d(u) = \min \{ d(v) : v \in V \setminus S \} \quad d(u) = \min \{ d(v) : v \in V \setminus S \}$$

$$d(u) = \min \{ d(v) : v \in V \setminus S \}$$

○ Ajouter u à S

○ Pour chaque voisin $v \in V$ de u :

si $d(u) + w(u, v) < d(v)$ alors :

$$d(v) \leftarrow d(u) + w(u, v)$$

$$\pi(v) \leftarrow u$$

2. Fin de l'algorithme :

○ À la fin, $d(v)$ contient la distance minimale de s à chaque $v \in V$

○ $\pi(v)$ permet de reconstruire le plus court chemin

Figure 2.5. Algorithme Dijkstra modèle mathématique

b) Algorithme Dijkstra en pseudo-code

```

Entrée : Graphe  $G = (V, E)$ , poids  $w(u, v) \geq 0$ , source  $s$ 
Sortie : distance minimale de  $s$  vers chaque nœud, et les chemins

1. Pour chaque nœud  $v$  dans  $V$  :
    $dist[v] \leftarrow \infty$ 
    $précédent[v] \leftarrow null$ 
2.  $dist[s] \leftarrow 0$  // distance du nœud source à lui-même
3.  $Q \leftarrow$  ensemble de tous les nœuds (non encore visités)

4. Tant que  $Q$  n'est pas vide :
    $u \leftarrow$  nœud de  $Q$  avec  $dist[u]$  minimale
   Retirer  $u$  de  $Q$ 
   Pour chaque voisin  $v$  de  $u$  :
     si  $v \in Q$  :
        $alt \leftarrow dist[u] + w(u, v)$ 
       si  $alt < dist[v]$  :
          $dist[v] \leftarrow alt$ 
          $précédent[v] \leftarrow u$ 
    
```

Fig.2.6. Algorithme Dijkstra en pseudo-code

Dès que chaque nœud a obtenu toutes les LSAs : Il refait la représentation graphique du réseau. Il se sert de l'algorithme de Dijkstra pour déterminer les itinéraires optimaux vers tous les autres nœuds. Le protocole OLSR modifie le modèle LSA et l'algorithme de Dijkstra en intégrant plusieurs perfectionnements particuliers :

2.2.2. Protocoles réactifs

Les protocoles de routage réactifs ne conservent pas constamment des tables de routage couvrant tout le réseau. L'approche basée sur la détermination des itinéraires à la demande contribue à diminuer le nombre de messages de contrôle, cependant elle peut provoquer des délais lorsqu'un nœud tente d'expédier un message pour la première fois vers un autre nœud ou si le parcours enregistré n'est plus disponible. (Turjman, F. Al., 2022).

Exemples : AODV (Ad hoc On-demand Distance Vector), DSR (*Dynamic Source Routing*)

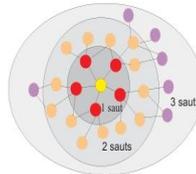


Figure 2.7 : Principe des zones en fonction du nombre des sauts dans le protocole FSR

2.2.2.1. Protocole AODV

Le protocole AODV (Ad Hoc On-Demand Distance Vector) est une méthode de routage employée dans les réseaux sans fil ad hoc. Il est conçu pour faciliter la communication entre les nœuds d'un réseau mobile, sans recourir à une infrastructure centralisée telle que des routeurs câblés ou des points d'accès. (Frikha, M. 2010).

1. Algorithme de découverte de trajet (Route Discovery) au sein d'AODV

L'algorithme AODV a recours à une découverte de route à la demande lorsque le nœud source ne possède pas de trajet valide vers la destination.

L'algorithme Route Discovery d'AODV fusionne des méthodes de diffusion intelligente, de suivi de séquence et de routage temporaire afin de proposer une solution performante et évolutive pour les réseaux ad hoc.

En termes de capacité de stockage, chaque nœud conserve une table de routage et un cache des requêtes précédemment traitées, entraînant une complexité spatiale de $O(n)$ par nœud.

Par conséquent, AODV propose un équilibre satisfaisant entre performance et économie de ressources, spécifiquement conçu pour les réseaux ad hoc à topologie modérément dynamique. (Nirbhay, C., 2025).

2.2.3. Protocoles hybrides

Les protocoles hybrides présentent une structure moins rigide que les deux types antérieurs. Olamide, K. (2024). par exemple, en ce qui concerne la capacité énergétique. Ils fusionnent les approches réactives et proactives pour améliorer les performances dans des contextes variés.

