



Formation JMeter

Plan

- Introduction
- Installation et configuration
- Création d'un plan de test
- Exécution d'un plan de test
- Analyse des résultats de tests
- Concepts avancés

Introduction

Introduction



- Initié par Stefano Mazzocchi en 1998
- Développé par l'Apache Software Foundation et distribué sous licence Apache v2
- 100 % pure Java
- Nombreux types de test supportés : HTTP/HTTPS, FTP, JDBC, LDAP, ...
- Très extensible (plugins, BeanShell, ...)

Installation et configuration

Installation

■ Pré-requis :

▶ JDK 1.5

1. Télécharger l'archive depuis

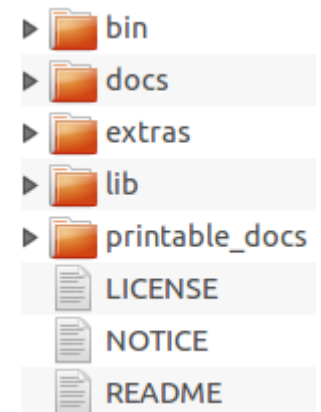
http://jmeter.apache.org/download_jmeter.cgi

2. Extraire l'archive

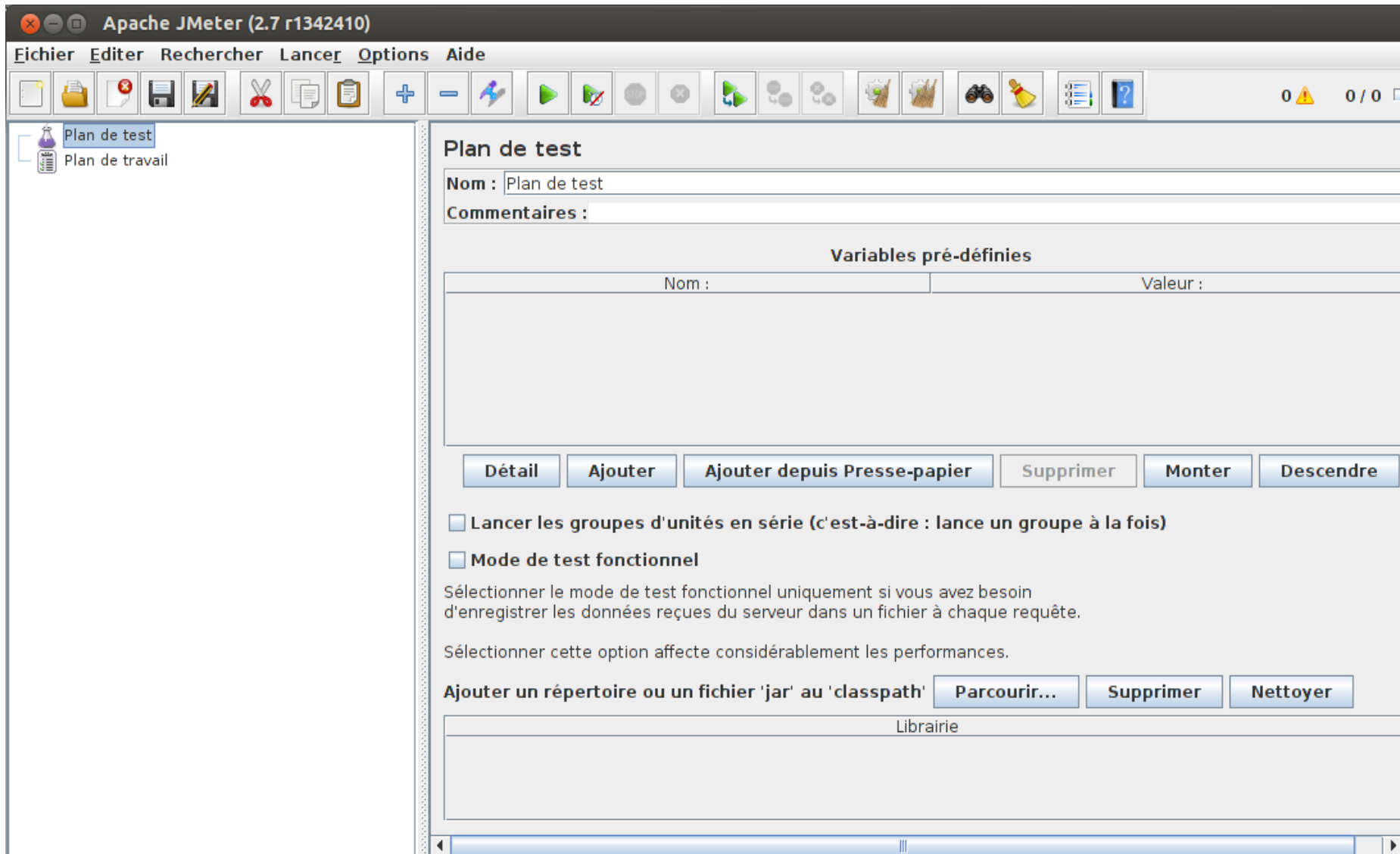
3. Exécuter JMeter

> JVM_ARGS="-Xms1024m -Xmx1024M"

> bin/jmeter



Installation et configuration



Configuration

■ bin/jmeter.properties

Propriété	Description	Obligatoire
remote_hosts	Liste des serveurs distants	Non
not_in_menu	Liste des composants qui ne doivent pas être	Non
jmeter.save.saveservice.o	Format dans lequel les résultats des tests	Non
user.properties	Nom du fichier contenant les propriétés	Non

Exercices

- Installer et configurer JMeter

Création d'un plan de test

Création d'un plan de test

