

Lab- Évaluation des Performances Infrastructure Virtuelle pour l'Entreprise XYZ

Introduction :

Les tests de performance sont des évaluations systématiques visant à mesurer et évaluer les capacités d'un système, d'une application ou d'un composant informatique dans des conditions spécifiques. Ces tests couvrent des aspects tels que la vitesse, la stabilité, la fiabilité, la capacité et la réactivité. Les résultats des tests aident à optimiser les systèmes, prendre des décisions éclairées et améliorer continuellement les performances. Les types de tests incluent la charge, le stress, la stabilité, la latence, la bande passante, la fiabilité et la scalabilité. Ils sont essentiels pour garantir que les systèmes répondent aux attentes des utilisateurs et évoluent efficacement. Les tests de performance sont intégrés au processus de développement logiciel et à la gestion de l'infrastructure informatique.

Contexte :

L'entreprise XYZ lance un projet visant à déployer un serveur web essentiel pour ses opérations.

L'objectif est de créer une infrastructure virtuelle efficace qui peut gérer la charge de trafic prévue tout en optimisant l'utilisation des ressources matérielles.

Objectifs du Lab :

1. Évaluation des Performances du CPU :

- Utiliser le programme `Sysbench` pour mesurer les performances du CPU sous différentes charges de travail simulées.

2. Simulation de Charge de Travail Intensive sur la RAM :

- Utiliser `Stress` pour simuler une charge de travail intense sur la mémoire.
- Observer comment l'infrastructure virtuelle réagit sous une pression maximale.

3. Évaluation des Performances du Stockage :

- Utiliser `Sysbench` pour mesurer les performances du stockage, en mettant l'accent sur les taux de lecture/écriture.

4. Évaluation des Performances DU Serveur Apache2

- utiliser `wrk` pour évaluer le nombre de connexion.

5. Analyse des Résultats :

- Utiliser les résultats des évaluations pour analyser la stabilité et les performances de l'infrastructure virtuelle.

Prérequis :

- Une machine virtuelle Ubuntu avec un serveur web (LAMP) installé et configuré.

Tâches :

Partie 1:

Installation de wrk, sysbench et stress pour les Tests de Performances.

Installation de wrk :

1. Installer les dépendances nécessaires :

```
sudo apt-get install build-essential libssl-dev git
```

Explication

build-essential : Contient les packages nécessaires pour la compilation.

libssl-dev : Dépendance pour les fonctionnalités SSL.

git : Outil de gestion de versions pour récupérer le code source.

2. Se déplacer vers le répertoire /opt :

```
cd /opt
```

3. Récupérer le code source de wrk depuis GitHub :

```
sudo git clone https://github.com/wg/wrk.git
```

Explication

git clone récupère le code source de wrk depuis le dépôt GitHub.

4. Accéder au répertoire wrk :

```
cd wrk
```

5. Compiler le code source avec make :

```
sudo make
```

Explication

make : est utilisé pour automatiser le processus de compilation.

6. Copier l'exécutable wrk vers /usr/local/bin :

```
sudo cp wrk /usr/local/bin
```

Explication

cp : copie l'exécutable vers un répertoire accessible globalement.

Installation de sysbench :

```
sudo apt install sysbench
```

Installation de stress :

```
sudo apt install stress
```

Remarque : Les parties de 2 à 4 de ce document constituent simplement une prise en main des outils sysbench, wrk et stress.

La partie 5 représente la phase où les véritables tests sont effectués.

Partie 2:

Test de Performances du serveur web avec wrk.

Exécutez un test de trafic avec wrk sur la page info.php :

```
wrk -t8 -c200 -d30s http://localhost/info.php
```

Explication

- -t8 : 8 threads.
- -c200 : 200 connexions simultanées.
- -d30s : durée du test, 30 secondes.

Partie 3:

Test de Performances avec sysbench

1. Test de performance du CPU avec sysbench

```
sysbench cpu --threads=4 run
```

Explication

- Ceci simule une charge CPU avec 4 threads.

