

Nama : Nissa 'Uzzulfa

Kelas : QE - B

Jelaskan jenis - jenis performance testing!

Jawab : Performance testing adalah salah satu jenis pengujian perangkat lunak yang bertujuan untuk mengukur kinerja suatu aplikasi atau sistem dalam menghadapi beban yang tinggi atau dalam situasi yang ekstrem.

Load Testing

Load testing adalah jenis pengujian kinerja yang digunakan untuk mengukur seberapa baik sebuah sistem dapat menangani beban kerja tertentu. Dalam pengujian ini, sistem diberikan beban yang meningkat secara bertahap hingga mencapai batas maksimumnya, untuk menguji kemampuan sistem dalam menangani beban tersebut.

Tujuan dari load testing adalah untuk menemukan batasan kinerja sistem, seperti waktu respon yang lambat, kapasitas penyimpanan yang tidak mencukupi, atau kegagalan sistem yang disebabkan oleh beban pengguna yang berlebihan. Pengujian ini juga dapat membantu mengidentifikasi bottlenecks kinerja pada sistem, seperti konfigurasi server yang tidak tepat, database yang tidak dioptimalkan, atau masalah jaringan yang mempengaruhi kinerja sistem.

Load testing harus dilakukan secara teratur untuk memastikan sistem selalu berkinerja optimal. Selain itu, hasil dari load testing dapat digunakan untuk melakukan perbaikan dan optimisasi sistem, sehingga sistem dapat menangani beban kerja yang lebih besar dan kompleks di masa depan.

- Scalability Testing

Scalability testing adalah jenis pengujian kinerja yang penting untuk mengevaluasi kemampuan suatu sistem dalam menangani peningkatan beban kerja dengan menambahkan sumber daya tambahan. Dalam pengujian ini, diperiksa apakah sistem dapat menambahkan sumber daya dengan lancar dan apakah kinerja sistem tetap stabil ketika beban kerja meningkat.

Tujuan dari scalability testing adalah untuk mengetahui sejauh mana sistem dapat bertahan dan berkinerja baik ketika diberikan beban kerja yang lebih besar. Dalam pengujian ini, diperiksa apakah sistem dapat menambahkan sumber daya dengan lancar dan apakah kinerja sistem tetap stabil ketika beban kerja meningkat.

- Volume Testing

Volume testing adalah jenis pengujian kinerja yang bertujuan untuk menguji kemampuan suatu sistem dalam menangani jumlah data yang besar. Dalam pengujian ini, sistem diberikan volume data yang meningkat secara bertahap hingga mencapai batas maksimum sistem.

Tujuan dari volume testing adalah untuk menemukan batasan kinerja sistem, seperti kapasitas penyimpanan yang tidak mencukupi atau waktu respon yang lambat ketika beban data meningkat. Pengujian ini juga dapat membantu mengidentifikasi bottleneck pada sistem, seperti database yang tidak dioptimalkan atau penggunaan memori yang tidak efisien.

- **Stress Testing**

Stress testing adalah jenis pengujian kinerja yang bertujuan untuk menguji kemampuan suatu sistem dalam menangani beban kerja yang ekstrem. Dalam pengujian ini, sistem diberikan beban kerja yang jauh melebihi kapasitas normalnya untuk melihat sejauh mana sistem dapat menangani beban kerja tersebut dan apakah sistem dapat pulih dengan cepat setelah beban kerja dilepaskan.

Tujuan dari stress testing adalah untuk menemukan batas maksimum sistem dan mengevaluasi kemampuan sistem dalam menangani beban kerja yang tidak terduga. Pengujian ini juga dapat membantu mengidentifikasi titik kegagalan atau batasan pada sistem dan memperbaikinya sebelum terjadi masalah pada sistem yang lebih besar.

- **Spike Testing**

Spike testing adalah jenis pengujian dalam performance testing yang bertujuan untuk menguji reaksi aplikasi atau sistem terhadap perubahan beban yang tiba-tiba atau naik turun secara signifikan dalam waktu yang cepat. Dalam spike testing, beban atau jumlah pengguna yang digunakan dalam pengujian dinaikkan atau diturunkan secara tiba-tiba dalam waktu yang singkat, sehingga dapat menguji kemampuan aplikasi atau sistem dalam menghadapi perubahan beban yang tidak terduga.

Tujuan utama dari spike testing adalah untuk menguji kinerja aplikasi atau sistem saat menghadapi situasi yang ekstrem atau lonjakan pengguna dalam waktu yang singkat, seperti ketika terjadi peristiwa atau kejadian yang dapat mempengaruhi beban pada sistem, misalnya saat peluncuran produk baru, promosi besar-besaran, atau acara khusus lainnya. Spike testing dapat membantu mengidentifikasi masalah kinerja yang mungkin muncul saat sistem diberikan beban yang tinggi.

secara tiba-tiba, seperti peningkatan waktu respons, penurunan throughput, atau penurunan performa lainnya.

