

Découvrez l'importance du test de charge dans la gestion de vos applications web



Date: Jeudi 9 octobre 2014
Début de présentation: 18h30
Lieu: MDN - Cap Omega - Montpellier

Plus d'information sur www.syloe.fr

./ Expert en système d'information



Plan de notre intervention

1. Pourquoi le **test de charge** ?
2. Les différents types de tests de charge
 - 1.1. Le test de performance
 - 1.2. Le test aux limites
3. Méthode
 - 2.1. Pré-requis
 - 2.2. Préparation des scénarios
 - 2.3. Lancement du test de charge
 - 2.4. Interprétation des résultats

Plan de notre intervention

4. Réalisation d'un test en direct avec Jmeter
 - 4.1. Généralités
 - 4.2. L'interface Jmeter
 - 4.3. Les éléments
 - 4.4. Le proxy HTTP
 - 4.5. Les sources de données
 - 4.6. Visualisation des résultats
 - 4.7. Exemple de scénario complet
5. Présentation de [Syloé](#)
6. Suite des présentations [Syloé](#) des logiciels libres

1. Pourquoi le test de charge

Introduction (1/2)

- Une application Web est un site hébergé sur des serveurs

Introduction (1/2)

- Une application Web est un site hébergé sur des serveurs
- Même si celui-ci fonctionne bien, des ralentissements peuvent apparaître :
 - Passage en haute saison dans des cas où la fréquentation varie au cours de l'année
 - Augmentation de l'audience suite à un spot publicitaire ou un envoi de mailing

Introduction (2/2)

- Un test de charge consiste à solliciter le site Internet en lui appliquant un certain nombre de requêtes automatisées



Introduction (2/2)

- **Un test de charge** consiste à solliciter le site Internet en lui appliquant un certain nombre de requêtes automatisées
- Il permet d'observer son comportement au-delà de l'audience normale

Les tests classiques

- **Tests unitaires**
Tester de manière intrinsèque le programme

Les tests classiques

- Tests **unitaires**
Tester de manière intrinsèque le programme
- Tests de **non-régression**
Eviter l'introduction de bugs lors de la correction d'autres bugs ou l'ajout de nouvelles fonctionnalités

Les tests classiques

- Tests **unitaires**
Tester de manière intrinsèque le programme
- Tests de **non-régression**
Eviter l'introduction de bugs lors de la correction d'autres bugs ou l'ajout de nouvelles fonctionnalités
- Tests **fonctionnels**
Valider que les fonctionnalités attendues marchent

Limites

- Combien d'utilisateurs simultanés l'application peut tolérer ?

Limites

- Combien d'utilisateurs simultanés l'application peut tolérer ?
- Comment vont évoluer les temps de réponse avec l'augmentation de l'audience ?

Limites

- Combien d'utilisateurs simultanés l'application peut tolérer ?
- Comment vont évoluer les temps de réponse avec l'augmentation de l'audience ?
- Quel est le composant de l'application qui sera le facteur limitant à la montée en charge ?

Limites

- Combien d'utilisateurs simultanés l'application peut tolérer ?
- Comment vont évoluer les temps de réponse avec l'augmentation de l'audience ?
- Quel est le composant de l'application qui sera le facteur limitant à la montée en charge ?
- Comment vont réagir les composants à de nombreux accès concurrents ?

Préconisations

- Les tests de charge doivent être intégrés dans le processus de développement

Préconisations

- Les tests de charge doivent être intégrés dans le processus de développement
- Ils sont à prévoir avant tout déploiement en production d'une nouvelle application

Préconisations

- Les tests de charge doivent être intégrés dans le processus de développement
- Ils sont à prévoir avant tout déploiement en production d'une nouvelle application
- Chaque nouvelle version de l'application devrait également être testée

2. Les différents types de tests de charge

Le test de performance (1/2)

- On définit le nombre maximal d'utilisateurs virtuels en fonction de l'audience réelle attendue

Le test de performance (1/2)

- On définit le nombre maximal d'utilisateurs virtuels en fonction de l'audience réelle attendue
- On peut faire varier le nombre d'utilisateurs par paliers

Le test de performance (1/2)

- On définit le nombre maximal d'utilisateurs virtuels en fonction de l'audience réelle attendue
- On peut faire varier le nombre d'utilisateurs par paliers
- La durée de chaque palier peut se prolonger afin de valider le fonctionnement dans le temps

Le test de performance (2/2)

- Met en évidence des points sensibles et critiques de l'architecture technique

Le test de performance (2/2)

- Met en évidence des points sensibles et critiques de l'architecture technique
- Permet la prise de métriques de l'application en fonctionnement (temps de réponse, charge système, requêtes en base de données, etc.)

