

# DOCUMENT D'INSTALLATION

Grégory Runser – SAE3.02 – Développer un cluster de calcul



Université de Haute-Alsace, IUT de Colmar  
BUT Réseaux et Télécommunications

## Sommaire

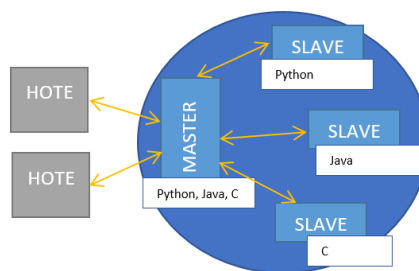
### Table des matières

Sommaire .....	1
Préambule .....	1
Téléchargement pour le Client.....	2
Télécharger le ZIP .....	2
Extraire les fichiers .....	2
Téléchargement pour les Serveurs.....	3
Installation pour le Serveur Maître .....	3
Installation pour un Serveur Esclave .....	3
Running Coté Serveur .....	4
Serveur Esclave.....	4
Serveur Maître.....	5
Running Coté Client - Hôte .....	6
Exécuter un code .....	6

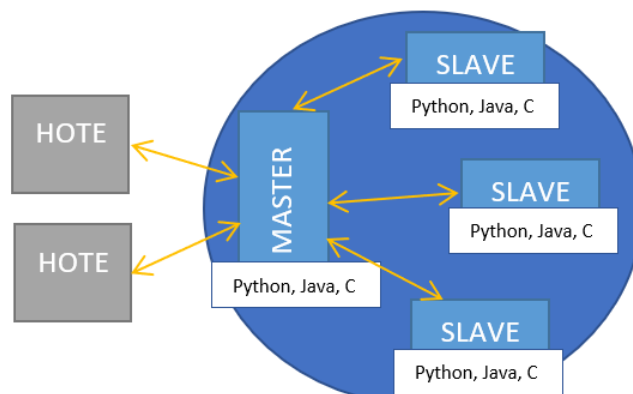
## Préambule

Ce document est un outil qui vous permettra de mettre en place le cluster de calcul. Il vous indiquera les étapes à suivre, pas à pas, que vous soyez novice ou expert en programmation.

Mon cluster a été fait de tel sorte qu'un script arrive sur le serveur maître, il peut être traité, quel que soit le langage, et renvoyé sans l'aide de serveur esclave. Le serveur peut être activé si vous restreignez le nombre de tâche en simultanée du serveur maître. **Il est donc impossible de faire un cluster de ce genre :**



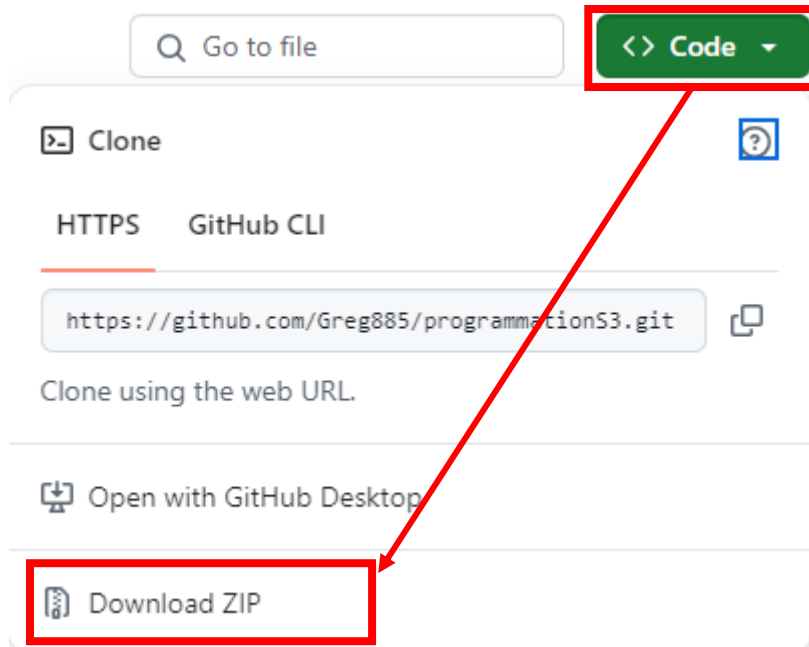
Il faut mettre en place un cluster de ce genre en installant Python, Gcc, ... sur chaque Serveur



