

1. Testarea de performanță – noțiuni de bază

Testarea de performanță reprezintă procesul de testare software în care are loc testarea vitezei de răspundere, stabilității, fiabilității, scalabilității și utilizării resurselor în cadrul unui anumit volum de lucru. Principalul scop al acestui proces este identificarea și eliminarea blocajelor de performanță apărute într-o aplicație software.

În linii generale, testarea performanței poate avea diferite scopuri, cum ar fi:

- Demonstrarea faptului că sistemul îndeplinește criteriile de performanță;
- Compararea a două sisteme pentru a găsi care operează mai bine;
- Stabilirea unor părți ale sistemelor care cauzează scăderea nivelului de performanță.

Pe lângă cele 3 criterii generale de testare a performanței mai pot fi adăugate două tipuri (Figura 1.1):

- 1) Testarea încărcării – lansării – supra-încărcării;
- 2) Testare de stres;
- 3) Testarea stabilității;
- 4) Testarea în impulsuri (spike testing);
- 5) Testare în configurație (configuration testing).

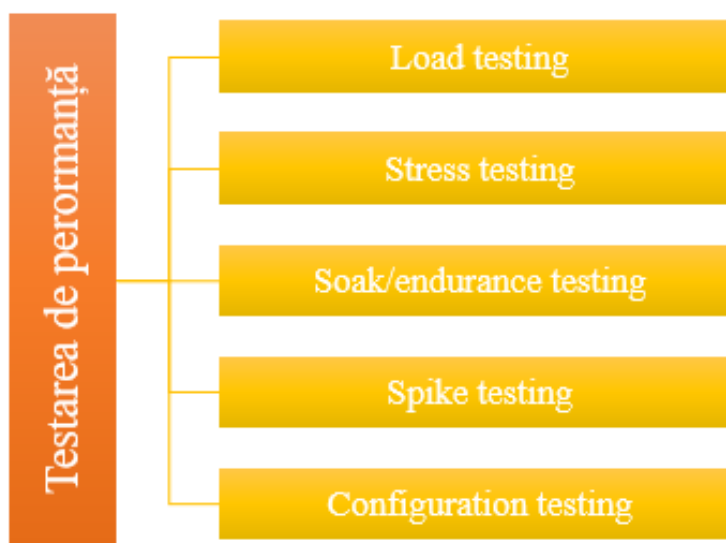


Fig. 1.1 Procesul de testare a performanței

Testarea în impulsuri. După cum sugerează și numele, se realizează prin simularea creșterii sau descreșterii bruște a sarcinii generate de un interval larg a numărului de utilizatori și evaluarea comportamentului sistemului. Scopul este de a determina dacă performanța va fi afectată, atunci sistemul va ieși din funcțiune sau dacă va putea face față schimbărilor supuse.

Testarea în configurație este o altă variantă a testării de performanță. În locul testării performanței prin perspectiva încărcării, se testează efectele schimbării de configurație din mediul programului asupra comportamentului și performanței.

2. Etapele testării performanței

Metodologia adoptată pentru testarea performanței are o varietate vastă pe când obiectivul principal rămâne același. Potrivit Microsoft Developer Network metodologia de testare constă din următoarele pași:

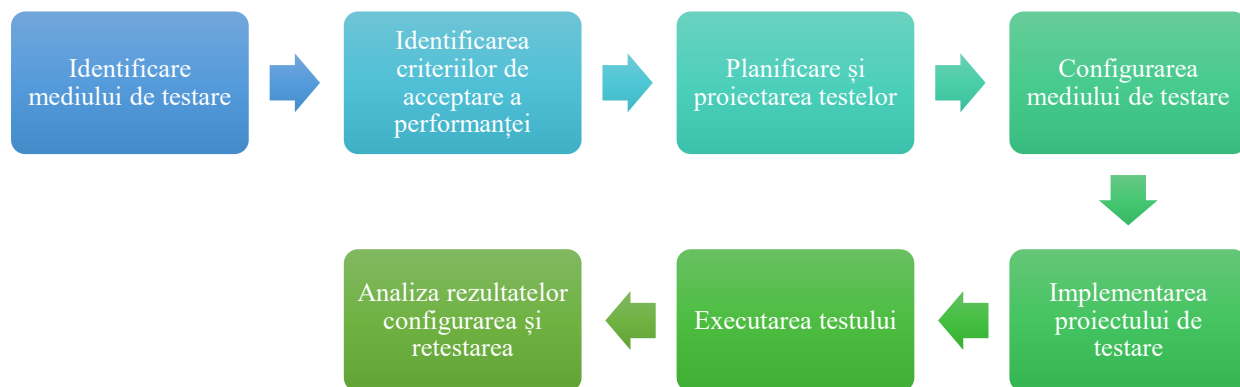


Fig. 1.2 Procesul de testare a performanței

1) Identificare mediului de testare.

