UNIVERSITATEA BABEȘ-BOLYAI
FACULTATEA DE ȘTIINȚE ECONOMICE ȘI GESTIUNEA AFACERILOR
ȘCOALA DOCTORALĂ ȘTIINȚE ECONOMICE ȘI GESTIUNEA AFACERILOR

TEZĂ DE DOCTORAT

-REZUMAT-

Contribuții la disponibilitatea serviciilor și securitatea datelor în cloud computing

Conducător Științific:
Prof. Dr. Nicolae Tomai

Doctorand:
Alexandru Butoi

Cluj-Napoca
2017

Cuprinsul rezumatului

Cuvinte cheie

Cloud computing, sisteme distribuite, toleranța la defecte, arbori de pene, sisteme multi-agent, public cloud, securitatea datelor, confidențialitatea datelor, schemă secretă partajată, protocol de securitate

Cuprinsul tezei de doctorat

Lista publicațiilor proprii

1. Lista publicațiilor referitoare la cuprinsul tezei de doctorat

1. Alexandru Butoi, G. C. Silaghi, *Fault Tree - based service availability model in cloud environments. A Failure Trace Archive approach*, In Economics of Grids, Clouds, Systems, and Services, Lecture Notes in Computer Science, Editors : Altmann, Jorn, Vanmechelen, Kurt, Rana, Omer F., 2016, Springer International Publishing - prezentat la GECON 2016, 20-22 September 2016, Conference held at Harokopio University, Athens, Greece;

2. Alexandru Butoi, A. Stan, G. C. Silaghi, *Autonomous Management of Virtual Machine Failures in IaaS Using Fault Tree Analysis*, In Economics of Grids, Clouds, Systems, and Services, Lecture Notes in Computer Science, Editors : Altmann, Jorn, Vanmechelen, Kurt, Rana, Omer F., 2014, ISSN 978-3-319-14608-9, http://dx.doi.org/10.1007/978-3-319-14609-6_14 , Springer International Publishing – presented at GECON 2014, 16-18 September 2014, Conference held at Cardiff University, UK (2 citări conform Google Scholar la 02.06.2016);

3. Alexandru Butoi, A. Stan, G.C. Silaghi, *Reliable Management of Virtualized Resources Using Fault Trees*, Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International Symposium on , pp.309-316, 22-25 Sept. 2014 doi: 10.1109/SYNASC.2014.49, http://dx.doi.org/10.1109/SYNASC.2014.49 IEEE Conference held at West University in Timisoara, Romania

4. Alexandru Butoi, N. Tomai, *Secret Sharing Scheme for Data Confidentiality Preserving in a Public-Private Hybrid Cloud Storage Approach*, Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on , pp.992-997, 8-11 Dec. 2014, doi: 10.1109/UCC.2014.163, http://dx.doi.org/10.1109/UCC.2014.163 (3 citări conform Google Scholar la 02.06.2016)

5. Alexandru Butoi, M. Moca, N. Tomai, *Data Confidentiality in Cloud Storage Protocol Based on Secret Sharing Scheme: A Brute Force Attack Evaluation*, In Trust Management IX, Vol. 454, 2015, ISBN: 978-3-319-18490-6, IFIP Advances in Information and Communication Technology, Editori Damsgaard Jensen, Christian and Marsh, Stephen and Dimitrakos, Theo and Murayama, Yuko, Springer International Publishing, pp. 177-184, http://dx.doi.org/10.1007/978-3-319-18491-3_13

6. Alexandru Butoi, G.A. Morar, A Ilea, *Two-Phased Protocol For Providing Data Confidentiality in Cloud Storage Environments* .In Proceedings of the BIS 2012 workshops, Lecture Notes in Business Information Systems, vol. 127, Springer, 2012, pp. 220-230, http://dx.doi.org/10.1007/978-3-642-34228-8_21

2. Lista altor publicații relevante din stagiul doctoral

1. Alexandru Butoi, N. Tomai, L. Mocean, *Cloud-Based Mobile Learning*, Informatică Economică, vol. 17, no. 2/2013, pp. 27-40, 2013. ISSN 14531305/17.2.2013.03, http://dx.doi.org/10.12948/issn14531305/17.2.2013.03 (20 citări – Google Scholar la 02.06.2017)

2. Alexandru Butoi, G. C. Silaghi, *A Survey On Security And Legal Issues In Public Cloud Services,* - Proceedings of 13th International Conference on Informatics in Economy, Bucureşti, 2014

