Projet simulation - Rapport

Maxime Gonthier (21500231) - Benjamin Guillot (21500545)

28 mai 2019

Table des matières

1 Introduction

L'objectif de ce projet est de simuler en temps discret des arrivées et des services dans un cyber café. On en déduira des mesures d'évaluations à l'aide du calcul du temps moyen d'attente et du 90ème percentile du temps d'attente.

2 Explication de la programmation

2.1 Entrée et sortie du progamme

En entrée le programme prend un fichier texte contenant les données que l'on veut faire varier dans notre application (ici lambda). On obtient en sortie 5 fichier :

— Result_modele1.txt
— Result_modele2.txt
— Result_modele3.txt
— resultE.txt
— result90.txt

Les 3 premiers fichiers contiennent pour chaque valeurs de lambda le temps moyen d'attente $E[A]$ et le 90 percentile du temps d'attente $t_{90}E[A]$.
Les deux derniers fichiers sont utilisés pour l'affichages des courbes obtenues après la simulation.

2.2 structure du progamme

Nous avons choisit pour la programmation de gérés les différents modèles dans les fonctions *Arrivee_Client* et *service_event*. Ces deux fonctions on 3 modes de fonctionnement passé en argument pour savoir quel modèle on est en train de simuler. Il y a un simulateur par modèles. Il sont appelés dans le main avec en argument :

— un fichier dans lequel écrire les données relatives a la simulation.
— la valeur de lambda en train d'être testée.

2.2.1 premier modèle

le premier modèle représente une M/M/N classique. on traite donc les clients dès qu'un serveur est libre.

2.2.2 second modèle

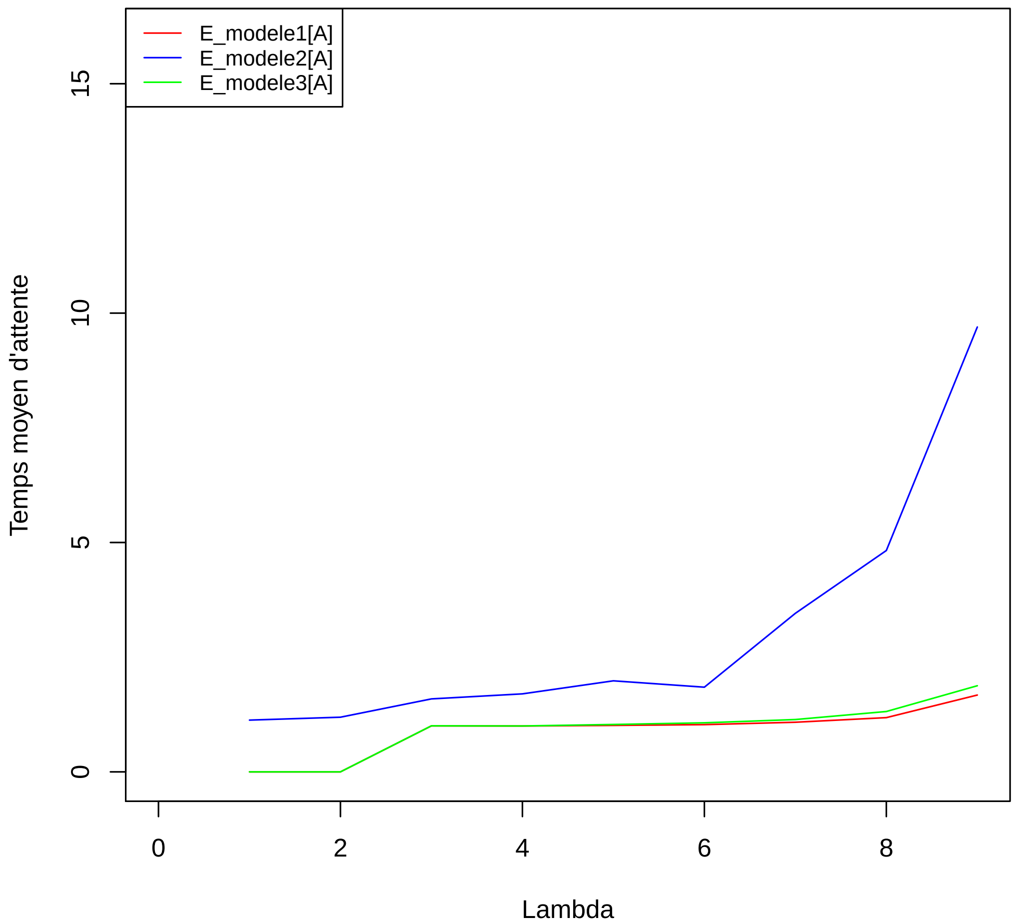
Le second modèle peut être simulé avec une M/M/1. On ajoute une condition aux arrivées de client qui est un pourcentage de chance d'arriver dans la file, comme on a 10 serveurs en theorie, un client arrive dans une file avec une probabilité $\frac{1}{10}$.

