analyse de performance et simulation - M1 II et ISIFAR

Examen du 19 mai 2009 – corrigé

Université Paris Diderot

Tous les documents sont autorisés
Durée : 2h30

Exercice 1 – *Génération aléatoire*

On dispose d'une fonction `double u()` qui renvoie un réel de double précision aléatoire de loi uniforme sur(0 ;1). Programmez les fonctions qui génèrent les variables aléatoires suivantes :

1. X une pièce courbée, qui tombe sur pile (0) avec probabilité 0,36 ; sur face (1) avec probabilité 0,6 ; et sur le bord (2) avec probabilité 0,04

   Correction.
   ```
   double piece()
           unif=u()
           if unif<0.36
             return 0
           else if unif <0.96
             return 1
           else
             return 2
   ```

2. Y le temps d' attente d'une météorite - - en années (loi exponentielle avec $\lambda = 0.04$)

   Correction. La fonction de distribution est :

   $$F(t) = 1 - e^{-\lambda t}, \quad (t > 0)$$

   On applique la méthode de l'inversion de la fonction de distribution : $Y = F^{-1}(U)$, ce qui équivaut à $F(Y) = U$. On résout cette équation :

   $$F(Y) = U \Leftrightarrow 1 - e^{-\lambda Y} = U \Leftrightarrow e^{-\lambda Y} = 1 - U \Leftrightarrow -\lambda Y = \ln(1 - U) \Leftrightarrow Y = -\ln(1 - U)/\lambda.$$

   On programme :
   ```
   double meteorite()
           lambda=0.04
           unif=u()
           Y=-(log(1-unif))/lambda
           return Y
   ```

3. Z de loi de Cauchy, avec la fonction de distribution $F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(x)$

   Correction. On applique la méthode de l'inversion de la fonction de distribution : $Z = F^{-1}(U)$, ce qui équivaut à $F(Z) = U$. On résout cette équation :

   $$F(Z) = U \Leftrightarrow U = \frac{1}{2} + \frac{1}{\pi}\arctan(Z) \Leftrightarrow \arctan(Z) = \pi(U - \frac{1}{2}) \Leftrightarrow Z = \tan\left(\pi(U - \frac{1}{2})\right)$$

   On programme :
   ```
   double cauchy()
           unif=u()
           Z=tan(pi*(unif-0.5))
           return Z
   ```

Exercice 2 – *File d'attente*

On considère une file d'attente M/M/3.

1. Expliquez brièvement le sens de cette notation.

   Correction. Selon les notations de Kendall, c' est une file d' attente avec
   –le processus d'arrivées poissonnien (les intervalles entre arrivées sont indépendantes d'une loi exponentielle)
   –les temps de service indépendants d'une loi exponentielle
   –3 serveurs identiques
   –on ne rejette jamais les requêtes (buffer non-borné)

2. Le temps de service moyen par un serveur est $\bar{S} = 6s$, le nombre de requêtes moyen est λ par seconde. Trouvez quelles valeurs de λ la file peut supporter en restant stable. Quel théorème utilisez-vous ?

   Correction. Par théorème de Loynes la condition de stabilité est $\rho < 1$ avec l' utilisation $\rho = \lambda\bar{S}/s$ avec s le nombre de serveurs. Ca donne :

   $$\lambda \cdot 6/3 < 1 \Leftrightarrow \lambda < 1/2$$

3. Formulez la loi de Little pour cette file, en expliquant tous les paramètres.

   Correction. En régime stationnaire (sous l' hypothèse de stabilité $\lambda < 1/2$) on a

   $$\bar{N} = \lambda\bar{R},$$

   où $\bar{N}$ est le nombre moyen de clients dans le système (en attente et en service) ; $\bar{R}$ est le temps de réponse moyen (entre l'entrée dans la file et la fin de service).

Exercice 3 – *Analyse de bottleneck*

On considère un réseau de files d'attentes qui représente un hôpital. On suppose qu'il y a toujours n patients dans l'hôpital. Chaque malade est d' abord reçu par l' accueil avec le temps de réflexion Z=10 minutes, ensuite il visite les cabinets A (en moyenne $V_A = 0.5$ fois), B (en moyenne $V_B = 0.25$ fois), C (en moyenne $V_C = 3$ fois). Les temps moyens de service dans chaque cabinet sont 10, 15 et 2 minutes respectivement.

1. Estimez (en utilisant les lois opérationnelles) comment le nombre de patients λ traités par minute dépend de n et dessinez un graphe.

