

Evaluarea Disponibilităţii Agregate a Sistemelor de Calcul Orientate pe Servicii

Iurie ȚURCANU, Emilian GUȚULEAC, Ion BALMUȘ

Technical University of Moldova

egutuleac@mail.utm.md, balmus@mail.utm.md

Abstract — În lucrare sunt considerate unele aspecte de modelare și evaluare a disponibilității sistemelor de calcul cu arhitecturi orientate pe servicii (SOA) prin rețele Petri stocastice generalizate (RPRS), care descriu interacțiunea și influența dinamică între defectările hardware și a erorilor software. Pentru a simplifica analiza procesului de prelucrare a cererilor de către sistemul SOA și a evalua unii indicatori QoS este folosită metoda de agregare, determinând respectiv rata echivalentă de cădere în pană și rata de reparare a sistemului.

Index Terms — agregare, disponibilitate evaluare, modelare, sisteme orientate servicii.

I. INTRODUCERE

Sistemele de calcul cu arhitecturi orientate pe servicii (eng. SOA - Service Oriented Architecture) în timp real actuale cunosc o dezvoltare rapidă, atât sub aspectul complexității sau performanțelor, cât și a ariei de răspândire [1]. Acest tip de sisteme prezintă un grad ridicat de complexitate, datorită numeroaselor interacțiuni dintre componentele acestora, cum sunt: concurența, sincronizarea, partajarea resurselor, așteptare etc. Ca urmare a acestor interacțiuni, atunci când sunt procesate o nouă aplicație sau reconfigurări de calcul, care trebuie prelucrate urgent, în sistem apar conflicte între componente, blocări ale acestora, fenomene de așteptare și incertitudini de funcționare a sistemului.

Caracteristica de bază al unui sistem SOA este timpul de răspuns, care reflectă perioada necesară sistemului pentru a genera un răspuns adecvat la un eveniment extern.

Dintre cerințele de performanță ce asigură Calitatea Serviciilor (eng. QoS – Quality of Service) impuse acestor tip de sisteme se disting două elemente, și anume:

- *Viteza de reacție a la evenimentele din exterior* – Sistemul trebuie să asigure timpi de răspuns care să se încadreze în limite posibile, de cele mai multe ori deosebit de severe.

- *Siguranța în funcționare* - Sistemul trebuie să aibă o fiabilitate, disponibilitate și o securitate deosebită. De asemenea, trebuie să aibă capacitatea de a evita situațiile critice și de a limita consecințele unor defecțiuni grave. După o întrerupere accidentală sistemul trebuie repus cât mai repede în funcțiune.

Principalele metode de evaluare a indicatorilor QoS, utilizabile în diverse etape ale duratei de existență a unui sistem SOA, sunt: *analiza pe baza modelelor matematice, modelele de simulare și măsurarea performanțelor*, inclusiv *experimentarea prototipurilor*. Primele două metode sunt, de obicei, utilizate pentru evaluarea QoS unui nou sistem și pentru studierea gradului de adaptabilitate la cerințele de performanță și cost impuse de diverse clase de aplicații [2]. Măsurarea performanțelor, presupune existența sistemului real, ale cărui caracteristici funcționale trebuie investigate și ameliorate.

O problemă majoră în evaluarea performanțelor sistemelor SOA constă în descrierea fenomenelor stocastice care au loc, cum sunt defectarea și repararea componentelor, variația timpurilor de prelucrare a aplicațiilor. În acest scop, pot fi utilizate tehnici de simulare a funcționării sistemului sau metode de modelare în baza formalismelor de *rețele Petri* și a diferitor tipuri de algebre ale proceselor stocastice [2]. Dezavantajul simulării constă în faptul că ea nu permite verificarea funcțională și obținerea unor rezultate exacte, necesită mult timp și este foarte costisitoare. Mai mult, rezultatele obținute prin simulare sunt valabile numai pentru cazuri particulare.

