

APLICAREA LANȚURILOR MARKOV ȘI MODELELOR MARKOV UNIFICATE (UMM) ÎN TESTAREA STATISTICĂ WEB

AUTOR: ANTON MIRONIUC

Conducător științific: magistru în TI, lect. sup. Mariana CATRUC

Universitatea Tehnică a Moldovei

Email: amironiuc@gmail.com, micatruc@yahoo.com

Abstract: Testarea Statistică și analiza fiabilității pot fi folosite eficient pentru a asigura calitatea aplicațiilor Web. Pentru a folosi această strategie, vom extrage informații de utilizare și informații de eșec din log-urile Web existente. Informațiile de utilizare sunt folosite pentru a construi modele pentru Testarea Statistică Web. Informațiile legate de eșec sunt utilizate pentru a măsura fiabilitatea aplicațiilor Web și potențialul de eficacitate a testării Statistice Web. Am aplicat această abordare pentru a analiza unele log-uri Web reale. Rezultatele au demonstrat viabilitatea și eficiența abordării selectate.

Cuvinte cheie: World Wide Web, Testarea Statistică, analiza fiabilității, lanțurile Markov, statistica de utilizare.

1. Introducere

Aplicațiile Web asigură accesul universal la resursele Web pentru un număr masiv de utilizatori.

Cu prevalența World Wide Web, asigurarea calității pentru Web devine din ce în ce mai importantă. Astăzi există diverse tehnici, pentru testarea generală a sistemelor software și măsurarea fiabilității lor. Cu toate acestea, mediul Web prezintă numeroase provocări noi, și necesită adaptarea tehnicilor existente sau dezvoltarea tehnicilor noi.

Din cauza că se pune accentul pe utilizator și din cauza dimensiunii mari a spațiului Web, o metodă bună pentru asigurarea efectivă a calității Web este testarea statistica și analiza fiabilității [1], [2].

Aceste tehnici pot să ajute la priorizarea eforturilor de testare bazându-se pe scenarii de utilizare pentru resurse Web individuale și a modelelor de navigare pentru a asigura fiabilitatea aplicației Web. Această abordare ierarhică pentru testarea statistica Web și analiza fiabilității bazată pe Modele Markov Unificate (UMMs) a fost prima dată implementată de J.Tian [3], [4].

O condiție prealabilă pentru astfel de testare statistică și analiza fiabilității este colectarea informațiilor de utilizare a Web și construirea modelelor de utilizare corespunzătoare.

Fiabilitatea site-ului Web, precum și eficacitatea unei astfel de tehnici de testare pot fi evaluate prin analiza datelor corespunzătoare de eșec. Pentru aplicații Web, diferite fișiere log sunt păstrate pe serverele Web.

2. Necesitatea Strategiei de Testare statistică a aplicațiilor Web

Tipurile de testări cum ar fi: Testarea funcționalităților (Functionality testing), testarea de încărcare (Load testing), testarea de utilizabilitate (Usability testing), folosesc ca criteriu de acoperire cu teste o scară mică. Cu alte cuvinte, se încearcă acoperirea anumitor zone sau caracteristici, cum ar fi conformitatea cu standardele HTML, sau verificarea de funcționalitate, utilizare, etc, în conformitate cu specificațiile produsului sau implementările lui [4]. Cu toate acestea, după cum s-a menționat anterior, din cauza numărului masiv de utilizatori și a dimensiunii enorme a rețelei Web, o astfel de acoperire, bazată pe testarea exhaustivă poate fi utilizată numai într-o zonă mică, restricționată local, în loc de o colecție rezonabilă de pagini Web. Pentru a face astfel de teste pe o scară largă, este necesară o testare selectivă pe baza unor scheme de priorități.

Testarea statistică îndeplinește aceste cerințe care ne permit să ne concentrăm pe caracteristici sau părți care sunt frecvent utilizate pentru a asigura fiabilitatea produsului [1], [2].

