

Évaluation de performances – TP1: Simulation discrète d'une file M/M/1

Elhadi SERRADJ

M1 SIRIS

Rapport :

Objectifs :

- Étude du comportement d'une file M/M/1 par simulation discrète.
- Programmation en java de la gestion de l'échéancier.

Utilisation :

1. pour compiler les fichier sources, utilisez la commande **make**
2. pour lancer un exemple de simulation en mode (avec/sans) debug :
 - **make run** (debug = 0)
 - **make run_debug** (debug = 1)
3. pour générer les courbes de la simulation :
 - lancer tout d'abord le **script_data.sh** avec **make gen_data**, ce dernier permet de générer les données nécessaires à la création des courbes.
 - et puis lancer le script **script_graph.p** de gnuplot avec **make gen_graph**
 - les courbes sont générées dans un fichier « **graphe_simulation.pdf** »

Introduction:

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 λ) et les temps de service suivent une loi exponentielle de paramètre μ . Les simulations seront effectuées avec $\lambda < \mu$, de manière à obtenir une stabilité en régime permanent.

I. comparaison des résultats obtenus par simulation et les résultats théoriques.

Après avoir simulé plusieurs fois la file MM1, on remarque que le comportement de la file MM1 varie énormément en fonction de paramètre λ et μ . Ces deux

paramètre jouent un rôle très important et affectent le comportement de la simulation.

Dans un premier temps, j'ai étudié le comportement de la simulation en faisant varier les paramètres λ et μ :

- cas 1: $\lambda < \mu$

```
serradj@serradj:~/Bureau/Master_1/s1/eval_perf/Serradj_Elhadi_TP1$ java MM1 5 10 1000 1
##### Traitement en cours... #####
Date : 0.0 Arrivee client #0
Date : 0.06628686369959541 Arrivee client #1
Date : 0.09220310153713403 Depart client #0 Arrive à t= 0.0
Date : 0.2272261694735609 Depart client #1 Arrive à t= 0.06628686369959541
Date : 0.7069803328176778 Arrivee client #2
Date : 0.7774622484282571 Arrivee client #3
Date : 1.0407850442421216 Arrivee client #4
Date : 1.1061224668440097 Arrivee client #5
Date : 1.1509187717666596 Depart client #2 Arrive à t= 0.7069803328176778
Date : 1.210796065442975 Arrivee client #6
Date : 1.2704000891371592 Depart client #3 Arrive à t= 0.7774622484282571
Date : 1.2780337756519324 Depart client #4 Arrive à t= 1.0407850442421216
Date : 1.285935424657185 Depart client #5 Arrive à t= 1.1061224668440097
Date : 1.3749240181839564 Arrivee client #7
Date : 1.4332446089130526 Depart client #6 Arrive à t= 1.210796065442975
Date : 1.5423522539691459 Depart client #7 Arrive à t= 1.3749240181839564
Date : 1.550199264047277 Arrivee client #8
Date : 1.585335004117977 Arrivee client #9
Date : 1.600345274419307 Depart client #8 Arrive à t= 1.550199264047277
Date : 1.6286196231462613 Depart client #9 Arrive à t= 1.585335004117977
Date : 1.692334058901007 Arrivee client #10
Date : 1.734783571601869 Depart client #10 Arrive à t= 1.692334058901007
Date : 2.0334246230120328 Arrivee client #11
Date : 2.047385113746868 Depart client #11 Arrive à t= 2.0334246230120328
Date : 2.1380367847951485 Arrivee client #12
Date : 2.152519527149915 Depart client #12 Arrive à t= 2.1380367847951485
Date : 2.5673181895922106 Arrivee client #13
```

```
Date : 5.107542951120984 Arrivee client #34
Date : 5.24809985441225 Arrivee client #35
Date : 5.4561017102956155 Arrivee client #36
Date : 5.508227551873616 Depart client #33 Arrive à t= 5.066224753718324
Date : 5.537823333131742 Depart client #34 Arrive à t= 5.107542951120984
Date : 5.631676289137065 Arrivee client #37
Date : 5.63953639438064 Depart client #35 Arrive à t= 5.24809985441225
Date : 5.7248372174874 Depart client #36 Arrive à t= 5.4561017102956155
Date : 5.899022112290267 Depart client #37 Arrive à t= 5.631676289137065
Date : 6.034762763053052 Arrivee client #38
Date : 6.074175391143994 Arrivee client #39
Date : 6.174949741557576 Depart client #38 Arrive à t= 6.034762763053052
Date : 6.284393732828026 Depart client #39 Arrive à t= 6.074175391143994
Date : 6.317718835261875 Arrivee client #40
Date : 6.376439555304901 Depart client #40 Arrive à t= 6.317718835261875
Date : 6.774462373010705 Arrivee client #41
Date : 6.803693010390802 Arrivee client #42
Date : 6.981641467891608 Depart client #41 Arrive à t= 6.774462373010705
Date : 7.106781314470573 Depart client #42 Arrive à t= 6.803693010390802
Date : 7.253065005973419 Arrivee client #43
Date : 7.326899700580783 Depart client #43 Arrive à t= 7.253065005973419
Date : 7.350292181455787 Arrivee client #44
Date : 7.394429315494585 Arrivee client #45
Date : 7.431560639701127 Depart client #44 Arrive à t= 7.350292181455787
Date : 7.467586039633885 Depart client #45 Arrive à t= 7.394429315494585
Date : 7.49799811035075 Arrivee client #46
Date : 7.63568154429778 Depart client #46 Arrive à t= 7.49799811035075
Date : 7.723818322714124 Arrivee client #47
Date : 7.857166829828964 Arrivee client #48
Date : 7.986586799678836 Arrivee client #49
```



```

-----
RESULTAT THEORIQUES
-----
lambda < mu : file stable
ro = 0.5
nombre de clients attendus (lambda x duree) = 5000.0
Prob de service sans attente (1 - ro) = 0.5
Prob file occupee (ro) = 0.5
Debit (lambda) = 5.0
Esp nb clients (ro/(1-ro)) = 1.0
Temps moyen de sejour (1/mu(1-ro)) = 0.2

-----
RESULTAT SIMULATION
-----
Nombre total de clients = 4996.0
Proportion clients sans attente = 0.4857886309047238
Proportion clients avec attente = 0.5142113690952762
Debit = 4.99280351165121
Nb moyen de clients dans systeme = 1.000640219135668
Temps moyen de sejour = 0.2132492268042337
##### Fin de la Simulation: Temps d'exécution: [0.456 Sec] #####

