

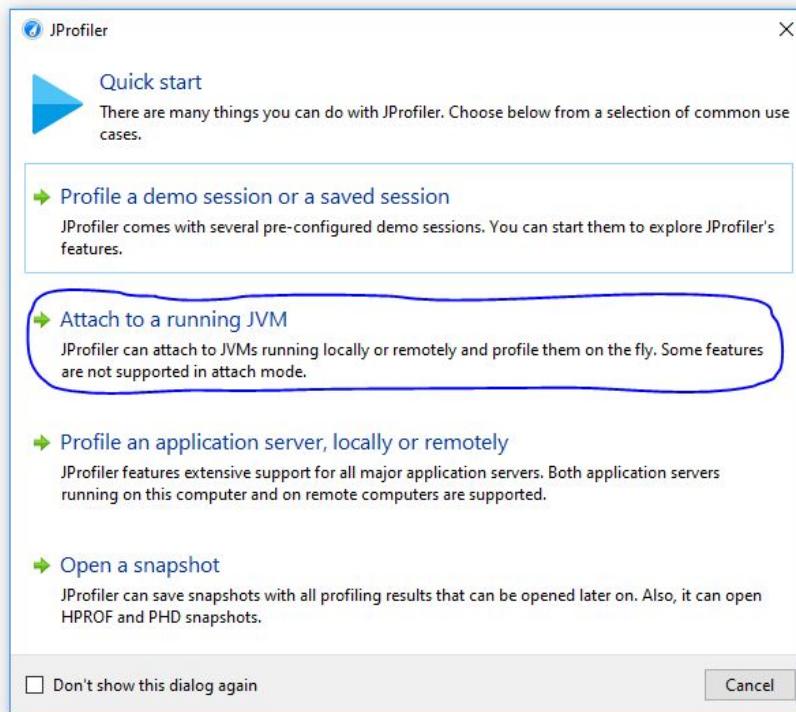
# Guide d'utilisation pour JProfiler.

## Installation et configuration

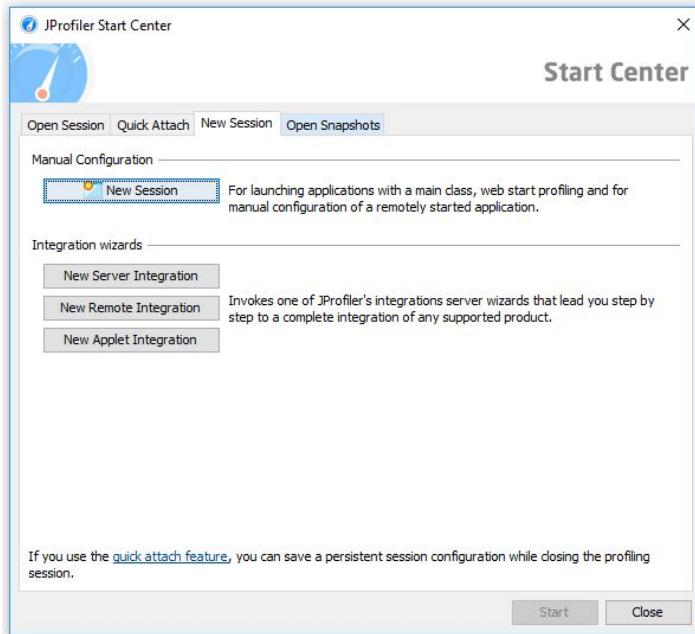
1. Si sur windows, vous ne pouvez pas installer docker, installer Docker Toolbox à partir du lien suivant: [https://docs.docker.com/toolbox/toolbox\\_install\\_windows/](https://docs.docker.com/toolbox/toolbox_install_windows/)
2. Installer jprofiler à partir du lien suivant (assurez-vous que la version dans le fichier *JProfilerDockerfile* corresponde à la version téléchargée sur le site web):  
<https://www.ej-technologies.com/download/jprofiler/files>
3. Installer Maven à partir du lien suivant si sur windows:  
<https://docs.wso2.com/display/IS323/Installing+Apache+Maven+on+Windows>
4. Ouvrir une et lancer la commande suivante dans la racine du répertoire du projet:  
`mvn clean package`
5. Exécuter Docker Quickstart Terminal (si vous avez installez Docker Toolbox) ou bien une console dans la racine du projet et lancer les commandes suivantes pour exécuter une instance de MongoDB et construire une image du repo:
  - `docker build --file JProfilerDockerfile -t jprofilerweka:tag .`
  - `docker pull mongo`
  - `docker-compose -f docker-compose-jprofiler.yml up`

N.B. Si vous avez installé Docker Toolbox, le terminal vous donnera un IP qu'il faudra utiliser pour accéder aux conteneurs dans le future. Assurez-vous de bien le garder en note. Sinon, localhost devra être l'adresse principale.