- Un plan de test se compose d'un ou plusieurs éléments (groupe d'unités, contrôleurs, récepteurs, etc.).
- Un plan de test simple contient normalement les éléments suivants :
 - ▶ un groupe d'unités,
 - ▶ un échantillon,
 - ▶ un récepteur.

Groupe d'unités (ThreadGroup)

- Un groupe d'unités représente un ensemble d'utilisateurs qui exécutent un scénario de test particulier.
- $100 \text{ utilisateurs} / 10 \text{ secondes} = 10 \text{ utilisateurs par secondes}$

Groupe d'unités (ThreadGroup)

- Les propriétés d'un groupe d'unités permettent de définir les nombres d'utilisateurs, la durée de la montée en charge et le nombre d'itérations.

Groupe d'unités	
Nom :	Groupe d'unités
Commentaires :	
Action à suivre après une erreur d'échantillon	
<input checked="" type="radio"/> Continuer <input type="radio"/> Démarrer itération suivante <input type="radio"/> Arrêter l'unité <input type="radio"/> Arrêter le test <input type="radio"/> Arrêter le test imm	
Propriétés du groupe d'unités	
Nombre d'unités (utilisateurs) :	1
Durée de montée en charge (en secondes) :	1
Nombre d'itérations	<input type="checkbox"/> Infini 1
<input type="checkbox"/> Programmeur de démarrage	

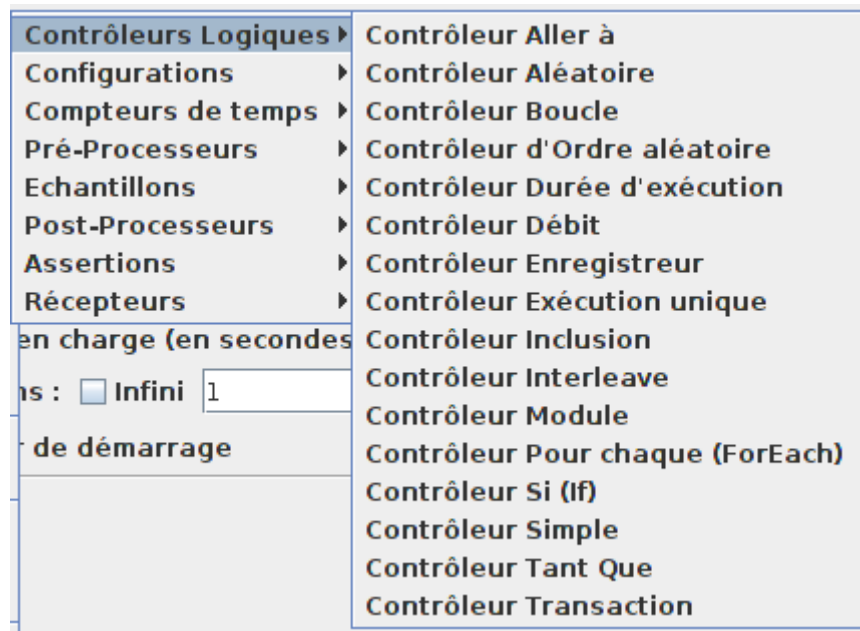
Échantillons (Samplers)