```

lorsque $\lambda < \mu$, on observe que le nombre des évènements de départ est aussi élevée que les évènements d'arrivée. Quand un client arrive, la file d'attente est presque à moitié vide donc le client passe directement au service sans attente ou avec une très courte durée d'attente, ainsi nous avons un nombre moyen de clients dans le système très bas et un temps moyen de séjour très court.

- cas 2: $\lambda > \mu$

```

serradj@serradj:~/Bureau/Master_1/s1/eval_perf/Serradj_Elhadi_TP1$ java MM1 10 5 1000 1
##### Traitement en cours... #####
Date : 0.0 Arrivee client #0
Date : 0.17239574951326037 Arrivee client #1
Date : 0.174601010590288 Depart client #0 Arrive à t= 0.0
Date : 0.21645138459274468 Arrivee client #2
Date : 0.23145140796147914 Arrivee client #3
Date : 0.27895702456828225 Depart client #1 Arrive à t= 0.17239574951326037
Date : 0.3433813876270822 Depart client #2 Arrive à t= 0.21645138459274468
Date : 0.3970635997622345 Depart client #3 Arrive à t= 0.23145140796147914
Date : 0.5634370675027297 Arrivee client #4
Date : 0.5663275712779555 Arrivee client #5
Date : 0.7244576356347135 Depart client #4 Arrive à t= 0.5634370675027297
Date : 0.7625527704266195 Arrivee client #6
Date : 0.8718698325979514 Arrivee client #7
Date : 0.9879515096615884 Arrivee client #8
Date : 1.020130155564795 Depart client #5 Arrive à t= 0.5663275712779555
Date : 1.0442409576674088 Depart client #6 Arrive à t= 0.7625527704266195
Date : 1.1744815313115218 Arrivee client #9
Date : 1.1990826005284685 Arrivee client #10
Date : 1.2707102984206509 Arrivee client #11
Date : 1.3537848080918082 Arrivee client #12
Date : 1.4104523839992336 Depart client #7 Arrive à t= 0.8718698325979514
Date : 1.4199448871792266 Arrivee client #13
Date : 1.5129558425334397 Depart client #8 Arrive à t= 0.9879515096615884