1. Protocol ZRP

En réalité, le protocole ZRP (Zone Routing Protocol) combine deux protocoles de routage distincts : l'un proactif et l'autre réactif.

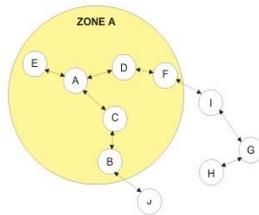


Figure 2.8 : la transition entre le protocole réactif et proactif

2.2.4. Protocoles géographiques (position-based)

Les protocoles de routage géographiques se distinguent des précédents en intégrant une information supplémentaire dans la quête des itinéraires : la localisation géographique des nœuds du réseau.

2.3. Administration de l'énergie dans les réseaux ad hoc

Dans un réseau ad hoc, où les dispositifs échangent directement sans intermédiaire central, la régulation de l'énergie est essentielle. Les dispositifs mobiles de ces réseaux disposent de capacités énergétiques restreintes, rendant ainsi essentielle une gestion efficace de cette énergie pour maximiser la longévité des batteries et garantir le fonctionnement optimal du réseau.

On fait appel à diverses méthodes pour améliorer l'efficacité énergétique de ces réseaux :

1. Routage écoénergétique,
2. Planification astucieuse
3. Gestion de l'alimentation dynamique
4. Protocoles MAC (Contrôle d'Accès au Support) à faible consommation énergétique Mise en œuvre efficace du cache et de la mémoire locale.

En unissant ces diverses méthodes, on peut considérablement optimiser l'efficacité énergétique des réseaux ad hoc. C'est crucial étant donné les restrictions liées à la batterie dans ces contextes mobiles et décentralisés sans fil.

Le défi majeur consiste à « offrir une connexion fiable tout en réduisant la consommation d'énergie ».

3. Le Cloud Computing et service web

Le modèle de service de cloud computing offre un accès sur demande à une collection partagée de ressources informatiques (telles que les serveurs, le stockage, les applications, etc.) par le biais d'Internet. Carlier, A. (2022).

Les modèles de service cloud les plus courants sont :

- IaaS (Infrastructure en tant que Service),
- PaaS (Plateforme en tant que Service),
- SaaS (Logiciel en tant que Service).

Parmi ses bénéfices, on peut citer :

- La souplesse et l'expansion possible,
- La facturation basée sur l'utilisation,
- La centralisation et la virtualisation des ressources,
- L'accessibilité et la résistance aux défaillances.

« Le cloud computing est un modèle qui offre un accès réseau sur demande à un ensemble partagé de ressources informatiques configurables (comme les réseaux, les serveurs, le stockage, les applications et les services) pouvant être provisionnées rapidement et distribuées avec peu de gestion ou d'interaction nécessaire avec le fournisseur de services. »

- Le Cloud Computing est donc une idée qui vise à transférer les stockages et traitements informatiques, généralement effectués sur des serveurs locaux ou sur l'ordinateur de l'utilisateur, vers des serveurs distants. L'idée est de fournir des services informatiques en tant que service à la demande, disponible partout, à tout moment et pour tout utilisateur.
- Le concept clé à garder en tête est que le Cloud ne se résume pas à une collection de technologies, mais plutôt à un modèle d'offre, de gestion et d'utilisation des services et ressources informatiques situés dans des Datacenter.

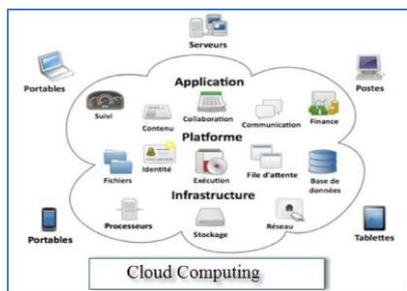


Figure 3.1 : Cloud Computing

3.1. Sécurité dans le Cloud Computing

Lorsqu'il s'agit de Cloud Computing, les responsables informatiques mettent souvent l'accent sur la sécurité et la conformité comme étant des préoccupations majeures, ces inquiétudes étant d'autant plus prononcées dans le contexte d'un Cloud public Khan, S. et al., (2023).