Le test aux limites (1/2)

- On définit un nombre d'utilisateurs virtuels sans cesse croissant

Le test aux limites (1/2)

- On définit un nombre d'utilisateurs virtuels sans cesse croissant
- On teste l'application avec une activité bien supérieure à l'activité normale

Le test aux limites (1/2)

- On définit un nombre d'utilisateurs virtuels sans cesse croissant
- On teste l'application avec une activité bien supérieure à l'activité normale
- On multiplie les accès concurrents aux mêmes processus métiers

Le test aux limites (2/2)

- Tester les réactions du système aux limites du modèle d'usage de l'application

Le test aux limites (2/2)

- Tester les réactions du système aux limites du modèle d'usage de l'application
- Déterminer la capacité maximale de l'ensemble dans une optique prévisionnelle

Le test aux limites (2/2)

- Tester les réactions du système aux limites du modèle d'usage de l'application
- Déterminer la capacité maximale de l'ensemble dans une optique prévisionnelle
- Mettre en évidence la limite de crash de l'application ou du système

3. Méthode de mise en place

Pré-requis

- Avoir à disposition un système de **supervision** déployé sur la plate-forme

Pré-requis

- Avoir à disposition un système de supervision déployé sur la plate-forme
- Connaître le type d'architecture et le nombre de serveurs

Pré-requis

- Avoir à disposition un système de **supervision** déployé sur la plate-forme
- Connaître le type d'architecture et le nombre de serveurs
- Connaître les mécanismes de cache en place

Pré-requis

- Avoir à disposition un système de **supervision** déployé sur la plate-forme
- Connaître le type d'architecture et le nombre de serveurs
- Connaître les mécanismes de cache en place
- Bouchonner les appels aux services externes pour tester uniquement le service désiré

Préparation des scénarios

- Définir un ou plusieurs déroulements d'actions typiques d'un utilisateur réel

Préparation des scénarios

- Définir un ou plusieurs déroulements d'actions typiques d'un utilisateur réel
- Identifier les requêtes au service que chaque action est censée faire

Préparation des scénarios

- Définir un ou plusieurs déroulements d'actions typiques d'un utilisateur réel
- Identifier les requêtes au service que chaque action est censée faire
- Définir des pauses qui correspondent aux temps de réaction de l'utilisateur réel

Préparation des scénarios

- Définir un ou plusieurs déroulements d'actions typiques d'un utilisateur réel
- Identifier les requêtes au service que chaque action est censée faire
- Définir des pauses qui correspondent aux temps de réaction de l'utilisateur réel
- Créer des comptes sur l'application pour chaque utilisateur virtuel

Lancement du test de charge

- Partir d'une base de données prédéfinie qui pourra être réinitialisée avant chaque test

Lancement du test de charge

- Partir d'une base de données prédéfinie qui pourra être réinitialisée avant chaque test
- Lancer un pré-test sur une durée courte pour valider qu'il n'y a pas de dysfonctionnement apparent

Lancement du test de charge

- Partir d'une base de données prédéfinie qui pourra être réinitialisée avant chaque test
- Lancer un pré-test sur une durée courte pour valider qu'il n'y a pas de dysfonctionnement apparent
- Lancer le test de charge

Interprétation des résultats (1/3)

- Séparer les requêtes statiques (images, css, etc.) des requêtes de contenu

Interprétation des résultats (1/3)

- Séparer les requêtes statiques (images, css, etc.) des requêtes de contenu
- Pour chaque type de requête, combiner sous forme de graphes :
 - le nombre de requêtes simultanées
 - les temps de réponse
 - les codes retours des requêtes (succès ou erreur)

Interprétation des résultats (2/3)

- Mettre en corrélation ces graphes avec les graphes systèmes récoltés par la supervision

Interprétation des résultats (2/3)

- Mettre en corrélation ces graphes avec les graphes systèmes récoltés par la supervision
- Mettre en évidence les points limites de l'application

Interprétation des résultats (2/3)

- Mettre en corrélation ces graphes avec les graphes systèmes récoltés par la supervision
- Mettre en évidence les points limites de l'application
- En expliquer les raisons à l'aide des informations remontées par la supervision