Modelarea și analiza sistemelor SOA este folosită pentru descrierea funcționării și evaluării unor indicatori ai QoS în stadii de elaborare sau de mentenanță. Mai mult, modelarea este unicul mijloc, care poate fi folosit de conceptor, când el trebuie să aleagă între mai multe opțiuni viabile ce sunt, la rândul lor inaccesibile la experimentare din cauza constrângerilor de cost și timp.

Strategiile de folosire a potențialului de calcul oferit de multitudinea resurselor depind de mulți factori: relațiile de precedență dintre procese, întârzierile de comunicație, frecvența comunicațiilor dintre elemente de prelucrare, strategia de folosire a resurselor specifice aplicației.

Modelele propuse pentru descrierea adecvată a a acestor tipuri de sisteme devin mult mai complicate, iar cercetarea lor implică noi abordări [1].

Interesul nostru s-a îndreptat înspre rețele Petri reconfigurabile stocastice (RPSG) [3] ca model de referință, deoarece acestea constituie un formalism grafic simplu și intuitiv de reprezentare a sistemelor cu evenimente discrete în care au loc fenomene de paralelism, de sincronizare, partajare a resurselor și restructurare în timp real.

În lucrare sunt considerate unele aspecte de modelare și evaluare a disponibilității sistemelor SOA prin rețele RPSG.

II. MODELAREA SISTEMULUI SOA PRIN REȚELE PETRI STOCASTICE GENERALIZATE

Disponibilitatea dinamică exprimă probabilitatea ca sistemul să funcționeze corect la un moment dat, indiferent dacă anterior au avut loc defectări și/sau

reparări. Astfel, pentru creşterea disponibilităţii unei componente se poate acţiona pentru evitarea defectării, pe de o parte, şi reducerea perioadei de remediere în caz de defectare, pe de altă parte.

Pentru modelarea funcţionării sistemului distribuit SOA şi a efectua evaluarea disponibilitatea dinamică a acestuia prin reţele RPSG este necesar de a lua în consideraţie astfel de fenomene cum ar fi: competiţia, sincronizarea, situaţii de conflict, excludere mutuală, aşteptare şi interdependenţa părţilor hardware şi software, deoarece defectarea unor componente ale părţii hardware influenţează direct starea de bună funcţionare a aplicaţiei software, care duce la rezultate incorecte.

În continuare, pentru a facilita înţelegerea de către utilizator a expunerii lucrării date, prezentăm succint unele elemente ale RPSG [2].

În RPSG legătura *cauză-efect* dintre evenimente este redată printr-o mulţime de relaţii de tipul *condiţii-evenimente*, condiţiile fiind asociate cu locaţiile reţelei, iar evenimentele cu tranziţiile respective. În reprezentarea grafică locaţiile P sunt redată prin cerceuleţe, iar tranziţiile T sunt redată prin bare subţiri sau dreptunghiuri negre, $P \cap T = \emptyset$ (vezi Fig.1). Locaţiile pot să conţină un număr întreg pozitiv de jetoane, reprezentate prin puncte negre, determină satisfacerea unei condiţii ce corespunde acestei condiţii. Relaţiile *cauză-efect* dintre condiţiile şi evenimente ce au loc în sistem sunt descrise de arce directe reprezentate prin săgeţi., arce inhibitor şi arce test, care determină logica conexiunilor şi relaţiile *producător-consumator* de resurse, redată de regulile de funcţionare a RPSG [7, 11]. Arcele inhibitorie şi/sau test nu consumă jetoane. Un arc inhibitor este reprezentat printr-o linie cu un cerceuleţ mic la sfârşit, iar un arc test este reprezentat printr-o săgeată cu linie întreruptă. Dacă cardinalitatea unui oarecare arc este egală cu 1, ea nu se menţionează explicit. Mulţimea tranziţiilor T este partiţionată în două clase diferite: mulţimea tranziţiilor *imEDIATE* T_0 şi a tranziţiilor *temporizate* T_r , $T = T_0 \cup T_r$ şi $T_0 \cap T_r = \emptyset$. O tranziţie *imediată* are o durată de *reţinere nulă* la declanşarea sa, dacă ea este validată de marcajul curent M , adică *declanşează în mod imediat*. Tranziţiile temporizate au o durată pozitivă de reţinere la declanşare, care este o variabilă aleatorie, distribuită conform legii exponenţial-negative cu rata $\lambda_j(M)$ respectiv de declanşare. Grafic, tranziţiile imediate sunt reprezentate prin bare subţiri, iar cele temporizate – prin bare groase.

