Systèmes embarqués temps réel (GIF-3004)
Département de génie électrique et de génie informatique
Hiver 2019

UNIVERSITÉ LAVAL

EXAMEN FINAL

Instructions : – Une feuille aide-mémoire recto verso manuscrite est permise ;
– Durée de l'examen : 1 h 50.

Pondération : Cet examen compte pour 25% de la note finale.

Aide-mémoire

Combinatoire

- Factorielle de n : $n! = 1 \times 2 \times 3 \times \cdots \times n \qquad (0! = 1)$
- Choisir k parmi n, selon un certain ordre (arrangement) : $A_n^k = \frac{n!}{(n-k)!}$
- Choisir k parmi n, sans ordre (combinaison) : $\binom{n}{k} = \frac{A_n^k}{k!} = \frac{n!}{k!(n-k)!}$

Loi exponentielle

- Densité de probabilité : $p(t) = \lambda \exp(-\lambda t)$, pour $t \geq 0$
- Loi de répartition : $P(X \leq t) = 1 - \exp(-\lambda t)$, pour $t \geq 0$
- Espérance : $\mathbb{E}(t) = \frac{1}{\lambda}$
- Variance : $\text{Var}(t) = \frac{1}{\lambda^2}$

Théorie des files d'attente

Modèle	$M/M/1$	$M/M/m$	$M/M/\infty$	$M/M/1/K$
Utilisation (ρ)	$\frac{\lambda}{\mu}$	$\frac{\lambda}{m\mu}$	$\frac{\lambda}{\mu}$	$\frac{\lambda}{\mu}$
Probabilité d'aucune tâche (π_0)	$1 - \frac{\lambda}{\mu} = 1 - \rho$	$\left[1 + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho}\right]^{-1}$	$\exp(-\rho)$	$\left[1 + \sum_{n=1}^{K} (\lambda/\mu)^n\right]^{-1} = \frac{1-\rho}{1-\rho^{K+1}}$
Probabilité de n tâches (π_n)	$(1-\rho)\rho^n$	$\pi_0 \frac{(m\rho)^n}{n!}$ pour $1 \leq n \leq m$ $\pi_0 \frac{m^m}{m!}\rho^n$ pour $n > m$	$\exp(-\rho)\frac{\rho^n}{n!}$	$\frac{1-\rho}{1-\rho^{K+1}}\rho^n$ pour $0 \leq n \leq K$ 0 pour $n > K$
Taille moyenne de la file ($\mathbb{E}(X)$)	$\frac{\rho}{1-\rho}$	$\sum_{n=0}^{\infty} n\pi_n = m\rho + \frac{(m\rho)^m}{m!}\frac{\rho}{(1-\rho)^2}\pi_0$	ρ	$\frac{\rho}{1-\rho^{K+1}}\left[\frac{1-\rho^K}{1-\rho} - K\rho^K\right]$
Temps système moyen ($\mathbb{E}(S)$)	$\frac{1/\mu}{1-\rho}$	$\frac{1}{\mu} + \frac{1}{\mu}\frac{(m\rho)^m}{m!}\frac{\pi_0}{m(1-\rho)^2}$	$\frac{1}{\mu}$	$\frac{\mathbb{E}(X)}{\lambda(1-\pi_K)}$

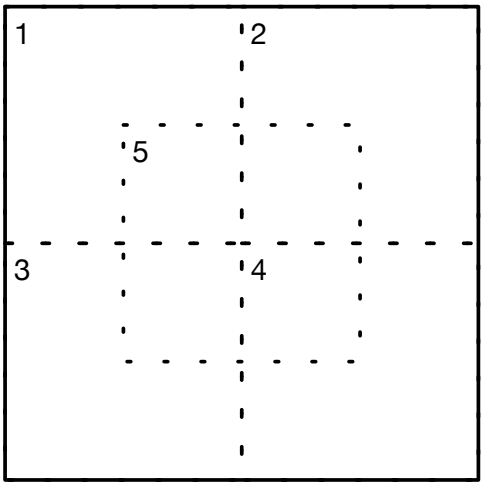
Fiabilité de systèmes complexes

- **Exactement** k composants fonctionnels sur n : $R_s = \binom{n}{k}[R]^k[1-R]^{n-k}$
- **Au moins** k composants fonctionnels sur n : $R_s = \sum_{i=k}^{n}\binom{n}{i}[R]^i[1-R]^{n-i}$