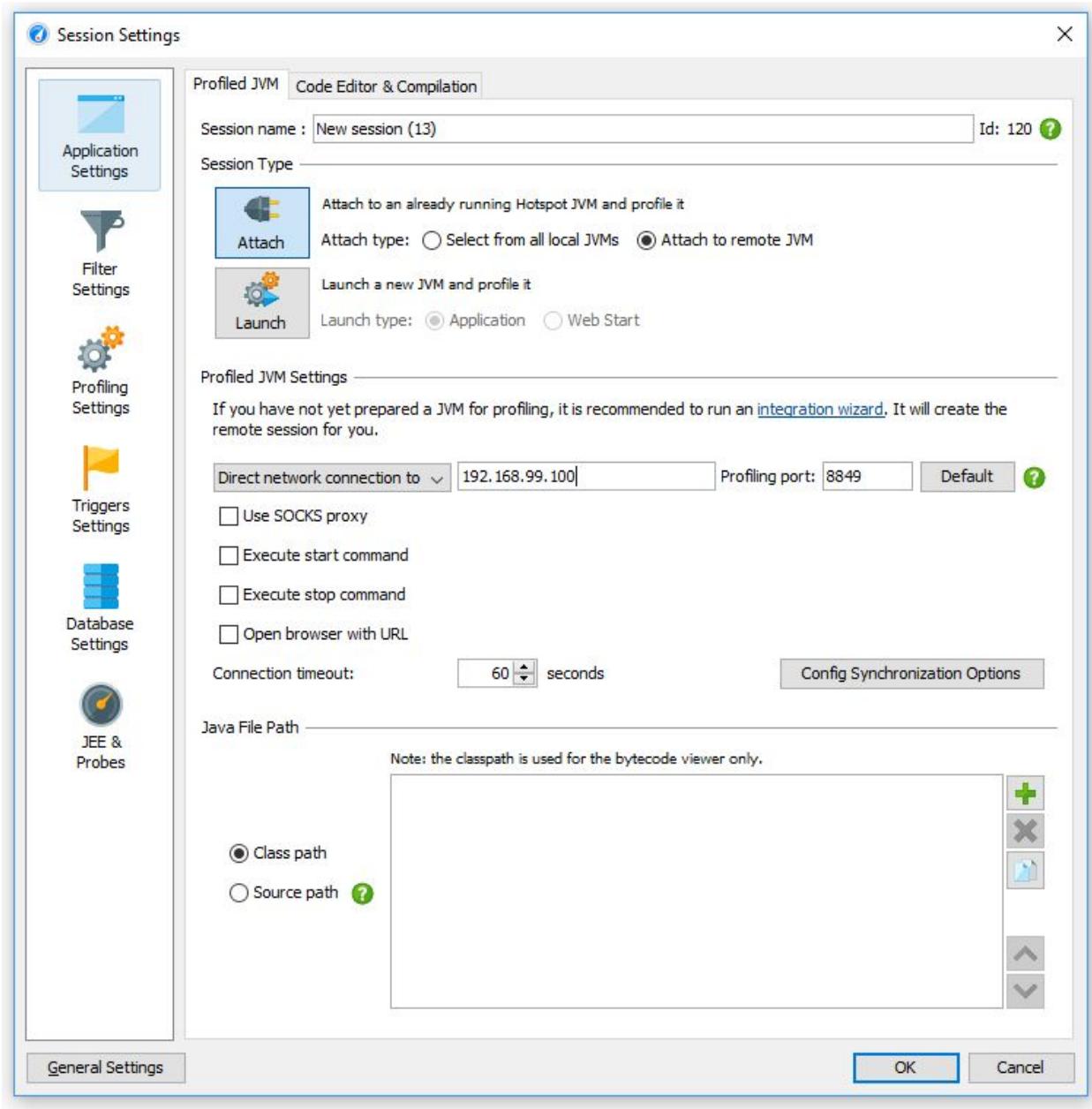
6. Exécuter JProfiler sur l'hôte et suivez les étapes suivantes:
  1. Choisir option “Attach to a running JVM”