Definirea RPSG este justificată prin faptul că într-un sistem de calcul au loc următoarele tipuri de procese:

- *processe rapide*, asociate cu activităţi logice de control pentru alocarea resurselor şi dirijarea interacţiunii evenimentelor de sincronizare şi excludere mutuală etc.;

- *processe lente*, asociate cu activităţi de procesare a datelor, a transferului de date, a ocurenţei evenimentelor de defectare sau depanare ale componentelor sistemului.

În acest caz durată medie $\bar{\tau}_R$ a derulării unui proces rapid este cu mult mai mică în comparaţie cu durată medie $\bar{\tau}_L$ a derulării unui proces lent, ($\bar{\tau}_R \ll \bar{\tau}_L$) şi deci $\bar{\tau}_R$ poate fi neglijată, fiindcă ea nu influenţează indicatorii de performanţă ale sistemului, însă se micşorează considerabil complexitatea analizei unui astfel de model.

Astfel, în RMG tranziţiile imediate reprezintă fie *controlul logic* al interacţiunii proceselor rapide, ce produc stări instabile, fie activităţi, durată cărora poate fi neglijată, ca fiind cu mult mai mică decât durată de declanşare a tranziţiilor temporizate care modelează derularea proceselor lente. Pentru RPSG, regula de declanşare a tranziţiilor validate $T(M)$ de către marcajul curent M se va lua în consideraţie $\Pr i(T_0) > \Pr i(T_r)$ relaţia de ordonare parţială, care determină o prioritate superioară de declanşare a tranziţiei imediate din T_0 faţă de tranziţia temporizată din T_r .

Dacă $T(M)$ conţine numai tranziţii temporizate, ele produc marcaje stabile în care durată de aflare în marcaj este nenulă. Implicit, $T(M_0) \neq \emptyset$. Dacă $T(M)$ conţine tranziţii temporizate şi imediate, atunci numai tranziţiile imediate pot fi declanşate, ceea ce poate produce marcaje instabile, adică durată de aflare în aceste marcaje este nulă.

Modelarea şi evaluarea disponibilităţii dinamice a unui nod al acestui tip de sistem a fost efectuată folosind abordarea descrisă în [2].

Modelul elaborat de RPSG ce descrie funcţionarea sistemului SOA, luând în consideraţie tipurile de defecte şi erori posibile, este prezentat în Fig. 1.

Pentru partea hardware, semnificaţia locaţiilor şi a tranziţiilor acestui model este următoarea:

- p_1 - denotă starea de bună funcţionare a hardware-lui;
- p_2 - apariţia unei disfuncţiuni; p_3 - defect intermitant;
- p_4 - eroarea depistată poate fi corectată de partea hardware; p_5 - defect permanent, hardware necesită lucrări de mentenanţă; p_6 - defectul intermitant devine permanent; p_7 - defectul intermitant se propagă către partea software; p_8 - depistarea tipului de eroare software;

- t_9 - tranziţie temporizată, denotă activitatea de reparare a hardware-lui cu rată medie λ_9 ; t_{10} - tranziţie imediată, denotă evenimentul detectării cu probabilitatea q_{10} a unui defect permanent şi cere reparare întregului hardware; t_{11} - tranziţie imediată, denotă evenimentul detectării cu probabilitatea $1 - q_{10}$ a unui defect intermitant; t_{12} - tranziţie temporizată, denotă cu rată medie λ_{12} ocurenţa unui defect hardware; t_{13} - tranziţie temporizată, denotă activitatea de remediere a defectului permanent cu rată medie λ_{13} ; t_{14} - tranziţie temporizată, denotă cu rată medie λ_{14} activitatea de transformare a defectului intermitant în cel permanent; t_{15} - tranziţie imediată, denotă transformarea cu probabilitatea q_{15} a defectului intermitant în cel permanent; t_{16} - tranziţie imediată, denotă cu probabilitatea $1 - q_{15}$ detectarea defectului intermitant ce poate să fie reparat de hardware; t_{17} , t_{18} - tranziţii temporizate, denotă respectiv cu rată medie λ_{17} ,

λ_{18} activitatea de propagare a defectului intermitent ce afectează partea software.

