

MTI840 - RESUME D'UN ARTICLE SCIENTIFIQUE

BAPTISTE VIERA | VIEB92370001

REFERENCES DE L'ARTICLE :

He, W., Guo, S., Liang, Y., & Qiu, X. (2019). Markov Approximation Method for Optimal Service Orchestration in IoT Network. IEEE Access, 7, 49538–49548.
<https://doi.org/10.1109/access.2019.2910807>

TABLE DES MATIERES

CONTEXTE GENERALE

PROBLEMATIQUE ET MOTIVATIONS

DESCRIPTION DE LA SOLUTION PROPOSEE

PRINCIPAUX RESULTATS OBTENUS

CRITIQUES ET TRAVAUX FUTURS

Contexte général

Dans un contexte d'une multiplication des applications IoT et de l'avancement rapide de la 5G, l'edge computing, les Network Functions Virtualization (NFV), les méthodes distribuées, et les exigences en matière de qualité d'expérience (QoE) sont et seront essentiels à l'avenir. Il existe un besoin de proposer des solutions efficaces, scalables et flexibles.

Problématiques et motivations

Des études ont été alors menées pour déterminer le nombre optimal de Virtual Network Function (VNF), le placement de VNF avec Plusieurs Instances (VPI), le chemin d'acheminement optimal du trafic avec des méthodes centralisées. Mais, ces travaux font face aux limites suivantes. Le placement de VPI est un problème NP-difficile et les algorithmes centralisés ont besoin d'énormes ressources de calcul et de mémoire pour une seule machine physique. L'article étudié présente alors une méthode distribuée pour le placement optimal des VPI pour minimiser les coûts, les délais et garantir l'équilibrage de la charge du réseau.

Description de la solution proposée

Les auteurs ont approximé le modèle d'optimisation suivant « $Min (coût(serveur) + coût(liens))$ » pour résoudre le problème de façon distribuée. Ils ont alors proposé un algorithme de placement de VNF à instances multiples basé sur la chaîne de Markov. L'algorithme nommé VPMIA se décompose en trois phases : initialisation de la procédure (1), transition / réglage du « timer » (2) et réinitialisation (3). Lors de l'étape (1), les instances de VNF sont placées aléatoirement dans les serveurs disponibles sur le chemin actuel en fonction de contraintes (CPU, mémoire, transmission, réplication) et sont associés à un « $cout(a)$ ». L'étape (2) augmente, si possible, le nombre d'instances de VNF par réplication qui est associé au « $cout(a')$ ». Un « timer » est généré pour déterminer s'il faut mettre à jour l'état « a' » à partir de « a ». Si le « timer » expire, on passe à l'étape (3). Les instances des VNFs et les chemins correspondants sont alors choisis comme schéma de réplication. Les étapes (2) et (3) sont répétées pour minimiser les coûts jusqu'à convergence.

Principaux résultats obtenus

VPMIA est comparé à deux algorithmes qui sont le Random Placement Algorithm (RPA) et le Single Path Algorithm (SPA). Par exemple, pour 120 chaînes de services (SC), le coût total de SPA et RPA sont respectivement 31% et 21% supérieur à celui de VPMIA qui planifie le nombre d'instance et utilise une méthode distribuée. Ensuite, les taux moyens d'utilisation des liens et serveurs, avec 100 SCs, sont environ, pour le SPA et RPA, respectivement 50% et 20% inférieurs à celui du VPMIA qui réussit à bien équilibrer la charge sur le réseau.

Critiques et travaux futurs

Les limites sont que VPMI est avantageux uniquement lorsque le nombre de SCs est élevé et que les résultats ont été obtenus à partir d'une simulation et non en conditions réelles. Comparer l'efficacité des techniques de reinforcement learning et considérer la résilience de l'intégration de la SC avec le concept de découpage en tranches pourraient être pertinent.