   Correction. On a deux contraintes sur λ (facile à obtenir, voir Le Boudec, ch. 10) :

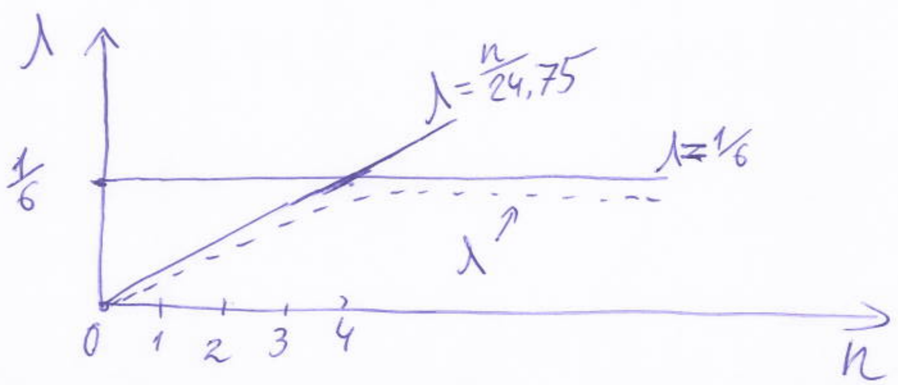
   $$\begin{aligned} \lambda &< \frac{1}{\max V_i \bar{S_i}} \\ \lambda &< \frac{n}{Z + \sum V_i \bar{S_i}} \end{aligned}$$

   Avec le paramètres du problème ça donne

   $$\begin{aligned} \lambda &< \frac{1}{\max\{0.5 \cdot 10, 0.25 \cdot 15, 3 \cdot 2\}} \\ \lambda &< \frac{n}{10 + 0.5 \cdot 10 + 0.25 \cdot 15 + 3 \cdot 2} \end{aligned}$$

   c-à-d

   $$\begin{aligned} \lambda &< \frac{1}{6} \\ \lambda &< \frac{n}{24.75} \end{aligned}$$



2. Mêmes questions pour le temps moyen de réponse.

Correction. On a deux contraintes sur $\bar{R}$ (facile à obtenir, voir Le Boudec, ch. 10) :

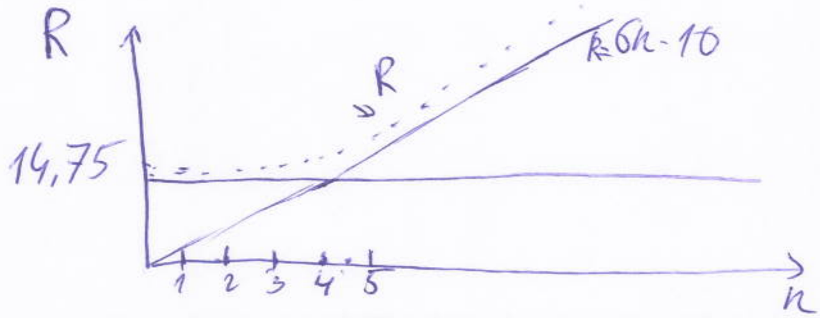
$$\begin{aligned} \bar{R} &\geq \sum V_i \bar{S_i} \\ \bar{R} &\geq n \max V_i \bar{S_i} - Z \end{aligned}$$

Avec le paramètres du problème ça donne

$$\begin{aligned} \bar{R} &\geq 0.5 \cdot 10 + 0.25 \cdot 15 + 3 \cdot 2 \\ \bar{R} &\geq n \max\{0.5 \cdot 10, 0.25 \cdot 15, 3 \cdot 2\} - 10 \end{aligned}$$

c-à-d

$$\begin{aligned} \bar{R} &\geq 14.75 \\ \bar{R} &\geq 6n - 10 \end{aligned}$$



3. Quel cabinet est le maillon faible (bottleneck) qui limite la performance de l'hôpital ?

Correction. Celui qui maximise $V_i \bar{S_i}$, à savoir C.