Beberapa langkah yang umum dilakukan dalam spike testing antara lain:

1. Menyusun skenario pengujian: Menentukan skenario pengujian yang akan digunakan, termasuk perubahan beban yang akan diberikan secara tiba-tiba dan durasi waktu untuk mengujinya.
2. Menerapkan perubahan beban: Mengubah beban atau jumlah pengguna yang diterapkan pada sistem secara tiba-tiba sesuai dengan skenario pengujian yang telah ditentukan.
3. Merekam dan menganalisis hasil: Merekam data kinerja sistem selama periode perubahan beban dan menganalisis hasil pengujian, termasuk melihat waktu respons, throughput, dan utilitas sumber daya sistem lainnya.
4. Mengidentifikasi masalah: Mengidentifikasi potensi masalah kinerja yang muncul saat perubahan beban diterapkan dan melakukan analisis untuk menemukan penyebab masalah tersebut.
5. Mengambil tindakan perbaikan: Jika ditemukan masalah kinerja, melakukan tindakan perbaikan yang diperlukan untuk meningkatkan kinerja sistem, seperti melakukan tuning konfigurasi, mengoptimalkan kode, atau meningkatkan kapasitas sumber daya.

Dengan melakukan spike testing, tim pengujian dapat mengevaluasi kemampuan aplikasi atau sistem dalam menghadapi perubahan beban yang tidak terduga dan mengidentifikasi masalah yang mungkin muncul saat sistem diberikan beban yang tinggi secara tiba-tiba. Hal ini dapat membantu meningkatkan kinerja sistem dan memastikan bahwa sistem mampu menjaga performa yang baik dalam situasi ekstrem.

- **Benchmark Testing**

Benchmark testing adalah jenis pengujian dalam performance testing yang bertujuan untuk membandingkan kinerja suatu aplikasi atau sistem dengan standar atau benchmark yang telah ditetapkan sebelumnya. Dalam benchmark testing, aplikasi atau sistem diuji dengan menggunakan serangkaian skenario atau beban pengguna yang telah ditentukan sebelumnya, dan hasilnya dibandingkan dengan nilai-nilai benchmark yang telah ditetapkan untuk mengukur kinerja sistem.

Tujuan utama dari benchmark testing adalah untuk mengukur kinerja suatu aplikasi atau sistem dalam memenuhi standar kinerja yang telah ditetapkan sebelumnya. Benchmark dapat berupa nilai-nilai kinerja tertentu yang telah

ditetapkan berdasarkan persyaratan atau kebutuhan bisnis, atau dapat merujuk pada nilai-nilai kinerja yang telah ditetapkan sebagai standar industri atau best practice.

Beberapa langkah yang umum dilakukan dalam benchmark testing antara lain:

1. Menyusun skenario pengujian: Menentukan skenario pengujian yang akan digunakan, termasuk beban pengguna, durasi pengujian, dan metrik kinerja yang akan diukur.
2. Menetapkan benchmark: Menetapkan standar atau benchmark kinerja yang akan digunakan sebagai acuan untuk membandingkan hasil pengujian.
3. Menjalankan pengujian: Menjalankan pengujian menggunakan skenario pengujian yang telah ditentukan, mengukur dan merekam kinerja sistem selama pengujian.
4. Menganalisis hasil: Menganalisis hasil pengujian dan membandingkannya dengan nilai-nilai benchmark yang telah ditetapkan, untuk mengevaluasi sejauh mana kinerja sistem memenuhi standar atau benchmark yang telah ditetapkan.
5. Melakukan tindakan perbaikan: Jika ditemukan perbedaan antara hasil pengujian dan benchmark yang ditetapkan, melakukan tindakan perbaikan yang diperlukan untuk meningkatkan kinerja sistem, seperti melakukan tuning konfigurasi, mengoptimalkan kode, atau meningkatkan kapasitas sumber daya.

Dengan melakukan benchmark testing, tim pengujian dapat memperoleh pemahaman yang lebih jelas tentang kinerja suatu aplikasi atau sistem dalam memenuhi standar atau benchmark kinerja yang telah ditetapkan sebelumnya. Hal ini dapat membantu dalam mengidentifikasi potensi masalah kinerja, mengoptimalkan kinerja sistem, dan memastikan bahwa sistem beroperasi sesuai dengan standar atau benchmark yang ditetapkan.