Identificarea mediului fizic de testare și mediului de producere are loc cu atât mai bine cu cât instrumentele și resursele sunt accesibile echipei de testare. Mediul fizic include elementele hardware, software și configurațiile network. Având cunoștințe profunde în ceea ce privește tot mediul de testare de la început permite o planificare și proiectare a testării mai eficiente și ajută la identificarea problemelor în primele etape ale proiectării. În careva situații, acest proces trebuie să fie repetat pe parcursul ciclului de viață al proiectului.

2) Identificarea criteriilor de acceptare a performanței.

Aceasta presupune identificarea timpului de răspundere, capacității de producție, precum și scopurile, și restricțiile de utilizare a resurselor. După regulă, timpul de răspundere reprezintă o problemă pentru utilizatori, capacitatea de producție este o problemă de afaceri, iar utilizarea resurselor este o problemă de sistem. În plus, determinarea criteriilor de succes al proiectului care nu pot fi luate în considerare de aceste scopuri și restricții. Spre exemplu, utilizarea testării performanței pentru a aprecia care combinație de setări ale configurației va influența obținerea unor caracteristici dorite.

3) Planificare și proiectarea testelor.

La această etapă se identifică scenariile-cheie, se determină schimbările între utilizatori reprezentativi și modul de simulare a acestor schimbări, se definesc datele testării și se stabilesc

metricile pentru colectare. Apoi, se consolidează informația în una sau mai multe modele de utilizare a sistemului pentru a fi implementat, executat și analizat.

4) Configurarea mediului de testare.

Are loc pregătirea mediului de testare, a instrumentelor și resurselor necesare pentru îndeplinirea fiecărei strategii, întrucât funcțiile și componentele devin valabile pentru testare.

5) Implementarea proiectului de testare.

Presupune dezvoltarea testării de performanță în conformitate cu proiectarea testării.

6) Executarea testului.

Are loc rularea și monitorizarea testării. Se execută testele validate pentru analiză, se monitorizează testul și mediul testării.

7) Analiza rezultatelor, configurarea și re-testarea.

Această etapă constă din analizarea, consolidarea și distribuirea datelor despre rezultatele obținute, modificarea setărilor și testarea repetată.

3. Instrumentele testării performanței

Din punct de vedere ale instrumentelor utilizate, testarea performanței se divizează în două categorii de bază: scenariul performanței și monitorizarea.

Scenariul performanței presupune crearea sau scrierea fluxurilor de lucru ale proceselor de afaceri. La această parte se utilizează o largă varietate de instrumente precum: HP LoadRunner, NeoLoad, Apache JMeter, Rational Performance Tester, Silk Performer, Flood ș.a.

Fiecare instrument menționat folosește sau limbaj de programare sau o formă de reprezentare vizuală de creare și simulare a fluxurilor de lucru ale utilizatorilor finali. Majoritatea conțin funcția ”Record & Replay” cu care tester-ul va lansa instrumentul de testare, îl va conecta cu un browser și va fixa toate tranzacțiile de rețea care apar între client și server. În acest sens, este dezvoltat un scenariu care poate fi îmbunătățit/modificat pentru a simula diferite scenarii de afaceri.

Monitorizarea performanței. Cu ajutorul acesteia, comportamentul și caracteristicile de răspundere ale aplicației sunt supuse controlului. Parametrii care sunt de obicei monitorizați în timpul testării sunt:

- Utilizarea CPU;
- Utilizarea memoriei;
- Utilizarea rețelei.

Pentru determinarea cauzei exacte a problemei, programatorul utilizează instrumente de profilare (stabilire în linii generale a obiectivului de bază al producției) pentru a stabili ce părți ale dispozitivului sau ale software-ului contribuie cel mai mult la performanță scăzută.

4. Studiu de caz

Acest studiu de caz reprezintă un scenariu de dezvoltare a testului de performanță pentru măsurarea utilizării procesorului folosind programul Maximo Asset Management implementat de compania IBM.

