Examen de « Parallélisme, systèmes distribués et grille »

Master ILC – M2 – 2010-2011

Tous documents autorisés

Durée : 2h00

Attention : faire les deux parties de l'examen sur des feuilles séparées.

I – Partie de S. Vialle (10 points)

Exercice 1 : Analyse de performances (4 points)

On considère un cluster de CPUs bi-cœurs : chaque nœud possède un CPU à 2 cœurs physiques non hyperthreadés. Le réseau d'interconnexion est un très bon réseau Gigabit Ethernet (très bon switch).

On a exécuté et mesuré les performances de deux applications très différentes sur ce cluster : une simulation de phénomène physique 3D, et un réseau de neurones d'inspiration biologique (réseaux de neurones tentant de mimer le fonctionnement de parties du cerveau). Dans les deux cas on a utilisé une parallélisation basée sur MPI, et on expérimenté le déploiement d'un processus MPI par nœud, et de deux processus MPI par nœud. Il n'y a pas de multithreading : dans le cas où l'on déploie un processus MPI par nœud, alors on n'utilise qu'un seul cœur par nœud.

On a obtenu les courbes de performances représentées sur les figures 1 et 2. Pour chaque application on a tracé son temps d'exécution (pour un problème de taille fixe) en fonction du nombre de nœuds utilisés, et en fonction du nombre de processus MPI lancés (l'argument de l'option « -np » de « mpirun »).

Rmq : Le problème du réseau de neurone est trop gros pour tenir sur un seul nœud (pas assez de mémoire). Il fallait au minimum 3 nœuds pour l'exécuter. C'est pourquoi les courbes de performances de ce problème ne commencent pas à 1 nœud.

Q1.1 : Est-ce que ces applications exploitent bien les nœuds multi-cœurs du cluster ? (justifiez votre réponse)

Q1.2 : Les calculs accomplis par ces applications sont-ils limités par les capacités de calcul des nœuds ? ou par leur capacité d'accès aux données en mémoire ? (justifiez votre réponse)

Q1.3 : Est-ce que le réseau d'interconnexion à l'air assez rapide ? (justifiez votre réponse)

Q1.4 : Un utilisateur veut exécuter ces applications sur de gros problèmes. Il demande à utiliser intensivement le cluster. Comment lui conseillez-vous d'utiliser les nœuds pour chaque problème ? (comment doit-il déployer ses processus ?)

