

Systèmes distribués: MPI et le lancé de rayon

Grégory Mounié

2012-11-15 jeu.

1 Mise en place

Générez une clef ssh et activez l'autorisation de connection sur toutes les machines :

```
ssh-keygen -t rsa # avec une passphrase !
ssh-copy-id localhost
ssh-add # pour donner la clé (avec passphrase à votre agent)
```

Vérifiez que vous pouvez vous connecter sur une autre machine sans taper de mots de passe.

2 Prise en main

Créez un fichier `hostfile.txt` contenant la liste de machine que vous utiliserez (typiquement localhost au départ).

Tapez, compilez (`mpicc`) et lancez le programme (`mpirun`) d'anneau à jeton suivant sur votre liste de machine:

```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char **argv) {
    int taille, rang, hostlen;
    char hostname[MPI_MAX_PROCESSOR_NAME]={};
    double message=rang+123;
    int TAG = 123456;
    MPI_Status status;

    MPI_Init(& argc, &argv);
```

```

MPI_Comm_size(MPI_COMM_WORLD, &taille);
MPI_Comm_rank(MPI_COMM_WORLD, &rang);
MPI_Get_processor_name(hostname, &hostlen);

if (rang == 0) {
    MPI_Send(& message, 1 , MPI_DOUBLE,
             (rang +1)%taille, TAG, MPI_COMM_WORLD);
    MPI_Recv(& message, 1, MPI_DOUBLE,
             (taille -1), TAG, MPI_COMM_WORLD, & status);
    printf("Message %g reçu ! %d parmi %d sur %s!\n",
           message, rang, taille, hostname);
}
else {
    MPI_Recv(& message, 1, MPI_DOUBLE,
             (rang -1), TAG, MPI_COMM_WORLD, & status);
    printf("Message %g reçu ! %d parmi %d sur %s!\n",
           message, rang, taille, hostname);
    MPI_Send(& message, 1 , MPI_DOUBLE,
             (rang +1)%taille, TAG, MPI_COMM_WORLD);
}

MPI_Finalize();
}

```

3 Lancé de rayon

Téléchargez l'archive depuis ensiwiki puis décompressez l'archive.

3.1 Exercice

1. Ecrivez une version distribuée du code de lancé de rayon.
2. Assurez vous que tout fonctionne (attention, l'exécution n'est pas forcément déterministe).
3. Tracez quelques courbes d'accélération, en particulier pour les exemples *s3.txt* et *s4.txt*