Depuis quelques années, ce point vulnérable dans le Cloud Computing est l'objet de recherches approfondies. Un organisme a été constitué pour établir des normes de sécurité dans le domaine du Cloud Computing. On désigne ce groupe sous le nom de CSA (Cloud Security Alliance). Certaines méthodes actuelles visant à renforcer la sécurité du Cloud Computing ont été mises en évidence grâce aux travaux de cette organisation. On peut mentionner parmi ces méthodes :

- La multi-localisation;
- Le cryptage;
- L'isolement des machines virtuelles.

La sécurité totale n'est pas une réalité, par conséquent, la question de sécurité est généralement une question de confiance entre le prestataire de services et l'utilisateur de ces services.

3.2. Les services web et les services de Cloud

On confond fréquemment les deux concepts scientifiques que sont les services web et le « cloud service » (ou service de cloud). Comme le démontre la (figure 3.1) (qui provient de la (figure 3.2)), un diagramme de Venn illustre les liens entre l'architecture orientée services, le cloud computing et les services web. Il est évident que le cloud computing est englobé par les services web, cependant, l'architecture orientée services peut exister sans eux, même si ces derniers constituent la meilleure option pour développer une SOA.

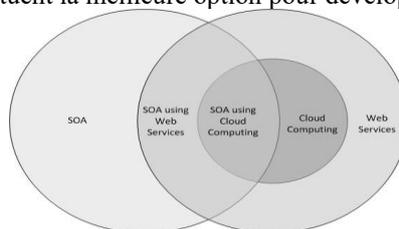


Figure 3.2 : Diagramme de Venn entre les services Web, SOA et le cloud computing

On considère le cloud computing comme la prochaine phase de l'évolution d'internet. En français, le terme « cloud », signifiant « nuage », indique que tous les éléments seront centralisés dans ce nuage : l'équipement, les applications, le stockage et même la mise en place du réseau. Matthieu Zarouk. (2013). L'objectif étant de fournir aux utilisateurs des services leur permettant d'accéder à leurs données et applications où et quand ils le souhaitent. Selon les utilisateurs visés, on distingue le cloud public accessible à tous, le cloud privé réservé à un usage exclusif et le cloud hybride.

Le projet fait appel aux services de [ex. Amazon Web Services (AWS)], essentiellement pour les motifs suivants :

- Une vaste gamme de services (calcul, base de données, stockage)
- Disponibilité élevée et présence à l'échelle mondiale.
- Intégration directe avec des outils DevOps et d'intelligence artificielle.

3.2.1. Architecture technique dans le cloud

L'architecture mise en place repose sur les éléments suivants :

- Des instances virtuelles pour le backend
- Un stockage d'objets pour les fichiers
- Une base de données dans le cloud (par exemple, PostgreSQL géré)

- Un équilibreur de charge pour la distribution des requêtes
- [Facultatif] Un système de conteneurs utilisant Kubernetes ou Docker.

3.2.2. Le Cloud Computing et les Acteurs

Le marché du cloud computing se compose de différents intervenants : les éditeurs et les fournisseurs.

1. Éditeurs :

Les entreprises qui proposent des solutions Cloud sont appelées éditeurs. Un éditeur n'est pas nécessairement un prestataire de services, en d'autres termes, son rôle ne consiste pas à proposer un service Cloud, mais plutôt à fournir une technologie permettant d'héberger une solution Cloud.

2. Fournisseurs :

Les fournisseurs de services de Cloud Computing sont des hébergeurs, Ils mettent à disposition des infrastructures physiques proposant une plate-forme de Cloud. Il serait bien trop conséquent d'analyser tous les acteurs du Cloud Computing présents sur le marché actuel. Nous survolerons les principaux acteurs:Salesforce.com, Amazon, Google, VMware et Microsoft. (Plouin, G., 2022).

3.2.3. Composants de l'Architecture Cloud

Une structure cloud standard se compose de plusieurs strates opérationnelles :

1. Interface utilisateur (Client)

Cette couche représente l'interface visible et utilisable par l'utilisateur final.

- Interfaces web ou mobiles (navigateur, application),
- APIs pour l'échange d'informations entre les services,
- Méthodes d'accès : HTTP/HTTPS

2. Partie serveur (Back-End)

Ici se trouve le noyau de l'application. Il inclut :

- Serveurs applicatifs : logiques de traitement, opérationnelles.
- Bases de données : relationnelles (MySQL, PostgreSQL) ou NoSQL (MongoDB, Firebase)
- Conservation d'objets : pour les fichiers, images, vidéos (par exemple : AWS S3)

3. Infrastructure de réseau

- Équilibreurs de charge : distribuent les requêtes entre différentes instances afin de répartir la charge,
- Réseau de diffusion de contenu (CDN) : améliore la distribution du contenu statique.
- VPN et pare-feu : assurent la sécurité des transmissions de données et l'accès aux ressources.