2.2.3 troisième modèle

Le troisième modèle est simulé avec 10 M/M/1. On stocke dans un tableau le nombre de client par file et a chaque arrivée de client, on vérifie quelle est la file la moins rempli. Le client est envoyé sur cette file. On stocke dans l'évènement la file sur laquelle se trouve le client afin de pouvoir décrémenter le nombre de client de cette file après un service.

3 Analyse des résultats

Avant tout de chose, notons que pour le modèle 2, les valeurs que nous avons obtenus pour le temps moyen d'attente correspondent à 1/10 des clients d'une M/M/1. Ce qui revient au final à créer 10 M/M/1. Nous avons choisis cette ordre de grandeur plus petit pour pouvoir dessiner les 3 courbes sur le même graphe sans avoir à déformer énormément l'echelle. En effet sinon nous aurions dû multiplier par 10 toutes les valeurs du temps moyen d'attente pour le modèle 2.

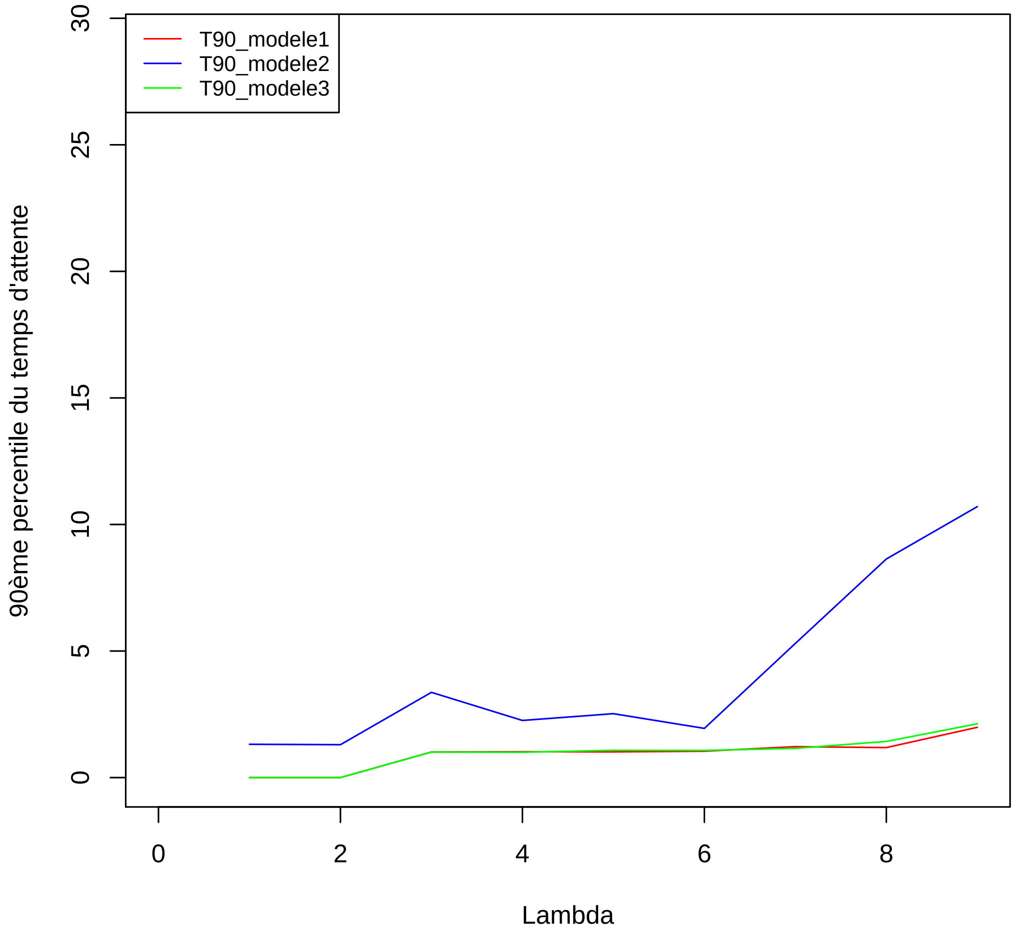


On observe que le temps moyen d'attente du modèle 2 augmente significativement à partir de lambda = 6. En effet il passe de moins de 3 à lambda = 6 à plus de 5 à lambda = 8. Cela s'explique par la manière dont se calcule le temps moyen d'attente dans une M/M/1. En effet on fait :

$E[A] = \rho * \frac{\frac{1}{\mu}}{1-\rho}$
avec : $\rho = \frac{\lambda}{\mu}$
Donc plus lambda augmente, plus la valeur de rho est grande est donc plus le temps moyen d'attente est grand. A condition bien evidemment que μ ne soit pas trop élèvé, ce qui rendrait toute les valeurs de E[A]quasiments similaires à cause du $\frac{1}{\mu}$

