

Un portail, un intranet, un site marchand, constituent des applications sensibles pour accéder à des services, des données. Pour s'assurer de la robustesse des applications web, il faut bien procéder à des batteries de tests dès la conception des pages et du code. Avant sa mise en production, une catégorie de tests peut se révéler cruciale : les tests de charge.

Crash test **Mettre en place des tests de charge** pour les applications web ?



Cela consiste à faire subir à son application les conditions réelles d'utilisation et d'exploitation avant sa mise en production. Ces tests permettent de prédire le comportement du système, du serveur, du réseau, de l'application en utilisation et de diagnostiquer les problèmes, dysfonctionnements, crashes éventuels de l'application mais aussi de l'infrastructure. Car tester avec quelques utilisateurs et simuler plusieurs centaines, voire milliers d'utilisateurs (pour la montée en charge de l'application et donc simuler la charge d'utilisation) ce n'est pas réellement la même chose. Les tests de montée en charge et de performance sont indispensables pour optimiser la qualité et la disponibilité des applications et des services Web. Plus ils débutent tôt dans le cycle de vie de l'application, et plus leur retour sur investissement est important.

Cette pratique, qui doit faire partie intégrante de tout cycle de développement d'un projet Web, requiert de disposer de ressources logicielles (méthodologie et outils), matérielles et humaines. Un bon outillage va permettre d'enregistrer facilement les scénarii de test, par exemple par une simple navigation sur l'application Web, de les personnaliser, et de les rejouer avec des Utilisateurs Virtuels (utilisateurs simulés), mais aussi de générer automatiquement des rapports de test détaillés, servant à l'administrateur, aux développeurs pour rectifier les problèmes, les erreurs.

Dans cet article, nous verrons comment l'équipe de Web Performance, éditeur de Web Performance Suite, met des « best practices » que vous pourrez aisément mettre en place pour vos projets web.

➤ *Exécution d'un test de charge avec Web Performance Suite 3.5*

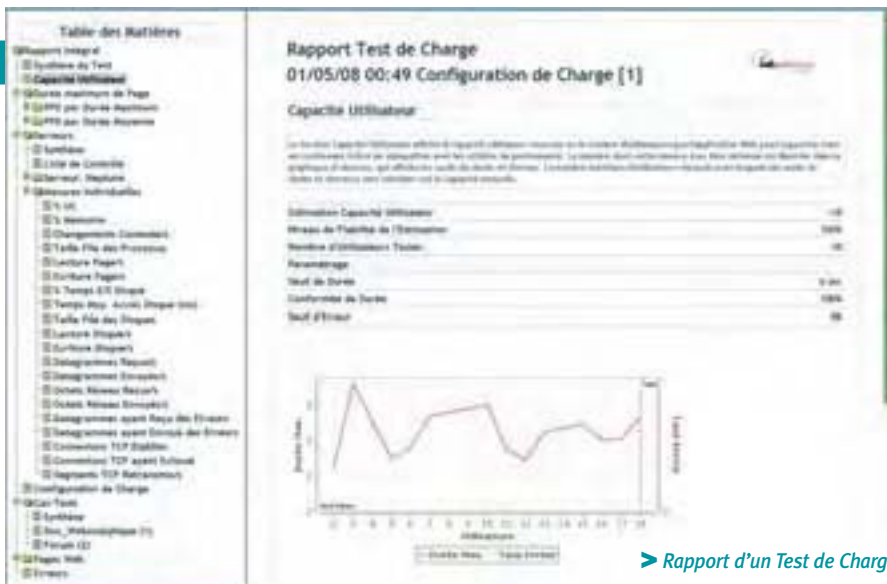
Une stratégie globale : simple et pratique

Les règles décrites ci-dessous pourront vous paraître simples, mais on oublie souvent de les appliquer, de les utiliser. Et pourtant, elles font gagner en efficacité.