Compania „A” plănuiește să implementeze programul Maximo Asset Management cu o personalizare amplă. Pentru a asigura o implementare de succes, compania „A” dezvoltă și rulează teste de performanță.

Descriere

Compania „A” are ca scop să folosească programul dat pentru gestiunea activelor, cumpărări, urmărirea comenzilor efectuate. Din cauza proceselor specifice de afaceri, compania „A” deține o implementare personalizată care utilizează fluxuri automatizate de lucru. Utilizatorii companiei folosesc următoarele componente:

- Active;
- Cereri de achiziție;
- Comenzi de achiziție;
- Urmărirea comenzilor.

Primul pas: Determinarea obiectivelor de măsurare

Echipele de implementare al companiei „A” iau în considerare întrebarea-cheie de afaceri și acordă prioritate riscurilor, recompenselor și costurilor în implementare. Pe baza cercetărilor, echipa de implementare stabilește că utilizatorilor nu le place atunci când tranzacțiile din aplicațiile web durează mai mult pentru a se efectua. Interviuurile realizate cu grupuri anumite determină faptul că utilizatorii devin frustrați atunci când o tranzacție durează mai mult de 2 secunde pentru a răspunde.

Echipele de gestionare a serverelor de la Compania „A” stabilește că, dacă utilizarea procesorului rămâne sub 80% pentru o încărcare a utilizatorilor țintă pe sistem, atunci procesorul poate furniza resurse adecvate pentru ca aplicațiile să funcționeze la nivelul dorit. Această valoare poate gestiona, de asemenea, creșteri ocazionale în procesare, cum ar fi în timpul procesării pentru sfârșit de lună, fără a influența asupra timpului de răspuns. Pe baza mărimii companiei, echipa de gestionare a serverului identifică 950 de utilizatori ca sarcină simultană țintă.

Pasul doi: Dezvoltarea cazurilor de utilizare

Echipele de implementare ia în considerare comportamentul utilizatorilor pe parcursul zilei. Aceasta identifică că, în general, utilizatorii se conectează după ce sosesc dimineața. Utilizatorii realizează de obicei un set de activități de lucru și apoi se deconectează. Echipa de implementare

estimează că fiecare utilizator finalizează un caz de utilizare de aproximativ 20 de ori într-un interval de o oră.

Pentru a aproxima comportamentul de conectare și deconectare, echipa de implementare intenționează să creeze cazuri de utilizare în care utilizatorii de test automat se conectează, rulează șase iterații ale unui caz de utilizare și apoi se deconectează. Utilizatorii de test automat se conectează din nou și repetă ciclul. O pauză de 5 până la 10 secunde este încorporată în pașii din scenariului pentru a reprezenta ratele reale de procesare ale utilizatorilor.

Echipa de implementare identifică cazurile de utilizare care sunt necesare pentru a testa implementarea. Echipa atribuie, de asemenea, factori de volum pentru fiecare caz de utilizare. Factorul de volum reprezintă numărul de utilizatori automatizați care rulează fiecare caz de utilizare.

Tabelul 4.1 Cazuri de utilizare identificare pentru testare în Compania „A”

Identificator caz. de utiliz.	Descriere	Factor volum, %
AS01	Căutarea activelor și examinarea informațiilor despre siguranță	20
PO01	Crearea unei cereri de achiziție, apoi crearea unei comenzi de achiziție pe baza cereri de achiziție.	5
PO02	Modificarea statutului a comenzii ”În curs de desfășurare”	5
PO03	Primirea comenzii	5
PO04	Închiderea comenzii de achiziție	5
WF01	Crearea unei comenzi de lucru și direcționarea acesteea prin aplicația Workflow	12
WF02	Vizualizarea unei comenzi de lucru și direcționarea acesteea prin aplicația Workflow pentru aprobare	12
WF03	Emiterea unui element al unei comenzi de lucru și direcționarea acestuia prin aplicația Workflow	12
WF04	Adăugarea forței de muncă la o comandă de lucru și direcționarea comenzii prin aplicația Workflow pentru finalizare	12
WF05	Trecerea unei comenzi de lucru prin aplicația Workflow pentru închidere.	12

Pasul trei: Elaborarea testelor

Echipa de implementare înscrie fiecare caz de utilizare într-un caz de testare. Fiecare caz de test listează fiecare etapă necesară pentru a rula testul. Tabelul următor oferă un exemplu de caz de testare pentru cazul de utilizare AS01, care caută active și apoi examinează informațiile despre siguranță.