```

```

Date : 12.799866682711505 Arrivee client #137
Date : 12.977831154259135 Arrivee client #138
Date : 13.010922034484317 Arrivee client #139
Date : 13.189027847428445 Arrivee client #140
Date : 13.211610094949913 Depart client #59 Arrive à t= 5.517152068884688
Date : 13.21928503921443 Arrivee client #141
Date : 13.248572485164175 Arrivee client #142
Date : 13.304193590311682 Depart client #60 Arrive à t= 5.6147617464128174
Date : 13.605157258990602 Depart client #61 Arrive à t= 5.6611855692319715
Date : 13.648386454905863 Arrivee client #143
Date : 13.722460623431287 Arrivee client #144
Date : 13.89817001695713 Arrivee client #145
Date : 13.946972999561886 Arrivee client #146
Date : 14.017499775090839 Arrivee client #147
Date : 14.254505039953075 Arrivee client #148
Date : 14.339510328758237 Arrivee client #149
Date : 14.366309491179777 Arrivee client #150
Date : 14.404140164801104 Depart client #62 Arrive à t= 5.67987868612088
Date : 14.40434498268869 Arrivee client #151
Date : 14.490483822371766 Arrivee client #152
Date : 14.531991436256467 Arrivee client #153
Date : 14.555471051962638 Depart client #63 Arrive à t= 5.771987299275031
Date : 14.737814679782375 Depart client #64 Arrive à t= 5.7781859637796895
Date : 14.863050958793599 Depart client #65 Arrive à t= 5.88164705539658
Date : 14.896387800660333 Depart client #66 Arrive à t= 5.960491999951435

```

```

-----
RESULTAT THEORIQUES
-----
lambda < mu : file instable
ro = 2.0
nombre de clients attendus (lambda x duree) = 10000.0
Prob de service sans attente (1 - ro) = -1.0
Prob file occupee (ro) = 2.0
Debit (lambda) = 10.0
Esp nb clients (ro/(1-ro)) = -2.0
Temps moyen de sejour (1/mu(1-ro)) = -0.2

-----
RESULTAT SIMULATION
-----
Nombre total de clients = 10006.0
Proportion clients sans attente = 0.002198680791525085
Proportion clients avec attente = 0.9978013192084749
Debit = 4.940526137082198
Nb moyen de clients dans systeme = 2.0252903683471644
Temps moyen de sejour = 516.7319516215492
##### Fin de la Simulation: Temps d'exécution: [150.821 Sec] #####

```

On observe que le nombre des évènements d'arrivée est plus élevée que le départ, ce qui nous donne un nombre moyen de clients dans le système plus élevé que dans le cas précédent d'où un temps de séjour très long.

- cas 3: $\lambda \cong \mu$ (très proche)

```
serradj@serradj:~/Bureau/Master_1/s1/eval_perf/Serradj_Elhadi_TP1$ java MM1 5 6
1000 1
Traitement en cours...
Date : 0.0 Arrivee client #0
Date : 0.14370254142293956 Depart client #0 Arrive à t= 0.0
Date : 0.15811312246464554 Arrivee client #1
Date : 0.16053287446231865 Depart client #1 Arrive à t= 0.15811312246464554
Date : 0.31977555779532485 Arrivee client #2
Date : 0.3453819182497661 Arrivee client #3
Date : 0.39619183347195874 Depart client #2 Arrive à t= 0.31977555779532485
Date : 0.5830646525358002 Arrivee client #4
Date : 0.6257808753705135 Depart client #3 Arrive à t= 0.3453819182497661
Date : 0.6932927123234123 Depart client #4 Arrive à t= 0.5830646525358002
Date : 0.9957085946253612 Arrivee client #5
Date : 1.0637348895801604 Arrivee client #6
Date : 1.106395802106145 Depart client #5 Arrive à t= 0.9957085946253612
Date : 1.1134325115904016 Depart client #6 Arrive à t= 1.0637348895801604
Date : 1.6997968852423173 Arrivee client #7
Date : 1.7236129468614165 Arrivee client #8
Date : 1.7469692702155448 Depart client #7 Arrive à t= 1.6997968852423173
Date : 1.7799364882582178 Depart client #8 Arrive à t= 1.7236129468614165
Date : 1.8254077808109492 Arrivee client #9
Date : 1.8558693096134191 Depart client #9 Arrive à t= 1.8254077808109492
```

```
Date : 972.8289945197462 Depart client #4977 Arrive à t= 972.4937815160589
Date : 973.0378171858381 Arrivee client #4981
Date : 973.1641518028383 Arrivee client #4982
Date : 973.2562085915644 Arrivee client #4983
Date : 973.2579531206597 Arrivee client #4984
Date : 973.3055950880049 Arrivee client #4985
Date : 973.3344487665626 Arrivee client #4986
Date : 973.3686949066926 Arrivee client #4987
Date : 973.5490824433625 Arrivee client #4988
Date : 973.9300423167925 Depart client #4978 Arrive à t= 972.6051022272078
Date : 973.9882651659105 Depart client #4979 Arrive à t= 972.7566400866558
Date : 973.990615446623 Depart client #4980 Arrive à t= 972.8282149195212
Date : 974.0744881047694 Depart client #4981 Arrive à t= 973.0378171858381
Date : 974.1169437587769 Depart client #4982 Arrive à t= 973.1641518028383
Date : 974.1453287906888 Depart client #4983 Arrive à t= 973.2562085915644
Date : 974.2549119951989 Arrivee client #4989
Date : 974.3417678945814 Depart client #4984 Arrive à t= 973.2579531206597
Date : 974.5162759833115 Depart client #4985 Arrive à t= 973.3055950880049
Date : 974.8106692826472 Depart client #4986 Arrive à t= 973.3344487665626
Date : 974.880478155146 Depart client #4987 Arrive à t= 973.3686949066926
Date : 974.9072622388056 Arrivee client #4990
Date : 975.0703631364768 Depart client #4988 Arrive à t= 973.5490824433625
Date : 975.1023305818528 Arrivee client #4991
```

```

.....
RESULTAT THEORIQUES
.....
lambda < mu : file stable
ro = 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
.....
RESULTAT SIMULATION
.....
Nombre total de clients = 5115.0
Proportion clients sans attente = 0.1536656891495601
Proportion clients avec attente = 0.8463343108504399
Debit = 5.112430759022193
Nb moyen de clients dans systeme = 4.114497452171205
Temps moyen de sejour = 1.006511685432507
Temps d'execution: 0.333s

