

Rapport projet de simulation :

BERNOT Yoann 21403002

Résumer du projet :

L'objectif du projet est de simulation 3 modèle différent de file d'attente :

- une M /M/N
- une M/M/1 x10
- un mixe entre les 2 premiers

il y a deux manières d'interpréter le model 3 :

- soit le client choisi le serveur qui a fait le moins attendre ses client jusque la le temps d'attente que le serveur a imposé à ses clients et le temps qu'il lui reste a imposé n'étant pas lie. Ce choix revient à choisir au hasard
- soit il choisit le serveur qui sera disponible le plus tôt ce qui est la même chose que le model 1 à la différence que au lieu d'attendre dans une file centralisé il attend dans la file du serveur.

Exécution et code

Nous avons choisi de faire 3 programmes indépendants afin de pouvoir les lancer ensemble ou séparément avec le MakeFile. Nous avons toutefois décidé d'utiliser le même nom pour les variable afin de ne pas se perdre lors de la lecture des différents programme.

Le fichier MakeFile permet de compiler et exécuter le ou les programme(s) facilement sans avoir à rentrer la commande pour compiler et exécuter manuellement. On a aussi ajouter la possibilité de créer des graphes avec le logiciel R pour voir visuellement l'allure des courbes.

Les programmes nécessitent la présence du fichier « lambda.txt » qui contient tous les valeurs de lambda que nous voulons tester. Nous avons choisi de tester 8 valeurs de lambda pour que l'exécution des 3 programmes ne soit pas trop longue.

A la fin de d'une des simulation, le programme va créer un fichier nommé « resultatx.txt » (remplace x par le model) qui contient :

- Les valeurs de lambda
- $E[A]$ (version expérimental)
- t_{90} (version expérimental)
- $E[A]$ (version théorique)
- t_{90} (version théorique)

Une fois les fichier générer, on peut lancer le scripte R pour générer plusieurs graphe (explication plus détail des résultats par la suite):

- Le temps moyen d'attente en fonction de lambda des 3 courbes (fichier « tempsMoy.jpg »)
- Le t_{90} du temps d'attente en fonction de lambda des 3 courbes (fichier « t90.jpg »)

A cause des résultats, nous avons aussi ajouter des courbes afin de voir ce qu'on aurai du avoir :

- Le temps moyen d'attente expérimental et théorique en fonction de lambda pour chaque courbe (fichiers « attenteM1.jpg », « attenteM2.jpg » et « attenteM3.jpg »)

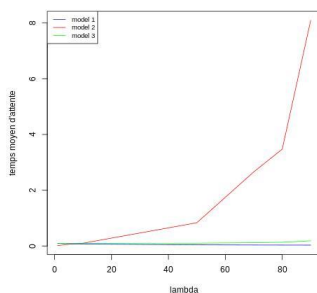
Nous n'avons pas réussi à calculer les valeurs théorique pour le model 3, toutes les valeur théorique ont été fixé à 0.

Ligne de commande à utiliser pour faire fonctionner le programme :

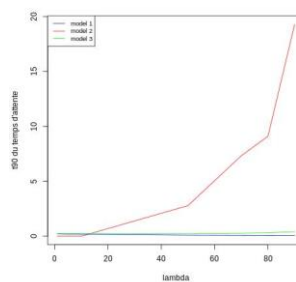
- *make runX* remplacer X par 1,2 ou 3 pour exécuter le model 1, 2 ou 3
- *make runR* va exécuter les 3 modèle puis exécutera le script R
- *make R* va juste exécuter le script R

Explication des courbes

Nous avons des complications dans le model 1 et 3 qui ont faussé les résultats expérimentaux



« tempsMoy.jpg »



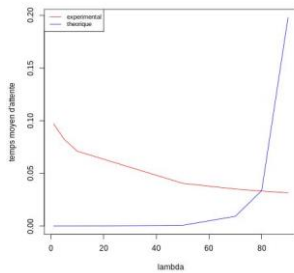
« t90.jpg »

Comme on peut le voir sur les 2 graphes, les courbes du model 1 n'est pas bonne. De plus, à cause de la courbe 2 on ne voit pas bien la courbe du model 3

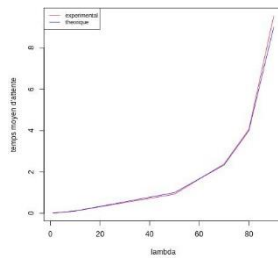
La courbe du model 2 et 3 sont croissantes en fonction de lambda. Cette augmentation est logique car plus on augmente lambda, plus l'écart entre 2 clients est petit et plus il y a de clients, et donc la file se remplit et le temps d'attente moyen d'attente augmente. De même avec le t_{90} , comme il y a plus de clients dans la file, il y a plus de clients qui attendent et donc le temps d'attente des clients augmenter ce qui fait augmenter t_{90} .

Résultat théorique

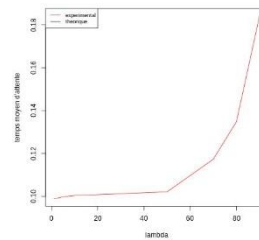
Comme dit précédemment les résultats expérimentaux du model 1 et 3 ne sont pas bonne, nous avons quand même pu étudier le fonctionnement du model 1 en observant les résultats théoriques.



« attenteM1.jpg »



« attenteM2.jpg »



« attenteM3.jpg »

Dans le fichier « attenteM1.jpg », on voit clairement que les résultats expérimentaux ne sont pas bonne car ce n'est pas logique d'ajouter plus de client dans le système et de voir que le temps moyen d'attente des clients et réduit. Cependant, grâce à la courbe théorique, on peut déduire que l'allure de la courbe est la même que celle du model 2 sauf que le temps moyen pour le model 1 quand lambda est grand est plus petit que celle du model 2.

La courbe du model 3 a la même allure que les 2 autres model, en regardant plus particulièrement l'échelle des ordonnées, celle du model 3 est plus petite que celle du model 1. On peut en déduire que le model 3 est plus performant que le model 1.

Pour comprendre pourquoi, on s'appuie sur une file m/m/1 comme exemple (1 seul serveur).

Pour que le système soit stable (et donc testable), il faut qu'il y ait moins d'arrivées que de sorties.

L'augmentation du temps d'attente vient des irrégularités dans les statistiques.

Dans une période, il peut y avoir subitement plus de clients qui arrivent ou bien ils peuvent être servis plus lentement, ce qui génère de l'attente.

Il est aussi possible que le serveur serve vite et qu'il attende sans client, c'est du temps d'attente négatif (qui n'a pas de sens dans la vraie vie)

Dans le cadre d'une m/m/10

les irrégularités seront gommées par le nombre de serveur.

Là où il fallait 1 irrégularité pour faire grandir la file d'attente (pour m/m/1)

il en faut maintenant 10, ce qui réduit drastiquement les probabilités que cela arrive.

Sans irrégularité, les clients arriveraient tous les $1/\lambda$ et seraient servis tous les $1/\mu$ lambda étant plus petit que mu, ils n'attendraient jamais.

Le mode de fonctionnement 2 ne bénéficie pas de cette protection car les clients sont placés aléatoirement sur des serveurs.

Voyons le problème différemment: si on donne à chaque client

son numéro de file avant qu'il ne commence à attendre, on se retrouve alors avec 10 files distinctes $m/m/1$ et le temps d'attente moyen de notre système est la moyenne du temps d'attente moyen de ces 10 files.

Pour le mode de fonctionnement 3, on améliore le système d'une $M/M/10$ grâce au choix de la file la moins pleine pour le client.