Tabelul 4.2 Cazuri de testare de căutare a activelor și examinare a informației despre siguranță

Operațiune	Descriere	Rezultatul așteptat
AS01_01_D_Launch	Porniți Maximo Asset Management	Este afișat ecranul Bun venit la Maximo
AS01_02_D_Logon	Introduceți numele de utilizator ASSET0001 și parola maxasset. Faceți clic pe Sign In	Se afișează Start Center
Începerea buclei pentru elemente multiple de lucru		
AS01_03_D_GoTo	Faceți clic pe Go To	Se afișează meniul Go To
AS01_04_D_LaunchAssets	Selectați Assets > Assets	Este afișată aplicația Assets
AS01_05_D_EnterAsset Prefix	În câmpul Asset, introduceți CAC și apăsați tasta Tab	Fundalul pentru câmpul Asset se schimbă în alb. Cursorul se deplasează la câmpul următor
AS01_06_D_FindAsset	Faceți clic pe pictograma Filter Table	Este afișată o listă care listează toate activele care au CAC (Customer acquisition cost) în numele lor
Crearea buclelor de până la 9 ori pentru a selecta o pagină aleatorie de date		
AS01_07_D_NextPage	Faceți clic pe pictograma Next Page	Este afișată următoarea pagină cu rezultatele activelor
Finisarea buclei pentru datele paginii		
AS01_08_D_SelectAsset	Selectați un număr de activ aleatoriu	Sunt afișate detaliile pentru activul selectat
AS01_09_D_TabSafety	Selectați fila Safety	Se afișează fila Safety
AS01_10_D_ReturnTo StartCenter	Faceți clic pe Start Center	Este afișat Start Center
Finisarea buclei pentru elemente multiple		
AS01_11_D_Logoff	Faceți clic pe Sign out	Deconectarea este finalizată. Este afișat ecranul Bun venit la Maximo.

Pasul patru: Definirea mediului de testare

În etapele inițiale de planificare, echipa de implementare discută dacă costul unui mediu de testare identic cu mediul de producție este o cheltuială admisibilă. În cele din urmă, echipa de implementare decide că riscurile unui mediu de testare inadecvat depășesc orice economii

potențiale de costuri. Prin urmare, mediul de testare al companiei „A” este un duplicat exact al mediului de producție.

În pregătirea pentru implementare, datele existente din sistem pe care Maximo Asset Management este programat să le înlocuiască sunt migrate în mediul de testare. Migrarea datelor asigură faptul că echipa este capabilă să migreze datele existente și oferă, de asemenea, un volum real și o structură a datelor din baza de date, care este apoi utilizată pentru testarea performanței.

După implementarea inițială în producție, echipa poate testa modificări suplimentare în mediul de testare. Asemănările dintre mediile de testare și producție oferă un grad ridicat de încredere că rezultatele similare pot fi obținute atunci când modificările suplimentare sunt mutate în mediul de producție.

Pasul cinci: Rularea testelor

Echipa de implementare poate înregistra acum testul pentru exemplul de caz de testare și poate repeta procesul pentru a dezvolta cazuri de testare pentru toate cazurile de utilizare. Echipa de implementare utilizează un instrument de testare a performanței pentru a crea cazurile de testare. După ce toate testele sunt înregistrate și depanate, echipa de implementare rulează testele.

Pentru a afla cum funcționează sistemul la diferite niveluri de încărcare, echipa de implementare începe testul cu 750 de utilizatori simultani. Încărcarea utilizatorului este mărită cu încă 50 de utilizatori după un interval de 30 de minute. Creșterea numărului de utilizatori se repetă până când 950 de utilizatori simultani sunt pe sistem. Testul este configurat pentru a conecta un utilizator virtual la fiecare 2000 de milisecunde până când utilizatorii de la fiecare nivel de încărcare sunt conectați. Acest proces este destinat să elimine procesarea suplimentară necesară pentru a crește încărcarea.

