Mineure "Calcul parallèle"

Contrôle de classement (2 heures)

Eric Goubault

27 mars 2000

- les questions 1, 2 et 7 sont indépendantes,
- les questions 4 et 5 dépendent de la question 3 et d'un minimum de compréhension de la section 1,
- la question 6 dépend de 4 et 5,
- les questions 8 et 9 dépendent de la compréhension de ce qui est demandé à la question 7,
- la question 10, subsidiaire, dépend des questions 8 et 9.

1 Suites bitones

Soit $X = (x_0, \cdots, x_{n-1})$ une suite de n entiers. On supposera dans toute la suite du problème que $\underline{n = 2^k}$ (k étant un entier positif). On dit que X est *bitone* si il existe $j < n$ et un entier l, $j + n - 1 \geq l \geq j$ tels que,

- $x_{j \bmod n} \leq \cdots \leq x_{l \bmod n}$,
- $x_{l+1 \bmod n} \geq \cdots \geq x_{j+n-1 \bmod n}$

Question 1 Soit $X = (-5, -9, -10, -5, 2, 7, 35, 37)$. Montrer que X est bitone.

On définit maintenant deux opérateurs prenant en argument une suite bitone:

- $L(X) = (\min(x_0, x_{\frac{n}{2}}), \min(x_1, x_{\frac{n}{2}+1}), \cdots, \min(x_{\frac{n}{2}-1}, x_{n-1}))$,
- $R(X) = (\max(x_0, x_{\frac{n}{2}}), \max(x_1, x_{\frac{n}{2}+1}), \cdots, \max(x_{\frac{n}{2}-1}, x_{n-1}))$.

Question 2 Montrer que (pour le X de la question 1), $L(X)$ et $R(X)$ sont toutes deux des suites bitones telles que chaque élément de $L(X)$ est inférieur à tout élément de $R(X)$ (on ne demande pas de le prouver en toute généralité).

En fait cette propriété est vraie pour toutes les suites bitones (on n'essaiera pas de montrer ce résultat).

2 Tri de suites bitones

On veut maintenant écrire les algorithmes correspondants.

Question 3 Déduire de la section 1 un algorithme séquentiel récursif selon le schéma "diviser pour régner" permettant de trier une suite bitone.

Question 4 On suppose dans cette question uniquement que l'on dispose d'un réseau de stations travaillant sous PVM, ayant toujours suffisament de processeurs pour que chaque tâche nouvelle s'exécute sur un processeur nouveau. Dans l'écriture des pseudo-codes PVM demandés, on se contentera d'une écriture raccourcie des principales primitives de communication. On pourra écrire par exemple `spawn(escl ave,argv,k,tid)` (où `argv` représente une liste d'arguments passés à la création des processus esclaves, comme pour `pvm_spawn`), `send(tid[i],p(j) / 0 <= j <= n)` et `recv(tid[i],q)` pour dire respectivement, créer `k` copies du programme `esclave` dont on range les identificateurs respectifs dans le tableau `tid`, envoyer (même sémantique que `pvm_send`) la liste de valeurs `p(0),p(1), ...,p(n)` au `i`ème esclave, et enfin, recevoir (même sémantique que `pvm_recv`) une valeur de `tid[i]` à ranger dans la variable locale (éventuellement un tableau) `q`.

Ecrire un algorithme parallèle par décomposition fonctionnelle sur le modèle de l'algorithme de la question 3.

Question 5 Ici on suppose que l'on dispose d'une machine parallèle à p processeurs qui communiquent par mémoire partagée. On veut trier une suite bitone X à n éléments stockée dans la mémoire partagée. Comment peut-on paralléliser par décomposition de domaine l'algorithme de la question 3 sachant que p divise $\frac{n}{2}$? (indication: chaque processeur peut maintenant s'occuper d'une portion de X) Pour cela on écrira un pseudo-code (ressemblant au `C` mais avec en plus une barrière de synchronisation dont la sémantique est similaire à `pvm_barrier`) par processeur (P_0 à P_{n-1}) où on considérera que les écritures et les lectures de variables distinctes, ou d'éléments de tableaux différents, s'effectuent en parallèle.

Question 6 En négligeant les temps de communication et les temps correspondants à `pvm_spawn` (dans le cas de l'algorithme de la question 4) et les temps de lecture/écriture (dans le cas de l'algorithme de la question 5), et en supposant que le coûts de calcul sont uniquement dûs aux opérateurs min() et max() (coût unitaire),

(a) quel est le coût pour l'algorithme de la question 4 avec suffisament de processeurs pour qu'un seul processus soit affecté à chaque processeur,

(b) quel est le coût pour l'algorithme de la question 5 avec p processeurs,

(c) comparer ces deux temps de calcul.

On fera bien attention à argumenter les résultats, et à représenter en particulier les actions exécutées en parallèle, en supposant les processeurs de même rapidité.

3 Tri général

On part maintenant d'une suite quelconque $X = (x_0, \cdots, x_{n-1})$. On va construire une suite de suites $X^0 = X, X^1, \cdots, X^k$, avec X^k triée grâce aux remarques suivantes:

Question 7 On demande de prouver successivement:

(a) toute paire de valeurs, telles (x_0, x_1), (x_1, x_2) etc., est bitone,

(b) ordonnons donc ces paires (x_{4p}, x_{4p+1}) $(0 \leq p \leq \frac{n}{4})$ dans l'ordre croissant donnant ainsi (x^1_{4p}, x^1_{4p+1}), et les paires (x_{4p+2}, x_{4p+3}) dans l'ordre décroissant, donnant ainsi (x^1_{4p+2}, x^1_{4p+3}). Prouver que $(x^1_{4p}, x^1_{4p+1}, x^1_{4p+2}, x^1_{4p+3})$ est une suite bitone. Soit X^1 la suite à n valeurs, concaténation de ces quadruplets.

(c) a chaque étape, pour $q < k$, on obtient ainsi par tri de $(x^q_{2^{q+1}p}, \cdots, x^q_{2^{q+1}p+2^q-1})$ (sous-suite de X^q) dans l'ordre croissant, une suite $(x^{q+1}_{2^{q+1}p}, \cdots, x^{q+1}_{2^{q+1}p+2^q-1})$, et de $(x^q_{2^{q+1}p+2^q}, \cdots, x^q_{2^{q+1}p+2^{q+1}-1})$ (sous-suite de X^q) dans l'ordre décroissant $(x^{q+1}_{2^{q+1}p+2^q}, \cdots, x^{q+1}_{2^{q+1}p+2^{q+1}-1})$.

Prouver que $(x^{q+1}_{2^{q+1}p}, \cdots, x^{q+1}_{2^{q+1}p+2^q-1}, x^{q+1}_{2^{q+1}p+2^q}, \cdots, x^{q+1}_{2^{q+1}p+2^{q+1}-1})$ est bitone. On appelle ainsi la suite résultante X^{q+1}.

On appelle X^k le résultat du tri (de la section 2) appliqué à la suite X^{k-1}.

Question 8 En utilisant l'algorithme de la question 4 comme une procédure "boîte noire" (c.a.d. que vous pouvez supposer l'avoir écrite même si vous ne l'avez pas fait), écrire un pseudo-code PVM permettant de trier toute suite X grâce aux remarques de la question 7.

Question 9 En utilisant l'algorithme de la question 5 comme une procédure "boîte noire", écrire un pseudo-code pour notre machine parallèle à mémoire partagée permettant de trier toute suite X grâce aux remarques de la question 7.

Question 10 subsidiaire Quelle est la complexité de l'algorithme de la question 8? de la question 9?