Figure 3.3 : Modèle d'une infrastructure cloud

C'est l'infrastructure d'une entreprise déployée en mode Cloud. Nous disposons de bases de données, de serveurs pour le traitement, d'associés, d'entreprises extérieures et de divers lieux, en plus des flux et concepts liés à la sécurité. Les clients utilisent les navigateurs web pour consulter l'information.

3.3. Structure Physique

L'infrastructure physique du Cloud consiste en une combinaison de serveurs, de dispositifs de stockage et d'éléments réseau agencés pour favoriser une expansion progressive dépassant celle qu'on obtient avec les infrastructures traditionnelles. Il est nécessaire de choisir ces éléments en fonction de leur aptitude à satisfaire les critères d'évolutivité, d'efficacité, de solidité et de sécurité.

La partie IaaS du Cloud Computing se compose de trois éléments clés :

- Section réseau comprenant des routeurs, des commutateurs et des pare-feu ;
- Section de stockage SAN qui englobe principalement des baies ;
- Section informatique comprenant des châssis qui hébergent des serveurs de type blade.

4. Intégration du Cloud et du Réseau Ad Hoc

L'intégration du cloud et du réseau ad hoc constitue ce que l'on nomme le Cloud Mobile Ad hoc (MAC) ou le Cloud Computing Ad hoc. Dans ce modèle :

- Les nœuds mobiles du réseau ad hoc ont la possibilité de mettre en commun leurs ressources locales (processeur, mémoire, énergie, bande passante) pour réaliser des tâches déportées,
- Le cloud distant joue un rôle dans le stockage, la planification globale et l'augmentation de la capacité de calcul.

Cette configuration hybride favorise :

- La diminution du temps de latence,
- L'économie en termes de bande passante,
- Une distribution améliorée de la charge de calcul,
- Une résilience renforcée face aux déconnexions.

4.1. Proposition d'un Architecture Cloud-Ad Hoc-web service

L'article présente un modèle d'intégration entre le web service, l'informatique en nuage, et le réseau ad hoc, connu sous le nom de Mobile Ad hoc Cloud. Dans ce modèle, les ressources du réseau local (nœuds mobiles) collaborent avec celles du cloud distant. Cela garantit une continuité du service tout en diminuant la latence dans les contextes où la connexion Internet est instable ou sporadique.

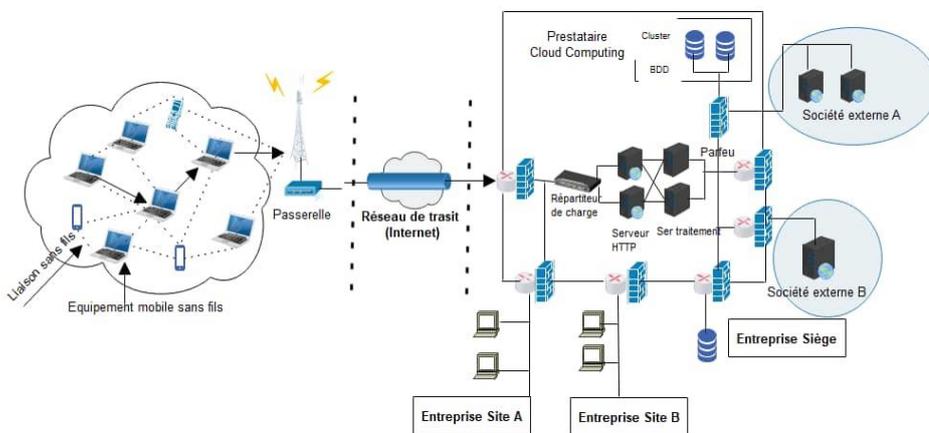


Figure 4.1. Architecture à intégration de services (réseau ad-hoc-cloud computing-service web)

4.2. Equilibrage de Charge dans un Environnement Multi-Queue

Dans un système à multiples files d'attente, plusieurs queues et serveurs gèrent des demandes en même temps. L'enjeu principal est de répartir judicieusement les tâches pour prévenir l'engorgement de certains serveurs et optimiser le délai de réponse général.