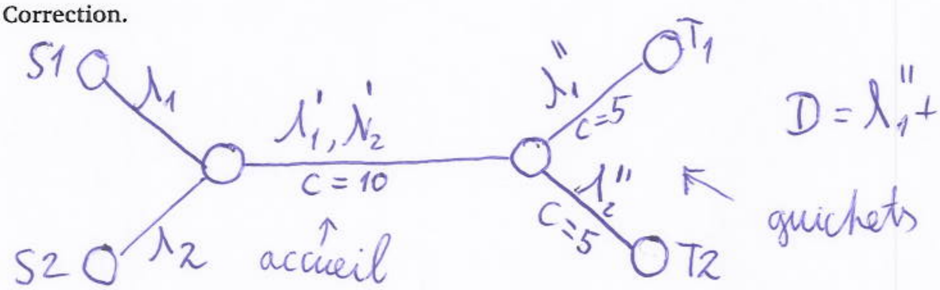
Exercice 4 – *Collapse*

Les clients de type 1 et 2 arrivent à l'accueil de la banque (respectivement $\lambda_1 = 5$ et λ_2 clients par minute). L'accueil traite $c_a = 10$ clients par minute et les renvoie aux guichets G1 (pour le type 1) et G2 (pour le type 2). La capacité de chaque guichet est $c_{G1} = c_{G1} = 5$ clients par minute.

1. En utilisant la modélisation fluide représentez le système comme un réseau de flux. Faites un dessin et indiquez clairement les sources, les destinations et les capacités des "tuyaux".

Correction.

2 flux: S1→T1; S2→T2



2. Etudiez le fonctionnement du système pour $\lambda_2 = 1$, pour $\lambda_2 = 5$, pour $\lambda_2 = 10$, $\lambda_2 = 100$ et trouvez le débit D (nombre de clients traités par minute) .

Correction.

$\lambda_2 = 1$. Tout le trafic passe sans problème : on a $\lambda_1' = 5; \lambda_2' = 1$; ensuite $\lambda_1'' = 5; \lambda_2'' = 1$; et finalement $D = \lambda_1'' + \lambda_2'' = 6$.

$\lambda_2 = 5$. Tout le trafic passe sans problème : on a $\lambda_1' = 5; \lambda_2' = 5$; ensuite $\lambda_1'' = 5; \lambda_2'' = 5$; et finalement $D = \lambda_1'' + \lambda_2'' = 10$.

$\lambda_2 = 10$. L'accueil est saturé, avec la règle proportionnelle on a : $\lambda_1' = \frac{5 \cdot 10}{5+10} = 3\frac{1}{3}$ et $\lambda_2' = \frac{10 \cdot 10}{5+10} = 6\frac{2}{3}$. Ensuite les clients de type 1 passe sans problème au G1, tandis que G2 est saturé : $\lambda_1'' = 3\frac{1}{3}; \lambda_2'' = 5$; et finalement $D = \lambda_1'' + \lambda_2'' = 8\frac{1}{3}$.

$\lambda_2 = 100$. L'accueil est saturé, avec la règle proportionnelle on a : $\lambda_1' = \frac{5 \cdot 10}{5+100} = \frac{10}{21}$ et $\lambda_2' = \frac{100 \cdot 10}{5+100} = 9\frac{11}{21}$. Ensuite les clients de type 1 passe sans problème au G1, tandis que G2 est saturé : $\lambda_1'' = \frac{10}{21}$; $\lambda_2'' = 5$; et finalement $D = \lambda_1'' + \lambda_2'' = 5\frac{10}{21}$.

3. Essayez de trouver une formule pour D en fonction de λ_2.

Correction. Il y a 2 cas possibles :

$\lambda_2 =\leq 5$. Tout le trafic passe sans problème : on a $\lambda_1' = 5; \lambda_2' = \lambda_2$; ensuite $\lambda_1'' = 5; \lambda_2'' = \lambda_2$; et finalement $D = 5 + \lambda_2$.

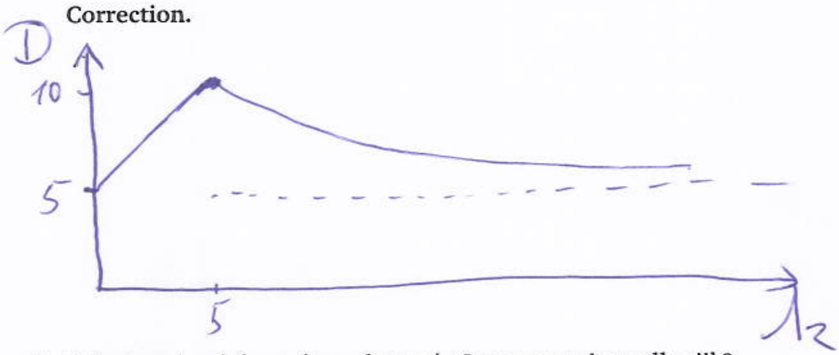
$\lambda_2 > 5$. L'accueil est saturé, avec la règle proportionnelle on a : $\lambda_1' = \frac{5 \cdot 10}{5+\lambda_2} < 5$ et $\lambda_2' = \frac{\lambda_2 \cdot 10}{5+\lambda_2} > 5$. Ensuite les clients de type 1 passe sans problème au G1, tandis que G2 est saturé : $\lambda_1'' = \frac{50}{5+\lambda_2}; \lambda_2'' = 5$; et finalement $D = 5 + \frac{50}{5+\lambda_2}$.