Question 1 (16 points sur 100)

Projetez-vous dans quelques années d'ici, où vous avez la tâche de concevoir un système embarqué effectuant du traitement d'image à l'aide de réseaux de neurones artificiels. Le système que vous concevez comporte un processeur standard, mais également un coprocesseur neuronal qui est utilisé pour faire du traitement d'image (un réseau de neurones utilisé en inférence seulement).

Supposons que l'on veut paralléliser le traitement en divisant chaque image en K régions. Cependant, pour que le traitement corresponde à ce qu'on veut faire avec l'image complète, il faut utiliser M sous-images, avec certaines superposées à une région et d'autres comprenant des recoupements entre régions ($M \geq K$). Par exemple, si on veut séparer une image en $K = 4$ régions, il faut utiliser $M = 5$ sous-images, une pour chaque région et une sous-image supplémentaire centrée sur l'image, comme illustré ci-bas.



Supposez que le coprocesseur neuronal peut traiter jusqu'à quatre sous-images en parallèle. De plus, faites l'hypothèse que le temps de traitement est directement et uniquement proportionnel au nombre de pixels des sous-images. Supposez également que les régions et les sous-images sont toutes de même taille, soit la taille de l'image d'origine divisée par le nombre de régions. Par exemple, lorsqu'une image comprend N pixels, chaque sous-images comportera N/K pixels.

(10) (a) Donnez une équation permettant de déterminer l'accélération (*speed-up*) parallèle possible selon cette modélisation pour le cas général.

(6) (b) Calculez également l'accélération avec les configurations suivantes et désignez la configuration ayant le temps de traitement le plus bas :

(i) $K = 2$ régions avec $M = 3$ sous-images ;

(ii) $K = 4$ régions avec $M = 5$ sous-images ;

(iii) $K = 6$ régions avec $M = 8$ sous-images ;

(iv) $K = 9$ régions avec $M = 13$ sous-images.

Question 2 (24 points sur 100)

Toujours avec le système embarqué avec coprocesseur neuronal présenté à la question précédente. Supposons maintenant que le système reçoit en entrée des séquences vidéos avec un taux d'image fixe (*framerate*), mais qu'un prétraitement est effectué sur le processeur pour déterminer les images de la séquence nécessitant de plus amples analyses sur le coprocesseur neuronal.

Supposons qu'une modélisation de type théorie des files d'attente est effectuée en établissant que la distribution d'arrivée des images à traiter et la distribution de service (traitement) par le coprocesseur neuronal suivent chacune un processus de Poisson. De plus, le coprocesseur neuronal comporte également quatre cœurs permettant un traitement parallèle, chaque coprocesseur pouvant traiter une image distincte à la fois, et qu'il y a suffisamment d'espace mémoire pour stocker toutes les images en attente de traitement.

(6) (a) Supposons un taux d'arrivée de 5 images à traiter par seconde par le coprocesseur neuronal, calculez le temps de traitement (service) moyen par image pour maintenir le taux d'utilisation du coprocesseur à moins de 50 % (plus de la moitié du temps sans traiter d'image).

(6) (b) Supposons maintenant que le taux d'arrivée est de 2 images par seconde et que le temps de traitement (service) moyen est de 400 ms par image, calculez la probabilité instantanée qu'il y ait une image ou plus en attente de traitement dans le système.

(6) (c) Toujours avec un modèle avec un taux d'arrivée de 2 images à traiter par seconde et un temps de traitement moyen de 400 ms par image, calculez le temps système moyen nécessaire pour traiter chaque image.