- ✓ **Un résultat raisonnablement précis maintenant vaut mieux qu'un résultat très précis plus tard.** Tout le monde s'accorde sur le fait que les changements tardifs sont beaucoup plus coûteux pour le projet, donc plus les problèmes de performances sont identifiés rapidement, plus ils sont facilement corrigés, et à moindre coût.
- ✓ **Testez largement, puis en profondeur.** Le test d'une large palette de scénarii d'une manière simple est préférable au test en profondeur d'un nombre réduit de cas-tests (Lorsque nous décrivons la façon dont un utilisateur interagit avec une application Web, nous nous référons à un scénario. Une fois que le scénario a été enregistré et que la configuration commence pour la simulation, alors nous nous référons à un cas-test). Tôt dans un projet, une simulation approximative du monde réel est tout à fait acceptable. Le temps passé à obtenir des cas-tests pour imiter exactement le scénario prédit du monde réel ou à tester des dizaines de variations du même scénario sera mieux dépensé en testant une large palette de scénarii.
- ✓ **Testez dans un environnement contrôlé.** Le test sans des serveurs dédiés et une bonne gestion de la configuration donnera des résultats qui ne seront pas reproductibles. Et si vous ne pouvez pas reproduire les résultats, alors vous ne pourrez pas mesurer précisément les améliorations entre deux versions du système. ■





➤ Rapport d'un Test de Charge

1-Sélection du scénario

Lors de la sélection des scénarii à tester à partir de tous les scénarii possibles, les consultants de Web Performance évaluent chaque scénario en fonction de deux aspects: **Difficulté de Simulation** et **Importance**. Malheureusement pour les débutants, il est difficile de juger de la difficulté de simulation sans une certaine expérience des outils de test de charge et de la compréhension des détails sous-jacents d'implémentation de l'application. Mais en général, les scénarii qui sont courts et faciles à vérifier sont également la plupart du temps faciles à simuler. La *Vérification* d'un scénario consiste à confirmer que le scénario est correct et qu'il effectue bien l'action attendue dans le système lorsqu'il est simulé par l'outil de test. Un scénario qui consiste à envoyer un courriel est facile à vérifier, puisqu'il suffit de vérifier que le courriel a bien été envoyé. De même, un scénario qui effectue une opération d'ajout qui peut ensuite être interrogée est facilement vérifiable.

Les scénarii ayant été évalués selon leur niveau de difficulté de simulation et leur importance, il faut alors les trier. Les scénarii faciles à simuler et très importants arrivent en tête de liste. Ensuite il faut choisir les scénarii qui sont soit faciles à simuler mais avec une relative importance, ou qui ne sont pas faciles à simuler mais très importants. L'idéal est alors de les alterner, en sachant que si le planning est serré la priorité sera alors aux scénarii faciles. La faveur aux scénarii faciles est donnée car l'expérience montre tout simplement que les facteurs qui causent des problèmes de performances sont rarement en relation avec l'importance du scénario. En testant autant de scénarii que possible très tôt, cela donne plus de temps aux développeurs pour se concentrer sur la résolution des anomalies et la mise au point de l'application.

2-Configuration : bien la gérer, c'est être efficace

Une bonne gestion de configuration est essentielle pour obtenir des résultats de tests de charge cohérents (attention : la cohérence des tests est vitale). Si l'environnement de test change fréquemment, il sera difficile de comparer les résultats d'un test à un autre et de résoudre des problèmes. En effet, comment déterminer alors les facteurs environnementaux qui affectent les résultats des tests ?

3-Matériel de Test

Il faut veiller à ce que le matériel soit dédié au test. Le test sur des systèmes en production est une erreur courante. Dans la mesure où il est généralement impossible de dire ce que les "vrais" utilisateurs font, il devient alors très difficile de savoir si les anomalies de performances sont dues à l'application, à la configuration de tests ou bien aux autres utilisateurs sur le système.

4-Base de Données

Il est important que les données de l'application (base de données) soient dans le même état pour chaque test. Plus les bases de données sont alimentées de données, plus les requêtes deviennent lentes. En général la base de données doit être restaurée dans un état connu avant chaque test dans un souci d'efficacité ; cela permet de faire des suppositions sur la construction et la configuration de cas-tests et donc d'économiser beaucoup de temps et d'efforts.

5-Changer un élément à la fois

Il est recommandé de ne changer qu'une seule chose à la fois dans l'environnement de test. Changer plus d'un facteur à un moment donné rend alors difficile d'évaluer l'impact du changement.