- Un échantillon permet d'envoyer une requête et d'attendre la réponse.

Echantillons	▶ Action test
Post-Processeurs	▶ Appel de processus système
Assertions	▶ Echantillon BeanShell
Récepteurs	▶ Echantillon BSF
Temps (en secondes) : 1	Echantillon Débogage
Temps d'arrêt : 1	Echantillon Journal d'accès
Arrêt	Echantillon JSR223
	Echantillon Lecteur d'email
	Requête AJP/1.3
	Requête FTP
	Requête HTTP
	Requête Java
	Requête JDBC
	Requête JMS Abonnement
	Requête JMS Point-à-point
	Requête JMS Publication
	Requête JUnit
	Requête LDAP
	Requête LDAP étendue
	Requête SMTP
	Requête SOAP/XML-RPC
	Requête TCP
	Requête WebService (SOAP)

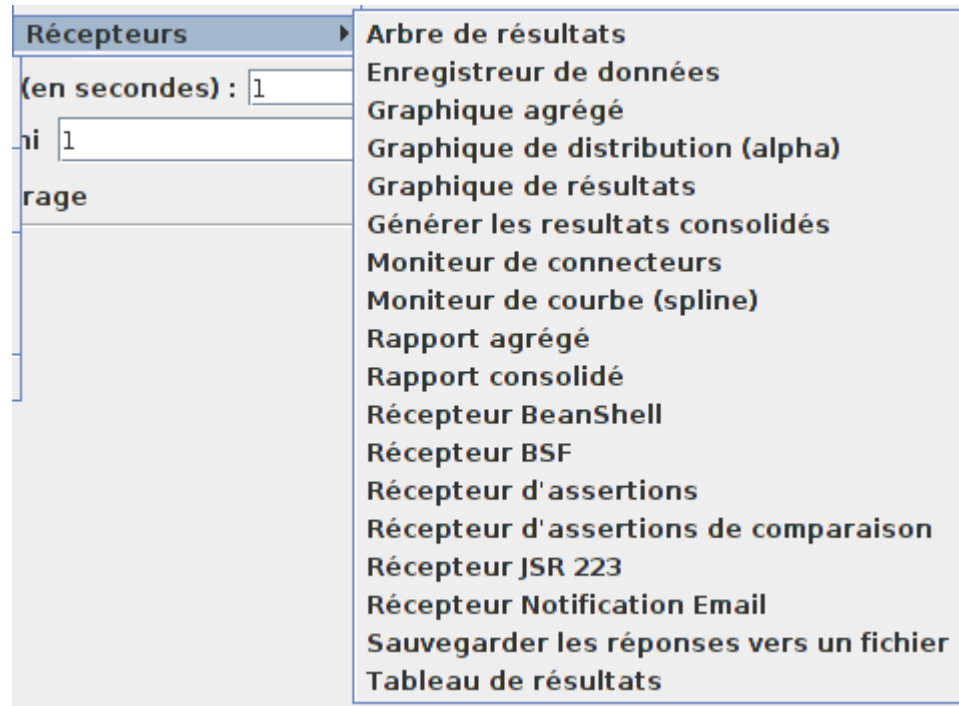
Contrôleurs logiques

- Un contrôleur logique permet de personnaliser la logique utilisée par JMeter pour décider quand envoyer une requête.



Récepteurs (Listeners)

- Un récepteur permet d'accéder aux informations collectées par JMeter lors de l'exécution des tirs de charge.