```

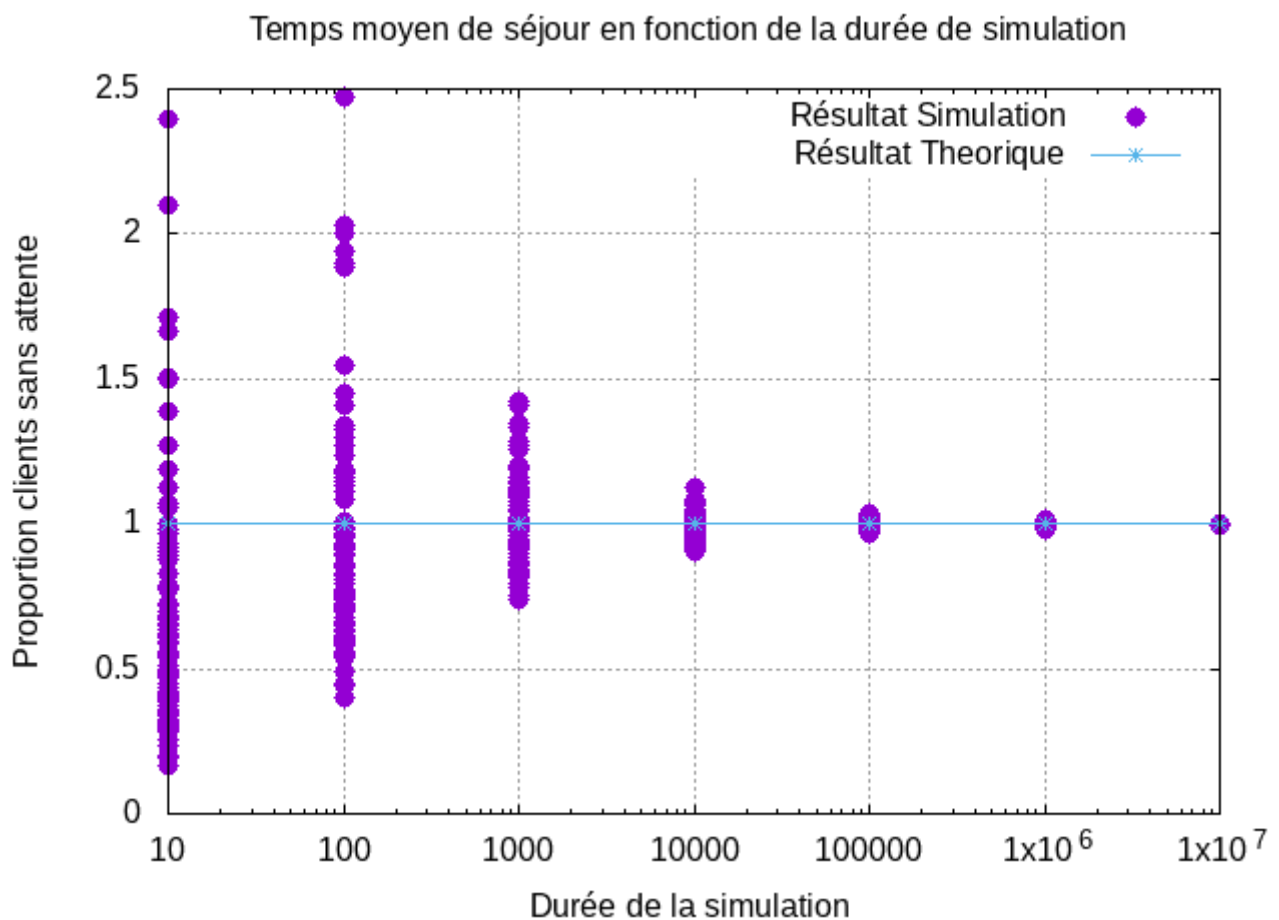
Dans ce dernier cas d'étude, on observe que le nombre des évènements d'arrivée est presque identique aux évènements de départ. Malgré que le nombre moyen de clients dans le système augmente, le temps moyen de séjour reste moyen.

