

# MPIS TP<sub>3</sub>

## THÉORIE DES FILES D'ATTENTE

Ballot Corentin, Simon Camille

05/12/2016

Ce compte-rendu présente l'implémentation d'une file d'attente.

L'archive MPIS-TP3-CorentinBallot-CamilleSimon.zip contient :

- Les fichiers java des classes
- Un fichier nommé *gnuplot* contenant les commandes pour générer le(s) graphique(s)
- Des fichiers textes contenant les données pour gnuplot
- Les graphiques utilisés pour ce compte-rendu

Le sujet du TP est accessible ici : [Sujet TP3](#)

### 1 Rappel du sujet

Les files d'attentes sont rangées en plusieurs catégories en fonction de plusieurs critères. La notation de Kendall permet d'écrire de façon condensée les caractéristiques de ses files. Cette notation est de la forme suivante :

$$A / S / C / K / M / Z$$

où :

- A est la loi suivis par les processus entrant dans la file d'attente
- s est la loi suivis par les processus sortant de la file d'attente
- C est le nombre de serveurs
- K est la capacité total du système, comprenant la file d'attente et les serveurs
- m est le nombre de processus, il peut être infini
- Z indique le mode de service

Nous nous intéressons ici à la modélisation d'une file M/M/1, c'est à dire que ses processus d'entrée et de sortie suivent une loi exponentielle et qu'il n'y a qu'un serveur. La capacité total du système n'étant pas spécifié, elle est supposé infini de même pour le nombre de processus. La discipline de service par défaut est *FIFO*, "premier arrivé, premier servi".