3. Nicolae Tomai, Alexandru Butoi, D. Mican, *E-Learning and M-Learning in Cloud,* - Proceedings of 13th International Conference on Informatics in Economy, Bucureşti, 2014

4. Alexandru Butoi, N. Tomai, D. Mican, G. C. Silaghi, *Designing Effective Web-Based M-Learning Systems,* in 12th International Conference on Informatics in Economy, ASE Bucureşti 2013, pp. 126-130, ISSN 2284-7472

Introducere

Noua paradigmă cloud computing a schimbat și continuă să schimbe modul în care computația este livrată și gândită în toate domeniile de la procesarea intensivă de date până la utilizarea acestor tehnologii în mediul de afaceri, instituțional sau personal. R. Buya a văzut ca și finalitate în procesul de adopție al acestei tehnologii un factor decisiv în crearea unei a cincea utilități, alături de cele uzuale ca și apă, gaz, electricitate și telefonie (Rajkumar Buyya, 2009). În prezent suntem tot mai aproape de această finalitate pe măsură ce telefonia mobilă este automat livrată la pachet cu o conexiune de date mobile precum și cu anumite servicii de tip cloud. Adopția acestei tehnologii la nivel organizațional și instituțional implică anumite provocări legate de securitate și disponibilitate a serviciilor: conform unui studiu de referință IDCI printre managerii IT ai unor organizații de top, 74% dintre respondenți au considerat securitatea datelor ca și principală barieră în adopția la scară largă a serviciilor de tip cloud în mediul de business (Subashini & Kavitha, 2011). Externalizarea proceselor de management al sistemului informațional al unei afaceri către un furnizor de servicii cloud implică un risc asociat securității datelor mai ridicat (Dorey & A., 2011). Externalizarea acestor procese la nivel organizațional implică multe provocări, mai ales în domeniul securității, implicând o tranziție către un model de securitate a datelor centrat pe date și mai puțin pe drepturi asupra manipulării datelor atribuite utilizatorilor. Principalul avantaj recunoscut al acestor tehnologii cloud este acela al unei soluții de reducere a costurilor asociate cu stocarea și procesarea volumelor tot mai mari de date pe care organizațiile îl produc. Având în vedere principalele tipuri de livrare a serviciilor cloud, conform clasificării furnizate de către NIST (cloud privat, public și hibrid), cea mai mare reticență întâlnită în adopția acestor servicii în mediul de afaceri este în cazul serviciilor de tip public cloud unde riscul de securitate asociat este cel mai mare, iar nivelul de control al organizației asupra datelor este minim. Problema reticenței adoptării serviciilor de public cloud în mediul organizațional se reduce astfel la o oarecare lipsă de încredere între consumatorul de servicii public cloud și furnizorul de astfel de servicii. Întrebarea care se pune aici este: cum putem să abordăm provocările tehnice și nu numai care generează această lipsă de încredere dintre potențialii consumatori de servicii de cloud și furnizorii de astfel de servicii? Studiul literaturii de specialitate oferă două posibile soluții: Service-level-Agreements (SLA) și anumite caracteristici ale serviciilor de tip public cloud livrate într-o manieră ”by-design”. Adopția și externalizarea infrastructurii și proceselor de prelucrare a datelor în cloud pot avea implicații serioase în continuitatea activității. De exemplu dacă o afacere utilizează un sistem de management și stocare a datelor într-un cloud public, un eveniment de indisponibilitate sau securitate al acestui serviciu va afecta în mod direct activitatea organizației deoarece angajații nu vor mai avea acces la date și fișiere pe perioada în care serviciul de tip cloud este indisponibil. Din punctul de vedere al furnizorului de servicii de cloud, continuitatea activității se traduce în problema disponibilității datelor și funcționalităților asociate serviciului la parametrii optimi de

funcționare, aspecte formalizate ca și angajamente contractuale reunite într-un acord Service Level Agreement.