Interprétation des résultats (3/3)

- Auditer et préconiser des optimisations :
 - Applicatives (code, base de données)
 - Systèmes (configurations, tuning)
 - Architecture (scalabilité)

4. Jmeter

Généralités

- Outil fourni par la Fondation Apache sous licence Apache

Généralités

- Outil fourni par la Fondation Apache sous licence Apache
- Ecrit en Java

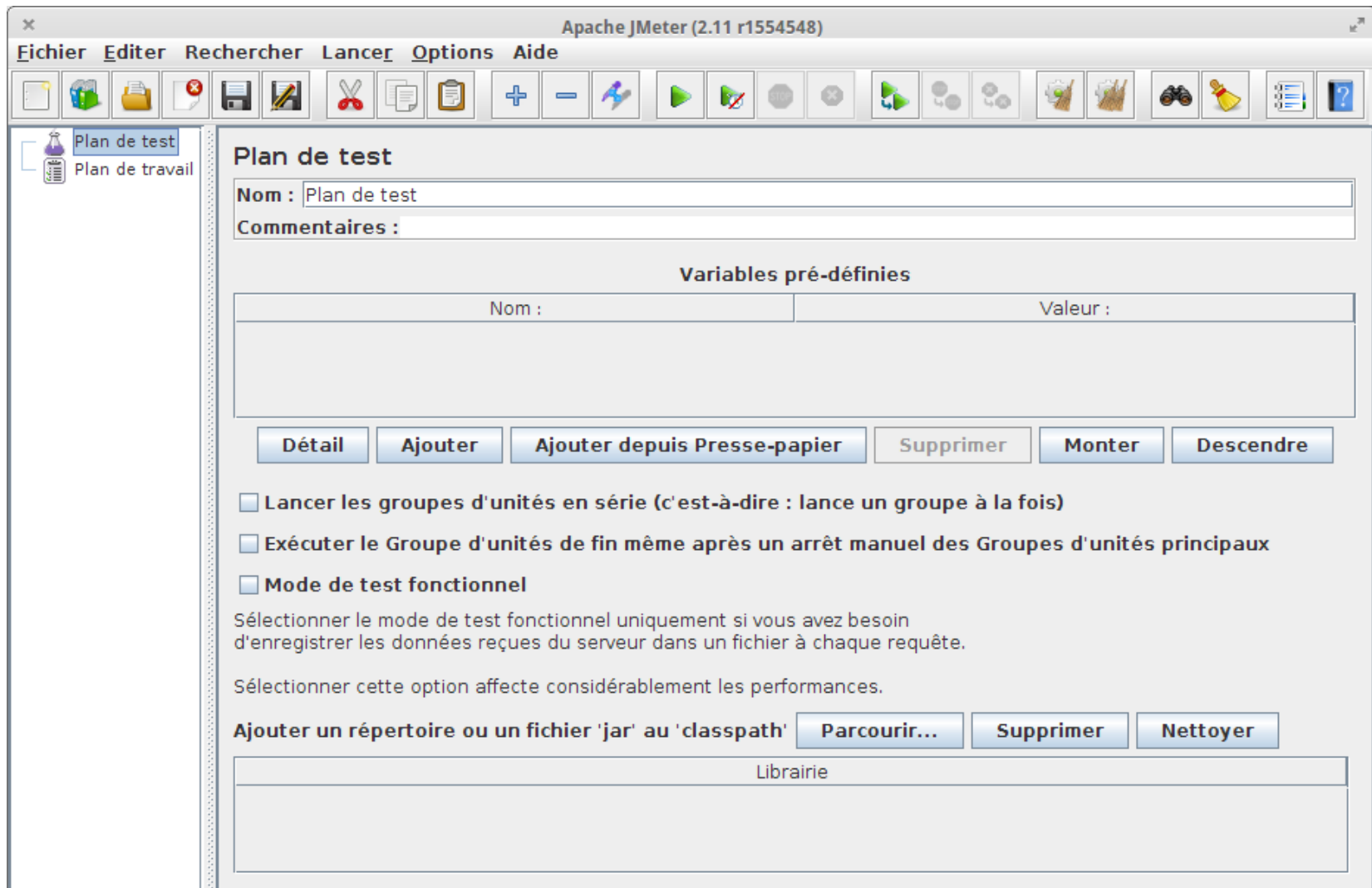
Généralités

- Outil fourni par la Fondation Apache sous licence Apache
- Ecrit en Java
- Fonctionne en mode standalone ou client/serveur

Généralités

- Outil fourni par la Fondation Apache sous licence Apache
- Ecrit en Java
- Fonctionne en mode standalone ou client/serveur
- Permet de tester de nombreux types de serveurs : web, bases de données, LDAP, mail, etc.