## Téléchargement pour le Client

### Télécharger le ZIP

Dans un premier temps vous devrez télécharger l'ensemble du fichier ZIP GitHub présent sur ce lien ci : <https://github.com/Greg885/programmationS3.git>



### Extraire les fichiers

Vous venez à présent de télécharger un dossier compresseur, il faut donc extraire les fichiers pour pouvoir les utiliser : (Par étapes)

1. Windows + E (ouverture de l'explorateur de fichier)
2. Aller dans la section « Téléchargements »
3. Clic droit sur le dossier « programmationS3-main.zip »
4. Extraire les fichiers

Vous venez de rendre les fichiers utilisables.

## Téléchargement pour les Serveurs

### Installation pour le Serveur Maitre

Sur une VM Debian 12 : (par étape)

1. Lancé une Machine Virtuelle via Oracle ou VMware
2. Ouvré un CLI<sup>1</sup>
3. Passer en mode super user<sup>2</sup>
4. Puis taper les commandes suivantes<sup>3</sup> :
  - a. *apt update*
  - b. *apt install python3*
  - c. *apt install git*
5. Cloner le répertoire GitHub avec la commande :  
*git clone https://github.com/Greg885/programmationS3*
6. Vérifier le clonage du répertoire avec la commande :  
*ls*
7. Vous devriez apercevoir : *./ProgrammationS3*

### Installation pour un Serveur Esclave

Sur une VM Debian 12 : (par étape)

1. Lancé une Machine Virtuelle via Oracle ou VMware
2. Ouvré un CLI
3. Passer en mode super user
4. Puis taper les commandes suivantes :

*apt update*

*apt install python3*

*apt install git*

5. Cloner le répertoire GitHub avec la commande :

*git clone https://github.com/Greg885/programmationS3*

6. Vérifier le clonage du répertoire avec la commande :

*ls*

7. Vous devriez apercevoir : *./ProgrammationS3*

---

<sup>1</sup> Command Line User

<sup>2</sup> Avec la commande : *su -*

<sup>3</sup> Toute les lignes en *Italique* sont des commandes à rentrer dans le CLI

## Running Coté Serveur

### Serveur Esclave

Vous venez de cloner le répertoire où se trouve le fichier python permettant le lancement du serveur **ESCLAVE**.

Pour le lancer : (Par étapes)

1. Se déplacer à l'endroit où se trouve le fichier :

```
cd programmationS3/SAE3.02/Graphique/PyQt/
```

2. Cré un environnement python :

```
apt install python3.11-venv
python3 -m venv venv
source venv/bin/activate
```

3. Récupérer l'adresse IP du serveur

```
ip a
```

4. Noter l'adresse IP
5. Modifier l'@IP et port pour chaque serveur esclave (a refaire sur chaque VM Slave)

*Nano slave.py*

Se déplacer jusqu'à la ligne 13 (vers le début)

```
def __init__(self, host="127.0.0.1", port=10001): #Modification port ici
    Modifier l'@IP avec l'IP de la VM et le port : Slave1 = 10001, Slave2 = 10002, ...
    Sauvegarder et quitter l'éditeur de texte
    Attention ces informations sont importantes notez les il vous les faudra plus tard
```

6. Lancer le serveur :

```
pip install psutil
python3 slave.py Y4
```

7. Vérifier que depuis votre Hôte vous communiquer avec le serveur (Ping), si cela n'est pas le cas vérifier les réglages de votre VM (Bridge, Nat, Host-Only)

---

<sup>4</sup> Y= Numéro du port du Serveur Esclave (Slave1 = 10001) /\ A remplacer par un nombre

## Serveur Maitre

Vous venez de cloner le répertoire où se trouve le fichier python permettant le lancement du serveur **MAITRE**.