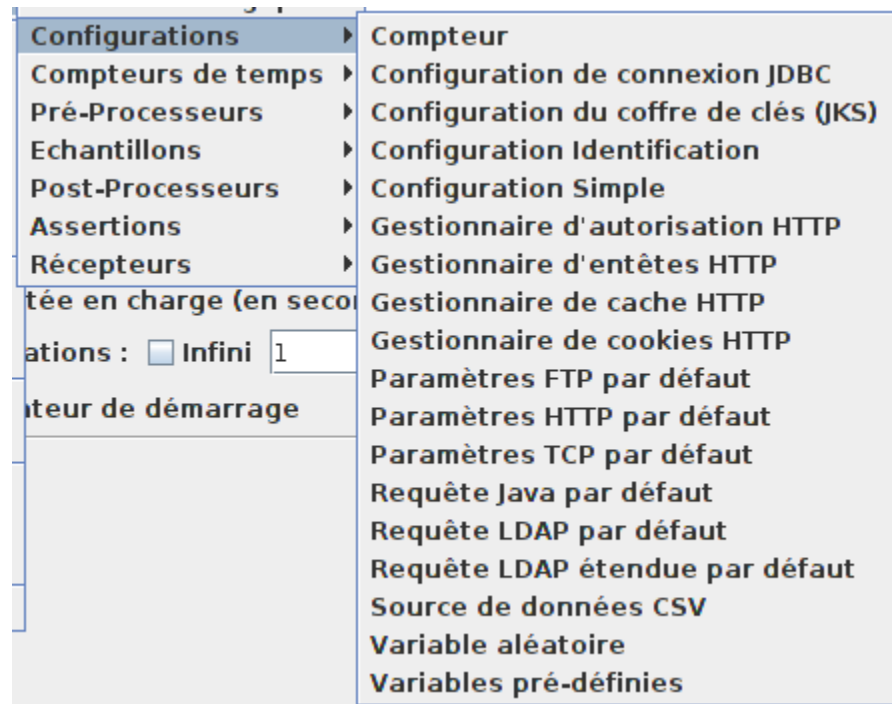


Exercices

- Créer un plan de test simple avec les éléments suivants :
 - ▶ Groupe d'unités
 - ▶ Requête HTTP (Échantillons)
 - ▶ Paramètres HTTP par défaut (Élément de configuration)
 - ▶ Graphique agrégé (Récepteur)

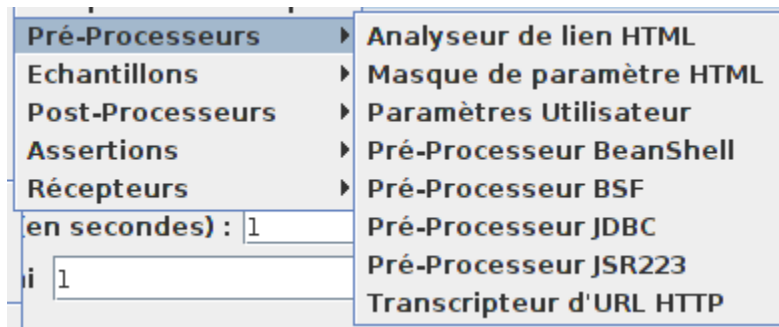
Éléments de configuration

- Un élément de configuration permet de définir des variables.