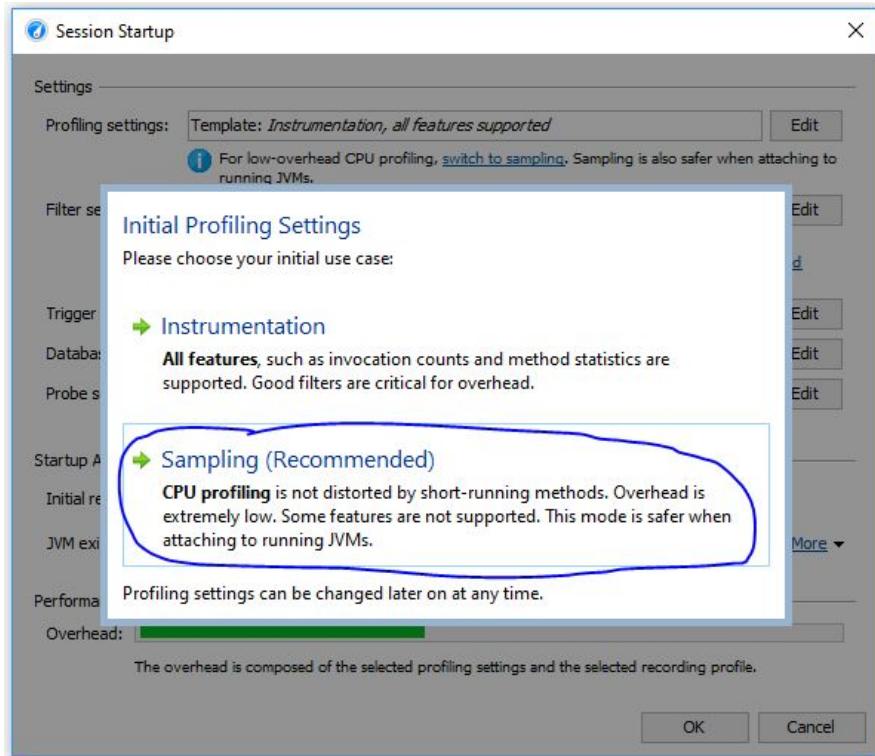
2. Sélectionner le tab *New Session* et ensuite sélectionner l'option *New Session*:



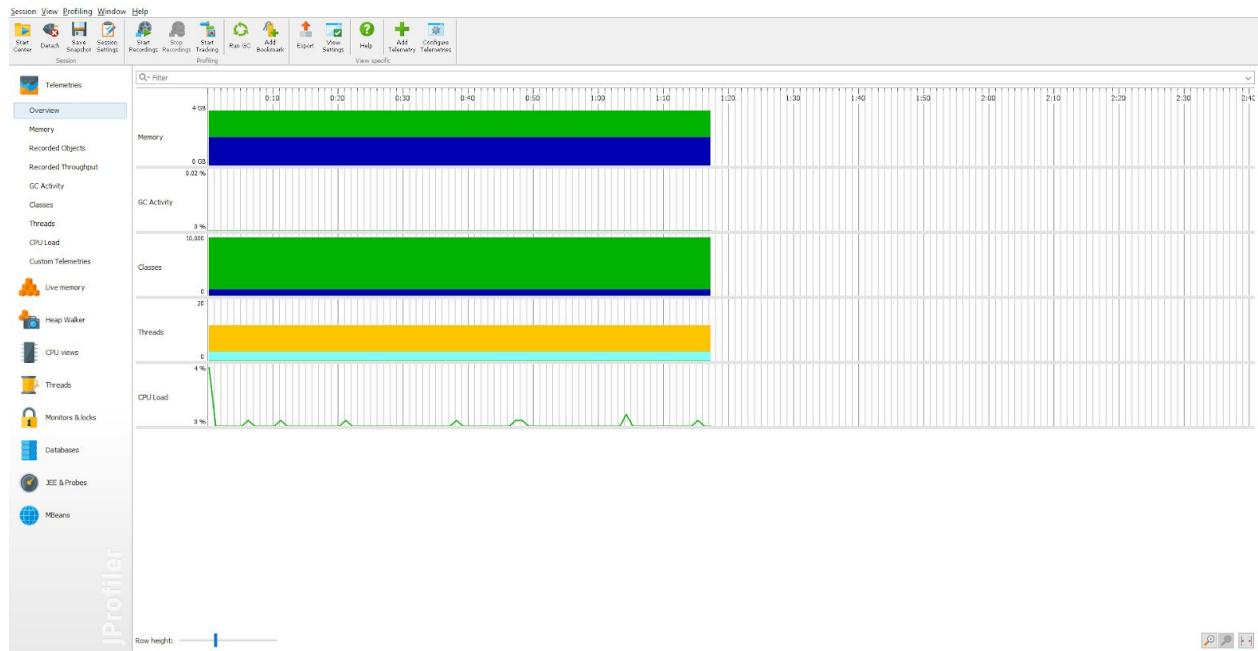
3. Sélectionner l'option *Attach to remote JVM* dans la section *Session Type* et l'option *Direct network connection to* et inscrire l'adresse pour accéder au conteneur. Le port doit être 8849. Ensuite appuyer sur le bouton *OK*.



4. Sélectionner l'option *Sampling (Recommended)* et ensuite appuyer sur le bouton **OK**:



#### 4. Vous aurez maintenant accès à l'interface suivante:



# Scénarios

Nom	Grosseur de la requête	Nombre de requêtes
Réduite	376Ko	1
Moyenne	3.48Mo	5
Augmentée	3.48Mo	10
Augmentée exceptionnelle	52.9Mo	1

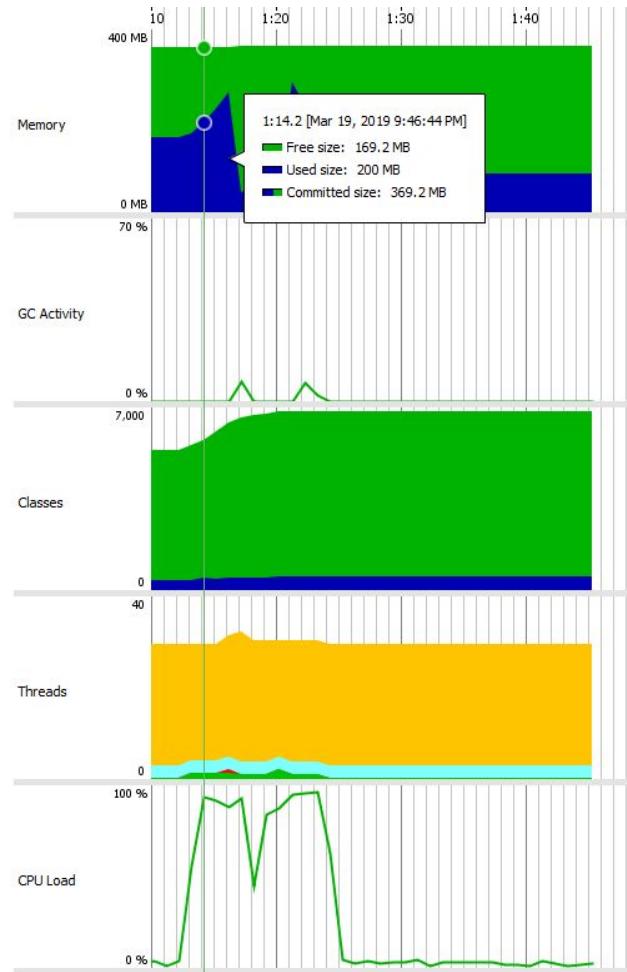
## Exécution des scénarios

Nous pouvons lancer autant de requêtes que nous voulons en utilisant un *application node*. Dans le répertoire profiler/resp, ouvrir un command prompt et lancer les commandes suivantes pour exécuter un scénario.