II. Résultat Pratique vs Théorique :

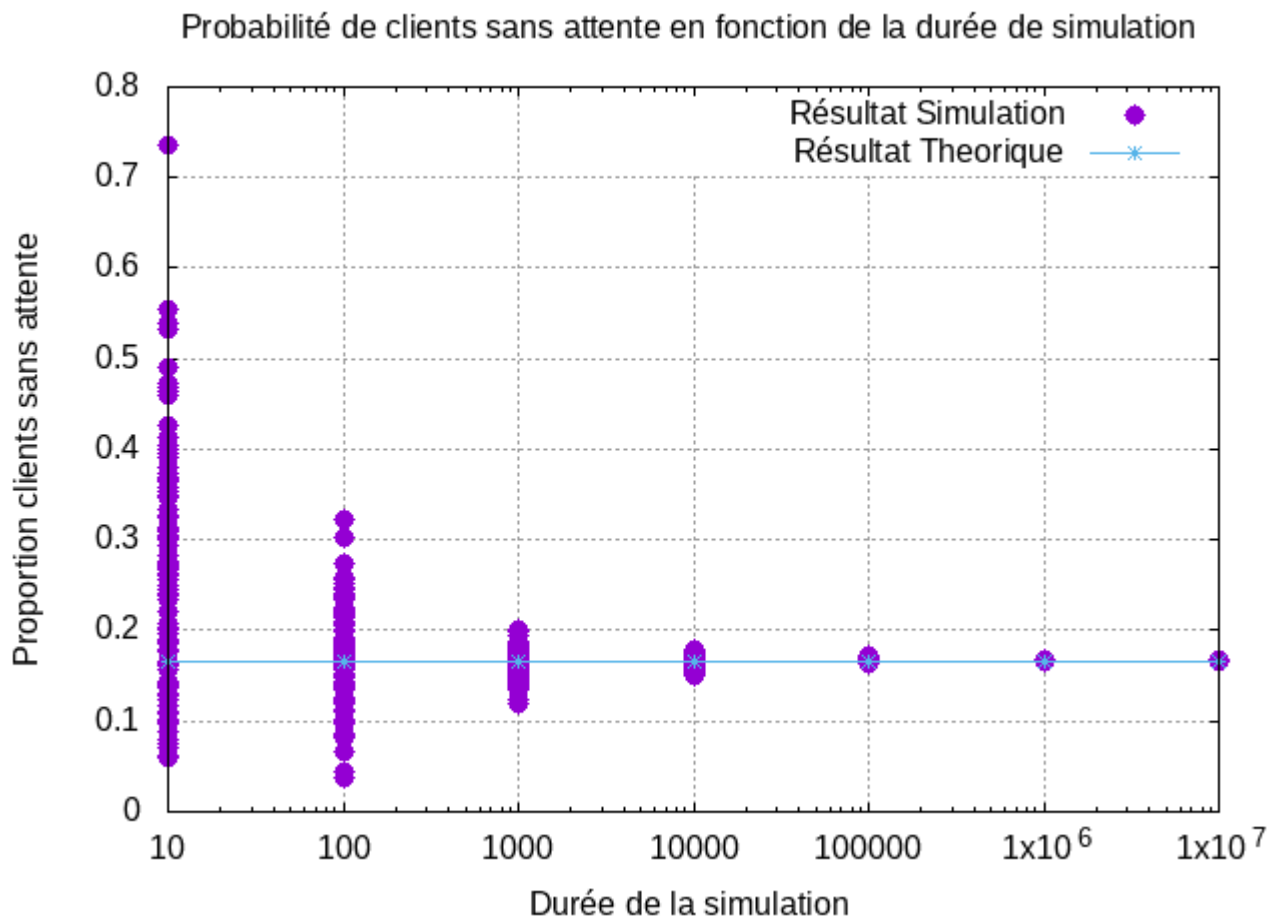
dans cette partie, j'ai lancé la simulation 100 fois pour chaque durée ($t=10$, $t=100$, $t=1000$, $t=10000$, $t=100000$, $t=1000000$, $t=10000000$) avec $\lambda=5$ et $\mu=6$ car quand on réalise une expérience aléatoire, on remarque que les résultats obtenus sont différents (pas toujours les mêmes), c'est la fluctuation d'échantillonnage. Donc on constate que, plus on répète une expérience un plus grand nombre de fois, plus la régularité des résultats est grande. Donc t doit être suffisante pour laisser se dégager des tendances.