L'équilibrage de charge dans un contexte cloud-ad hoc associe :

- Des algorithmes adaptatifs de planification basés sur la disponibilité des nœuds,
- Des stratégies dynamiques d'attribution des tâches,
- Et des processus de notification et de transfert de tâches entre le réseau ad hoc et le nuage.

On vise à réduire :

- Le délai moyen d'attente des tâches,
- La proportion de paquets perdus,
- La congestion dans le réseau,
- Et la consommation énergétique au sein du réseau mobile.

Le routage dans les réseaux ad hoc (MANETs) nécessite une communication décentralisée où chaque nœud transfère les paquets, ce qui est compliqué du fait de la mobilité et de l'absence d'infrastructure. En revanche, il est probable que les réseaux 6G (sixième génération) utiliseront une technologie de routage sophistiquée et des protocoles hybrides (proactifs, réactifs) pour gérer un vaste éventail d'appareils et diverses formes de trafic, en incorporant essentiellement des fonctionnalités d'intelligence artificielle.

5. Défis de routage dans les réseaux ad hoc

1. Défis : Rendre le routage complexe est dû à la topologie dynamique, l'absence d'une structure centralisée et la mobilité incessante. Lorsqu'il n'est pas possible d'atteindre directement le nœud de destination, il est nécessaire de tracer des itinéraires en passant par des nœuds intermédiaires.

2. Procédures :

- Proactifs : Les nœuds actualisent constamment leurs tables de routage pour rester informés de la configuration du réseau, ce qui facilite des choix de routage rapides.

- Réactifs : Les tables de routage sont élaborées sur demande, ce qui peut occasionner des temps d'attente plus importants, mais prévient la surcharge du réseau.
- Hybrides : Fusion des protocoles proactifs et réactifs.

Illustrations : Les protocoles AODV (Ad hoc On-Demand Distance Vector) et OLSR (Optimized Link State Routing) figurent parmi les plus répandus.

6. Application d'Affectation de requêtes et équilibrage de charge

Cette nouvelle architecture d'intégration (cloud computing, service web et réseau ad-hoc), permet l'utilisation de l'application d'équilibrage de charge et de gestion de multi-file d'attente, dans le contexte de réseau ad-hoc cloud, cette application facilite l'affectation équilibré des requêtes afin de désengorger le système.

5.1. Algorithme d'affectation de requêtes et équilibrage de charge en python

```
# Début de L'algorithme en Python

# Entrée du nombre d'éléments
n = int(input("Entrer le nombre d'éléments n : "))

# Déclaration des tableaux
v = [0] * n
N_i = [0] * n
n_i = [0] * n

# Lecture de La constante N
N = float(input("Entrer la valeur de N (constante) : "))

# Lecture des valeurs v[i] et N_i[i]
for i in range(n):
    v[i] = float(input(f"Entrer v[{i+1}] : "))
    N_i[i] = float(input(f"Entrer N_i[{i+1}] : "))

# Calcul de La somme
somme = 0
for i in range(n):
    somme += v[i] * N_i[i]

# Calcul des n_i[i]
for i in range(n):
    n_i[i] = (v[i] * N) / somme

# Affichage des résultats
print("\n==== Résultats =====")
print("Somme :", somme)
for i in range(n):
    print(f"n_i[{i+1}] = {n_i[i]}")
```

Figure 6.1. : Algorithme d'affectation de requêtes en Python

5.2. Test de l'algorithme avec données d'essai.

```
Entrer le nombre d'éléments n : 3
Entrer la valeur de N (constante) : 100
Entrer v[1] : 2
Entrer N_i[1] : 5
Entrer v[2] : 3
Entrer N_i[2] : 4
Entrer v[3] : 1
Entrer N_i[3] : 6

==== Résultats =====
Somme : 28.0
n_i[1] = 7.142857142857142
n_i[2] = 10.714285714285714
n_i[3] = 3.571428571428571
```

Figure 6.2. : Test de l'algorithme et résultat.

L'utilisation de cette application dans l'intégration de ces services est une contribution au problème d'affectation de requêtes et équilibrage de charge dans le réseau ad-hoc cloud. Ce résultat montre comment l'application pourra affecter les requêtes à n serveur à travers le réseau ou le cloud selon leur catégorie.

Conclusion

L'intégration de ces paradigmes : **réseaux ad hoc**, **services web** et **cloud computing** constitue une démarche prometteuse pour l'équilibrage de charge dans des systèmes à **multi file d'attente**. Cette symbiose permet de tirer profit des avantages de ces paradigmes tel que : la flexibilité et de l'autonomie des réseaux ad hoc tout en exploitant la puissance et l'élasticité offertes par le cloud à travers le service web.