On observe que les modèles 1 et 3 ont des valeurs similaires de temps moyen d'attente. Cela s'explique par la similarité de leurs fonctionnement. En effet le modèle 1 il y a une file avec 10 serveurs et dès qu'un serveur est libre on fais un service. Dans le 3 il y a 10 files et les clients vont dans la file la plus vide. Ainsi dans les deux cas les clients sont servis dès qu'une place est libéré, que ce soit un serveur ou une file qui se libère. Cela explique bien la ressemblance des deux courbes. Nous notons aussi que les temps moyen d'attente sont plus faibles pour le modèle 1 et 3 que pour le modèle 2. Cela vient du fait que le patron choisit aléatoirement un ordinateur, cela peut donc de temps en temps ralentir le service car un client sera mis sur un ordinateur ou il y a déjà des clients qui attendent. Et donc cela augmentera le temps moyen d'attente de manière significative pour un grand nombre de clients.

Le graphique des 90ème percentiles :



On observe les mêmes résultats que pour le graphique précedent à l'exception de ce sursaut pour le modèle 2 à lambda = 3. Cela s'explique par une "malchance" qui a fais que plusieurs clients ont été mis sur des files dont le temps d'attente fût long comme expliqué au dessus, ce qui a donc augmenter le nombre de valeurs extrèmes dans le temps d'attente des clients. Or le 90 ème percentile prend la valeur dont 90% des valeurs sont inférieures, donc si 10% des valeurs des temps d'attente sont extrèmes, alors le 90ème percentile le sera aussi.

4 Explication des résultats théoriques

5 Conclusion

Pour ce projet, nous aurions pu simplifier le code en utilisant un seul simulateur pour les 3 modèles, en ayant plusieurs cas d'usage. Nous aurions également pu découper le code en plus de fonctions afin de rendre celles-ci plus courtes et plus facilement compréhensible.