L'interface Jmeter (1/2)




L'interface Jmeter (2/2)

- **Le plan de test** peut être divisé en groupes d'unités et contient les éléments des scénarios

L'interface Jmeter (2/2)

- **Le plan de test** peut être divisé en groupes d'unités et contient les éléments des scénarios
- **Le plan de travail** contient tous les éléments hors scénario (configuration du proxy, tableaux de résultats, etc.)

L'interface Jmeter (2/2)

- **Le plan de test** peut être divisé en groupes d'unités et contient les éléments des scénarios
- **Le plan de travail** contient tous les éléments hors scénario (configuration du proxy, tableaux de résultats, etc.)
- Le test de charge se lance en cliquant sur l'icône 

Les éléments (1/2)

- **Les échantillons** sont les requêtes qui seront envoyées au serveur

Les éléments (1/2)

- **Les échantillons** sont les requêtes qui seront envoyées au serveur
- **Les contrôleurs** permettent de structurer les échantillons (si, boucles, aléatoire)

Les éléments (1/2)

- **Les échantillons** sont les requêtes qui seront envoyées au serveur
- **Les contrôleurs** permettent de structurer les échantillons (si, boucles, aléatoire)
- **Les compteurs de temps** sont utiles pour insérer des pauses dans le scénario

Les éléments (2/2)

- **Les pré- ou post-processeurs** peuvent générer ou extraire des données d'un échantillon

Les éléments (2/2)

- **Les pré- ou post-processeurs** peuvent générer ou extraire des données d'un échantillon
- **Les récepteurs** offrent plusieurs moyens de présenter les résultats (graphiques, tableaux, arbres, etc.)

Le proxy HTTP (1/3)

- Il permet de simplifier l'écriture des scénarios

Le proxy HTTP (1/3)

- Il permet de simplifier l'écriture des scénarios
- Proxy HTTP en écoute sur le port 8080

Le proxy HTTP (1/3)

- Il permet de simplifier l'écriture des scénarios
- Proxy HTTP en écoute sur le port 8080
- Configuration du proxy dans le navigateur

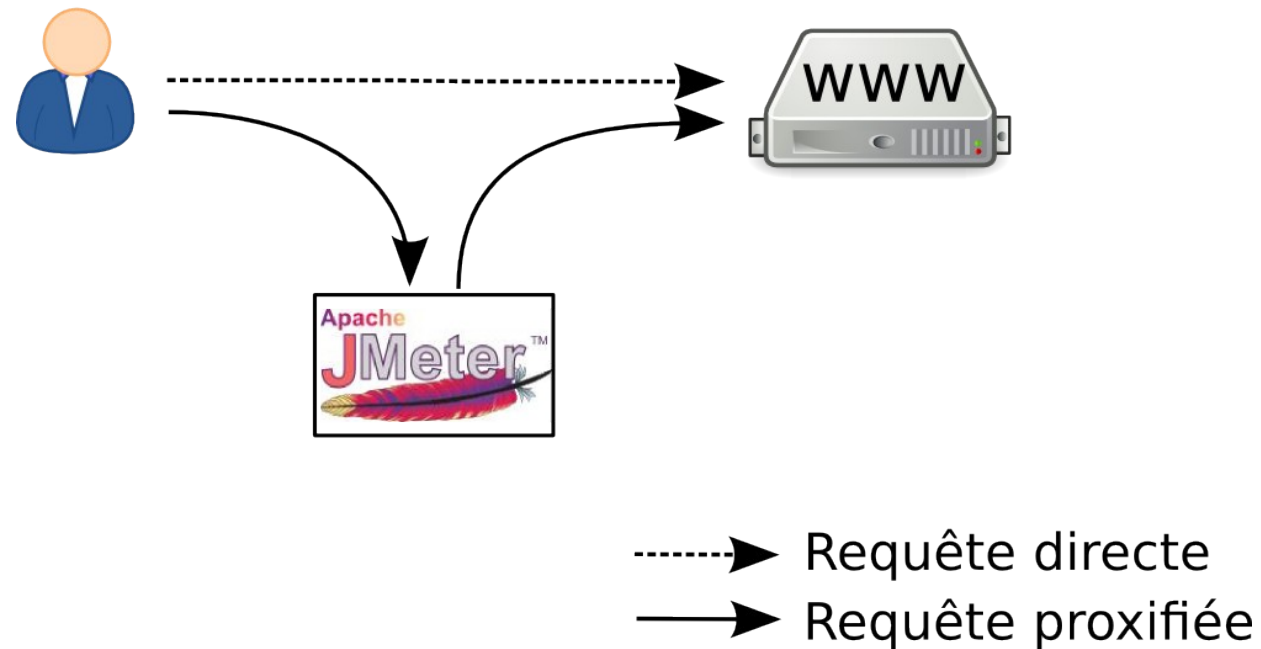
Le proxy HTTP (1/3)