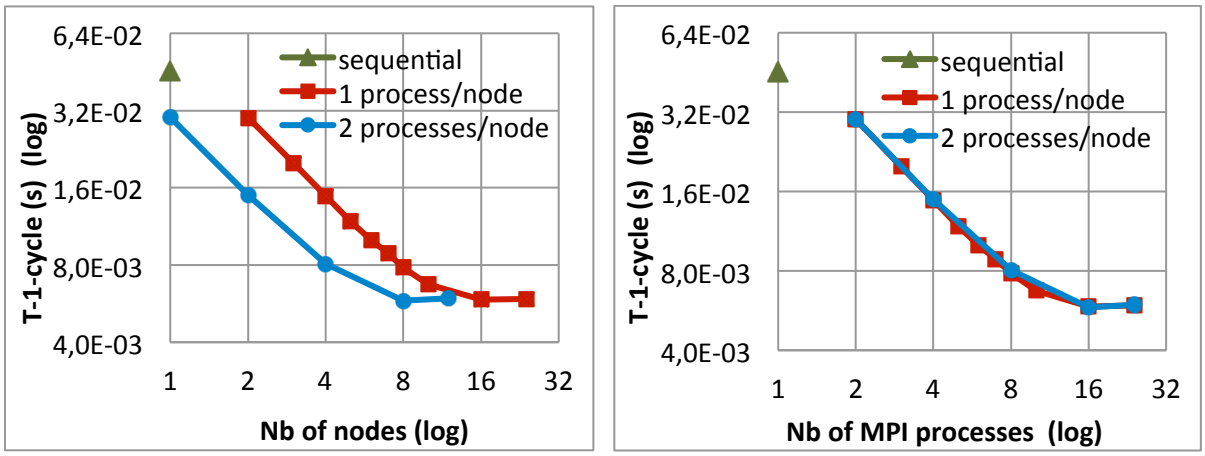


Fig 1 : Simulation de phénomène physique 3D

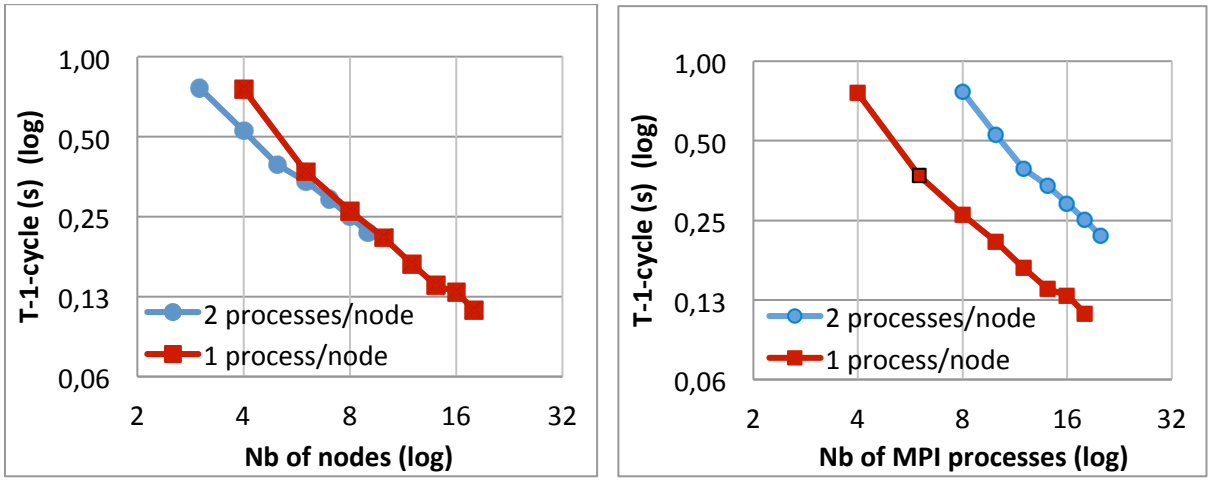


Fig 2 : Réseau de neurones (d'inspiration biologique)

Exercice 2 : Distribution de calculs sur un cluster (6 points)

On considère un cluster de P PCs, sur lequel est installé MPI, et on considère la fonction de calcul séquentiel suivant :

```
//-------------------------------------
extern double f(double tab1[L], double tab2[L]);
double Tab[N][L];
void calcul()
{
  double res = 0.0;
  for (int i = 0; i < N-1; i++) {
     res += f(Tab[i],Tab[i+1]);
  }
  printf("Final result = %f\n",(float) res);
}
//-------------------------------------
```

Rmq : La fonction « f » ne fait que lire le tableau Tab, elle ne le modifie pas.

Rmq : On suppose que le tableau Tab est présent en totalité en mémoire sur le processeur 0.

Rmq : On suppose que l'on déploie un seul processus MPI par nœud.

Q2.1 : Quelles données faut-il envoyer sur le processus « i » (0 <= i < P) pour qu'il puisse faire ses calculs et que la charge de travail soit équitablement répartie sur les P processus ?

Rmq : On suppose que N est un multiple de P.

Q2.2 : Ré-écrivez la fonction « calcul » pour qu'elle s'exécute sur les « P » PCs du cluster :

- le processus 0 envoie une partie des données à chaque autre PC, et les autres PC attendent de recevoir leurs données,

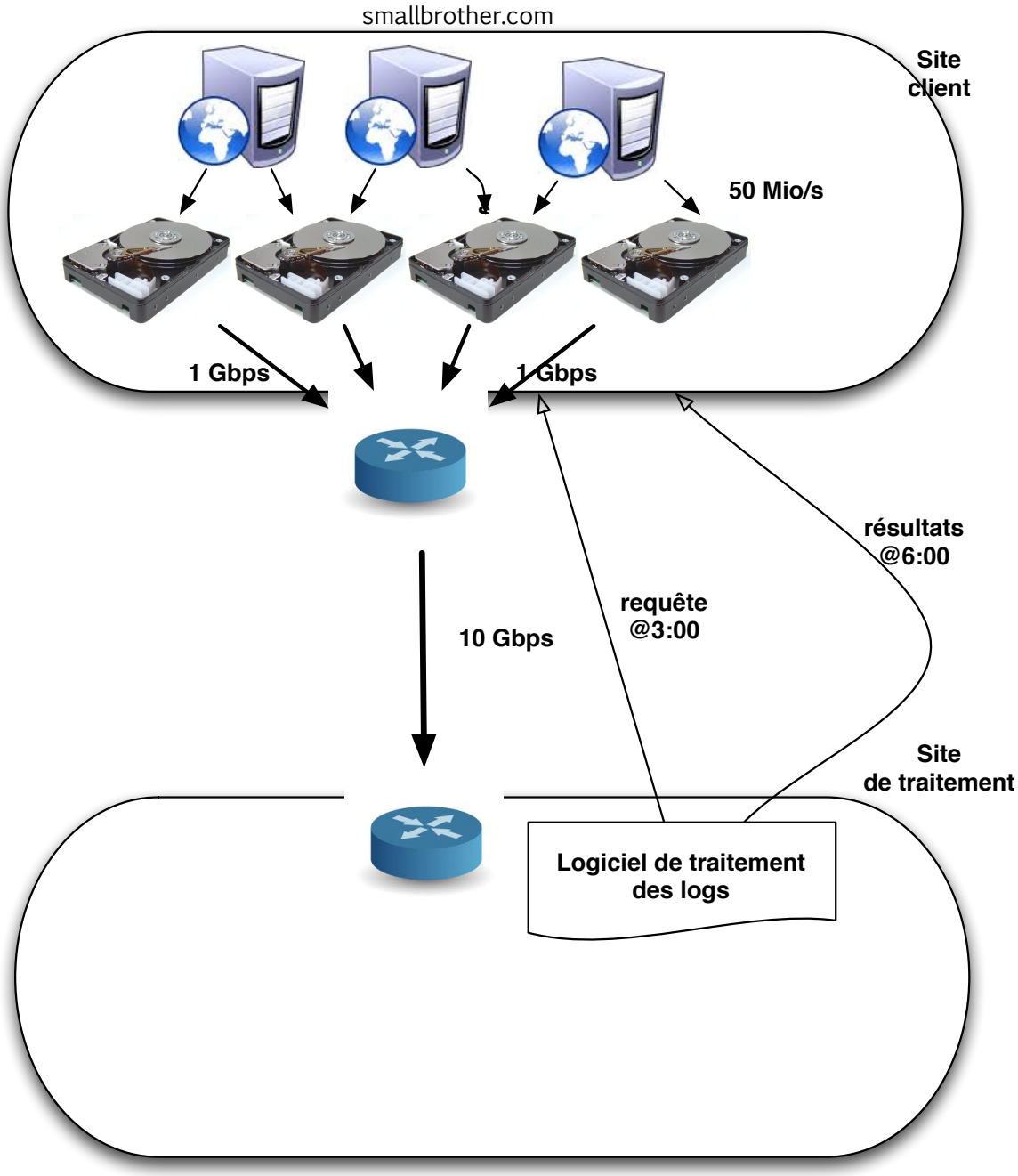
- puis chaque processus fait sa part de calculs,

