



Durée : 2h — Documents autorisés

■ ■ ■ CUDA — 10 points

1– Soit le kernel suivant :

10pts

```
5  __global mon_kernel(...)
6  {
7    for(...)
8    {
9      if (condition) {
10         /* Traitement A */
11         ...
12       }
13     else {
14         /* Traitement B */
15         ...
16     }
17 }
18 }
```

et son lancement :

```
43 dim3 block_size;
44 block_size.x = 10;
45 block_size.y = 10;
46 dim3 grid_size;
47 grid_size.x = 100;
48 grid_size.y = 100;
49 mon_kernel<<<grid_size,block_size>>>(...)
```

Pour aider au débogage, on veut réaliser des *statistiques* sur l'exécution du kernel par une thread :

- ▷ savoir si la thread a exécuté le «Traitement A» ou le «Traitement B» ;
- ▷ savoir combien de fois elle a réalisé chacun de ces différents «Traitements» :
 - ◊ il y a une boucle en ligne 7 ;
 - ◊ la condition de la ligne 9 peut varier au cours de l'exécution de la boucle for.

Questions

- a. Combien de threads vont être exécutées et comment vont elles être organisées ? (1pt)
Vous décrierez comment la grille est définie et comment sont réparties les différentes threads.
- b. Proposez une structure de données qui permet de calculer les statistiques d'exécution : (1pt)
Vous décrierez la structure de données et la correspondance avec une thread et un bloc.
 - ◊ identifier précisément la thread ; (1pt)
Vous donnerez la formule de correspondance entre une thread et la structure de données proposée.
 - ◊ indiquer pour une thread donnée, le nombre de fois où chacun des traitements A et B a été fait. (1pt)
Vous indiquerez le code à ajouter dans l'écriture du kernel en indiquant entre quelles lignes s'insère votre code.
 - ◊ indiquer pour un bloc donné le nombre de fois où chacun des traitements A et B a été fait. (1pt)
Vous indiquerez le code à insérer ou vous réécrirez le kernel.
- c. Comment va être créée cette structure de données dans le «device» ? (2pts)
Comment va être fournie la référence vers cette structure de données au kernel ?
Comment ces statistiques vont être récupérées sur le «Host» ?
Est-il possible de cumuler des statistiques sur plusieurs exécutions successives du kernel en optimisant le temps d'exécution et comment faire ?
- d. Pouvez vous fournir une version utilisant la mémoire partagée du bloc ? (2pts)
À quoi faut-il faire attention ? *Vous indiquerez le code à insérer ou vous réécrirez le kernel.*
- e. On voudrait mettre au point un mécanisme d'exception permettant à une thread de signaler une erreur : (1pt)
 - ◊ est-il possible d'interrompre le travail des autres threads au moment où l'erreur survient ?
 - ◊ comment pouvoir identifier la thread sur laquelle l'erreur s'est produite ?
 - ◊ comment prévenir le Host que si une erreur s'est produite sur une thread, le résultat complet du lancement du kernel est faux ?*Vous discuterez de la faisabilité et des solutions possibles, sans proposer de code.*

■■■■ OpenMP — 10 points

2– Une société de livraison de fruits et légumes veut optimiser le traitement de ses produits lors de leur acheminement : **10pts**

- chaque produit est stocké dans un carton identifié par un numéro, un prix au kilo et un poids ;
- un camion ne peut transporter au plus que 1000Kg ;

Les données du problème sont stockées dans un tableau à deux dimensions, dont l'indice correspond au numéro de carton :

Produit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	N
Poids	6	3	6	6	3	2	4	6	5	8	2	6	2	2	...	7
Prix	38	49	34	11	46	39	43	47	16	11	34	12	29	44	...	30

L'optimisation consiste à explorer **toutes les possibilités** de chargement du camion en maximisant le produit de sa vente :

i. on construira une proposition de chargement :

On utilisera une relation d'appartenance pour exprimer une proposition de chargement de tous les cartons :

Produit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	N
Chargé	1	1	0	0	0	0	0	0	0	0	0	1	1	0	...	0

- ii. vérifier que le chargement est possible, c-à-d que la somme des poids est inférieure à 1000Kg ;
- iii. si le chargement est possible, évaluer le prix global de la proposition ;
- iv. recommencer pour toutes les propositions possibles et sélectionner celle qui a un prix global le plus élevé ;

Questions :

- a. Décrivez les structures de données que vous utiliserez pour trouver le meilleur chargement. (1pt)
- b. Combien de propositions doivent être construites sachant qu'il y a N cartons ? (3pts)
 Quelle est la complexité de l'algorithme de génération ?
 Donnez une méthode de génération parallèle de toutes les propositions de chargement.
 La complexité de cette méthode est-elle optimale ? Est-ce du parallélisme de contrôle ou de données ?
- c. Donnez une méthode parallèle de vérification de la faisabilité du chargement et du prix global de vente. (3pts)
 Est-il possible de proposer une méthode efficace améliorant l'équilibrage de charge ?
Vous justifierez votre réponse

Pour certain produits frais, le prix baisse s'il n'a pas été vendu le jour de production suivant une somme indiquée dans le tableau suivant :

Produit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	N
Malus	-2	-2	0	0	-1	0	-1	-1	-1	-1	-1	-2	0	-2	...	-1

On considérera que les cartons déjà vendus sont remplacés à l'identique chaque jour (le carton dispose du même poids et même contenu mais à un prix différent si un «malus» est appliqué.

- d. La société voudrait une projection de vente sur 4 jours, c-à-d la recherche du chargement lui rapportant le plus jour après jour après application du malus et la somme des profits de chacune de ces journées. (3pts)
 Comment allez vous calculer la somme de ces profits pour les 4 jours ? Devez-vous prendre des précautions particulières ?
 Est-ce du parallélisme de contrôle ou de données ?
 Sachant qu'elle exploite une machine équipée d'un «quadcore», donnez une solution parallèle optimisant l'équilibrage de charge.