Din punct de vedere tehnic, serviciile de tip cloud au la bază un conglomerat de tehnologii agregate în mod complex. În acest context, apare o altă întrebare: cum poate un furnizor de servicii cloud să își asume în mod rațional și în cunoștință de cauză anumite angajamente legate de parametrii de funcționare și securitate a serviciilor livrate, având în vedere complexitatea sistemelor și factorilor care pot influența multitudinea de componente și care nu pot fi controlate în totalitate prin simpla natura probabilistică a evenimentelor de pană/defect care pot apărea la nivelul sub-sistemelor? Un posibil răspuns ar putea fi reprezentat de necesitatea introducerii unor mecanisme tehnice de predicție și control al acestor evenimente capabile să acționeze eficient și autonom în cazul în care aceste evenimente de pană apar. Problema poate fi abordată de partea furnizorului de servicii cloud într-o manieră ”availability-by-design”, prin asigurarea unor mecanisme tehnice automate menite să asigure și să păstreze un anumit nivel ușor cuantificabil de disponibilitate a serviciului. Aceste mecanisme pot fi bazate pe tehnici de replicare, migrare, recuperare și balansare a sarcinii. Din punct de vedere al demersului de cercetare, problema se rezumă la modul în care aceste mecanisme tehnice ar putea fi agregate și implementate astfel încât probabilitatea apariției stării de indisponibilitate a serviciului să fie minimizat.

De asemenea disponibilitatea datelor și a serviciilor poate fi în mod direct afectată de apariția unor incidente de securitate care au ca urmări limitarea accesului la date, ștergerea sau alterarea acestora. Dacă în cazul mecanismelor de prevenire a evenimentelor de indisponibilitate, implicarea factorului uman poate fi redusă la minim, în cazul securității datelor implicarea factorului uman reprezintă cea mai mare provocare.

O altă provocare în calea adopției serviciilor de tip public cloud în mediul de business, o reprezintă securitatea datelor și în special problemele legate de confidențialitatea lor, precum și aspecte legale care privesc protecția datelor confidențiale. La fel, aceste aspecte pot fi acoperite într-un acord de tip SLA. Provocarea în acest caz se transformă în următoarea întrebare: în ce manieră putem specifica într-un SLA aceste chestiuni legate de securitatea datelor, și cum putem cuantifica probabilistic nivelul de securitate a datelor angajat de către furnizorul de servicii? Problema poate fi abordată într-o maniera ”by-design” – security-by-design. Această strategie cuprinde procese și protocoale de securitate menite să securizeze automat datele trimise și stocate în cloud, fără vreo intervenție explicită, în scopul minimizării probabilității asociate evenimentului de acces și modificare neautorizată a datelor.

Demersul nostru de cercetare vine să întâmpine aceste două provocări mai sus descrise, și anume ”asigurarea disponibilității serviciilor de tip cloud” și ”securitate datelor” (cu accent pe protecția datelor confidențiale) într-o manieră unificată, urmând o strategie ”by-design” orientată SLA.

Principalele obiective ale cercetării noastre au fost:

1. Crearea un model autonom într-o manieră ”by-design” care să rezolve problemele de disponibilitate ale serviciilor de tip cloud cauzate de erori survenite la nivelul nodurilor.
2. Crearea unei scheme de securitate pentru protejarea datelor confidențiale în medii de stocare de tip public cloud

Principalele premise de la care am pornit au fost:

1. Autonomia în decizii și acțiuni a proceselor tehnice implementate;
2. Modelele să fie ușor de integrat și adaptat în ecosistemul de tip cloud;
3. Disponibilitatea și securitatea datelor sunt considerate caracteristici esențiale ale unui serviciu de tip cloud pentru a putea fi utilizat în condiții optime;
4. Bazându-ne pe existența unor SLA-uri care trebuie respectate, abordarea noastră a fost: soluția se adaptează SLA-ului și nu invers

Lucrarea curentă utilizează concepte din domeniul toleranței la defecte în sisteme distribuite, securitatea datelor și cloud computing, și este poziționată la intersecția a trei domenii de cercetare: managementul defectelor în sisteme distribuite pentru asigurarea disponibilității serviciilor, dezvoltarea modelelor de securitate a datelor cu aplicații directe în domeniul cloud computing. Principalul instrument teoretic de modelare și investigare a problemelor a fost Teoria Arborilor de Pene.

Sumarul capitolelor din teza de doctorat este următorul:

Capitolul 1: *Aplicații ale arborilor de pene în știința calculatoarelor* – prezintă principalele concepte ale teoriei arborilor de pene cu aplicații directe în știința calculatoarelor. Capitolul trece în revistă cele mai reprezentative rezultate recente din domeniu precum și urmărește metodologia de aplicare a conceptelor fundamentale ale teoriei arborilor de pene în medii computaționale.