- Temps moyen de séjour dans le système

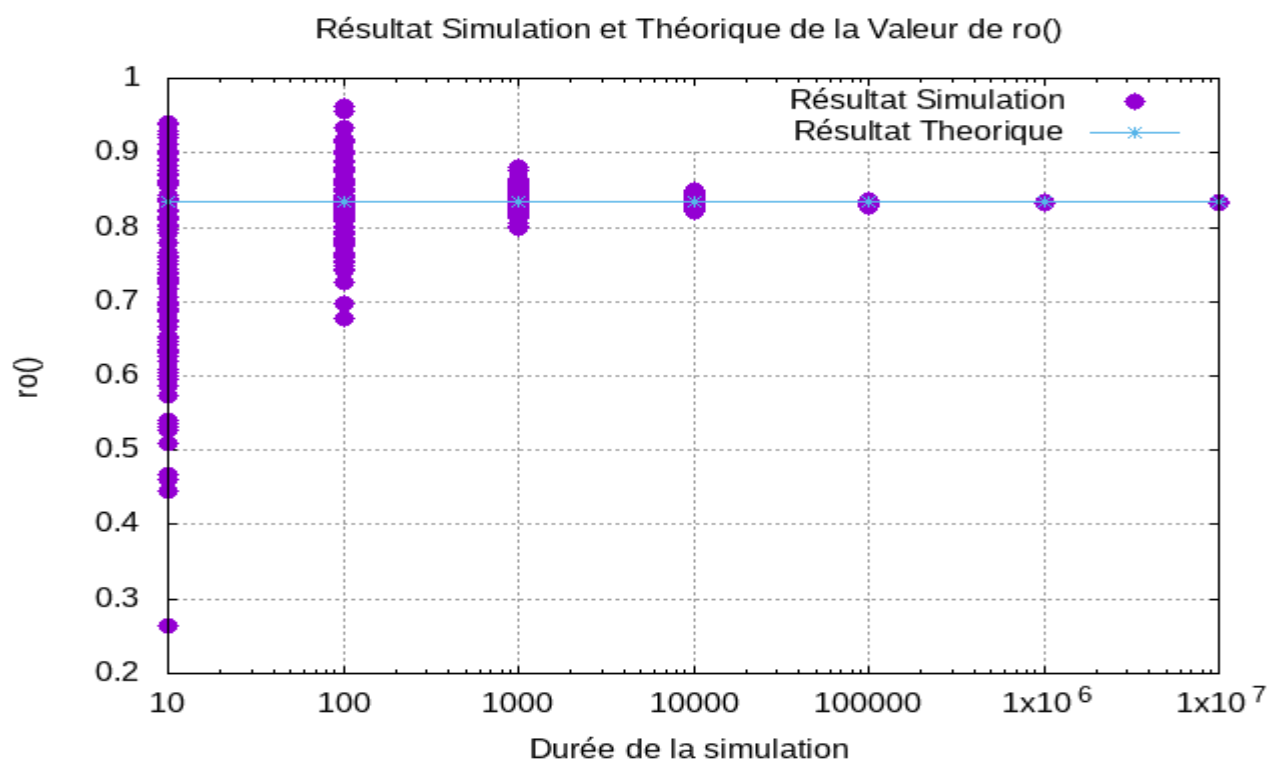
la figure ci-dessous nous présente les courbes correspondantes au temps moyen de séjour théorique et simulation. On observe que plus la durée de simulation est grande, plus on s'approche de la valeur théorique, donc on peut dire que le temps de séjour suit un régime stationnaire au fil de temps, Ceci est donc attendu puisque la relation entre le temps de séjour moyen et le nombre de clients dans le système est donné par la formule de Little: donc on retrouve bien la formule de little $L = R \cdot \Delta \Rightarrow (1/\mu) + (\rho/\mu(1-\rho))$ en régime permanent quand $(t \rightarrow \infty)$.



- Probabilité de clients sans attente :



- Probabilité d'une file occupée ou $Ro()$:



d'après les trois graphes ci-dessus. On en conclut que la simulation est très proche de la réalité quand la durée de simulation est assez grande car les résultats de simulation suivent bien l'allure de résultats théoriques. On peut donc dire que pour $t = 100000 \rightarrow 10\text{Millions}$, on a une meilleure répétabilité. Alors on peut déduire que la simulation tend vers un état stationnaire.