- Il permet de simplifier l'écriture des scénarios
- Proxy HTTP en écoute sur le port 8080
- Configuration du proxy dans le navigateur
- Déroulement du scénario souhaité à la main

Le proxy HTTP (1/3)

- Il permet de simplifier l'écriture des scénarios
- Proxy HTTP en écoute sur le port 8080
- Configuration du proxy dans le navigateur
- Déroulement du scénario souhaité à la main
- Interception à la volée de toutes les requêtes :
 - URL
 - En-têtes HTTP
 - Paramètres POST et GET

Le proxy HTTP (2/3)



Le proxy HTTP (3/3)

Requête HTTP

Nom : /

Commentaires :

Serveur web

Nom ou adresse IP : Port :

Délai expiration (ms)

Connexion : Réponse :

Requête HTTP

Implémentation : Protocole [http] : Méthode : Encodage contenu :

Chemin : /

Rediriger automat. Suivre les redirect. Connexion persist. Multipart/form-data Entêtes compat. navigateur

Paramètres **Corps de la requête**

Envoyer les paramètres avec la r

Nom :	Val

- Plan de test
- Plan de travail
- Enregistreur script de test HTTP(S)
 - /css/style.css
 - /widgets.js
 - /images/asf-logo.png
 - /images/logo.jpg
 - /images/twitter.png
 - /widgets/follow_button.1409790579.html
 - /widgets/tweet_button.1409790579.html