Pour le lancer : (Par étapes)

1. Se déplacer à l'endroit où se trouve le fichier :

```
cd programmationS3/SAE3.02/Graphique/PyQt/
```

2. Cré un environnement python :

```
apt install python3.11-venv
```

```
python3 -m venv venv
```

```
source venv/bin/activate
```

3. Modifier la liste des serveur esclave

```
Nano master.py
```

Se déplacer jusqu'à la ligne 211 (vers la fin)

Modifier la ligne `slave_ips`

Syntaxe : `[(@IP_VMslave1, Port_slave1), (@IP_VMslave2, Port_slave2)]`

Sauvegarder et quitter l'éditeur de texte

4. Lancer le serveur :

```
pip install psutil
```

```
python3 master.py X5
```

5. Récupérer l'adresse IP du serveur

```
ip a
```

6. Noter l'adresse IP

7. Vérifier que depuis votre Hôte vous communiquer avec le serveur (Ping), si cela n'est pas le cas vérifier les réglages de votre VM (Bridge, Nat, Host-Only)

---

<sup>5</sup> X= Nombre de tache exécuter parallèlement avant l'appel du serveur esclave /!\ A remplacer par un chiffre

## Running Coté Client - Hôte

Sur l'hôte nous avons extrait précédemment les fichiers.

Vous pouvez alors lancer un client de cette manière : (par étapes)

1. Ouvrir un CLI (PowerShell)
2. Se déplacer à l'endroit où se trouve le fichier

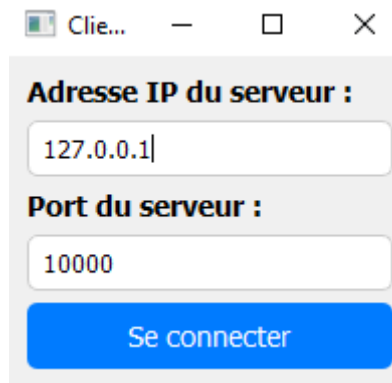
```
cd [...]\Downloads\programmationS3-main\SAE3.02\Graphique\PyQt
```

3. Lancer le client : (veillez à avoir python d'installer)

```
pip install PyQt5
```

```
python client.py
```

4. Ouverture de l'interface du client
  - a. Renseigner la bonne adresse IP (Celle du Serveur maitre)

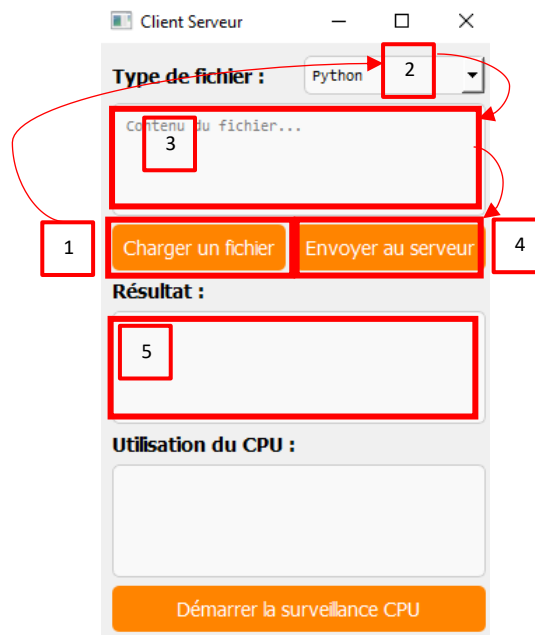


The screenshot shows a window titled 'Clie...' with two input fields. The first field is labeled 'Adresse IP du serveur :' and contains the text '127.0.0.1'. The second field is labeled 'Port du serveur :' and contains the text '10000'. Below these fields is a blue button labeled 'Se connecter'.

- b. Se connecter

## Exécuter un code

Vous pouvez désormais exécuter différents codes.



The screenshot shows a window titled 'Client Serveur' with several components. A red box labeled '1' highlights the 'Charger un fichier' button. A red box labeled '2' highlights the 'Type de fichier' dropdown menu, which is set to 'Python'. A red box labeled '3' highlights the 'Contenu du fichier...' text area. A red box labeled '4' highlights the 'Envoyer au serveur' button. A red box labeled '5' highlights the 'Résultat :' text area. At the bottom, there is a large orange button labeled 'Démarrer la surveillance CPU'.

1. Charger un fichier
  - a. Des fichiers Test sont disponibles dans le répertoire  
    \programmationS3-main\SAE3.02\ScriptTest
2. Choisir le type de fichier en Python, Java et C
3. Modifier si besoin le script
4. Envoyer au serveur (Indisponible pendant l'exécution de la tâche)
5. Visualisation du résultat

Il est possible de lancer plusieurs clients en parallèle, il suffit d'ouvrir un autre CLI sur votre client et de resuivre l'ensemble des étapes de la partie « Running coté client », si c'est sur un hôte différent il faut reprendre les étapes de la partie « Téléchargement pour le Client »