A. RAKOTONIRINA, DSI - INSERM

« Avant 2005, toutes les applications nationales mises en production sans Web Performance Load Tester ont subi des interruptions de services à cause de la charge qui a entraîné de fortes dégradations de performance. Depuis 2005, la mise en place du Bureau de la Recette et surtout depuis la mise en place des tests de charge et performances systématiques avec Web Performance Load Tester sur toutes les nouvelles applications nationales, aucune interruption de service des applications nationales n'a été enregistrée à cause de problèmes de charge ou de performances.

Ces fonctionnalités ont considérablement évolué depuis 2005. La lecture des résultats de la charge ou d'un stress d'une application est devenue très explicite et les problèmes sont rapidement mis en exergue grâce aux différents tableaux de bord qui sont proposés depuis la version 3.0...

L'atout majeur de Web Performance reste son service après-vente. La solidité du support technique et la réactivité de l'équipe support donnent une grande crédibilité à l'outil auprès des chefs de projets qui, très naturellement, en cas de mauvais résultats, accusent d'abord Web Performance de tous les maux (instabilité, incompatibilité, etc.) avant de lui faire confiance et de se rendre compte des anomalies liées à l'application elle-même ! »

Une partie de la gestion des données de l'application (base de données) consiste à l'alimenter. Lorsque vous disposez du bon outil de test (comme Web Performance Suite), vous pouvez facilement alimenter la base de données de façon représentative avec une relative facilité. Dans la mesure où les données sont entrées via la même interface que celle que les utilisateurs réels utiliseraient, vous avez l'assurance que les données passent par toutes les règles de nettoyage et de vérification de l'application. Une fois que vous avez créé et configuré un cas-test pour ajouter des utilisateurs à votre application, vous pouvez par exemple alimenter la base de données avec des centaines, voire des milliers d'utilisateurs. Vous pouvez également créer différentes populations,

enregistrer chacune d'entre elles, de façon à pouvoir faire des tests de comparaison.

Créer un cas-test

Il est possible d'avoir plusieurs cas-tests pour un même scénario. Par exemple, des variations mineures peuvent être considérées comme étant un même scénario, mais elles requièrent de créer des cas-tests séparés pour les besoins du test de charge. Le processus de création et de personnalisation d'un cas-test est généralement conforme à ces étapes :

- 1 - Enregistrement du scénario en utilisant un navigateur Web pendant que le logiciel de test enregistre les actions
- 2 - Configuration du cas-test pour simuler plusieurs identités utilisateurs (avec un login et un mot de passe)
- 3 - Personnalisation du cas-test pour fournir différentes entrées (recherche de mots clefs par exemple)
- 4 - Rejeu du cas-test pour vérifier que la simulation est correcte

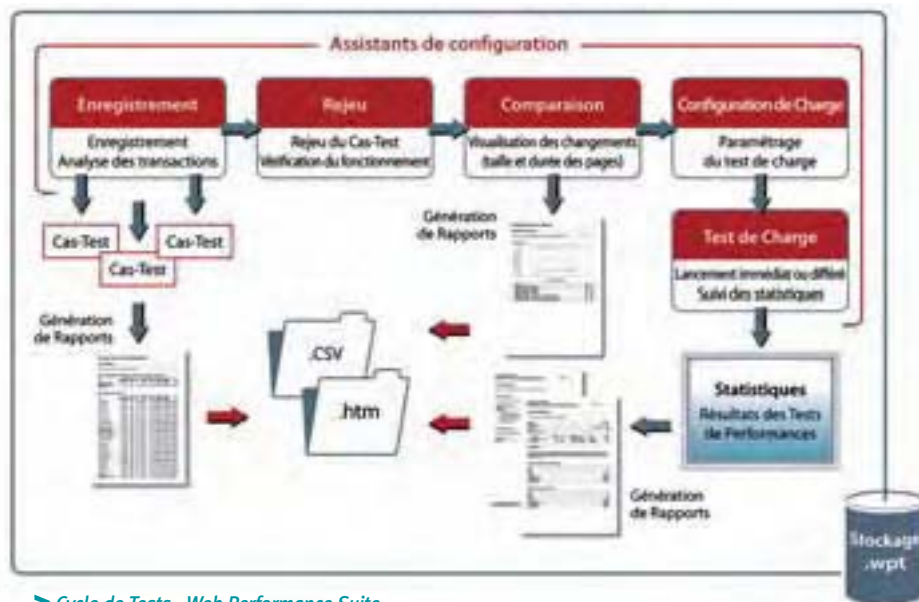
Il est à noter que lorsque l'on travaille pour la première fois avec une nouvelle application, le Rejeu doit être effectué après les étapes 1 et 2 avant d'aller plus loin.

