

M1 SIRIS	Evaluation de performances
<p style="text-align: center;">TP</p> <p style="text-align: center;">Simulation discrète d'une file M/M/1</p>	

**Objectifs :** Etude du comportement d'une file M/M/1 par simulation discrète. Programmation en java de la gestion de l'échéancier. Recoupement par simulation avec les grandeurs théoriques vues en cours pour ce type de file d'attente.

## Généralités

La chaîne de Markov d'une file d'attente M/M/1 est un processus de naissance et de mort. Les arrivées sont déterminées par un processus de Poisson (inter-arrivées suivant une loi exponentielle de paramètre  $\lambda$ ) et les temps de service suivent une loi exponentielle de paramètre  $\mu$ .

Les simulations seront effectuées avec  $\lambda < \mu$ , de manière à obtenir une stabilité en régime permanent.

## Durées suivant une loi exponentielle

Rappels : la loi exponentielle de paramètre  $\lambda$  a pour densité de probabilité

$$p(X = x) = \lambda e^{-\lambda x} (x > 0)$$

Une propriété intéressante est qu'une variable aléatoire  $X$  suivant une loi exponentielle de paramètre  $\lambda$  a pour espérance  $E(X) = 1/\lambda$ .

Pour tirer des durées suivant une loi exponentielle de paramètre  $\lambda$ , on utilise la fonction de répartition  $F(X) = 1 - e^{-\lambda X}$ . On sait que  $F(X) \in ]0, 1[$ . Il suffit donc de tirer une variable aléatoire  $R = F(X)$  entre 0 et 1, et de trouver le  $X$  correspondant. On a alors

$$R = 1 - e^{-\lambda X} \Leftrightarrow 1 - R = e^{-\lambda X} \Leftrightarrow \ln(1 - R) = -\lambda X \Leftrightarrow X = -1/\lambda \cdot \ln(1 - R)$$

En définitive, les inter-arrivées ou durées de service suivant une loi exponentielle de paramètre  $\lambda$  peuvent être exprimées par :

$$t_{i+1} = t_i + \frac{-\log(1 - x)}{\lambda} (x : v.a \text{ uniforme dans } ]0, 1[)$$

## Evénements et échéancier

Dans le cas d'une file M/M/1, il y a deux types d'événements : les arrivées et les départs. Pour initialiser l'échéancier, on y placera simplement un événement de type arrivée à  $t=0$ .

Le traitement d'un événement de type arrivée pour le client  $n$  appelle la création de deux nouveaux événements :

- l'arrivée future du client  $n+1$  (inter-arrivées suivant une loi exp. de paramètre  $\lambda$ , voir paragraphe précédent). L'un de vos paramètres de simulation sera la durée simulée : il ne faut donc pas créer d'événements d'arrivée dont la date serait ultérieure à la date de fin de simulation, sans quoi la simulation serait sans fin.
- la sortie future du client  $n$  (durée de service suivant une loi exp. de paramètre  $\mu$ , voir paragraphe précédent). Pour prédire l'heure de sortie, deux cas de figure
  - o soit la file est vide : dans ce cas c'est facile car il suffit de tirer une durée de service et de l'ajouter à l'heure actuelle (c'est-à-dire l'heure de l'arrivée du client  $n$ )
  - o soit la file n'est pas vide : la date de sortie du client  $n$  dépend donc de la date de sortie du client  $n-1$ . Pour simplifier les choses, puisque les arrivées sont traitées dans l'ordre, vous pourrez utiliser une variable permettant de mémoriser la date la plus tardive de sortie calculée jusqu'à présent.

La simulation se termine lorsque l'échéancier est vide.

## Implémentation

Vous implémenterez le processus de simulation en langage java, en créant les classes suivantes :

- Evt : classe des événements
- Ech : classe de l'échéancier. Vous pourrez utiliser la classe *LinkedList* de l'API java pour stocker les objets Evt dans une liste chaînée.
- Stats : classe regroupant la collecte et l'affichage des résultats de simulation
- Utile : classe regroupant toutes les méthodes statiques utiles (tirage aléatoire, loi exponentielle, ...)

Les paramètres d'appel seront  $\lambda$ ,  $\mu$ , la durée de simulation et un commutateur permettant d'entrer en mode débogage (c'est-à-dire afficher chaque événement traité).

