

---

# Systèmes distribués et grilles : TP1

**Thibaut Ehlinger, Benjamin HERB** Université de Strasbourg

---

19 septembre 2016

**L**e but de ce TP est de mettre en évidence l'impact de l'utilisation de *clusters* de PC dans l'exécution d'un algorithme très gourmand en calcul, ici un produit de matrices. Dans un premier temps, nous présenterons rapidement notre résolution du problème pour le code manquant dans le projet. Puis nous présenterons les résultats des différents essais; résultats que nous analyserons ensuite afin d'essayer de trouver quel est le choix le plus judicieux pour exécuter un produit de matrices de  $4096 \times 4096$  lignes.

## Résolution du problème

Dans un premier temps, nous devons compléter trois parties spécifiques du code, qui concernaient l'initialisation MPI, le calcul de l'*offset* de départ dans la matrice pour chacun de processus en fonction de leur identifiant MPI, et enfin les différentes étapes de la boucle principale, c'est à dire l'échange des données et le calcul des sous-parties de la matrice résultat par chaque processus. Cette partie se décompose en deux sous-fonctions : `ComputationAndCirculation` et `OneStepCirculation`.

## Initialisation MPI

```
1 #include <stdio.h>
2 #define N 10
3
4 /* Initialisation of processor coordinates
5  */
6
7 void ProcessorInit(void){
8
9 }
```

## Calcul de l'offset pour un processus Me donné

```
/* Compute the current step offset, in the
   MPI program, to access right C lines
   */ OffsetStepLigC = ((Me + step) *
   LOCAL_SIZE) % SIZE;
```

## Boucle principale

```
1 void ComputationAndCirculation()
2 {
3     unsigned long step;
4 }
5 /* Elementary circulation of A and B.
6
7  */
8 void OneStepCirculation(unsigned long step
9 )
10 {
11     MPI_Status status;
```

## Résultats et interprétations