Strategia generală poate fi efectuată în trei etape:

- **Pasul 1.** Construirea modelelor statistice de testare bazate pe scenarii reale de utilizare și frecvențe corespunzătoare.
- **Pasul 2.** Utilizarea acestor modele pentru proiectarea cazului de test, selecția, și execuția.
- **Pasul 3.** Analiza rezultatelor testelor pentru evaluarea fiabilității, predicției, și ajutorului pentru luarea deciziilor.

Diferite tipuri de modele de utilizare pot fi construite pentru a suporta testarea statistică și analiza fiabilității Web, inclusiv modele bazate pe folosirea lanțurilor Markov [2].

O condiție prealabilă pentru această strategie este de a colecta informații de utilizare pentru a construi model de utilizare. Astfel de strategii de testare statistică, de asemenea, ne dau datele care pot fi folosite pentru a evalua fiabilitatea sistemului pus în aplicare, precum și a evalua eficacitatea potențială a tehnicii de testare. Colectare automată a informației de utilizare și informației asociate de eșec ar putea duce la o punere în aplicare efectivă și eficientă a acestei strategii.

3. Modele Markov Unificate pentru Testarea Statistică Web

Informația de utilizare poate fi folosită pentru a construi Modelul Markov Unificat (UMMs) [3] pentru a ghida testarea de ansamblu a paginii Web [4]. Aceste UMMs pot fi folosite pentru modelarea comportamentului și utilizării sistemului, astfel că eșecurile care au probabilitatea să apară la utilizatorii Web vor putea fi detectate și eliminate prin astfel de testare. De asemenea poate fi obținută evaluarea realistă a performanțelor și fiabilității sistemului.

3.1 Conceptul Modelelor Markov Unificate

UMMs intră în posesia informațiilor despre fluxul de execuție (fluxul de control), fluxul de informații (fluxul de date și dependență), procesarea tranzacțiilor (crearea volumului de muncă, de manipulare și finalizare), precum și informații de utilizare probabilistice asociate. Această informație este reprezentată în UMMs ca un set de lanțuri ierarhice Markov, care pot fi folosite pentru testarea statistică, evaluarea performanțelor, și analiza fiabilității [3].

Lanțul Markov de nivel superior, reprezintă la nivel înalt unități operaționale (stări), conexiunile asociate (tranziții), și probabilitățile de utilizare (cantități numerice scrise pe link-uri). Suboperațiunile diferite pot fi asociate cu o stare individuală și ar putea fi modelate de modele mai detaliate. Colecția din aceste lanțuri Markov, la diferite nivele de detaliere formează o ierarhie care joacă un rol important în strategia de testare statistică Web.

În unele din aceste lanțuri Markov, suma probabilităților de tranziții de la o anumită stare poate fi mai mică de 1, deoarece destinațiile externe (și sursa) sunt omise pentru ca să se păstreze modele simple. Înțelegerea implicită a acestor UMMs este că lipsesc probabilitățile care duc la destinații externe.

3.2 Testarea bazată pe frecvențe de utilizare în UMMs

Cazurile de testare pot fi generate urmărind stările și tranzițiile de stări în UMMs pentru a alege unitățile operaționale individuale (stări) și a le corela (tranziții) pentru a forma o operațiune generală de tip end-to-end. Cazurile posibile de testare cu probabilități mai mari de praguri specifice pot fi generate pentru a acoperi operațiunile utilizate frecvent. În aplicațiile practice, pragurile pot fi ajustate pentru a controla numărul de cazuri de test care vor fi generate și executate.

În acest model se folosește în primul rând un prag de probabilitate global pentru operațiunile complete end-to-end pentru a se asigura că operațiunile utilizate frecvent de către clienți țintă sunt reglementate și testate în mod adecvat. Dacă această probabilitate este peste pragul dat, cazul de test corespunzător va fi selectat și executat. Pentru testarea Web, probabilitățile acestea de branșament de asemenea, pot fi folosite direct pentru a selecta dinamic pagina(*i*) următoare pentru a le vizita.

Dacă lanțul Markov este staționar, se poate ajunge la un echilibru, în cazul în care probabilitatea de staționare π_i pentru a fi în starea *i* rămâne aceeași înainte și după tranzițiile de stare de-a lungul timpului. Setul $\{\pi_i\}$ poate fi obținut prin rezolvarea următorului set de ecuații:

$$\pi_j = \sum \pi_{ij} p_{ij}, \quad \pi_i \geq 0, \quad \text{și} \quad \sum \pi_i = 1,$$

unde p_{ij} este probabilitatea tranziției din starea i în starea j .