- npm install
- node profiler.js #1 #2 url
  - #1 est un chiffre entre 1 et 4 inclusivement
  - #2 est le nombre de requêtes à envoyer
  - Url = l'url trouver à l'étape 5 + port 8081 ex: <http://192.12.312.13:8081> ou bien localhost si vous n'utilisez pas Docker Toolbox.

# Résultat de JProfiler

## Scénario 1:



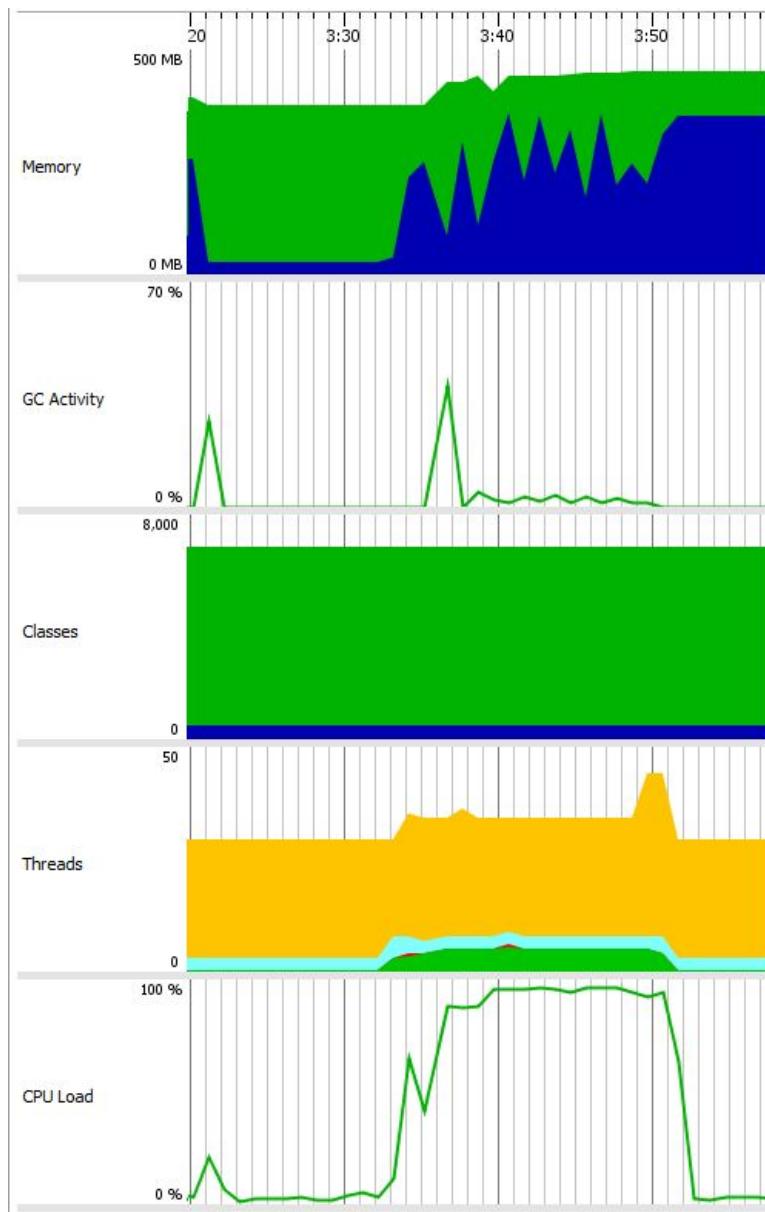
**Quantité de mémoire utilisé: 289Mo**

**CPU maximum: 98.5%**

**Nombre de thread maximum: 33**

**Temps d'exécution de la tâche: environ 13 secondes.**

## Scénario 2:



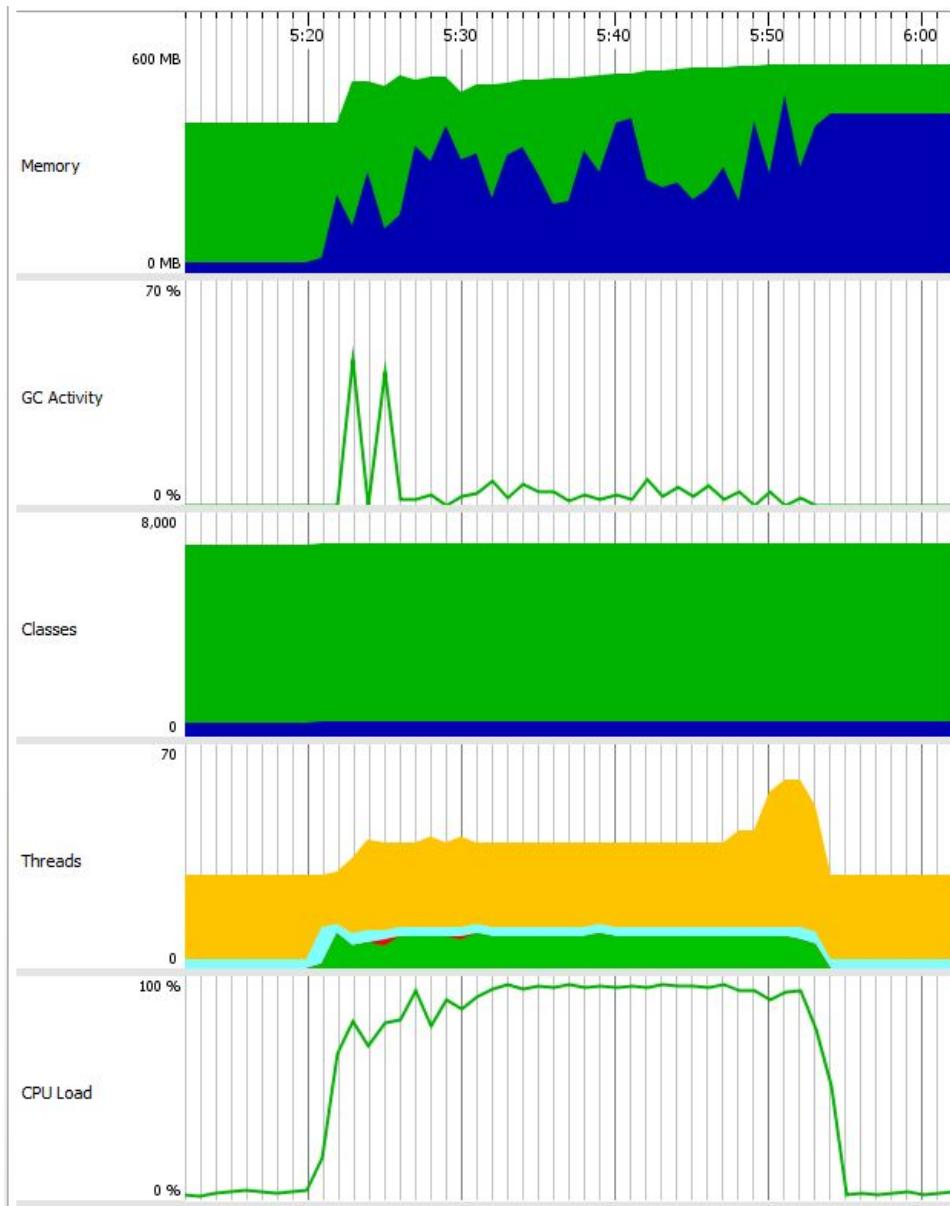
**Quantité de mémoire utilisé: 363Mo**

**CPU maximum: 98.5%**

**Nombre de thread maximum: 45**

**Temps d'exécution de la tâche: environ 20 secondes.**

### Scénario 3:



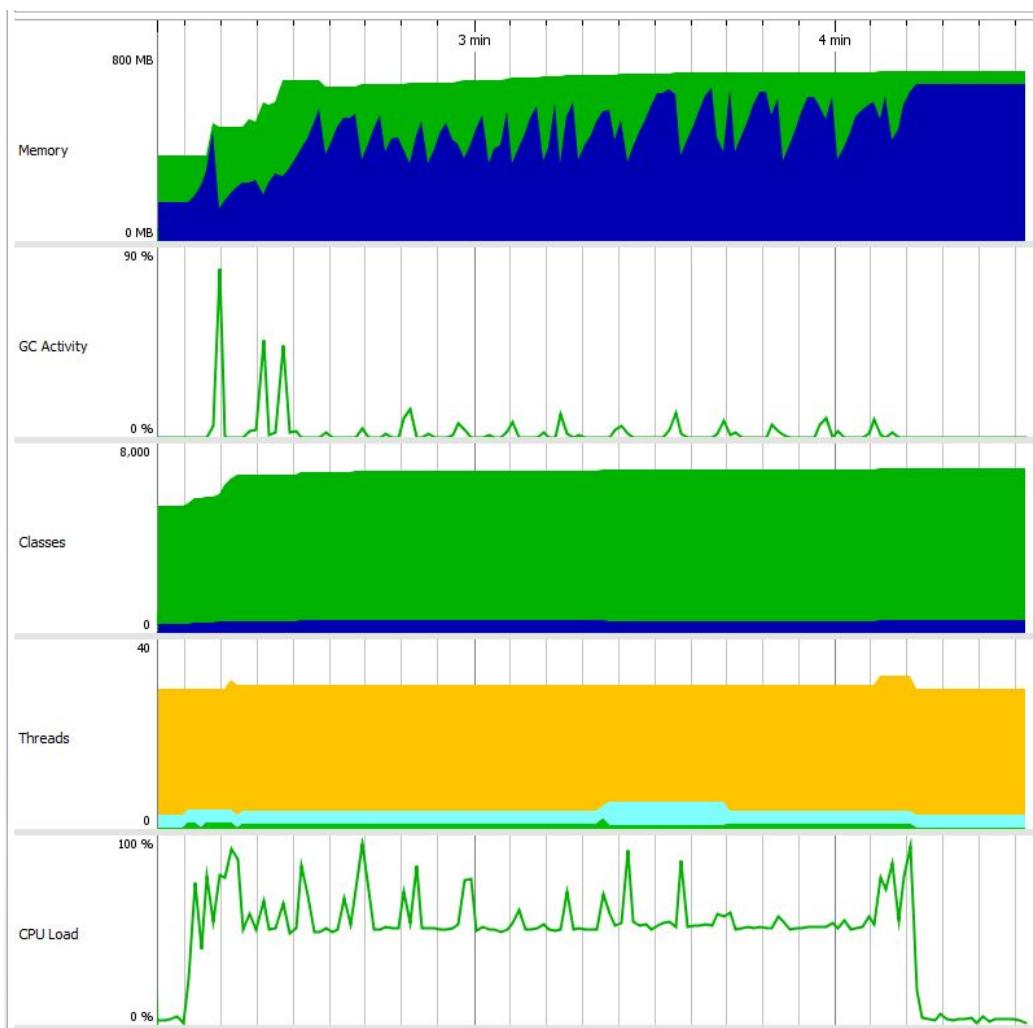