2. Test de Performance du Stockage avec sysbench

- Utilisez Sysbench pour mesurer les performances du stockage
Exécutez la phase de préparation pour créer les fichiers nécessaires. Utilisez la commande suivante :

```
cd ; sysbench fileio --file-total-size=2G --file-test-mode=rndrw --time=300 prepare
```

Explication

- fileio: L'option qui indique à Sysbench d'utiliser le module FileIO pour le test.
- --file-total-size=5G: Spécifie la taille totale des fichiers pour le test. Dans cet exemple, elle est fixée à 5 gigaoctets.
- --file-test-mode=rndrw: Définit le mode du test. Ici, le mode est défini sur "rndrw" pour des opérations de lecture/écriture aléatoires.
- --time=300: Définit la durée de la phase de préparation en secondes. Dans cet exemple, la durée est fixée à 300 secondes (5 minutes).
- prepare: Indique à Sysbench d'effectuer la phase de préparation, qui consiste à créer les fichiers nécessaires pour le test.

La phase de préparation crée les fichiers spécifiés avec les caractéristiques définies (taille, mode de test, etc.) afin que le test de performances puisse être exécuté de manière cohérente et reproductible. Une fois que la phase de préparation est terminée, vous pouvez exécuter la

commande originale (sans prepare) pour effectuer réellement le test de performances.

Une fois la phase de préparation terminée, exécutez la commande de test :

```
sysbench fileio --file-total-size=2G --file-test-mode=rndrw --time=300 run
```

Explication

Cette commande exécute le test avec les fichiers créés lors de la phase de préparation.

Partie 4:

Test de la RAM avec stress

```
stress --vm 1 --vm-bytes 2G --timeout 30s
```

Explication

- `--vm 1` : Cette option spécifie que vous souhaitez générer une charge de travail de type mémoire virtuelle. L'argument "1" indique le nombre de tâches de mémoire virtuelle à créer.
- `--vm-bytes 1G` : Cette option spécifie la quantité de mémoire virtuelle à allouer par processus. "2G" signifie 2 gigaoctet de mémoire virtuelle.
- Chaque tâche de mémoire virtuelle créée par stress utilisera 2 gigaoctet de mémoire.
- `--timeout 120s` : Cette option spécifie la durée pendant laquelle le test sera exécuté.
- Dans cet exemple, le test s'exécutera pendant 30 secondes.

C'est à vous de jouer...

Partie 5:

1. Tests & Analyse des Résultats

Lors des tests de charge, vous augmenterez progressivement la charge sur le système jusqu'à atteindre ou dépasser ses limites spécifiées. Cela peut vous aider à identifier plusieurs points clés :

1. **Seuil de capacité maximale** : Le niveau de charge auquel le système atteint sa capacité maximale tout en maintenant des performances acceptables.
2. **Seuil de dégradation** : Le niveau de charge au-delà duquel les performances commencent à dégrader de manière significative.
3. **Seuil de rupture** : Le point auquel le système ne peut plus fonctionner correctement et commence à échouer.

En identifiant ces seuils, vous pouvez déterminer la capacité maximale du système, planifier les capacités en conséquence, et ajuster les configurations si nécessaire pour répondre aux exigences du cahier des charges.

Voici un exemple simplifié d'un tableau pour un cahier des charges où vous devez définir les niveaux des trois seuils (CPU, RAM, seueur web Apache2) lors des tests de charge.

Ces valeurs peuvent être ajustées en fonction des besoins spécifiques de votre application et de votre VM.

Exemples :

- **Seuil de Capacité Max :**
 - 80% d'utilisation du CPU,
 - 90% de l'utilisation de la RAM,
 - 1000 connexions simultanées pour Apache2.
 - 500 Mo d'utilisation du stockage.
- **Seuil de Dégradation :**
 - 90% d'utilisation du CPU,
 - 95% de l'utilisation de la RAM,
 - 1500 connexions simultanées pour Apache2.
 - 1 Go d'utilisation du stockage.
- **Seuil de Rupture :**
 - 95% d'utilisation du CPU
 - 98% de l'utilisation de la RAM
 - 2000 connexions simultanées pour Apache2.
 - 2 Go d'utilisation du stockage.

Ces valeurs sont données à titre d'exemple et doivent être ajustées en fonction des spécifications réelles de votre système.

Important :Lors des tests de charge, observez les métriques pertinentes pour déterminer ces seuils.