Capitolul 2: *Disponibilitatea serviciilor și a datelor în cloud și sisteme distribuite* – abordează prima direcție de cercetare a lucrării noastre și analizează rezultatele prezentate în literatura de specialitate recentă împreună cu conceptele și metodele utilizate.

Capitolul 3: *Securitatea datelor și a serviciilor în sisteme de tip cloud* – abordează a doua direcție de cercetare asumată, asigurând o trecere în revistă a principalelor provocări în ceea ce privește securitatea în cloud computing, așa cum sunt ele prezentate în literatura de specialitatea. De asemenea se face un rezumat al principalelor probleme legate de aspectele legale ale stocării și procesării datelor cu caracter confidențial în medii cloud.

Capitolul 4: *Managementul autonom al penelor pentru servicii de tip cloud* - prezintă contribuția originală din domeniul disponibilității serviciilor. În acest capitol am

construit un model de agent capabil să anticipeze viitoarele stări de pană ale componentelor virtualizate și să ia decizii autonome de replicare sau migrare live în scopul prezervării stării de disponibilitate a serviciului și asigurarea țintei de procent de disponibilitate angajat în SLA. Abordarea este una probabilistică, agentul utilizând arbori de pene în procesul de predicție și de luare a deciziilor autonome.

Capitolul 5: *Protecția datelor confidențiale în medii de stocare cloud utilizând arbori de pene* – abordează cea de-a doua parte a contribuției științifice originale în domeniul securității datelor stocate pe infrastructuri de tip cloud. Capitolul introduce un protocol de securitate menit să stocheze în mod securizat datele confidențiale pe o infrastructură de cloud public, protejându-le astfel de acces și modificare neautorizate din partea furnizorului de cloud, venind totodată se preîntâmpine furtul de informație, baze de date sau know-how. Noutatea acestei abordări este faptul că nu utilizează mecanisme criptografice ci se bazează pe o schemă de partajare a secretului compusă din: algoritmi de împărțire a informației și algoritmi de distribuire a informației în cloud astfel încât probabilitatea ca o persoană neautorizată să poată descifra și accesa informația secretă să fie cât mai mică posibil. De asemenea problema este abordată atât probabilistic cât și utilizând arbori de pene.

Capitolul 6: *Concluzii și dezvoltări viitoare* – conține concluziile demersului nostru complex de cercetare, rezultatele obținute și elementele de noutate aduse în domeniile abordate împreună cu posibile dezvoltări ulterioare.

Capitolul 1. Aplicații ale arborilor de pene în știința calculatoarelor

Acest capitol își propune prezentarea principalului instrument de modelare utilizat în demersul nostru științific, și anume Analiza Arborilor de Pene, prin investigarea conceptelor specifice, a metodologiei de aplicare a acestora, a avantajelor și dezavantajelor utilizării acestui instrument în analiza sistemelor distribuite și de tip cloud.

Arborii de pene au fost introduși pentru prima dată în 1961 de către H. A. Watson iar în 1975 Comisia de Reglementare în Domeniul Nuclear a Statelor Unite a adoptat această tehnică de analiză ca și principal instrument în studiile de siguranță a reactoarelor. Aceiași comisie definește *modelul bazat pe arbori de pene* ca fiind un instrument analitic în care sunt descrise stările neconforme de funcționare ale unui sistem și sub-sistem, care apoi sunt analizate în contextul mediului în care sistemul există pentru a identifica toate posibilitățile în care starea neconformă de funcționare a sistemului poate surveni (Haymes Y., 2005). Practic arborii de pene sunt capabili să reprezinte stările de funcționare conformă sau neconformă a unei componente dintr-un sistem și implicit a întregului sistem. Fiecare sub-sistem sau componentă este reprezentat printr-un nod în arbore iar interdependențele dintre noduri sunt reprezentate prin legături de agregare utilizând porți logice (AND/OR) care au ca și ieșiri noi noduri agregate.

În lucrarea de față pornim de la două scenarii de bază utilizate în teoria arborilor de pene și anume: procedura de analiză a unui sistem serial și procedura de analiză a unui sistem paralel.

În cazul **analizei sistemului serial,** axioma de la care se pornește este următoare: *dacă cel puțin un sub-sistem devine nefuncțional, întreg sistemul devine nefuncțional*. Reprezentarea grafică a sistemului este prezentată în Figura 1. În termeni de probabilități, dacă $P(S_1)$ este probabilitatea de pană a sub-sistemului 1 iar $P(S_2)$ este probabilitatea de pană a sub-sistemului 2, probabilitatea ca întregul sistem să se afle într-o stare de pană este $S_1 \cup S_2$ dacă evenimentele S_1 și S_2 sunt independente: $P(S) = P(S_1) + P(S_2) – P(S_1)P(S_2)$.