Probabilitatea (statică) π_i indică frecvența relativă de vizitare a unei anumite stări i după ce lanțul Markov atinge acest echilibru. Prin urmare, testarea stărilor care depășesc un anumit prag este menită să se concentreze pe operațiunile individuale utilizate frecvent. Din păcate, cele mai multe lanțuri Markov nu sunt statice, inclusiv majoritatea celor care sunt construite pentru testarea Web. Cu toate acestea, ideea de a se concentra pe stările de testare cu probabilitatea înaltă de staționare poate fi încă folosită în aplicațiile practice, prin aproximarea probabilităților statice cu frecvențele relative înregistrate a vizitei.

3.3 Testarea bazată pe ierarhiile UMM

Structura ierarhică a UMMs, de asemenea, ne dă flexibilitatea de a atinge o acoperire selectivă și posibilitatea de a îmbunătăți eficiența testării prin evitarea execuțiilor redundante din momentul când o subcategorie a fost vizitată deja. În plus, se potrivește bine cu procesele incrementale, iterativ, și de dezvoltare spirală, în cazul în care noi trepte de software sau subsisteme pot fi tratate ca un nod nou de nivel superior în UMMs, care poate fi adăugat, extins, și testat, în timp ce restul modelului rămâne în esență același. Refolosirea componentelor software sau utilizarea componentelor COTS (comerciale off-the-shelf) poate fi suportată, modelată, și testată în mod similar.

Acoperirea, importanța sau nivelul critic, dimensiunea componentei și complexitatea sa, precum și alte informații pot fi, de asemenea, utilizate în conjuncție cu UMMs pentru a genera cazuri de test. Ca un principiu de bază, toate funcțiile puse în aplicare ar trebui să fie testate cel puțin o dată și să fie constatate satisfăcătoare înainte de a fi lansat produsul [6]. Natura ierarhică a UMMs ne permite să atingem o acoperire de stări și de tranziții pentru lanțuri Markov la nivel superior și la nivele aproape de nivelul superior utilizate frecvent, și mai selectiv pentru alte nivele (low-level) a lanțurilor Markov. Testarea paginilor Web individuale poate fi efectuată de către instrumentele existente pentru testarea HTML.

Preocupările de eficacitate pot cere ca diferite funcții sau componente să fie acoperite cu un număr minim de cazuri de testare. Acest lucru este valabil mai ales atunci când există numeroase suboperațiuni comune între operațiuni diferite de tip end-to-end. Atunci când revizităm anumite stări, repetarea exactă a executării stărilor care au fost vizitate anterior puțin probabil să ne dezvăluie noi probleme. În UMMs prezentate, lanțurile Markov asociate pentru părțile revizuite pot fi utilizate pentru a permite ca căile diferite de nivel inferior să fie acoperite.

Cazurile de testare, de asemenea, trebuie să fie documentate, pentru a ajuta eforturile viitoare de testare, pentru a urmări progresele înregistrate de testare, pentru a reproduce orice alte erori, și a integra strategia de testare în activitățile normale de dezvoltare și de întreținere. Structura ierarhică a UMMs oferă o modalitate ușoară de a documenta o mare parte din aceste informații.

4. Măsurare și analiză folosind log-uri Web

Monitorizarea utilizării Web este o sarcină importantă care trebuie să fie executată pentru a menține aplicațiile Web în stare operațională. Fișierele Log sunt utilizate în mod obișnuit, un click se înregistrează de fiecare dată când un client face o solicitare de a trece pe o pagină. Atunci când există unele probleme asociate cu accesul la Web, informațiile asociate cu aceste probleme, de asemenea, sunt înregistrate în registrele de eroare (error logs). Analiza informațiilor stocate în aceste fișiere din registru poate să ajute la dezvoltarea strategiei de testare statistică eficiente pentru a asigura calitatea aplicațiilor Web.