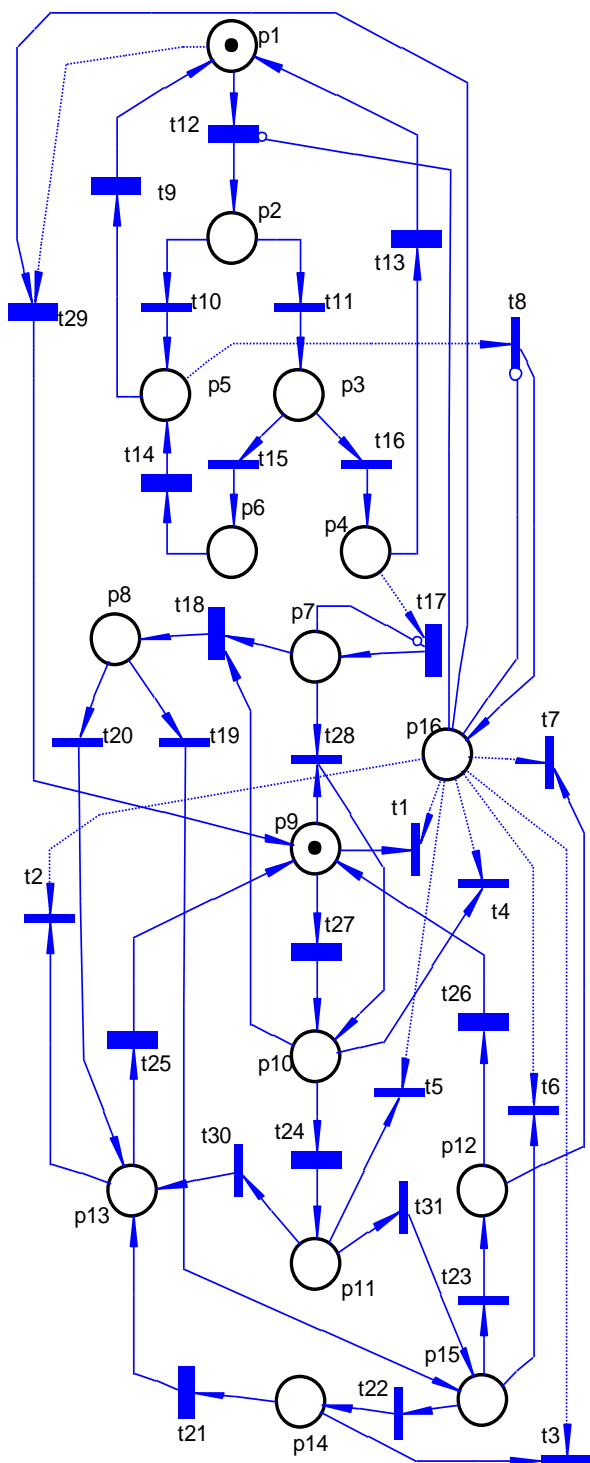


Figura 1. Modelul RPSG1 ce descrie funcționarea sistemului SOA.