On obtient le résultat

$$D = \begin{cases} 5 + \lambda_2 & \text{si} \quad \lambda_2 \leq 5 \\ 5 + \frac{50}{5+\lambda_2} & \text{si} \quad \lambda_2 > 5 \end{cases}$$

4. Dessinez le graphe de dépendance de D en fonction de λ_2.

Correction.



5. Décrivez le phénomène observé. Comment s'appelle t'il ?

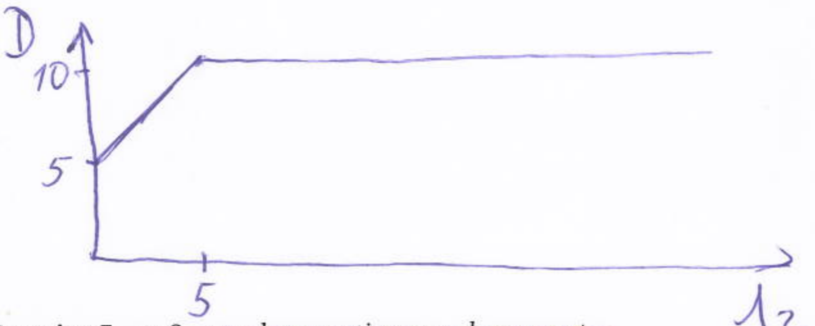
Correction. Lorsque λ_2 augmente de 0 à 5, le débit D augmente aussi, jusqu'à 10. Pour $\lambda_2 > 5$ le système est saturé, et sa performance baisse avec l'augmentation de λ_2 (pour $\lambda_2 \to \infty$ on a $D \to 5$). Ce phénomène désagréable est le collapse de congestion ;

6. Comment peut-on éviter ce phénomène grâce à un accueil intelligent ? Proposez les règles optimales pour l'accueil et dessinez le graphe de débit résultant.

Correction. Il suffit que l'accueil n'accepte pas plus que 5 clients de type 2 par minute. On aura

$$D = \begin{cases} 5 + \lambda_2 & \text{si} \quad \lambda_2 \leq 5 \\ 10 & \text{si} \quad \lambda_2 > 5 \end{cases}$$

(bottleneck, mais pas de collapse)



Exercice 5 – *ns2* - ~~pas de correction pour le moment.~~

Lisez le programme ns2 suivant. Sur le recto de la page du programme commentez le code. Sur le verso dessinez le réseau simulé par ce code. Insérez le programme dans votre copie.

(AVEC LES COMMENTAIRES)

```
set ns [new Simulator] ;# Création d'un objet Simulator

# Définition des couleurs pour la simulation
$ns color 1 Blue
$ns color 2 Red

# Association de la tr\^ace nam de la simulation à un fichier .nam
set nf [open out.nam w]
$ns namtrace-all $nf

# Procédure de terminaison
proc finish {} {
        global ns nf ;# Inclusion des variables globales utilisées

# Écriture de la trace nam du programme (stockée dans la variable $nf) sur le fichier associé
        $ns flush-trace

        close $nf ;# Fermeture de la variable associée à la trace
exec nam out.nam & ;# Exécution de l'outil nam pour la simulation
        exit 0 ;# Sortie
}

# Création des noeuds
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# Création des liens physiques
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms SFQ

# Orientation des liens dans la représentation graphique de nam
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

# Monitorage de la file d'attente associée aux liens entre les noeuds 2 et 3
$ns duplex-link-op $n2 $n3 queuePos 0.5

# Définition d'un agent UDP émetteur et son association au noeud $n0
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0

# Définition d'un générateur de trafic constant et son association à l'agent $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

# Définition d'un agent UDP émetteur et son association au noeud $n1
set udp1 [new Agent/UDP]
$udp1 set class_ 2
$ns attach-agent $n1 $udp1

# Définition d'un générateur de trafic constant et son association à l'agent $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1

# Définition d'un agent Null et son association au noeud $n3
```

```
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0

# Établissement des connexions entre les agents émetteurs et l'agent $null0
$ns connect $udp0 $null0
$ns connect $udp1 $null0

# Programmation des évènements de la simulation
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"

# Lancement de la simulation
$ns run
```