Complexité du Cas-Test

De façon à obtenir des résultats rapidement, il est préférable de garder les cas-tests aussi simples que possible tout en reflétant l'utilisation attendue du système.

Une façon de faire est de modéliser des variations du scénario sous forme de cas-tests distincts. Par exemple si certaines variations d'un scénario nécessitent que des utilisateurs visitent une page de confirmation alors que ce n'est pas le cas pour les autres, dans ce cas il faut créer deux cas-tests distincts. La raison étant que la modélisation avec la logique hypothétique (Si, alors) dans un cas-test ajoute de la complexité et nécessite généralement un outil de test basé sur les scripts. Or, maintenir des scripts est une tâche fastidieuse et l'impact sur le coût des tests n'est pas négligeable. De plus, à moins que la mesure de deux variations ne soit critique, il est préférable d'opter pour seulement un des scénarii tôt dans la phase de test afin de pouvoir le tester plus largement, et en phases finales, les variations restantes seront alors adressées. ■

Chris MERRILL (Web Performance, Inc) et
Sandrine BOARQUEIRO-VERDUN (Kapitec
Software SAS, distributeur exclusif)



➤ Cycle de Tests - Web Performance Suite

Optimisez vos Analyses avec des fonctionnalités avancées de Reporting

L'outil de test de charge permet d'enregistrer et de personnaliser des scénarii, de simuler un ou plusieurs cas-tests au sein d'un même test de charge avec n utilisateurs, et de collecter les mesures de performance. Mais il doit aussi pouvoir fournir automatiquement une analyse détaillée.

Une solution comme Web Performance Suite génère automatiquement des rapports décisionnels en français, organisés en sections et sous-sections avec un niveau de détail sans précédent, facilitant la compréhension des problèmes de performance détectés et leur localisation. Les rapports peuvent être paramétrés, imprimés, et exportés intégralement au format html en conservant la navigation au sein de la table des matières. Les données peuvent être exportés au format CSV.

L'outil offre trois groupes de rapports : Rapports de Cas-Tests, Rapports de Référence et Rapports de Test de charge. Par exemple, le Rapport Capacité Utilisateur affiche la capacité utilisateur mesurée ou le nombre d'utilisateurs que l'application Web peut supporter, tout en continuant à être en adéquation avec les critères de performance. De même, le Rapport Liste de Contrôle Serveur est un atout majeur car il met en évidence les mesures de performance clés pour chaque serveur surveillé avec les limites recommandées ; des icônes d'avertissement mettent en évidence les mesures collectées qui peuvent être problématiques pour les performances, et qui nécessitent d'être étudiées en détail. ■

Web Performance Load Tester

Dès 2000, Web Performance Load Tester (connu également sous le nom de Web Performance Trainer) s'est distingué sur le marché du test de performance Web grâce à sa simplicité d'utilisation, sa précision, son rapport qualité-prix et son service de support technique (disponible en anglais et en français). En 2006, l'interface graphique a été complètement refondue (basée sur Eclipse RCP), l'outil s'est vu enrichi d'une dizaine de nouvelles fonctionnalités dont la génération automatique de rapports. Depuis, de nouveaux modules (dont Web Performance Advanced Server Analysis pour la surveillance de serveurs) sont venus enrichir l'offre pour en faire une offre de test de performance Web complète mais qui ne déroge pas à ses principes de base : Simplicité, Précision, Prix abordable et Qualité du support. Web Performance Suite a su conquérir de nombreux clients en France, et pour répondre aux attentes du marché français une version francophone sera disponible prochainement. ■

Depuis 2000, Kapitec Software est Partenaire-Distributeur Exclusif pour la France, la Belgique et la Suisse de la solution de test de performance Web Performance Suite, éditée par Web Performance, Inc. Kapitec Software assure la promotion, la commercialisation, le support technique avant-vente et après-vente de Web Performance Suite. www.kapitec.com