Les sources de données

- Pré-charger des variables à partir d'un fichier CSV ou d'une base de données

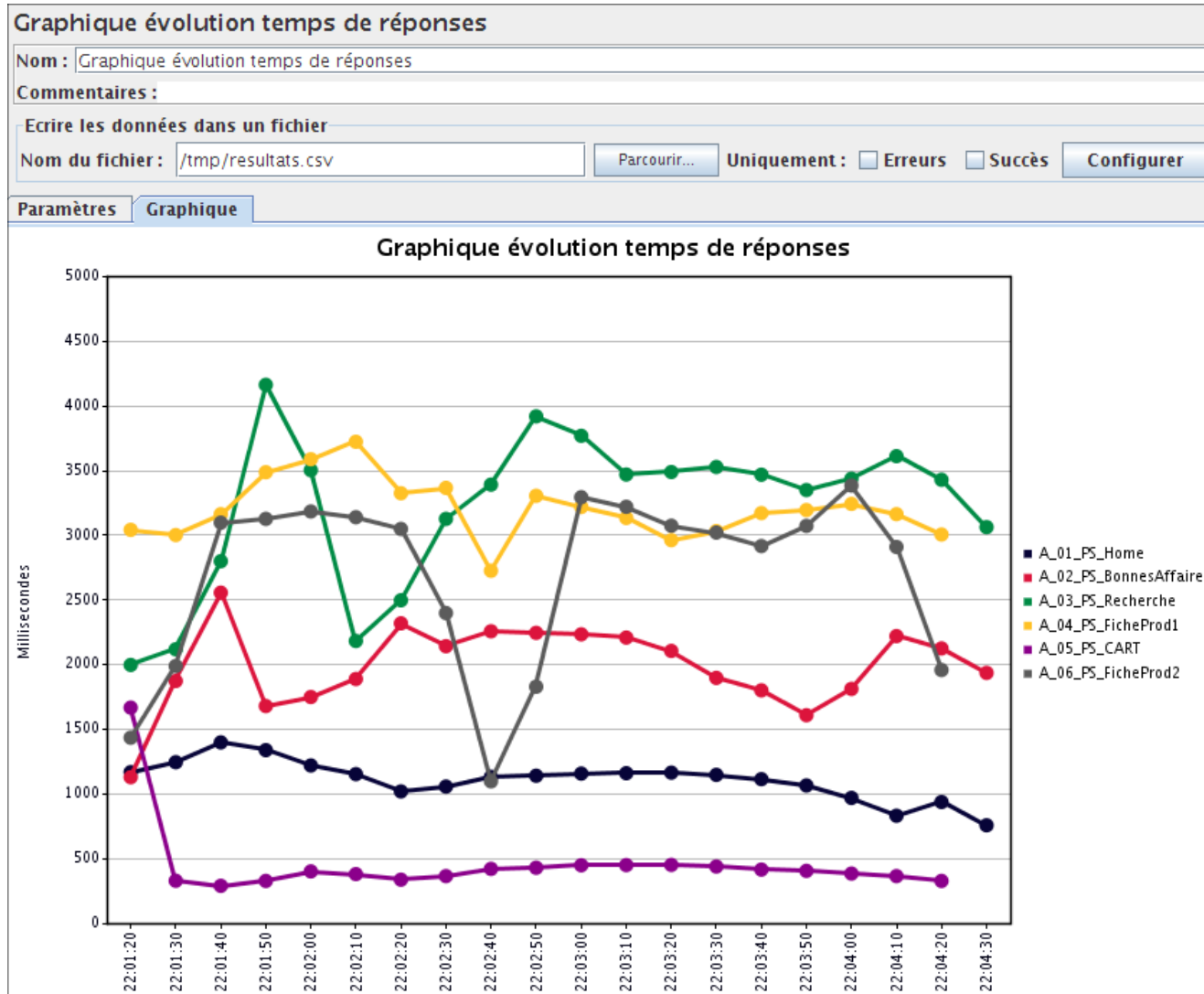
Les sources de données

- Pré-charger des variables à partir d'un fichier CSV ou d'une base de données
- Effectuer de manière itérative une requête pour chaque élément (ex : créer 10 comptes prédéfinis au sein de l'application)

Les sources de données

- Pré-charger des variables à partir d'un fichier CSV ou d'une base de données
- Effectuer de manière itérative une requête pour chaque élément (ex : créer 10 comptes prédéfinis au sein de l'application)
- Tirer un élément aléatoirement et l'utiliser dans un scénario (ex : choisir un compte et lui faire exécuter des actions)

Visualisation des résultats



5. Présentation de Syloé

Syloé : Expertise Linux

- Services :
 - Audit des SI de production
 - Audit de sécurité
 - Test de charge Applicatif
 - Supervision fine des applications Web critiques
 - Intégration d'outils systèmes sur mesures
 - Formation autour des technologies Linux et réseaux
- **Support en Crédit Temps** en mode pompier

Syloé : Expertise Linux

- Hébergement et infogérance :
 - 2 Datacenters
 - Plateformes en Haute Disponibilité
 - Failover, Loadbalancing, Clustering
 - Cloud , SaaS, Virtualisation
- Travail collaboratif
 - Zimbra, BlueMind, Open-Xchange..

Suite des présentations Syloe des logiciels libres

Sujets Transverses:

- Présentation : Bonnes pratiques d'installation et d'infogérance d'un serveur sur internet
- Présentation : Concevoir et réussir un test de charge applicatif avec scénarios de navigation des applications Web;
- Présentation : Autour de La supervision : Quoi superviser ? Quelle metrique, jusqu'a quel détails , quel processus organisationnel ?
- Présentation des différentes composantes de la lutte Anti-Spam sous Linux;
- Présentation de GIT GitLab ...;
- Présentation des nouveautés de Jboss7 et Tomcat 7;

Suite des présentations Syloe des logiciels libres

Sujets Techniques:

- Présentation des différentes solutions de High Availability de MysqLs ;
- Présentation des alternatives à MysqL de Oracle;
- Comment sécuriser un serveur Web , coté système ;
- Présentation de deux solutions de messagerie opensource : Zimbra -Bluemind;
- Présentation des solutions de Haute-Disponibilité sous Linux;
- Présentation d'une solution à base de brique opensource d'automatisation (Industrialisation) des mises en production;
- Présentation de Cfengine comme outils de centralisation des configurations;
- Présentation : Que faire suite à un hack d'un serveur internet ?
- Présentation : comment choisir une solution de virtualisation ?

Vos Questions ...

Nous contacter : www.syloe.fr

Support Linux professionnel <http://www.services-linux.fr>

contact@syloe.info