

# Master d'informatique 1ère année — Devoir Surveillé

## Programmation des Architectures Parallèles

Durée : 1h30 — Sans document

### 1 Cohérence des caches

On considère une machine quadripcesseurs à bus mémoire unique dans laquelle la cohérence des caches (assurée par *snooping*) utilise le protocole MESI à quatre états : modified (M), exclusive-clean (E), shared (S) et invalid (I).

**Question 1** On considère les accès successifs à une même variable  $x$  initialement en mémoire centrale<sup>1</sup>. Dessinez, pour chacune des étapes ci-dessous, les états des caches de chacun des processeurs et indiquez la nature des éventuelles transactions sur le bus mémoire (BusRd ou BusRdX) :

1.  $P_0$  lit  $x$
2.  $P_1$  modifie  $x$
3.  $P_2$  lit  $x$
4.  $P_2$  modifie  $x$
5.  $P_3$  lit  $x$

**Question 2** On considère maintenant une architecture NUMA à 16 processeurs constituée par l'interconnexion de quatre blocs quadripcesseurs SMP. Les processeurs  $P_0, P_1, P_2$  et  $P_3$  appartiennent au premier bloc.  $P_4, P_5, P_6$  et  $P_7$  appartiennent au second bloc, etc.

Rappelez le principe de gestion de la cohérence des caches par utilisation de "répertoires". Dans quelle(s) situation(s) un accès mémoire en écriture peut-il se révéler très coûteux ?

On considère la suite d'actions suivante :

1.  $P_0$  lit  $x$
2.  $P_2$  lit  $x$
3.  $P_6$  modifie  $x$

Discutez, suivant la localisation de  $x$ , du coût de chacune de ces trois opérations.

### 2 Protocoles de communication "zéro-copie"

**Question 1** Rappelez le principe général d'une communication utilisant un protocole "zéro-copie". Dans quelles circonstances cherche-t-on à mettre en œuvre un tel protocole ? Expliquez.

**Question 2** Rappelez ce que signifie l'expression "réaliser les communications en mode utilisateur". Expliquez précisément la difficulté de réaliser des communications à la fois en mode utilisateur et en utilisant un protocole zéro-copie. Imaginez une évolution des cartes réseau (e.g. un circuit que l'on pourrait leur ajouter) qui permettrait de résoudre le problème et qui demanderait juste une extension "raisonnable" du système d'exploitation...

**Question 3** Le protocole TCP/IP n'est pas vraiment réputé pour sa capacité à effectuer des communications en mode utilisateur. Est-ce fondamentalement dû à son interface ou est-ce simplement un "problème" d'implémentation ? Compte-tenu de l'utilisation de ce protocole au-dessus de réseaux à longue distance (i.e. haut débit mais latence importante), cette "limitation" vous paraît-elle justifiée ?

---

<sup>1</sup>et initialement dans aucun cache de processeur, of course.

### 3 BIP/Myrinet

**Question 1** Décrivez les deux protocoles utilisés par le protocole BIP/Myrinet pour acheminer respectivement des messages courts et des messages longs. Expliquez précisément comment il faut s’y prendre, lorsque l’on développe une bibliothèque telle que BIP, pour déterminer le seuil à partir duquel on considère qu’un message est “long” ?

Dans les deux cas de figure, pourquoi peut-on perdre des messages avec BIP si l’on ne prend pas de précautions particulières ?

**Question 2** On souhaite développer au-dessus de BIP une interface d’envoi de messages capable d’envoyer, de manière fiable, des messages constitués de plusieurs segments non contigus (en mémoire virtuelle). Concrètement, il s’agirait les fonctions habituelles `send` et `recv` qui auraient cette fois pour arguments des tableaux de couples (`adresse`, `longueur`). Appelons-les `send_multi` et `recv_multi`.

Sans écrire de code, faites des petits dessins pour décrire le protocole que vous utiliseriez pour implémenter `send_multi` et `recv_multi` en utilisant `bip_isend` et `bip_irecv`. Discutez en fonction de la taille des segments (e.g. tous les segments sont gros/tous les segments sont petits/certains segments sont gros et d’autres petits).

Dans le cas où les segments du message à transférer sont tous gros, y a-t-il un gain à utiliser les primitives `send_multi/recv_multi` par rapport à `send_recv` au niveau de l’application ? Que faudrait-il pouvoir faire au niveau de BIP pour améliorer la situation ?

### 4 Version distribuée de l’application du voyageur de commerce (TSP)

On se place dans le cadre de l’application du voyageur de commerce (*Travelling Salesman Problem*) étudiée en travaux dirigés. On considère l’implémentation distribuée (MPI) de cette application. On s’intéresse plus précisément à la gestion de la variable “partagée” indiquant en permanence la longueur du plus court chemin trouvé jusqu’alors.

**Question 1** Donnez plusieurs stratégies possibles concernant la mise à jour de la copie locale de la variable `valeur_plus_court_chemin` sur chacun des processus et discutez des avantages/inconvénients de chacune.

**Question 2** Indiquez les deux principales façons de traiter la réception des messages de mise à jour, l’une utilisant une primitive de réception et l’autre non. Y a-t-il une solution meilleure que l’autre dans l’absolu ? Discutez en fonction du comportement de l’application.