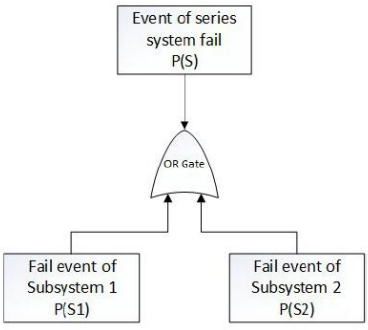

Figura 1 Reprezentare sistem serial

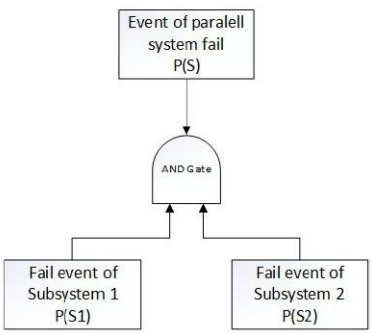

Figura 2 Reprezentare sistem paralel

În cazul **analizei sistemului paralel** axioma de la care se pornește este următoarea: *dacă la același moment de timp t toate sistemele care funcționează în paralel sunt nefuncționale, atunci întreg sistemul este considerat nefuncțional*. Figura 2 prezintă reprezentarea grafică a unui astfel de sistem iar în termeni de probabilități, calculul probabilității de pană a întregului sistem poate fi exprimat ca $P(S) = P(S_1) \bullet P(S_2)$ dacă evenimentele S_1 și S_2 sunt independente.

Bill Vesely definește Analiza Arborilor de Pene ca fiind un proces deductiv, sistematic și stilizat pentru identificarea riscului cu ajutorul unui arbore de pene menit să ilustreze succesiunea logică a unui șir de evenimente nedorite (Vesely, 2016). Metodologia de modelare utilizând arbori de pene urmează ca și etape principale:

1. Definirea clară a evenimentelor și stărilor de pană probabile, precum și definirea clară a limitelor sistemelor și sub-sistemelor;
2. Alegerea setului de instrumente conceptuale și metodologice specifice analizei arborilor de pene potrivite pentru modelarea problemei: fault nodes, porți logice, indicatori probabilistici sau indicatori de robustețe ai sistemului;
3. Necesită definirea clară a spațiului succes-eșec, funcționare normală – funcționare defectuoasă a sistemului;

Același autor, Bill Vesely prezintă și importanța definirii corecte și clare a spațiului succes-eșec, funcționare normală – funcționare defectuoasă.

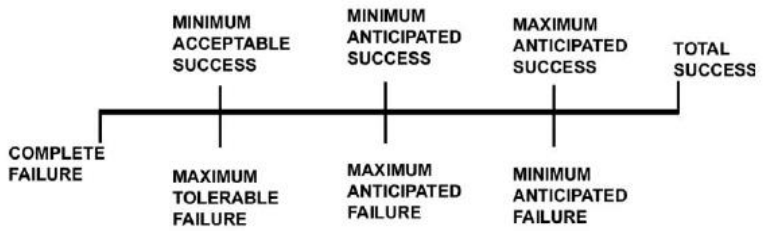

Figura 3 Limitele spațiului funcționare normală - funcționare defectuoasă

Aplicat problemei noastre, legată de disponibilitatea serviciilor cloud, aceste limite se pot extrage direct din specificațiile SLA; de exemplu dacă avem un SLA de 99% disponibilitate a serviciului, rata maximă tolerabilă de indisponibilitate va fi de 1%.