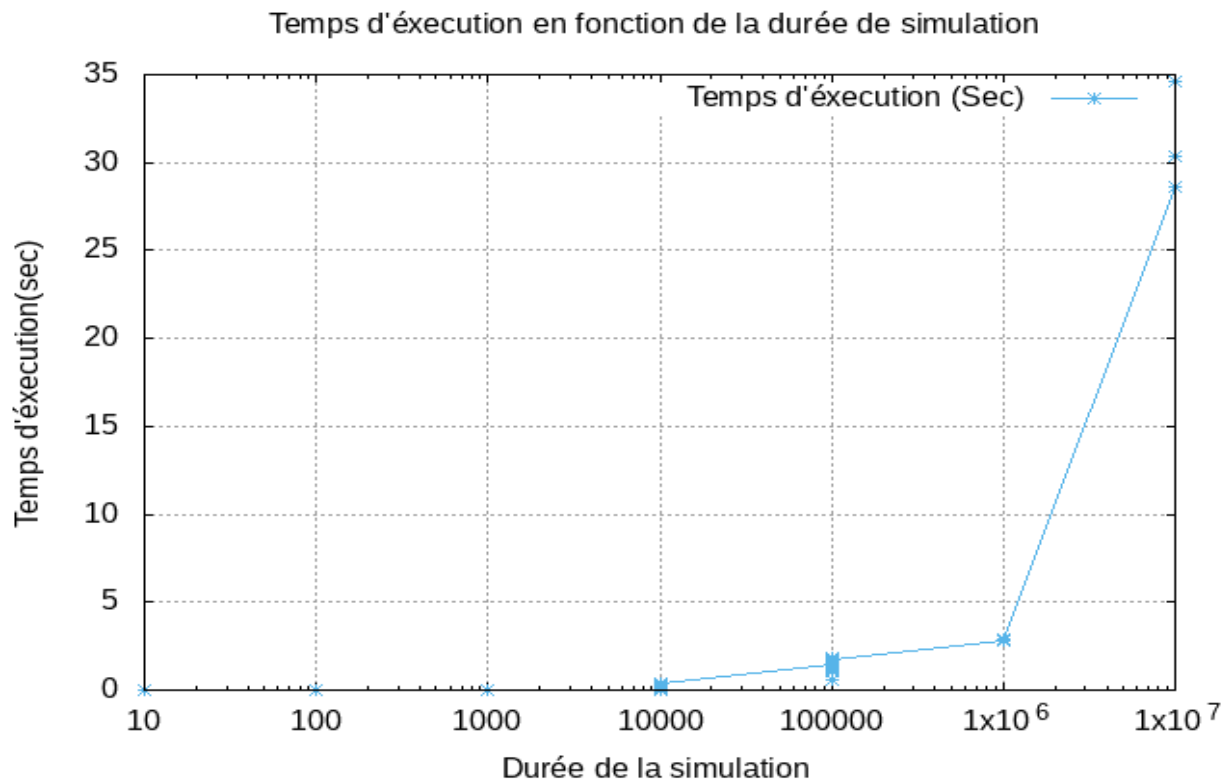
Lorsque la simulation est stable et que λ et μ sont très proches sur une longue durée, on remarque que la plupart des temps de service sont très courts. C'est la caractéristique typique de la distribution exponentielle.

Si on a une simulation instable ($\lambda > \mu$), le nombre de clients explose et la simulation dure plus longtemps qu'une file stable.