(6) (d) Supposons maintenant que le temps de traitement (service) est déterministe, où le traitement d'une image par un cœur du coprocesseur prend exactement 200 ms, calculez le taux d'arrivée maximal que le système peut supporter.

Question 3 (20 points sur 100)

Supposons maintenant que l'on utilise une version particulière du coprocesseur neuronal fonctionnant à faible voltage pour des économies d'énergie considérables, mais avec des erreurs intermittentes dans les calculs survenant à des moments aléatoires.

(8) (a) Supposons que l'on soit capable de détecter si une erreur est survenue lors d'une opération et que le taux d'erreur de l'opération est de 10^{-4}, soit une erreur pour 10 000 exécutions de l'opération. Sachant que le temps pour effectuer une opération est de 200 ms et que le temps pour refaire une nouvelle exécution d'une opération fautive est de 500 ms. Calculez le temps d'exécution moyen résultant, incluant les reprises de faute, qui ont le même taux de d'erreur. Vous pouvez supposer que les cas avec plus de deux fautes pour une opération sont très rares, donc négligeables pour nos calculs. Indiquez également si ce temps d'exécution moyen reflète bien l'impact que peuvent avoir les erreurs intermittentes sur le fonctionnement d'un système embarqué temps réel.

(12) (b) Supposons maintenant que les erreurs ne sont pas directement détectables, qu'il faille effectuer la même opération plusieurs fois pour détecter et corriger les fautes, en utilisant la réponse majoritaire comme résultat. Supposons pour l'opération qui nous intéresse, le taux d'erreur est toujours de 10^{-4} par exécution d'opération. Estimez le nombre d'exécutions d'une même opération nécessaire pour assurer une exécution fiable du système, qui correspond dans notre cas à un taux d'erreur global inférieur à 10^{-9}.

Question 4 (40 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

(4) (a) Expliquez de quelle façon dans Unix sont conçus les programmes selon la règle de la compositionnalité, en insistant en particulier sur la façon dont ceux-ci communiquent entre eux.

(4) (b) *Premature optimization is the root of all evil* (en français : l'optimisation prématurée est la racine de tous les maux) est un dicton célébre en informatique, attribué à Donald Knuth. Expliquez ce qu'on entend généralement par celui-ci.

(4) (c) Pour la gestion des interruptions dans le noyau Linux, expliquez comment les opérations doivent se partager entre la moitié supérieure (*top-half*) et la moitié inférieure (*bottom-half*).

(4) (d) Expliquez pourquoi la librairie C standard n'est pas disponible lorsqu'on développe du code s'exécutant dans le noyau.

(4) (e) Pour votre laboratoire 4, portant sur l'implémentation d'une pilote de périphérique, indiquez pourquoi le pilote est implémenté comme un dispositif de caractères, plutôt qu'un dispositif de blocs.

(4) (f) Expliquez pourquoi les réseaux routés, comme le réseau sans-fil de l'université, ne sont pas adaptés à la communication entre dispositifs d'un système temps réels.

(4) (g) IEEE 1394 (FireWire) et CANbus sont deux standards de bus de communication temps réels. Expliquez les principaux éléments qui les distinguent.

(4) (h) Pour votre laboratoire 5 portant sur la transmission audio sans fil entre deux unités embarquées, indiquez si une synchronisation des horloges entre les unités est nécessaire.

(4) (i) Expliquez pourquoi le choix du niveau de granularité des tâches correspondant à une décomposition parallèle d'une application se décline généralement comme un compromis, une granularité très élevée ou très faible n'étant généralement pas souhaitable.

(4) (j) La courbe en forme de baignoire permet de décrire l'évolution du taux de panne d'une grande variété de composants. Expliquez pourquoi utilise-t'on souvent une loi exponentielle pour modéliser la distribution de défaillance de composants dont le taux de panne suit la forme d'une courbe en baignoire.