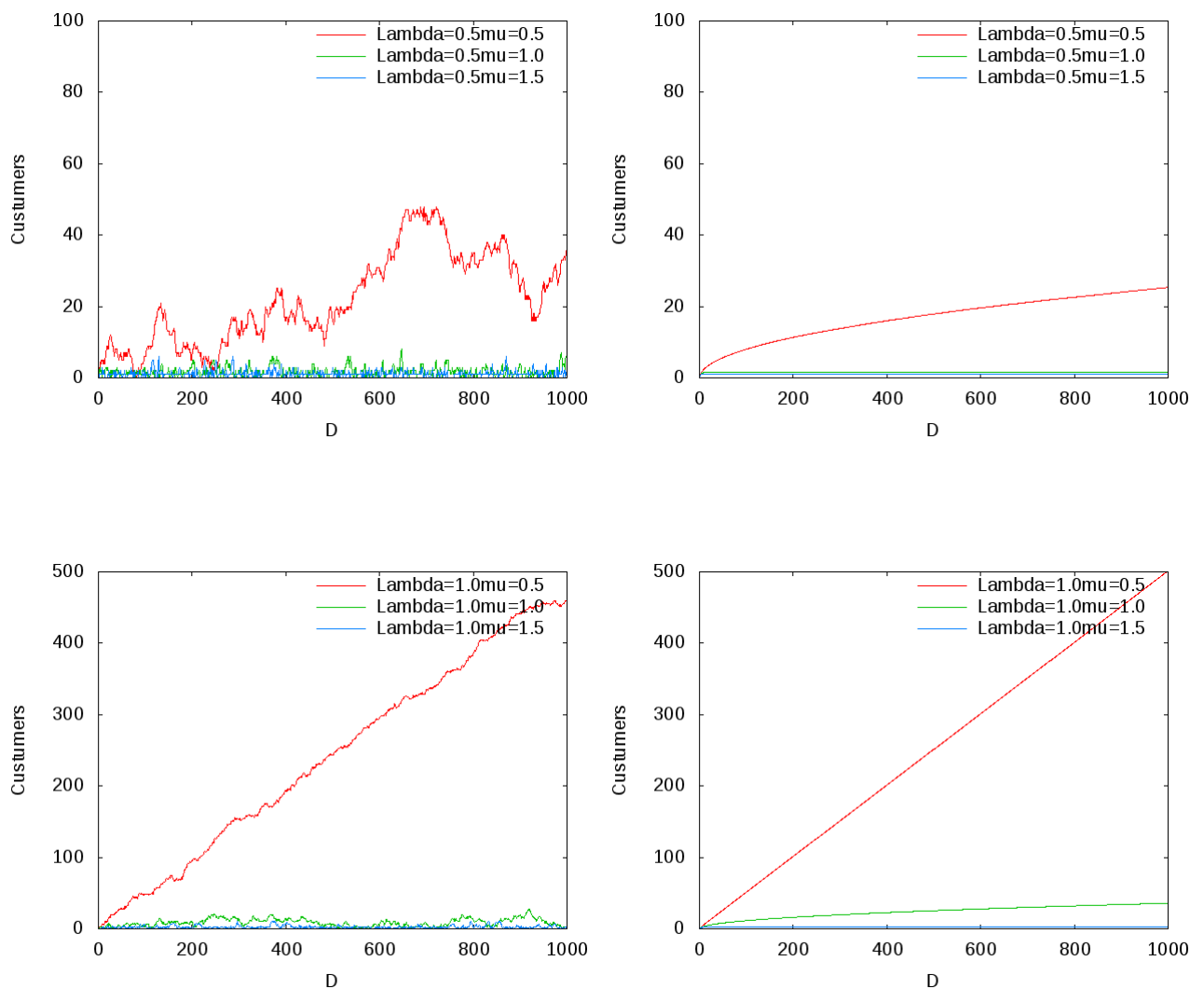
## 2 Modélisation de la file d'attente

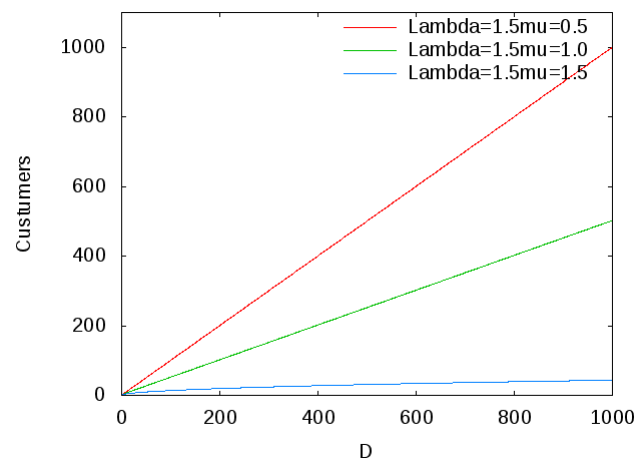
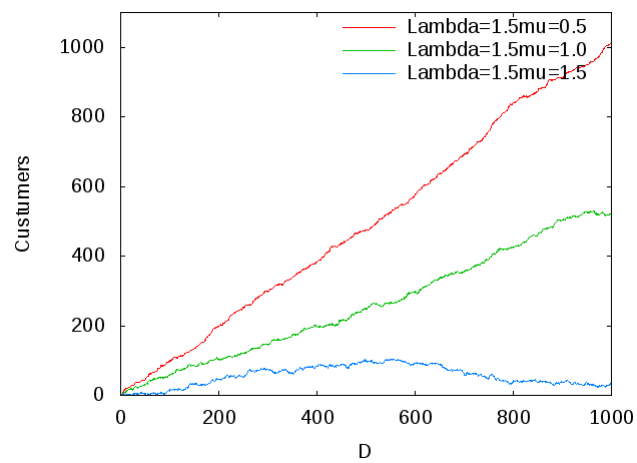
Dans cet exercice, nous modélisons la file d'attente par un processus de naissance et de mort.  $\lambda$ , le taux de naissance, correspond à l'arrivée d'un processus dans la file d'attente et  $\mu$ , le taux de mort, au taux de service.

Ses deux taux suivent une loi exponentielle, nous avons donc réutilisé la classe java du TP précédent qui modélisait la loi exponentielle.

La méthode *compute* génère une nouvelle arrivée ou un nouveau départ. Elle est utilisée dans la méthode *generate* qui génère l'ensemble des états de la file d'attente. Cette dernière crée un tableau où chaque case contient le nombre de processus en attente dans la file à chaque instant. Le nombre de période de temps est passé en paramètre de la fonction.

En faisant varier  $\lambda$  et  $\mu$ , on obtient les courbes suivantes :





### **3 Conclusion**

Au cours de ce TP, nous n'avons pas rencontré de problème particulier. N'ayant pas d'exemple de courbes, nous ne sommes pas sûr du résultat obtenu. Cependant nous avons suivis l'énoncé du TP et nous avons choisis d'afficher les courbes de l'état de la file d'attente.

Par la suite, on peut imaginer utiliser d'autre loi pour les processus d'entrée et de sortie de la file d'attente.