A travers cette étude, il a été prouvé que l'intégration de ces technologies favorise une meilleure répartition des tâches et équilibrage de charge, à l'aide de l'application mise en place pour une éventuelle diminution de délai d'attente, de la congestion dans le réseau et une rationalisation de l'utilisation des ressources clouding disponibles. Elle ouvre ainsi la voie à des environnements distribués plus efficace et résilients, capables de répondre aux besoins croissants d'utilisation dans des systèmes dynamiques et mobiles.

Dans l'avenir, l'évolution vers des réseaux ad hoc de 6G et l'amélioration continue des structures cloud donneront encore plus de possibilités pour des solutions innovantes d'équilibrage de charge et de gestion multi-file d'attente, boostant l'efficacité et la qualité des services dans des systèmes distribués complexes.

REFERENCES

- [1]. Benaini, Y. (2023). *Cloud-Ad hoc integration for 6G systems*, IEEE Communications Surveys & Tutorials.
- [2]. Carlier, A. (2022). *Déployer dans le cloud computing*. Paris : Éditions AFNOR / Eyrolles.
- [3]. Dordoigne, J. (2019). *Réseaux informatiques : notions fondamentales (protocoles, architectures, réseaux sans fil, virtualisation, sécurité, IPv6...)*. Saint-Herblain : Éditions ENI.
- [4]. Fauvarque-Cosson, B., & Zolynski, C. (2014). *Le cloud computing : l'informatique en nuage*. Paris : Société de Législation Comparée.
- [5]. Frikha, M. (2010). *Réseaux ad hoc : routage, qualité de service et optimisation*, Paris : Hermès Science Publications / Lavoisier.
- [6]. Ghandour, F. (2013). *Auto-organisation et routage dans les réseaux ad hoc*. Bruxelles : Presses Universitaires Européennes.
- [7]. Ghandour, F. (2015). *Les réseaux ad hoc sans fil*. Saarbrücken : Éditions Universitaires Européennes.
- [8]. Gupta, M. and Singh, R. (2024). *Ad hoc cloud computing for distributed IoT systems*, Journal of Cloud Computing.
- [9]. Khan, S. et al., (2023). *Load balancing in mobile cloud computing: A survey*, IEEE Access, vol. 10.
- [10]. Loïc Volant, (et autres). *Le réseau avec Microsoft Azure : Déployez, hybridez et sécurisez vos réseaux dans le cloud*, Éditions ENI.
- [11]. Matthieu Zarouk. (2013). *Cloud computing Maîtrisez les services Web d'Amazon (AWS)*, Éditions ENI (Collection Epsilon).
- [12]. Nirbhay Chaubey, (2025). *Sécurisation du protocole de routage AODV dans la conception de réseaux mobiles ad hoc*, Éditeur : Éditions Notre Savoir.
- [13]. Olamide, K. (2024). *Protocole de routage hybride pour gérer la congestion dans les réseaux Ad-hoc*, Éditions Notre Savoir.
- [14]. Oliveira, A. L. (2025). *Évaluation des stratégies d'équilibrage de charge : stratégies maître-esclave pour les clusters et les grilles informatiques*. Paris : Éditions Notre Savoir.
- [15]. Plouin, G. (2022). *Cloud et transformation digitale : SI hybride, protection des données, anatomie des grandes plateformes*, Éditeur : Dunod (Collection InfoPro). 6^e édition,
- [16]. RISSON, F., et GAONA, N., (2004) *Le routage au sein des Réseaux AdHoc*. Ed. Université de peau et de pays d'Adour., 68.
- [17]. Rivard, F. (2012). *Cloud Computing : le système d'information sans limite*. Paris : Hermès Science Publications.
- [18]. Sondi Obwang, P. (2023). *Conception et évaluation de protocoles de routage ad hoc*. Londres : ISTE Éditions.
- [19]. Turjman, F. Al., (2022). *AI-based task allocation in mobile ad hoc networks*, Computer Networks,
- [20]. Xavier, B. & Henri, G. C. (2003). *SAN et NAS : solutions de stockage, sécurité, infrastructures*, Édition Dunod,
- [21]. Zhang, L.(2025). *Multi-queue load balancing in distributed systems, Future Generation Computer Systems*.