## Résultats

L'objectif de cette simulation est de retrouver de manière statistique certaines valeurs théoriques vues en cours pour la file M/M/1 :

- probabilité de service sans attente (file vide lors d'une l'arrivée) :  $1 - \frac{\lambda}{\mu}$
- taux d'utilisation = probabilité que la file soit occupée :  $\frac{\lambda}{\mu}$
- débit :  $\lambda$
- nombre moyen de clients dans le système :  $\frac{\rho}{(1 - \rho)}$
- temps moyen de séjour dans le système :  $\frac{1}{\mu(1 - \rho)}$

Les résultats de simulation ne sont pas exacts : vous pourrez le constater en faisant varier la durée de simulation et en mesurant la différence aux résultats théoriques, et en calculant les variances ou écarts-types des résultats obtenus de plusieurs simulations.

## Exemple d'exécution

```
>java MM1 5 6 1000 1
Date=0.0 Arrivee client #0
Date=0.03627054952825975 Depart client #0 arrive a t=0.0
Date=0.3129462895233852 Arrivee client #1
Date=0.3229906252413475 Depart client #1 arrive a t=0.3129462895233852
Date=0.424299442998096 Arrivee client #2
Date=0.575492786687346 Depart client #2 arrive a t=0.424299442998096
Date=0.8832839796147136 Arrivee client #3
Date=1.0334555357145838 Arrivee client #4
Date=1.0470496555022897 Arrivee client #5
Date=1.2295069510488126 Arrivee client #6
[...]
Date=999.7903751479797 Depart client #4992 arrive a t=998.0708779291505
Date=999.790414310723 Depart client #4993 arrive a t=998.1454090318407
Date=999.8132179246271 Arrivee client #5002
Date=1000.0011646615608 Depart client #4994 arrive a t=998.1885168192186
Date=1000.1336310931819 Depart client #4995 arrive a t=998.3940000355528
Date=1000.8910479897529 Depart client #4996 arrive a t=998.5175124230215
Date=1001.0462822420868 Depart client #4997 arrive a t=998.8305172502718
Date=1001.1506224887903 Depart client #4998 arrive a t=999.3667698695706
Date=1001.4527386204617 Depart client #4999 arrive a t=999.4639073372005
Date=1001.5500147799029 Depart client #5000 arrive a t=999.5846147300545
Date=1001.5950008671978 Depart client #5001 arrive a t=999.7242365292011
Date=1001.7227333696205 Depart client #5002 arrive a t=999.8132179246271
```

### RESULTATS THEORIQUES

```
-----
lambda<mu : file stable
ro (lambda/mu) = 0.8333333333333334
nombre de clients attendus (lambda x duree) = 5000.0
Prob de service sans attente (1 - ro) = 0.16666666666666663
Prob file occupee (ro) = 0.8333333333333334
Debit (lambda) = 5.0
Esp nb clients (ro/1-ro) = 5.0000000000000002
Temps moyen de sejour (1/mu(1-ro)) = 1.0000000000000002
```

### RESULTATS SIMULATION

```
-----
Nombre total de clients = 5003
Proportion clients sans attente = 0.17489506296222265
Proportion clients avec attente = 0.8251049370377773
Debit = 5.003
Nb moyen de clients dans systeme = 3.98552146993969
Temps moyen de sejour = 0.7966263181970213
```

## Que faut-il rendre ?

Vous déposerez votre travail sur moodle, sous la forme d'une archive contenant :

### ***Code source java***

L'ensemble de vos fichiers java commentés ; inutile de fournir les fichiers compilés .class.

### ***Rapport***

Un rapport au format PDF développant les points suivants :

- la comparaison des résultats obtenus par simulation et les résultats théoriques vus en cours. Vous pouvez fournir des figures commentées montrant l'impact de certains paramètres sur la précision des résultats de simulation obtenus,
- les performances de votre simulateur : expliquez ce qui vous a permis de réduire le temps de simulation (temps de référence sur turing avec  $\lambda=5$ ,  $\mu=6$ ,  $\text{duree}=10\,000\,000$ ,  $\text{debug}=0$  : **22 secondes**).