Pentru partea software, semnificația locațiilor și a tranzițiilor este următoarea: p_9 - denotă starea de bună funcționare a software-ului; p_{10} - activarea unei erori; p_{11} - sfârșit detectare eroare; p_{12} - eroare intermitentă și ea este înlăturată de partea software; p_{13} - eroarea depistată este permanentă ce necesită resetarea software-ului;

p_{14} - denotă o eroare intermitentă nedetectată, care apoi este detectată; p_{15} - denotă o eroare intermitentă nedetectată; p_{16} - denotă căderea în pană a sistemului;

t_{19} - tranziție imediată, denotă cu probabilitatea q_{19} a defectului intermitent este o eroare permanentă ce necesită resetarea software-ului; t_{20} - tranziție imediată, denotă cu probabilitatea $1-q_{19}$ a defectului intermitent este o eroare intermitentă ce trebuie identificată; t_{21} - tranziție temporizată, denotă cu rata medie λ_{21} activitatea de identificare a erorii intermitente în cea permanentă ce cere resetarea software-ului; t_{22} - tranziție imediată, denotă cu probabilitatea q_{22} că eroarea este identificată ca intermitentă; t_{23} - tranziție imediată, denotă cu probabilitatea $1-q_{22}$ că eroarea intermitentă identificată și ea poate fi autoremediată; t_{24} - tranziție temporizată, denotă cu rata medie λ_{24} identificarea tipului erorii; t_{25} - tranziție temporizată, denotă cu rata medie λ_{12} activitatea de resetare a software-ului; t_{26} - tranziție temporizată, denotă cu rata medie λ_{12} activitatea de autoremediere a erorii software-ului; t_{27} - tranziție temporizată, denotă cu rata medie λ_{27} ocurența unei erori software; t_{28} - tranziție imediată, denotă că tipul erorii nu este identificată; t_{29} - tranziție temporizată, denotă cu rata medie λ_{29} activitatea de depanare, în rezultatul căruia sistemul este în bună funcționare; Tranzițiile imediate $t_1, t_2, t_3, t_4, t_5, t_6, t_7$ și t_8 descriu respectiv ocurențele evenimentelor de acțiune a părții hardware asupra părții software.

Funcționarea modelului RPSG1 este descrisă de către lanțul Markov în timp continuu (LMTC1) inclus [2], care este construit și analizat în mod automat, folosind produsul program instrumental VPNP [4].

S-a constatat că, pentru diferite mărimi ale ratelor de declanșare ale tranzițiilor temporizate și ale probabilităților de selectare a tranzițiilor imediate, cele mai relevante din punct de vedere cele mai mari valori ale probabilităților $\pi_i = \text{Pr ob}(stare_M_\tau = M_i)$ de aflare la momentul τ în stările LMTC1 al modelului RPSG1 din Fig. 1 sunt stările, redate de următoarele marcate:

$$M_0 = (p_1, p_9), M_1 = (p_1, p_{10}), M_2 = (p_1, p_{13}),$$

$$M_3 = (p_1, p_{12}), M_4 = (p_4, p_9), M_5 = (p_1, p_{16}),$$

$$M_6 = (p_5, p_{16}).$$

Starea M_0 denotă faptul că părțile hardware și software se află în bună funcționare. Starea M_2 denotă că părțile hardware și software se află în bună funcționare, iar cea software necesită de a fi restabilită. Starea M_5 denotă că a fost terminată reparația părții hardware, iar cea software necesită de a fi restartată. Starea M_6 denotă că partea

hardware este în pană și este necesară reparația întregului sistem.

În tabelul 1 (respectiv tabelul 2) este prezentat efectul defectării părții hardware (respectiv software) asupra disponibilității sistemului A, funcție de λ_{12} , λ_{27} și de MTTF- durata medie de funcționare până la defectare.

TABELUL 1. EFECTUL DEFECTELOR HARDWARE LA DISPONIBILITATEA SISTEMULUI

MTTF (ore)	λ_{12} ($\times 10^{-7} / s$)	A	π_0	π_3
200	13.80	0.995814	0.995432	0.000112
400	6.95	0.997942	0.997718	0.000095
600	4.65	0.998564	0.998362	0.000093
800	3.50	0.998941	0.998793	0.000085
1000	2.75	0.999078	0.998960	0.000078
1200	2.30	0.999271	0.999132	0.000082
1400	1.95	0.999382	0.999245	0.000084
MTTF (ore)	λ_{12} ($\times 10^{-7} / s$)	A	π_5	π_6
200	13.80	0.995814	0.000301	0.003775
400	6.95	0.997942	0.000152	0.001807
600	4.65	0.998564	0.000103	0.001241
800	3.50	0.998941	0.000072	0.000901
1000	2.75	0.999078	0.000065	0.000771
1200	2.30	0.999271	0.000039	0.000619
1400	1.95	0.999382	0.000047	0.000523

