TP1 : MPI, Introduction

Bram De Wachter

23 février 2005

1 Introduction

Le MPI (Message Passing Interface) est un standard pour une bibliothèque utilisée pour écrire des programmes en Fortran et C parallèles, utilisant le modèle de programmation par passage de messages (message passing). Elaboré dès 1992 par le Forum MPI (MPIF) réunissant aussi bien les constructeurs (Cray, NEC, IBM,...) que des représentants du milieu académique, ce standard a été implémenté sur des architectures à mémoire partagée ainsi que celles à mémoire distribuée.

Dans un modèle de parallélisme par échange de messages chaque processus exécute éventuellement des parties différentes d'un programme (écrit dans un langage classique comme le Fortran ou le C). Comme ce modèle de programmation parallèle est explicite, la gestion des communications et des synchronisations est donc entièrement à la charge du développeur. Ce modèle concerne les applications possédant plusieurs traitements indépendants. Le parallélisme découle de la réalisation simultanée de différents traitements sur les données.

Les variables sont alors privées et résident dans la mémoire locale de chaque processus. Une donnée ne peut être échangée entre deux ou plusieurs processus qu'au travers d'appels à des sous-programmes spécialisés constituant une bibliothèque de routines d'échanges de messages. PVM (Parallel Virtual Machine) et MPI (Message Passing Interface) sont des bibliothèques standards, dont les applications sont portables sur les machines parallèles actuelles. Il faut noter que PVM s'applique aussi aux réseaux de machines hétérogènes ce qui n'est pas le cas de MPI-1. SHMEM est une bibliothèque propre à SGI/CRAY. Enfin, avec l'arrivée de la norme MPI-2 les fonctionnalités de la bibliothèque MPI sont proches de celles de PVM tout en étant plus stables. Par conséquent il est préférable de développer plutot avec MPI qu'avec PVM.

2 Exercices

Ce qu'on vous demande c'est d'écrire un exemple `Hello world !` , utilisant MPI sur *lit5*. Deux versions de `Hello world !` doivent être réalisées.

Premierement, écrivez un programme MPI qui réalise N processus identiques, tous disant `Hello world !` à l'aide de la fonction C *printf*. Deuxiemement, changez le programme pour que le processus 0 devienne recepteur des messages `Hello world !` venant des autres N-1 processus. C'est le processus 0 qui affiche ces messages.

References :
http ://www.ulb.ac.be/di/ssd/bdewacht/i152