

L'ordonnancement de tâches occupe une place importante dans les réseaux fog, le but est de placer les tâches dans le réseau de façon à minimiser la latence, le temps d'exécution et dans certains cas, la consommation énergétique. La latence étant le temps de réponse du réseau et le temps d'exécution peut être celui d'une tâche ou d'un ensemble de tâches.

La logique floue quant à elle, tente de se rapprocher de la pensée humaine en définissant des variables caractérisées par des modalités subjectives ("petit", "grand", etc.). Ces modalités sont représentées par des intervalles de valeurs distincts, par exemple :

- "petit" < 165cm,
- 160cm < "moyen" < 175cm,
- 170cm < "grand".

Le fait de convertir des valeurs discrètes en variables floues est appelé la *fuzzification*, le processus inverse est appelé la *défuzzification*. Des règles viennent ensuite s'ajouter à fin d'attribuer une sortie à une ou plusieurs entrées. La logique floue vise à prendre en compte l'incertitude se trouvant entre ces intervalles, un individu mesurant 163cm serait à la fois dans l'intervalle "petit" et "moyen" avec un certain taux d'appartenance à chaque intervalle. Ces règles sont définies en langage naturel et se présente généralement sous la forme "SI _ est _ [ET/OU _ est _] ALORS _ est _".

Dans le cadre d'une expérience d'*offloading* ("déchargement" en français) de tâches fog à l'aide de logique floue j'ai utilisé le logiciel QtFuzzylite 4.0. Le but de cette expérience est de déterminer le lieu d'exécution d'une tâche, c'est-à-dire, si l'exécution se fait en local sur l'appareil mobile ou si elle est envoyée sur le réseau fog.

Grâce à l'interface graphique de QtFuzzylite, j'ai commencé par configurer les différentes variables d'entrée et de sortie ainsi que les règles de logique. Après cela, j'ai effectué des tests en manipulant les curseurs et je me suis vite rendu compte des limites du logiciels. Ce processus étant entièrement manuel, générer un grand nombre de résultats allait prendre du temps, de plus les règles établies étaient trop simples pour refléter correctement la nature du problème.

J'ai donc décidé de créer un script Python pouvant générer automatique un grand nombre de résultats. Pour cela, j'utilise la librairie pyfuzzylite, me permettant de créer une application de logique floue, et la librairie Numpy me permettant générer des valeurs aléatoires pour chacune des variables existantes. J'ai commencé par tester le script avec les règles existantes avant d'essayer d'établir des règles plus complexes.

Afin de créer de nouvelles règles plus complexes, j'ai établi une carte mentale à l'aide du logiciel Xmind dans le but de visualiser le plus de combinaisons possibles pour ensuite les implémenter dans le script Python. Des tests plus poussés sont en cours.