Avantaje ale utilizării analizei arborilor de pene:

1. Reprezentare vizuală a relațiilor cauză-efect în lanțuri complexe de evenimente ce induc stări de nefuncționare în sistem;
2. Aplicabil pentru sisteme de complexitate ridicată
3. Abordare probabilistică, formală, matematic fundamentată
4. Reprezentare arborescentă, poate fi ușor implementată în procese de calcul computerizate

Dezavantaje ale arborilor de pene:

1. Model probabilistic – are prin natură o dimensiune estimativă și nu una exactă
2. Informația incompletă referitoare la evenimentele luate în analiză poate afecta acuratețea estimărilor
3. Utilizarea modelelor de arbori dinamici poate duce la fenomenul de explozie a spațiului de stări generând modele de arbori foarte mari și costisitoare din punct de vedere al procesării;
4. Abstractizarea evenimentelor referitoare la nodurile frunze poate fi dificilă și în funcție de calitatea acestor abstractizări depinde în mare măsură acuratețea estimărilor la nivelul nodurilor agregate superioare;

Studiul literaturii de specialitate arată un efort constant în aplicarea conceptelor teoretice referitoare la arborii de pene în știința calculatoarelor precum:

1. Aplicații teoretice în știința calculatoarelor:
    - Fourneau și Pekergin propun un model de arbore dinamic care vine să înlocuiască lanțurile Markov în analiza tranzițiilor dintre stări cu procese stocastice propunând și un set de algoritmi care să realizeze această decompoziție (Fourneau & Pekergin, 2016);
    - Similar Codetta Raiteri propune un model de conversie a unui arbore de pene dinamic într-o rețea Petri Net stocastică, utilizând transformări specifice grafurilor și evitând astfel problema exploziei spațiului de stări în arbori ce descriu sisteme complexe (Codetta-Raiteri, 2005);
    - Kabir et. al. extind teoria arborilor de pene introducând porți temporale și legi temporale pentru a permite analiza cantitativă în identificarea seturilor de succesiuni minimale în arborii de pene (Kabir, Walker, & Papadopoulos, 2015);
    - Ortmeier si Shelhorn abordează formalizarea arborilor de pene din perspectiva demonstrabilității completitudinii lor, a demonstrabilității că o succesiune de evenimente poate reprezenta un set minimal de stări astfel încât evenimentul de pană generală să se materializeze. De asemenea autorii introduc primul model de arbore de pene cu stări infinite pornind de la teorema setului minimal de stări (Ortmeier & Schellhorn, 2007);
2. Simulări computerizate în domeniul siguranței sistemelor:
    - Cha și Yoo prezintă un protocol de verificare a siguranței reactoarelor nucleare modelat și evaluat prin simulări computerizate (Cha & Yoo, 2012);

- Labib și Read introduc o platformă de simulare hibrid care integrează mai multe instrumente de evaluare a robusteții sistemelor precum FTA, Reliability Block Diagram sau Risk Priority Number, aplicând această platformă în cadrul unei retrospective a dezastrului produs de către uraganul Katrina (Labib & Read, 2015);

3. Sisteme expert:

    - Modelarea arborilor de pene în condițiile utilizării în analiză a informațiilor incomplete este abordată de către Chang și Cheng care propun un model de arbore de pene când informația disponibilă este incompletă (Chang & Cheng, 2009);

    - Arborii de pene de tip fuzzy sunt cei mai potriviți în analiza sistemelor când informația este incompletă (Kabir, Walker, & Papadopoulos, 2016)

4. Studii ale calității software:

    - Tekinerdogan et. al. analizează un sistem software complex din prisma componentelor arhitecturale și a interdependenței dintre ele studiind robustețea sistemului cu ajutorul arborilor de pene; (Tekinerdogan, Sozer, & Aksit, 2008)

5. Securitatea sistemelor informatice:

    - Grunske și Joyce introduc ideea conform căreia riscul de securitate al întregului sistem poate fi estimat pe baza atacurilor modulare și pe structura acestora (Grunske & Joyce, 2008);

    - Brooke și Paige introduc o nouă metodologie legată de modul în care arborii de pene pot fi utilizați ca și instrument de proiectare și implementare a sistemelor critice din punct de vedere al securității (Broke & Paige, 2003);

    - Rushdi si Ba-Rukab studiază problema aplicabilității arborilor de pene în domeniul securității cibernetice utilizând o abordare stocastică pentru estimarea incidentului critic de securitate (Rushdi & Ba-Rukab, 2005);

În ceea ce privește domeniul sistemelor distribuite și mai ales cel al cloud computing, nu am identificat exemple relevante de aplicații ale arborilor de pene. De aici vine și noutatea lucrării de față, care încearcă să aplice teoria arborilor de pene pentru a rezolva probleme complexe din domeniul managementului penelor și securității datelor în sisteme distribuite de tip cloud.