TABELUL 2. EFECTUL ERORILOR SOFTWARE LA DISPONIBILITATEA SISTEMULUI

MTTF (ore)	λ_{27} ($\times 10^{-7} / s$)	A	π_0	π_3
100	27.70	0.999145	0.999143	0.000363
200	13.8	0.999287	0.999106	0.000191
300	9.25	0.999361	0.999264	0.000128
400	6.90	0.999398	0.999296	0.000098
500	5.50	0.999416	0.999317	0.000081
600	4.60	0.999421	0.999339	0.000069
700	4.0	0.999432	0.999351	0.000054
MTTF (ore)	λ_{27} ($\times 10^{-7} / s$)	A	π_5	π_6
100	27.70	0.999145	0.000038	0.000457
200	13.8	0.999287	0.000042	0.000481
300	9.25	0.999361	0.000040	0.000476
400	6.90	0.999398	0.000039	0.000466
500	5.50	0.999416	0.000038	0.000478
600	4.60	0.999421	0.000038	0.000476
700	4.0	0.999432	0.000038	0.000476

Din tabelul 1 și 2 observăm că la descreșterea ratei căderii în pană a părții hardware, disponibilitatea sistemului gradual crește. În același timp probabilitatea erorilor software se micșorează încet până la o mărime stabilă. Cu descreșterea ratei apariției erorilor software, probabilitatea căderii în pană a soft-ului încetul cu încetul, iar disponibilitate sistemului gradual crește.

Pentru a simplifica analiza procesului de procesare a cererilor de către sistemul SOA și a evalua uni indicatori

QoS. vom folosi metoda de agregare, determinând respectiv rata echivalentă de cădere în pană și rata echivalentă de reparație a sistemului, în ipoteza că sistemul poate să se afle numai în două stări: în bună funcționare operațional sau în pană.

În Fig.2 este prezentat modelul RPSG2 simplificat

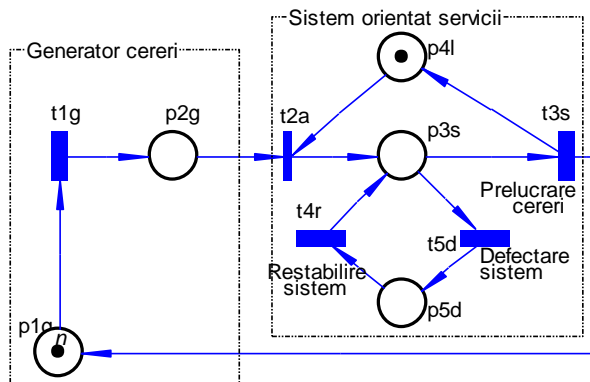


Figura 2. Modelul RPSG2 simplificat de procesare a cererilor de către SOA.

Semnificația locațiilor și a tranzițiilor ale RPSG2 este următoarea:

p_{1g} - populația cerilor în sistem ce trebuie prelucrate;
 p_{2g} - firul de așteptare a cererilor când sistemul este ocupat;
 p_{3s} - sistemul este în bună funcționare operațional și deservește o cerere dacă el este ocupat;
 p_{4l} - sistemul este liber în lipsa cererilor;
 p_{5d} - sistemul este în pană și cererea nu poate fi deservită.

t_{1g} - tranziție temporizată, redă procesul de generare a cererilor către sistem cu rata λ_{1g} ; t_{2a} - tranziție imediată, arbitru ce controlează încărcarea sistemului; t_{3s} - tranziție temporizată, redă activitatea de servire a unei cereri când sistemul este disponibil cu rata λ_{3s} ; t_{4r} sau t_{4r} - sunt tranziții temporizate, ce redau respectiv activitatea de reparare sau de cădere în pană a sistemului cu rata medie echivalentă λ_{4r} sau λ_{5d} , care sunt determinate prin agregarea modelului RPSG1 la 2 stări.

