

Voici une démonstration pour effectuer un benchmark pour chaque MPM (**prefork**, **worker**, **event**) avec Apache HTTPD. Nous allons configurer et tester chaque MPM individuellement en utilisant **Apache Benchmark (ab)** pour évaluer les performances.

Étape 1 : Préparation

Installer Apache Benchmark

Assurez-vous que **ab** est installé. Sur une distribution basée sur Debian/Ubuntu :

```
sudo apt-get install apache2-utils
```

Créer un fichier de test

Créez un fichier HTML simple pour servir de cible pour les benchmarks :

```
echo "<html><body><h1>Apache Benchmark Test</h1></body></html>" > /var/www/html/test.html
```

Démarrer Apache

Redémarrez Apache après chaque modification de configuration :

```
sudo systemctl restart apache2
```

Étape 2 : Configuration et Benchmark pour chaque MPM

MPM 1 : prefork

1. Activer **mpm_prefork** :

```
sudo a2dismod mpm_worker mpm_event
sudo a2enmod mpm_prefork
sudo systemctl restart apache2
```

2. Configurer **mpm_prefork** : Ajoutez dans **/etc/apache2/mods-available/mpm_prefork.conf** :

```
<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers   5
    MaxSpareServers   10
    MaxRequestWorkers 100
    MaxConnectionsPerChild 1000
</IfModule>
```

3. **Benchmark avec Apache Benchmark** : Exemple pour 1000 requêtes simultanées (100 à la fois) sur la page `/test.html` :

```
ab -n 1000 -c 100 http://localhost/test.html
```

4. **Résultat attendu** : Notez le temps total (`Time taken for tests`), les requêtes par seconde (`Requests per second`), et le taux d'erreur.

MPM 2 : worker

1. **Activer `mpm_worker`** :

```
sudo a2dismod mpm_prefork mpm_event
sudo a2enmod mpm_worker
sudo systemctl restart apache2
```

2. **Configurer `mpm_worker`** : Ajoutez dans `/etc/apache2/mods-available/mpm_worker.conf` :

```
<IfModule mpm_worker_module>
    StartServers      3
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadLimit        64
    ThreadsPerChild    25
    MaxRequestWorkers  150
    MaxConnectionsPerChild 10000
</IfModule>
```

3. **Benchmark avec Apache Benchmark** : Répétez le test avec `ab` :

```
ab -n 1000 -c 100 http://localhost/test.html
```

4. **Résultat attendu** : Notez les mêmes paramètres pour comparaison (**Time taken**, **Requests per second**, taux d'erreur).

MPM 3 : event

1. **Activer mpm_event** :

```
sudo a2dismod mpm_prefork mpm_worker
sudo a2enmod mpm_event
sudo systemctl restart apache2
```

2. **Configurer mpm_event** : Ajoutez dans `/etc/apache2/mods-available/mpm_event.conf` :

```
<IfModule mpm_event_module>
    StartServers      2
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadLimit        64
    ThreadsPerChild    25
    MaxRequestWorkers  150
    MaxConnectionsPerChild 10000
</IfModule>
```

3. **Benchmark avec Apache Benchmark** : Répétez le test avec **ab** :

```
ab -n 1000 -c 100 http://localhost/test.html
```

4. **Résultat attendu** : Notez les métriques pour comparaison avec les autres MPM.

Étape 3 : Analyse des Résultats

Comparer les résultats obtenus :

Créez un tableau pour documenter les performances de chaque MPM :

MPM	Temps total (s)	Requêtes par seconde	Taux d'erreur
prefork
worker
event

Étape 4 : Surveillance

1. **Surveiller les ressources système** : Pendant les tests, utilisez `htop` ou `top` pour voir l'utilisation CPU et mémoire :

```
htop
```

2. **Analyser les logs Apache** :

```
tail -f /var/log/apache2/access.log  
tail -f /var/log/apache2/error.log
```

Étape 5 : Résumé et Conclusion

- **Scénario à faible charge** : `prefork` peut suffire, mais est gourmand en mémoire.
- **Charge modérée** : `worker` offre de meilleures performances et une utilisation équilibrée des ressources.
- **Connexion Keep-Alive ou charge élevée** : `event` est optimisé pour ce type de trafic.