Capitolul 2. Disponibilitatea serviciilor și a datelor în cloud și sisteme distribuite

Acest capitol își propune prezentarea viziunii și soluțiile oferite de literatura recentă de specialitate în ceea ce privește problema disponibilității datelor și serviciilor în cloud computing și nu numai, astfel cuprinzând și domeniul sistemelor distribuite. În opinia noastră, există o relație de incluziune între sistemele de tip cloud și cele distribuite, un sistem de tip cloud fiind implicit proiectat ca și un mediu computațional distribuit pe baza diverselor arhitecturi existente, precum clustere virtuale, server farms, arhitecturi high-performance sau clustere de servere fizice. Specific mediilor de cloud este faptul că puterea de calcul este pusă la dispoziție utilizatorului final de cele mai multe ori în formă virtualizată urmând un model de cost ”pay-as-you-go”, resursele fiind livrate transparent și elastic în diferite forme de abstractizare la nivel de serviciu: IaaS, PaaS, SaaS.

Nabi et. al. definește disponibilitatea unui sistem ca fiind ”o cerință non-funcțională specificată ca procent de timp în care serviciul este disponibil” (Li, Lu, & et. al, 2013). Mai mult această cerință este subiectul a multor SLA-uri în care furnizorul își asumă responsabilitatea livrării serviciilor respective în parametrii specificați.

Studiul literaturii de specialitate definește disponibilitatea serviciilor și o poziționează în contextul SLA-urilor, noi identificând două mari direcții în definirea și studiul disponibilității în domeniul de interes curent:

1. **Abordarea pragmatică.** În abordarea pragmatică disponibilitatea unui sistem este definită de cele mai multe ori ca și o metrică ce exprimă procentual raportul dintre timpul total în care serviciul a fost activ, raportat la un interval de referință în care SLA-ul serviciului a fost activ și a avut semnatari activi. Această abordare este utilizată mai mult de către marii jucători din industrie precum Google, Amazon sau Microsoft.

   Abordarea este de asemenea întâlnită și în literatura academică pe alocuri.

2. **Abordarea probabilistică.** În abordarea probabilistică, disponibilitatea unui sistem este definită ca și ”probabilitatea sistemului de a fi operațional ori de câte ori este nevoie” (Toeroe & Tam, 2012). Această abordare este una mai generală, mai formală și vine să furnizeze estimări viitoare sau trecute legate disponibilitatea unui serviciu. Disponibilitatea în această abordare poate fi exprimată ca și estimare statistică în formă de raport între Timpul Mediu scurs până la pană și Timpul mediu scurs între pene (Batista & et., 2017) sau poate fi exprimată matematic utilizând teoria probabilităților (Undheim, Chilwan, & Heegaard, 2011).

Bauer și Adams, în cartea lor ”Reliability and availability of Cloud Computing” (Bauer & Adams, 2012) tratează pe larg subiectul disponibilității serviciilor în cloud, pornind de la ideea că diferitele pene sunt considerate elemente disruptive în capacitatea serviciilor de a răspunde la cererile primite la un anumit nivel de calitate. O singură eroare apărută în sistem poate conduce la un efect de domino dacă sistemul nu este capabil să trateze aceste erori și să se recupereze în timp scurt. Aceste situații pot avea ca și efect întreruperea parțială sau totală a funcționării serviciului. Impactul unei erori este direct proporțional cu durata de indisponibilitate a serviciului. Autorii identifică 8 ingrediente cheie care pot contribui la producerea penelor în cloud computing: componente hardware, alimentarea cu energie, condiții de mediu, rețeaua internă și canalele de transfer a datelor, componentele software, input de date neadaptate sau neconforme cu cerințele sistemelor de procesare, factorul uman precum și politicile de administrare și securitate. În aceiași carte se trec în revistă principalele trei formule de calcul ale disponibilității unui sistem:

1. formula de bază :

$$Availability = \frac{Uptime}{Uptime + DownTime}$$

2. formula practică:

$$Availability = \frac{ExpectedUptime - DownTime}{ExpectedUptime}$$

3. formula standard :

$$Availability = \frac{AgreedServiceTime - Downtime}{AgreedServiceTime} \cdot 100\%$$

Practic modelul nostru de asigurare a disponibilității serviciilor în cloud propune o abordare probabilistică care include și valorile incluse în SLA.

Capitolul de asemenea investighează literatura recentă din domeniu și punctează cele mai reprezentative abordări aplicabile domeniului de cercetare pe cele două direcții de studiu: abordarea pragmatică și abordarea probabilistică.

