

# Rapport Benchmarking DHT

Simon Wicky (260589)

Jeremy Mion (261178)

05.2018

## 1 Benchmarking : Implémentation et scénarios testés

### 1.1 Implémentation

L'implémentation de ce benchmarking en C est très brève. Elle consiste juste à chronométrer les phases réseau, c'est à dire la requête put ainsi que sa réponse, et la requête get, ainsi que sa réponse. Les différents scénarios sont ensuite exécutés par l'intermédiaire de script bash, détaillés ci-après.

### 1.2 Scénarios testés

Les différents scénarios ont été testés :

#### 1.2.1 Put, couple clé-valeur fixe, $N = 10$ , $W$ variable

Nous avons simulé 100 fois l'action de mettre une valeur sur la Hashtable, avec  $N$  valant 10, et  $W$  variant entre 1 et 10.

Le script `./test_put` effectue une action put, avec  $W$  partant de 1 jusqu'à 10. Le résultat est écrit dans un fichier.

Le script `./test_put_csv` se charge de lancer 100 fois `./test_put`, de collecter les résultats et de les fusionner dans un seul fichier au format csv, où chaque ligne correspond à une valeur de  $W$ .

Les résultats récoltés sont résumés dans la Figure 1.

#### 1.2.2 Get, couple clé-valeur fixe, $N = 10$ , $R$ variable

Nous avons ensuite simulé 100 fois l'action de demander une valeur sur la Hashtable, avec  $N$  valant 10, et  $R$  variant entre 1 et 10.

Le script `./test_get` effectue une action get, avec  $R$  partant de 1 jusqu'à 10. Le résultat est écrit dans un fichier.

Le script `./test_get_csv` se charge de lancer 100 fois `./test_get`, de collecter les résultats et de les fusionner dans un seul fichier au format csv, où chaque ligne correspond à une valeur de  $R$ .

Les résultats récoltés sont résumés dans la Figure 2.

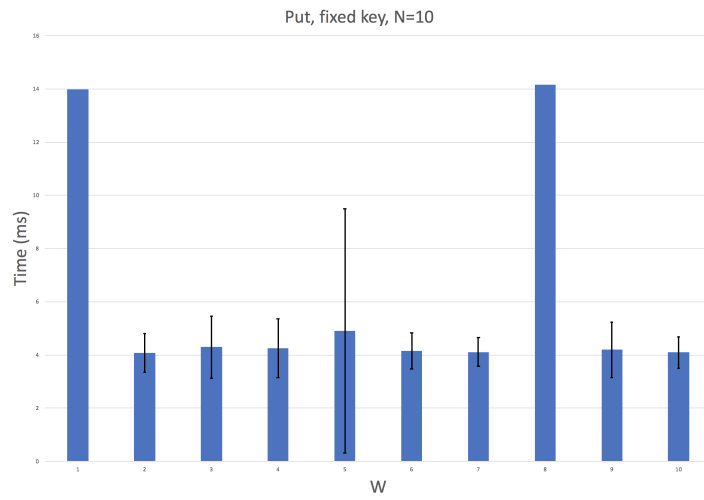


Figure 1: Graphique du scénarios 1.2.1

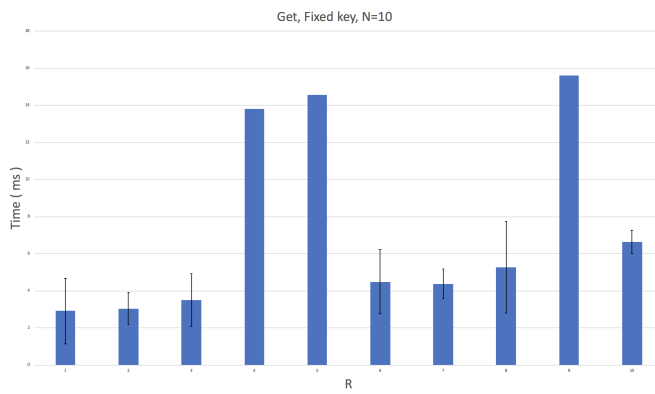


Figure 2: Graphique du scénarios 1.2.2