4.1 Analiza log-urilor de acces și construirea UMM

Prin extragerea și prelucrarea informațiilor din log-uri de acces, pot fi făcute evaluări diferite despre utilizarea Web, inclusiv: timpul, secvența și frecvența documentelor accesate, originile și intensitatea traficului de rețea, etc. Prelucrarea suplimentară este adesea necesară pentru a obține informații dorite din surse de date existente.

Există mai multe instrumente care lucrează cu o varietate de formate de fișiere log utilizate de diferite servere Web (FastStats, Analog etc.). Acestea analizează date în registrele de acces și generează următoarele rapoarte:

- statistici despre vizitatorii, pagini accesate de ei, etc;

- „Hyperlink Tree View” un graf care reprezintă arhitectura site-ului web, precum și informații directe de treceri de la pagini de nivelul înalt de ierarhie la pagini de pe nivele mai joase în ierarhia site-ului.

Se pot construi UMMs folosind aceste rapoarte generate. În primul rând pot fi identificate componentele care alcătuiesc întregul sistem și conexiunile între ele. Informațiile asociate cu click-urile, după prelucrare corespunzătoare, dau posibilitatea de a stabili probabilitățile de branșament pentru a finaliza UMMs.

Cu toate acestea, informațiile prezentate în aceste rapoarte nu pot fi utilizate așa cum sunt în crearea UMMs din cauza unor lipse de informații, precum și diferențele structurale și conținutul între rapoarte și modele diferite. De asemenea, când extragem ierarhiile UMM, trebuie să atragem atenție specială la bucle, și să grupăm pagini sau link-uri, în scopul de a produce UMMs simplificate și ușor de utilizat.

Abordarea aceasta în construcția UMM bazate pe Web log-uri este similară cu utilizarea tehnicilor de data mining pe Web, cu log-uri pentru evaluarea site-ului Web. Cu toate acestea, șabloanele de navigare și frecvențele [7] (extragerea datelor în structuri de tip arbore) formează o colecție eterogenă de rezultate care nu sunt legate cu structura site-ului Web. Accentul se pune pe atingerea obiectivelor de business sau pe intenții de design. În contrast, scopul primar al abordării noastre de măsurare a utilizării Web este de a construi modelele integrate, UMMs, în testarea Web statistică pentru analiza fiabilității.

4.2 Analiza log-urilor de erori și evaluarea fiabilității

Întrebările despre erorile apărute și distribuția lor, precum și de fiabilitate generală a site-ului Web, pot primi răspuns prin analiza log-urilor de eroare și informațiilor asociate. Cu toate acestea, deși există mai multe instrumente de analiză a logurilor Web disponibile astăzi, acestea oferă numai posibilități foarte limitate de analiză a erorilor. Problema este că log-urile de acces conțin doar informații generale despre subseturi de erori raportate în log-urile de eroare. În plus, dacă va depista un număr mare de erori similare într-un jurnal de acces (access log), asta ar însemna că ar putea exista și alte probleme (neînregistrate în jurnalul de acces, dar înregistrate în loguri de eroare) care nu se referă la aplicația sau serverul Web, dar aparute la, sau din cauza ISP (furnizorul de servicii Internet). Prin urmare, analiza jurnalelor de eroare este necesară pentru a ne ajuta să localizăm și să depanăm problemele și pentru asigurarea calității aplicațiilor Web. Pentru aceasta avem nevoie de programe care să extragă, de asemenea, diverse date de utilizare din jurnalele de acces. Astfel datele de utilizare și de eroare pot fi folosite pentru a oferi o evaluare obiectivă a fiabilității aplicației Web, prin ajustarea acestor date la diverse modele de fiabilitate [1].