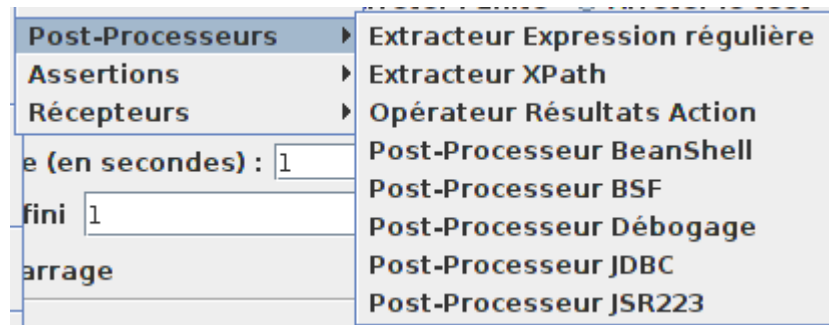


Pré- et Post-Processor

■ Pré-Processor

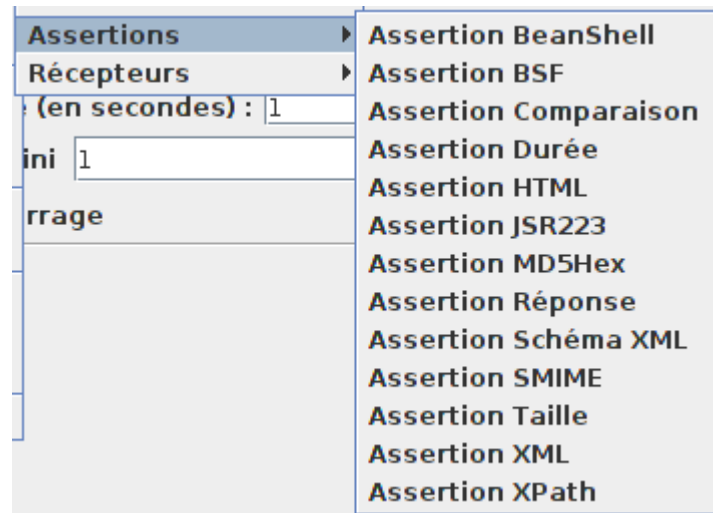


■ Post-Processor



Assertions

- Une assertion permet de tester les réponses renvoyées par le serveur.



Propriétés

- Elles sont définies :
 - soit dans le fichier `jmeter.properties` (ou préférablement `user.properties`),
 - soit sur la ligne de commande avec les options « `-J` » (local) ou « `-G` » (distant).
- Elles sont accédées au moyen de la fonction `__property` ou `__P`.
- Les propriétés sont globales.

Variables

- Les variables sont locales au groupe d'unités
- Variables pré-définies

Variables pré-définies

Nom :

Commentaires :

Variables pré-définies

Nom :	Valeur :	Description

Source de données CSV

- L'élément de configuration « Source de données CSV » permet de lire des lignes depuis un fichier CSV et de les séparer dans des variables.

Source de données CSV

Nom :

Commentaires :

Configuration de la source de données CSV

Nom de fichier :	<input type="text"/>
Encodage du fichier :	<input type="text"/>
Noms des variables (séparés par des virgules) :	<input type="text"/>
Délimiteur (utiliser 't' pour la tabulation) :	<input type="text" value=","/>
Autoriser les données avec des quotes ? :	<input type="text" value="False"/> ▼
Recycler en fin de fichier (EOF) ? :	<input type="text" value="True"/> ▼
Arrêter l'unité à la fin de fichier (EOF) ? :	<input type="text" value="False"/> ▼
Mode de partage :	<input type="text" value="Toutes les unités"/> ▼

Expressions régulières

- Le Post-Processor « Extracteur d'expression régulière » permet d'extraire des valeurs d'une réponse en utilisant une expression régulière.
- JMeter utilise Apache Jakarta ORO qui fournit des expressions régulières compatibles avec Perl5

Expressions régulières

Extracteur Expression régulière

Nom :

Commentaires :

Appliquer sur

☒ L'échantillon ☐ Les ressources liées ☐ L'échantillon et ses ressources liées ☐ Une variable :

Portée

☒ Corps ☐ Corps (non échappé) ☐ Entêtes ☐ URL ☐ Code de réponse ☐ Message de réponse

Nom de référence :

Expression régulière :

Canevas :

Récupérer la Nième corresp. (0 : Aléatoire) :

Valeur par défaut :

Exercices

- Variabiliser un plan de test

Exécution d'un plan de test

Exécution en mode non-GUI

■ `jmeter -n -t sample.jmx -l results.csv`

```
sducret@207NotK: ~/apache-jmeter-2.7/bin
sducret@207NotK:~/apache-jmeter-2.7/bin$ ./jmeter.sh -n -t examples/CSVSample.jm
x -l results.csv -j jmeter.log
Creating summariser <summary>
Created the tree successfully using examples/CSVSample.jmx
Starting the test @ Wed Jun 20 20:39:58 CEST 2012 (1340217598896)
Waiting for possible shutdown message on port 4445
summary =    12 in   3.0s =    4.0/s Avg:   227 Min:   103 Max:   352 Err:    0
(0.00%)
Tidying up ...    @ Wed Jun 20 20:40:02 CEST 2012 (1340217602071)
... end of run
sducret@207NotK:~/apache-jmeter-2.7/bin$
```