III. Performances :

- **Temps d'exécution :**

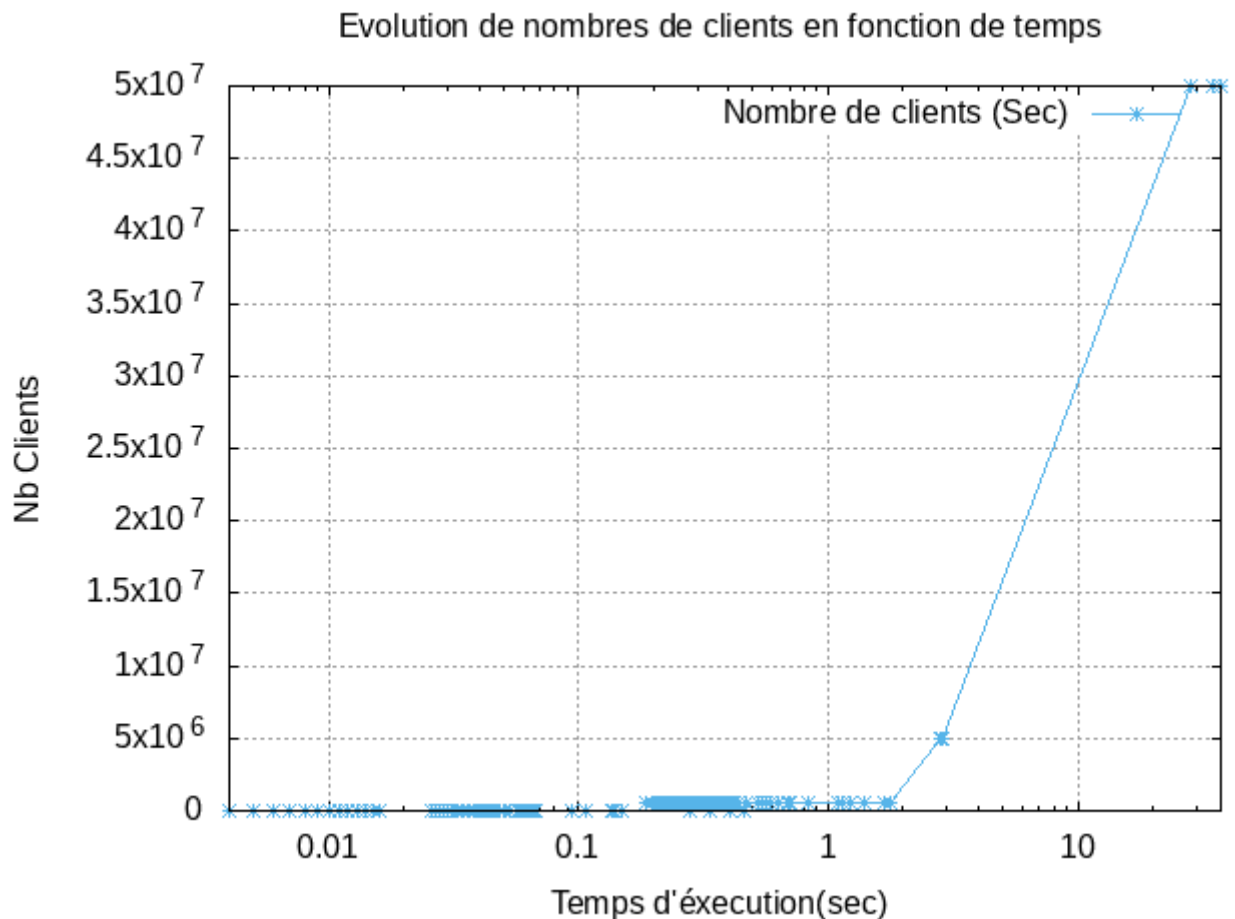
On peut voir sur la figure ci-dessous l'évolution de temps d'exécution en fonction de la durée de simulation, on observe que plus la durée de simulation est grande plus le temps d'exécution accroît. Sur cet exemple, on a une durée pour $t = 10\text{Millions}$ comprise entre $[28s, 34s]$ 'voir graphe ci-dessous' qui est raisonnable par rapport au temps mentionné dans le sujet de TP ($= 20s$). La courbe semble avoir l'allure d'une courbe exponentielle, cela confirme la linéarité du temps d'exécution en fonction de la durée de simulation.



Cette durée est calculé sur mon PC. Ma première version de programme calcule l'événement avec la date la plus courte en parcourant toute la liste mais cela prend très longtemps. Pour réduire ce temps d'exécution, j'ai donc procédé différemment pour l'événement à traité: si on a en premier une événement de type arrivé, on la supprime de l'échéancier car c'est cette événement qui a la date la plus courte et si on a un évènement de type départ, on regarde l'évènement suivant dans la liste si elle est une arrivée et si elle est ultérieur, on la supprime sinon on supprime l'évènement de départ.

- **Evolution de nombres de client au file de temps :**

En analysant le graphe ci-dessous, on voit qu'on a un ordre de 5×10^7 clients, donc pour un échantillon de cette taille, le temps de calcul est très intéressant.



IV. Conclusion :

L'analyse des files d'attente peut être un aspect important de la conception des systèmes. Les arrivées aléatoires des clients combinées à la variabilité des temps de service créent temporairement des congestions dans le système, d'où la création de files d'attente. Pour analyser des files d'attente, il est important de définir si la population de clients potentiels est infinie ou bien si elle se limite à un nombre fini de clients. En général, l'hypothèse émise dans le cadre de ce modèle est que le taux d'arrivée des clients sont distribués selon une loi de Poisson, alors que les temps de service suivent une loi exponentielle.