Folosirea datelor de utilizare și de eroare selectate în mod corespunzător pot fi acordate la diverse modele de fiabilitate pentru a oferi o examinare instantanee a fiabilității curente a aplicației Web. De exemplu, dacă un număr total de erori f este înregistrat pentru un număr n de click-uri, fiabilitatea R , conform modelului Nelson [8] poate fi obținută ca:

$$R = 1 - \frac{f}{n} = \frac{n-f}{n}$$

Când timpul de utilizare t_i este disponibil pentru fiecare click i , măsura sumară a fiabilității și timpul mediu între eșecuri (MTBF mean-time-between-failures), poate fi calculată ca:

$$\text{MTBF} = \frac{1}{f} \sum_i t_i$$

Când timpul de utilizare t_i nu este disponibil, putem folosi numărul de click-uri ca măsură estimativă de timp. În acest caz:

$$\text{MTBF} = \frac{n}{f}$$

În cazul în care eșecurile descoperite sunt fixate pe parcursul perioadei de observare, atunci creșterea fiabilității ca urmare a îndepărtării eșecului poate fi analizată prin utilizarea a diverselor modele software de

creșterea fiabilității (SRGMs Software Reliability Growth Models) [1]. De exemplu, în modelul Goel-Okumoto, unul dintre cele mai utilizate SRGMs, procesul de sosire a eșecului este presupus de a fi un proces neomogen de tip Poisson (NHPP - nonhomogeneous Poisson process).

Numărul cumulativ a eșecurilor care sunt de așteptat, (funcția de medie $m(t)$ în NHPP), în timp t este dat de formula:

$$m(t) = N(1 - e^{-bt})$$

în care constantele N (numărul total de eșecuri în sistem) și b (model curbura) trebuie să fi estimate din datele observaționale.

Atunci când informațiile referitor la fiabilitate pot fi obținute pentru fiecare stare sau tranzițiile pentru UMMs, fiabilitatea totală poate fi calculată pe baza acestor informații și a probabilităților a stărilor de tranziție. Avantajul acestei metode este mai multă flexibilitate pentru a extrapola rezultatele analizei fiabilității la situații diferite de utilizare sau medii. O evaluare cantitativă poate fi efectuată de către diverse modele analitice pentru lanțuri Markov [5].

5. Concluzii

Așa cum a fost menționat, diverse tehnici tradiționale de testare pot lucra numai pentru produsele de dimensiuni mici sau să furnizeze acoperiri aproximative în locul acoperirilor detaliate la nivelul de top pentru produse mari. Pentru orice site Web de dimensiuni rezonabile, testarea selectivă bazată pe scenarii și frecvențe de utilizare în loc de testarea exhaustivă este mai acceptabilă. Deși multe defecte pot fi descoperite, multe dintre ele sunt mai puțin probabil să fie observate de către clienții finali, contribuind astfel la îmbunătățirea nesemnificativă a fiabilității și viceversa. Rezultatul net este că utilizarea testării statistice, în general, rezultă în obținerea produsului mai fiabil, la sfârșitul testării în comparație cu metodele tradiționale de testare exhaustivă.

Bibliografie

1. J.D. Musa, *Software Reliability Engineering*, New York: McGraw-Hill, 1998.
2. J.A. Whittaker and M.G. Thomason, *A Markov Chain Model for Statistical Software Testing*, IEEE Trans. Software Eng., vol. 20, no. 10, pp. 812-824, Oct. 1994.
3. J. Tian and E. Lin, *Unified Markov Models for Software Testing, Performance Evaluation, and Reliability Analysis*, Proc. Fourth ISSAT Int'l Conf. Reliability and Quality in Design, Aug. 1998.
4. J. Tian and A. Nguyen, *Statistical Web Testing and Reliability Analysis*, Proc. Ninth Int'l Conf. Software Quality, pp. 263-274, Oct. 1999.
5. S. Karlin and H.M. Taylor, *A First Course in Stochastic Processes*, second ed. New York: Academic Press, 1975
6. B. Beizer, *Software Testing Techniques*, second ed. Boston, Mass.: Int'l Thomson Computer Press, 1990.
7. M. Spiliopoulou, *Web Usage Mining for Web Site Evaluation*, Comm. ACM, vol. 43, no. 8, pp. 127-134, Aug. 2000.
8. E. Nelson, *Estimating Software Reliability from Test Data*, Microelectronics and Reliability, vol. 17, no. 1, pp. 67-73, 1978.
9. Chaitanya Kallepalli and Jeff Tian, *Measuring and Modeling Usage and Reliability for Statistical Web Testing*, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 27, NO. 11, NOVEMBER 2001