

Examen partiel: jeudi le 27 octobre 2005 de 8h30 à 11h20

QUESTION 1 (15 points)

Skillicorn propose un modèle et une taxonomie des architectures de calcul parallèles dans laquelle on retrouve 28 types d'architectures. Soit une analogie basée sur le service aux clients en restauration: IP correspond à une serveuse, IM correspond à un client, DP correspond à un cuisinier et DM correspond aux endroits où sont stockés les ingrédients. Au niveau inférieur du modèle de Skillicorn, une machine à états finis décrit les étapes de traitement de IP et de DP. Par exemple, une étape de IP est "Instruct DP". Dans l'analogie ci-dessus, la serveuse informe le cuisinier sur le type de mets commandé par le client, par exemple une soupe.

- a) *Dites à quoi correspond chaque étape de traitement de IP et de DP dans cette analogie.*

- b) *Associez le type d'architecture le plus adéquat parmi les architectures de numéro 6, 8, 9, 14, 15 et 24 pour les types de restauration suivants (correspondance biunivoque): cantine mobile, casse-croûte, crèmerie, commande à l'auto en restauration rapide, restaurant de grand hôtel, marché public.*

QUESTION 2 (20 points)

Soit un algorithme séquentiel dont la proportion parallélisable p est égale à 0.5.

- a) *Calculez l'accélération obtenue en parallélisant cet algorithme sur 10 processeurs.*
- b) *Calculez la valeur des proportions séquentielle et parallèle de l'algorithme ainsi parallélisé.*
- c) *Montrez que l'accélération de Gustafson est la même que celle de Amdahl avec ces nouvelles proportions.*
- d) *Calculez le nombre de processeurs permettant d'atteindre 90% de l'accélération maximale de Amdahl.*

Examen partiel A-05

QUESTION 3 (10 points)

Nous avons présenté un modèle analytique du calcul parallèle permettant de faire ressortir l'importance du degré de parallélisation ou granularité du calcul (nombre de processus M) et des caractéristiques de l'architecture (ratio R/C et nombre de processeurs P). Nous avons vu que lorsque $P > 2$ et que chaque processus doit communiquer une fois avec chacun des autres processus, que ces communications se font séquentiellement et qu'elles ne se font pas en chevauchement avec les calculs, on obtient pour le temps de traitement (calculs et communications) de l'application (k_i est le nombre de processus alloués au processeur numéro i):

$$TC = \max(k_i)R + \left(M^2 - \sum_{i=1}^P k_i^2 \right) \frac{C}{2}$$

- a) Montrez que pour $M = 19$ et $P = 6$, une allocation équilibrée utilisant 5 des 6 processeurs donne un temps de calcul moindre qu'une allocation équilibrée utilisant 4 processeurs, ceci pour $R/C = 10$.

QUESTION 4 (10 points)

Soit le problème d'ordonnancement $PO = \{(20,6,1), (15,4,3), (10,4,5)\}$ où (T,c,b) sont les contraintes temporelles d'un processus, avec T sa période, c son temps de calcul et b sa contrainte de début en unités de temps. Pour ce PO , chaque processus est périodique et a une échéance égale à sa période. Le premier processus est le moins prioritaire et le troisième est le plus prioritaire. Le premier processus utilise la ressource Q de la deuxième à la cinquième unités de temps de son calcul. Le second processus utilise la ressource V aux deuxième et troisième unités de temps de son calcul. Le troisième processus utilise la ressource Q à la troisième unité de temps de son calcul et la ressource V à la quatrième.

- a) Montrez à quelle unité de temps se produit la première inversion des priorités en simulant l'ordonnancement. Donnez l'état des trois processus à cette unité de temps.

Examen partiel A-05

- b) *Calculez le nombre d'ordonnancements possibles sur la période d'ordonnancement, en considérant qu'à chaque unité de temps au plus un processus s'exécute avec une ressource ou sans ressource.*

QUESTION 5 (20 points)

Le routage adaptatif P-cube permet d'éviter les impasses. Il est une généralisation du routage "negative-first" dans les mailles 2-D. Ce dernier comporte deux phases: i) déplacements dans les directions négatives (sud et ouest) et ii) déplacements dans les directions positives (nord et est). Les directions de la maille 2-D correspondent aux dimensions du réseau. Dans la première phase, l'adresse diminue de 1 dans la dimension suivie (X ou Y) à chaque déplacement (saut ou "hop" négatif) vers un autre noeud le long du chemin. Dans la deuxième phase, l'adresse augmente de 1 dans la dimension suivie à chaque déplacement (saut ou "hop" positif). Le réseau binaire n-cube a n dimensions et l'algorithme P-cube utilise le même principe de routage en deux phases selon le signe des sauts. Il existe deux versions de l'algorithme P-cube: une version minimale et une version non-minimale. Soit un réseau hypercube 5-D dans lequel un message est envoyé du *noeud source* $S=01101$ au *noeud destination* $D=11010$.

- a) *Donnez le chemin suivi par l'algorithme de routage E-cube.*
- b) *Donnez tous les chemins possibles dans la version minimale du routage P-cube ainsi que leurs longueurs.*
- c) *Même chose pour la version non-minimale.*

Un programme parallèle comporte quatre processus concourants placés sur les noeuds 00001, 00101, 01001 et 01101. Chacun de ces processus doit faire une communication durant son traitement. Les sources et destinations de ces communications sont les suivantes (S:D): (00001:01111), (00101:01011), (01001:00111), (01101:00011). Le routage E-cube est utilisé avec la méthode de contrôle "store-and-forward".

- d) *Montrez que l'exécution de ce programme ne peut pas conduire à une impasse.*

Examen partiel A-05

QUESTION 6 (15 points)

Trois processus périodiques indépendants s'exécutent sur un processeur unique. Les contraintes temporelles sont sous la forme (T, c) avec T la période et c le temps de calcul.

- a) Calculez la valeur maximale de c_2 pour laquelle $PO = \{(15,3), (3,c_2), (10,2)\}$ est réalisable.
- b) Calculez le temps de réponse de chaque processus dans $PO = \{(15,3), (3,2), (10,2)\}$ si on utilise la stratégie rate-monotonic.

Examen partiel A-05

QUESTION 7 (10 points)

On désire effectuer une communication de type diffusion (“broadcast”) 1:31 à l’aide d’un algorithme parallèle optimal basé sur une topologie hypercube de 32 processus. Chaque processus ne peut faire qu’un transfert vers un autre processus à la fois. L’algorithme doit s’exécuter sur une architecture de réseau en arbre binaire sans perte de parallélisme.

- a) *Calculez le nombre total de transferts dans l’algorithme de diffusion.*

- b) *Calculez les valeurs minimum de la dilatation et de l’expansion (chaque processus de l’algorithme est placé sur un processeur différent de l’arbre).*