Summariser

■ bin/user.properties

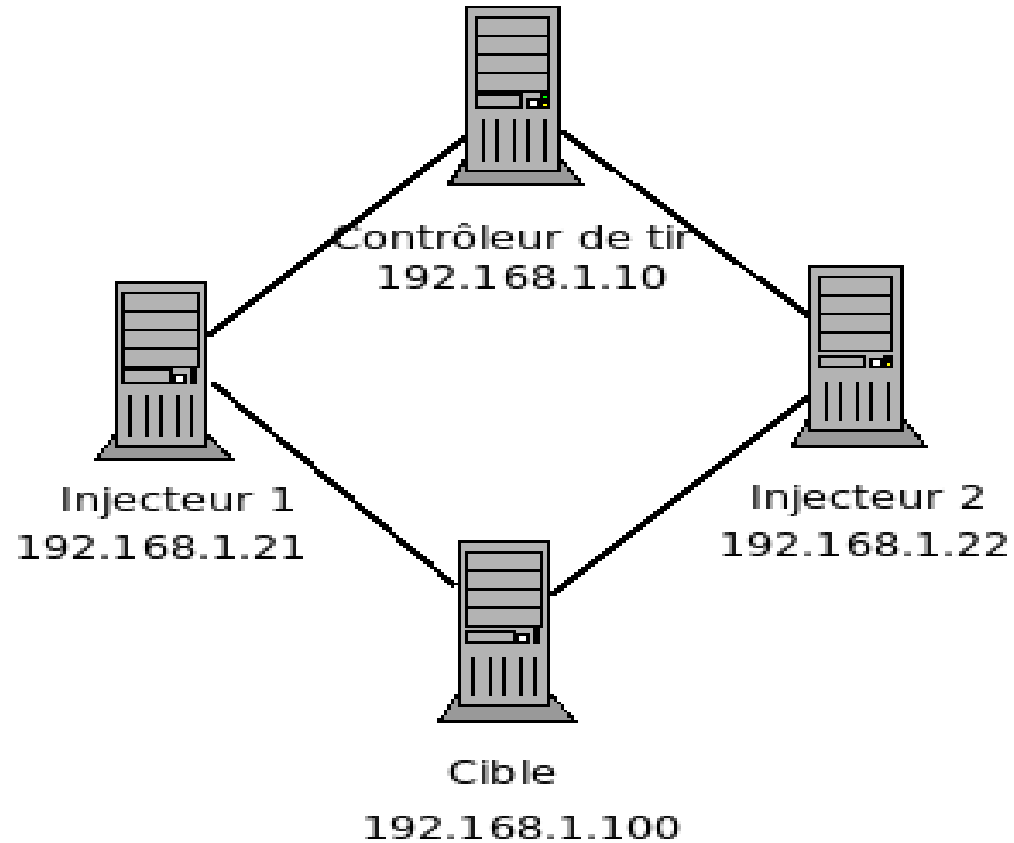
```
#-----  
# Summariser configuration (mainly applies to non-GUI mode)  
#-----  
# Summariser settings  
#  
# Define the following property to automatically start a summariser with that name  
# (applies to non-GUI mode only)  
summariser.name=summary  
#  
# interval between summaries (in seconds) default 3 minutes (180)  
summariser.interval=30  
#  
# Write messages to log file (c'est à dire jmeter.log)  
summariser.log=false  
#  
# Write messages to System.out  
summariser.out=true
```

Fichier de résultats

■ bin/user.properties

jmeter.save.saveservice.output_format=csv

Tirs distribués



Tirs distribués

- 1) Démarrer les injecteurs : `jmeter-server`
- 2) Ajouter les injecteurs dans le fichier de configuration du contrôleur de tir :
`remote_hosts = 192.168.1.21,
192.168.1.22`
- 3) Démarrer le contrôleur de tir : `jmeter -n -r
-X -t sample.jmx -l results.csv`

Analyse des résultats

Concepts avancés