Stările cele mai relevante care reprezintă ocurența defectelor hardware și/sau a erorilor software ale sistemului reparabil, descrise de lanțul LMTC1, sunt $M_3 = (p_1, p_{12})$,

$M_5 = (p_1, p_{16})$ și $M_6 = (p_5, p_{16})$.

Durata medie de aflare în pană a sistemului este:

$$\bar{t}_{4r} = \frac{[\pi_2 + \pi_5] / \lambda_{25} + \pi_6 / \lambda_9}{\pi_2 + \pi_5 + \pi_6} \quad (1)$$

În baza relației (1) pentru modelul agregat din Fig. 1, redat de tranziția temporizată t_{4r} , determinăm mărimea $\lambda_{4r} = 1 / \bar{t}_{4r}$ - rata medie echivalentă de reparare a sistemului (restabilire în bună funcționare).

În conformitate cu teoria sistemelor reparabile [3], rata medie echivalentă λ_{5d} de cădere în pană a sistemului este:

$$\lambda_{5d} = \frac{(1-A)}{A} \lambda_{4r} \quad (2)$$

De exemplu, folosind datele din tabelele 1 și 2 pentru $\lambda_{25} = 3,35 \times 10^{-3} / s$ și $\lambda_{25} = 2,8 \times 10^{-4} / s$ în conformitate cu relațiile (1) și (2) obținem respectiv: $\lambda_{4r} = 5,33 \times 10^{-4} / s$ și $\lambda_{5d} = 2,25 \times 10^{-6} / s$.

Pentru determinarea unor indicatori QoS ai SoA de procesare a cererilor a fost efectuată analiza modelului RPSG2 folosind produsul program VPNP [4].

În Fig.3 este prezentat un fragment al grafului de accesibilitate al modelului RPSG2 cu marcajul inițial $M_0 = (20 p_{1g}, p_{4l})$.

Reachability graph file. Generated by VPNP_ Fig2_QoS.pn2 - Visual Petri Net+ .
Totally found markings number = 41
PositionsList: p1;p2;p3;p4;p5:

- M0 = [20,0,0,1,0] [t1,t2>M1;
- M1 = [19,0,1,0,0] [t1>M2;t3>M0;t5>M3;
- M2 = [18,1,1,0,0] [t1>M4;t5>M5;t3,t2>M1;
- M3 = [19,0,0,0,1] [t1>M5;t4>M1;
- M4 = [17,2,1,0,0] [t1>M6;t5>M7;t3,t2>M2;
- M5 = [18,1,0,0,1] [t1>M7;t4>M2;
- M6 = [16,3,1,0,0] [t1>M8;t5>M9;t3,t2>M4;
- M7 = [17,2,0,0,1] [t1>M9;t4>M4;
- M8 = [15,4,1,0,0] [t1>M10;t5>M11;t3,t2>M6;
- M9 = [16,3,0,0,1] [t1>M11;t4>M6;
- M10 = [14,5,1,0,0] [t1>M12;t5>M13;t3,t2>M8;
- M11 = [15,4,0,0,1] [t1>M13;t4>M8;
- M12 = [13,6,1,0,0] [t1>M14;t5>M15;t3,t2>M10;
- M13 = [14,5,0,0,1] [t1>M15;t4>M10;
- M14 = [12,7,1,0,0] [t1>M16;t5>M17;t3,t2>M12;
- M15 = [13,6,0,0,1] [t1>M17;t4>M12;
- M16 = [11,8,1,0,0] [t1>M18;t5>M19;t3,t2>M14;
- M17 = [12,7,0,0,1] [t1>M19;t4>M14;
- M18 = [10,9,1,0,0] [t1>M20;t5>M21;t3,t2>M16;
- M19 = [11,8,0,0,1] [t1>M21;t4>M16;
- M20 = [9,10,1,0,0] [t1>M22;t5>M23;t3,t2>M18;
- M20 = [9,10,1,0,0] [t1>M22;t5>M23;t3,t2>M18;
- M21 = [10,9,0,0,1] [t1>M23;t4>M18;
- M22 = [8,11,1,0,0] [t1>M24;t5>M25;t3,t2>M20;
- M23 = [9,10,0,0,1] [t1>M25;t4>M20;
- M24 = [7,12,1,0,0] [t1>M26;t5>M27;t3,t2>M22;
- M25 = [8,11,0,0,1] [t1>M27;t4>M22;

Figura 3. Fragment al grafului de accesibilitate al modelului RPSG2.