Pour remplir le tableau, vous pouvez utiliser les commandes suivantes :

1. Sysbench pour le CPU :

Syntaxe de la commande :

```
sysbench --test=cpu --cpu-max-prime=<nombre> run
```

Attention ! remplacez <nombre> par le nombre que vous souhaitez utiliser pour le test de performance.

Exemple :

Seuil de Capacité Max

```
sysbench --test=cpu --cpu-max-prime=10000 run
```

Seuil de Dégradation

```
sysbench --test=cpu --cpu-max-prime=20000 run
```

Seuil de Rupture

```
sysbench --test=cpu --cpu-max-prime=30000 run
```

2. Stress pour la RAM :

Syntaxe de la commande :

```
stress --vm-bytes <RAM> --vm-keep -m <process> --timeout 30s
```

Remplacez <RAM> par la quantité de RAM souhaitée et <process> par le nombre de processus.

Exemple :

Seuil de Capacité Max

```
stress --vm-bytes 512M --vm-keep -m 1 --timeout 30s
```

Seuil de Dégradation

```
stress --vm-bytes 1G --vm-keep -m 2 --timeout 30s
```

Seuil de Rupture

```
stress --vm-bytes 2G --vm-keep -m 3 --timeout 30s
```

3. Wrk pour Apache2 (ou autre serveur web) :

Seuil de Capacité Max

```
wrk -t4 -c50 http://localhost/info.php
```

Seuil de Dégradation

```
wrk -t8 -c100 http://localhost/info.php
```

Seuil de Rupture

```
wrk -t16 -c200 http://localhost/info.php
```

4. Sysbench pour le stockage :

Seuil de Capacité Max

```
cd ; sysbench fileio --file-test-mode=rndrw --file-total-size=500M --time=300 prepare
```

```
sysbench fileio --file-test-mode=rndrw --file-total-size=500M --time=300 run
```

Supprimez les fichiers générés précédemment

```
rm test_file.*
```

Seuil de Dégradation

```
sysbench fileio --file-test-mode=rndrw --file-total-size=1G --time=300 prepare
```

```
sysbench fileio --file-test-mode=rndrw --file-total-size=1G --time=300 run
```

Supprimez les fichiers générés précédemment

```
rm test_file.*
```

Seuil de Rupture

```
sysbench fileio --file-test-mode=rndrw --file-total-size=2G --time=300 prepare
```


```
sysbench fileio --file-test-mode=rndrw --file-total-size=2G --time=300 run
```


Tableau à compléter et à rendre

Test de performance	Résultat Obtenu	Commentaires										
CPU avec sysbench	events per second Max : Dég : Rup :											
RAM avec stress	Quantité RAM Max : Dég : Rup :											
Stockage avec sysbench	<table><tr><td>Débit</td><td>Latence</td></tr><tr><td>Lecture/Écriture</td><td></td></tr><tr><td>Max :</td><td>Max :</td></tr><tr><td>Dég :</td><td>Dég :</td></tr><tr><td>Rup :</td><td>Rup :</td></tr></table>	Débit	Latence	Lecture/Écriture		Max :	Max :	Dég :	Dég :	Rup :	Rup :	
Débit	Latence											
Lecture/Écriture												
Max :	Max :											
Dég :	Dég :											
Rup :	Rup :											
Serveur HTTP (Web) avec wrk	QPS(Queries Per Second) : TPS: (Transactions Per Second): Max : Dég : Rup :											

Observez comment le système réagit aux tests et visualisez la consommation RAM et CPU sur l'interface Monitor

Pour accéder à l'interface du moniteur sur Ubuntu, suivez ces étapes:

1. Cliquez sur le menu **"Activités"** dans le coin supérieur gauche de l'écran.
2. Dans la barre de recherche qui s'affiche, tapez **"Monitor"**.
3. Lorsque l'icône du moniteur  apparaît dans les résultats de la recherche, cliquez dessus pour ouvrir l'interface du moniteur.

Dans l'interface du moniteur, sélectionnez l'onglet **"Ressources"** pour visualiser la consommation de votre CPU, et de la RAM .et l'onglet **"système de fichier"** pour la consommation disque dur.

