

TP n°1 : Installation et premiers exercices

1) Installation

Connexion sur plafrim

**ssh plafrim**

Il faut créer un répertoire pour accueillir les sources d'ExaStamp et les Tps :

**mkdir TPs\_In\_Situ**

**cd TPs\_In\_Situ**

Il faut récupérer les sources ExaStamp avec la commande :

**git clone /home/tcarrard/exaStamp.git**

Puis, il faut lancer le script afin de configurer l'environnement (modules, path, ...) :

**\$HOME/TPs\_In\_Situ/exaStamp/scripts/configure-plafrim.sh**

Taper *Return* pour accepter les défauts (chemins vers les répertoires de build et d'install), ensuite

**source \$HOME/build/setup-env.sh**

**salloc -proutage -n1 -c12 make -j24**

A partir de ce moment, vous devez avoir un binaire du code ExaStamp appelé : **xstampv2**

Le binaire **xstampv2** se trouve dans **\$HOME/build**

1.5) Vérifier que vous avez l'outil **vite** sur votre station ou portable

Pour initialiser l'environnement nécessaire à l'exécution de vite :

**source ~mfaverge/cisd/env\_vite.sh**

2) Premiers tests sur **plafrim** (Exercice N°1)

Créer un répertoire Exercice1 dans Tps\_In\_Situ et copier le fichier tutorial\_insitu\_histo\_par.msp

**cd \$HOME/TPs\_In\_Situ && mkdir Exercice1**

**cp exaStamp/data/samples/tutorial\_insitu\_histo\_par.msp Exercice1/**

Ce jeu de données va faire un calcul de DM avec un potentiel peu coûteux (Lennard-Jones) et une analyse simple de type histogramme sur la valeur de l'énergie des atomes, parallélisée en OpenMP.

Première exécution en **séquentiel** :

**cd \$HOME/TPs\_In\_Situ/Exercice1**

**salloc -proutage -n1 -c1 /bin/env OMP\_NUM\_THREADS=1 \**

**\$HOME/build/xstampv2 tutorial\_insitu\_histo\_par.msp**

Maintenant, on active la génération de trace vite en ajoutant **--profiling-vite trace.vite**

**salloc -proutage -n1 -c1 /bin/env OMP\_NUM\_THREADS=1 \$HOME/build/xstampv2 \**  
**tutorial\_insitu\_histo\_par.msp --profiling-vite trace.vite**

Pour exécuter la simulation en **parallèle** sur 4 threads :

```
salloc -proutage -n1 -c4 /bin/env OMP_NUM_THREADS=4 $HOME/build/xstampv2 \
tutorial_insitu_histo_par.msp --profiling-vite trace.vite
```

#### Exercices :

- 1 Modifier le « sample » de l'analyse histogram\_energy dans le jeu de données tutorial\_insitu\_histo\_par.msp
- 2 En observant les résultats obtenus dans les traces vite, que pouvez-vous dire sur l'utilisation de l'analyse histogram\_energy ?
- 3 copier le fichier ../exaStamp/data/samples/tutorial\_insitu\_histo\_seq.msp dans votre répertoire Exercice1 (il utilise une autre version de l'histogram **histoseq\_energy**). Après une première exécution du jeu de données sans le modifier, regarder les sorties **vite** pour des exécutions séquentielles et parallèles. Augmenter le nombre de cœurs utilisés dans la commande salloc. Que Remarquez-vous ?

Vous pouvez vous aider avec l'option --profiling-summary true du code :

```
salloc -proutage -n1 -c4 /bin/env OMP_NUM_THREADS=4 $HOME/build/xstampv2 \
tutorial_insitu_histo_seq.msp --profiling-vite trace.vite --profiling-summary
```

Si vous utilisez le jeu de données précédent qui utilise l'analyse **tutorial\_insitu\_histo\_par.msp** est-ce le même comportement ?

### 2,5) Avant de modifier les sources

à partir de maintenant, vous allez modifier les sources. Il est donc nécessaire que chacun crée une branche nouvelle dans le dépôt GIT.

```
cd $HOME/Tps_In_Situ/exaStamp
```

```
git checkout -b enseirb-$USER # $USER sera remplacé par votre login
après chaque changement fructueux (qui compile et fonctionne)
```

```
git commit -am « ce que j'ai changé »
```

### 3) Regardons un jeu de données et parallélisation d'une analyse (Exercice n°2)

**Exercice** : Parallélisons l'analyse histoseq\_energy utilisée par le fichier de données

**tutorial\_insitu\_histo\_seq.msp**

Le fichier à modifier se trouve dans **\$HOME/TPs\_In\_Situ/exaStamp/src/tutorial/histogram\_seq.cpp**

- 1 Première version parallèle de **histoseq\_energy** pour cela ajouter les directives OpenMP adéquates et si nécessaire modifier le code source.
- 2 Compiler et exécuter le nouveau code sur 1 à 8 threads et vérifier que l'analyse s'exécute bien en parallèle.

#### 4) Étude avec plusieurs analyses de type histogramme (Exercice N°3)

On utilise un nouveau fichier de données :

`~/TPs_In_Situ/exaStamp/data/samples/tutorial_insitu_histo2_par_freq.msp`

Maintenant copier ce fichier dans le répertoire **Exercice1**

#### **Exercices :**

- 1 Étudier le jeu de données. Combien d'analyses vont être exécutées et à quelles fréquences ?  
Vérifier votre hypothèse avec des traces vites.
- 2 Faites varier le nombre de threads. Que constatez-vous ?
- 3 Modifier la fréquence d'appel des analyses et éventuellement le nombre de ***samples*** et vérifier que les traces vite valident bien votre hypothèse.
- 4 Qu'est-ce qui peut perturber les mesures ?