În Fig.4 sunt prezentate dependențe ale umărului mediu de cereri $\bar{M}(p_{2g})$ în așteptare și a duratei medii $\bar{\tau}(p_{2g})$ de așteptare a unei cereri pentru fi deservită funcție de schimbarea mărimilor ratelor de declanșare:

$\lambda_{1g} \in [0,25/s, 1,25/s]$ cu pasul $\Delta\lambda_{1g} = 0,1/s$ și $\lambda_{3s} \in [1/s, 5/s]$ cu pasul $\Delta\lambda_{3s} = 1/s$ cu $\lambda_{4r} = 5,33 \times 10^{-4} / s$ și $\lambda_{5d} = 2,25 \times 10^{-6} / s$.

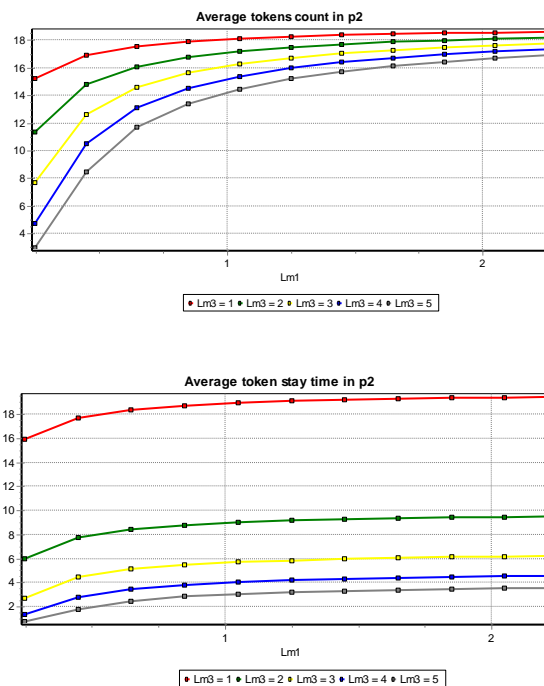


Figura 3. Numărul mediu de cereri $\bar{M}(p_{2g})$ în așteptare și durata medie $\bar{\tau}(p_{2g})$ de așteptare a unei cereri.

III. CONCLUSIE

În lucrare sunt prezentate 2 modele rețele Petri stocastice generalizate(RPRS) care descriu interacțiunea și influența dinamică între defectările hardware și erorile software. Aceste modele sunt folosite pentru evaluarea disponibilității și altor indicatori QoS ai sistemelor de calcul cu arhitecturi orientate pe servicii (SOA). Pentru a simplifica analiza procesului de prelucrare a cererilor de către sistemul SOA și a evalua unii indicatori QoS este folosită metoda de agregare care permite de a determina respectiv rata echivalentă de cădere în pană și rata de reparare a sistemului.

BIBLIOGRAFIE

- [1] M. Bell, Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture. Wiley, 2008.
- [2] E. Guțuleac, Evaluarea performanțelor proceselor de calcul prin rețele Petri generalizate stocastice. Meridian ingineresc, Nr. 1, 2008. Ed.: UTM, Chişinău, p.27-32.
- [3] K.S. Trivedi, Composite performance and dependability analysis. Performance Evaluation nr. 14, 1992, pp. 197-215.
- [4] E. Guțuleac, C. Boşneaga, A. Reilean, VPNP-Software tool for modeling and performance evaluation using generalized stochastic Petri nets. Proceedings of the 6-th International Conference on D&AS2002, 23-25 May 2002, Suceava, România, p. 243-248.