- enfin le processus 0 reçoit les résultats des autres PC et affiche le résultat final.

Vous pouvez écrire cette petite fonction parallèle à l'aide de communication en MPI_Bsend et MPI_Recv, ou en MPI_Ssend et MPI_Recv.

II – Partie de S. Genaud (10 points)

Q3 : 10 pt : Une entreprise, que l'on désignera par *le fournisseur* dans la suite, vient d'obtenir un contrat pour le traitement des logs d'un site internet à forte fréquentation nommé *SmallBrother*. On vous demande de proposer une solution d'ensemble pour mettre en place ce traitement. Cette solution peut consister en des préconisations d'équipements logiciels et ou matériels chez le client, plus une ou des solutions à mettre en place côté fournisseur.



Le volume de log généré par 24 heures est estimé à 1 million de lignes (format enrichi) pour un total de 1 To stockés dans environ 10000 fichiers. *SmallBrother* effectue une copie des logs de ses serveurs web Apache vers une batterie de disque vers minuit. Smallbrother accepte de donner un accès en

lecture à ces disques à partir de 3h00 du matin, du moment qu'ils reçoivent une requête dûment authentifiée du fournisseur. A partir de l'ordre de lecture, le délai de transfert vers le fournisseur peut être établi selon les caractéristiques suivantes : chaque disque peut être lu à 50 Mo/s, chaque disque contient une quantité égale de données, chaque machine hébergeant un disque peut transférer à 1 Gb/s vers le routeur connecté à l'extérieur. Le routeur et le lien vers l'extérieur ont une bande passante de 10 Gb/s.

Le traitement des logs consiste à fabriquer un index des sites référents pour SmallBrother à partir des lignes de logs. Rappelons que chaque visite d'une page web produit une ligne de log distincte, contenant l'IP du visiteur, le type de navigateur, la date, etc, ainsi que le référent, c'est-à-dire le site ayant renvoyé vers cette page. Le résultat du traitement est une liste de couples (*url,nombre*) , où *nombre* est le nombre de fois que le référent *url* apparaît dans les logs.

SmallBrother souhaite que ces traitements soient effectués dans un laps de temps de 3 heures, de façon à ce que les résultats puissent être utilisés dès 6h00.

On vous demande de faire une proposition de solution. Vous direz, entre autres :

a) Quel est le nombre de disques que vous préconisez pour la baie de disque recevant la copie des logs. Justifiez par une évaluation approximative des volumes à transférer et des temps de transferts.

b) Quel logiciel préconisez vous pour initier et piloter les transferts des logs vers le fournisseur.

c) Que conseillez vous pour programmer le traitement des logs ? Dites quels traitements on peut paralléliser. Ecrire le programme parallèle en pseudo-code dans le modèle de programmation que vous préconisez.

d) Esquissez la complexité de votre programme, en prenant en compte les paramètres suivants:,

- P : le nombre de processeurs,
- *f* : le temps constant de lecture ou écriture dans un fichier pour un bloc de données jusqu'à 1 Mo,
- c : le temps de communication constant pour un message envoyé d'un processeur à un autre pour des messages jusqu'à 100 Ko,
- *i* : le temps d'une opération élémentaire.

Considérez l'utilisation d 'un algorithme de tri de n éléments en $O(n \log n)$.