Pasul șase: Analiza rezultatele testelor

După executarea testelor și compilarea datelor, echipa de implementare extrage timpul de răspuns și rezultatele de utilizare a procesorului într-o foaie de calcul. Apoi se generează un grafic rezumat pentru a identifica dacă sunt îndeplinite criteriile de performanță. Următorul grafic prezintă un exemplu de rezultate ale utilizării:

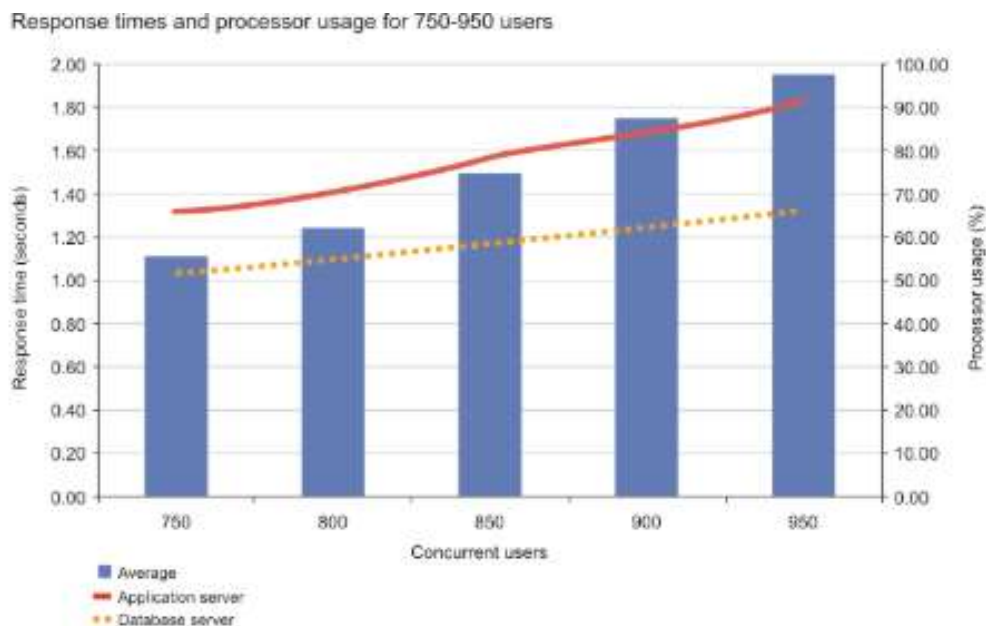


Fig.4.1 Exemplu de rezultate ale testelor de performanță pentru utilizarea procesorului

În graficul rezultat, timpul mediu de răspuns este sub 2 secunde. Utilizarea procesorului pe serverul bazei de date rămâne sub 80% la încărcarea țintă a 950 de utilizatori simultani. Cu toate acestea, procesorul serverului de aplicații depășește 80% utilizare cu o încărcare de 850 de utilizatori simultani. Prin urmare, criteriile testului de performanță nu au fost îndeplinite.

Echipa de implementare trebuie să investigheze pentru a determina dacă problema utilizării excesive a procesorului poate fi rezolvată prin reglarea setărilor legate de performanță sau modificări ale fluxurilor de lucru automatizate. Echipa de implementare poate decide, de asemenea, dacă sunt necesare resurse suplimentare de procesor pentru a îndeplini criteriile de performanță în implementarea producției.

Concluzie

Testarea performanței trebuie să fie o prioritate majoră înaintea lansării oricărui program (în dependență de domeniu). Aceasta trebuie să fie realizată încă la începutul etapei de dezvoltare pentru a detecta defectele mai devreme și pentru a satisface necesitățile utilizatorilor, economisind timp și bani.

Problemele care ar putea fi întâlnite pe parcursul testării sunt:

- Configurarea și gestionarea datelor;
- Crearea testelor de performanță care să corespundă cât mai îndeaproape comportamentului real al mediului înconjurător;

- Scalarea testelor pentru a corespunde capacității disponibile a mediului de testare și sarcinilor utilizatorului pentru a obține rezultate semnificative ale performanței;
- Modelarea proceselor de fundal care rulează pe mediul real și poate afecta performanța;
- Monitorizarea activității de mediu pentru a identifica situațiile în care sunt probleme de performanță sau „blocaje”;
- Furnizarea de informații semnificative și acționabile în rapoarte;
- Identificarea tipurilor corecte de testare pentru a rula.

Însă, pentru asigurarea maximă a vitezei de răspuns, scăderea numărului de greșeli, determinarea problemelor cu baza de date și crearea website-urilor sau aplicațiilor sigure și rapide, este necesar de a stabili obiective clare de testare, alegând între timp, instrumentele corecte de testare software, a infrastructurii și tehnologiilor.