**Quantité de mémoire utilisé: 481.9Mo**

**CPU maximum: 98.3%**

**Nombre de thread maximum: 60**

**Temps d'exécution de la tâche: environ 35 secondes.**

#### Scénario 4:



**Quantité de mémoire utilisé: 669.9Mo**

**CPU maximum: 97.4%**

**Nombre de thread maximum: 33**

**Temps d'exécution de la tâche: environ 2 minutes.**

## Résultat de la question 5

En principe, nous ne rencontrons pas les objectifs de performance définis lorsque nous utilisons seulement un conteneur. Dans la vidéo de la question 6, nous pouvons remarquer que le % du CPU excède bien 200% et la mémoire utilisé maximal est plus grand que 2.0G. Mais lorsque nous utilisons au moins 3 conteneurs avec la méthode “scale”, le pourcentage de CPU se stabilisent autours de 100% avec des fluctuations de 10 à 20%. La mémoire utilisée pour chaque conteneurs maximale demeurent autours de 1.1Go. Afin de répondre à nos objectifs, il faut au moins 5 conteneurs. Ceci nous permet de maintenir le pourcentage de chaque CPU à environ 100% et utiliser moins de 1Go pour chaque conteneurs. Donc, le seuil que nous pouvons utiliser est 5 conteneurs afin de répondre à nos objectifs.

Pour visionner les vidéos des scénarios 1 à 3 avec et sans scale, veuillez-vous référer au dossier DOCUMENTS/Question\_5