Un studiu interesant este condus de către Tu et. al. (Tu, Xu, Xia, & et. al., 2011) care investighează evoluția indicatorilor de disponibilitate specifice nodurilor într-un sistem distribuit atunci când gradul de replicare variază utilizând o modelare bazată pe grafuri. Se constată astfel că disponibilitatea sistemului se reduce liniar atunci când graful de replicare crește, disponibilitatea crește logaritmic atunci când gradul nodurilor crește și intuitiv crește atunci când gradul de replicare al fiecărui nod din sistem crește.

Mecanismele uzuale pe care tehnologiile cloud le utilizează pentru a asigura disponibilitatea serviciilor angajate în SLA-uri sunt trecute în revistă de către Cheraglou et. al. (Cheraghlou & et. al., 2016) printre care se numără tehnici de redundanță (replicare), politici de toleranță la defecte (reactive și pro-active), tehnici de balansare a sarcinii. Un aspect interesant îl reprezintă preferința diverselor proiecte de cercetare și nu numai către arhitecturile reactive, din simplul fapt că aceste tehnici sunt mai

simplu de aplicat și nu întâmpină provocarea de a anticipa posibilele pene ci doar au rolul de a lua măsurile necesare atunci când intervine un eveniment de indisponibilitate, în scopul minimizării timpului în care serviciul va fi indisponibil. Arhitecturile pro-active necesită implementări mai complexe menite să prevadă sau să anticipeze viitoare stări de indisponibilitate ale sistemului și sub-sistemelor acestuia. De asemenea am identificat câteva exemple de astfel de proiecte care implementează cele două tipuri de arhitecturi de toleranță la defecte:

1. *Arhitecturi Proactive:* MapReduce, FT-Cloud
2. *Arhitecturi Reactive:* Haproxy, BFT-Cloud, MPI, FTM, Magi-Cube, Candy

Modelul nostru de toleranță la defecte bazat pe arbori de pene combină cele două strategii proactivă și reactivă deoarece în mod continuu și autonom, pe baza principiilor teoriei arborilor de pene, estimează probabilitatea ca un eveniment de indisponibilitate să aibă loc în viitor, iar apoi pe baza acestor estimări probabilistice ia decizii autonome de replicare și migrare în scopul evitării evenimentului nedorit de indisponibilitate.

Nabi et. al. (Nabi, Toeroe, & Khendek, 2016) face o comparație între perspectiva academică și perspectiva industriei cloud computing asupra problemei disponibilității serviciilor cloud. De exemplu literatura academică exprimă disponibilitatea unui serviciu fie prin raportul dintre up-time și downtime incluzând sau excluzând intervalele de mentenanță (abordarea pragmatică), fie prin calcul de probabilități (abordarea probabilistică).

În ceea ce privește marii jucători din piața furnizorilor de servicii cloud avem:

1. Amazon are ca interval de raportare luna calendaristică iar calculul disponibilității diferă în funcție de tipul serviciului, în general disponibilitatea fiind calculată ca o diferență între 100% și procentul de indisponibilitate a serviciului în minute din luna de referință;
2. Google contabilizează doar perioadele de indisponibilitate, o astfel de fereastră de indisponibilitate este luată în considerare doar dacă depășește 5 minute consecutive.
3. Microsoft are o abordare mai riguroasă în calculul indicatorilor proprii de disponibilitate a serviciului, raportându-se la ferestrele maxime și minime de disponibilitate atinse. În general disponibilitatea este procentul de timp din timpul total al perioadei de referință în care sistemul este capabil să opereze la parametri optimi.

Capitolul 3. Securitatea datelor și a serviciilor în medii cloud

Studiul securității unui sistem începe prin identificarea caracteristicilor și comportamentelor specifice acelui sistem, urmată de identificarea principalelor provocări actuale, riscuri și vulnerabilități la care sistemul respectiv este expus. Cloud computing este o agregare de tehnologii software și hardware iar provocările legate de securitatea sistemelor de acest tip sunt prezente la toate nivelele sistemului cloud. Mai mult tehnologia virtualizării stă la baza livrării serviciilor de tip cloud asigurând elasticitatea specifică în alocarea resurselor cu efecte directe asupra izolării la nivel de utilizator și a managementului facil al resurselor . NIST dă o definiție larg acceptată în momentul de față care definește cloud computing ca fiind ”un model computațional ce facilitează ubicuitatea, conveniența și accesul facil la resurse computaționale partajate accesibile prin rețea și la cerere, ce pot fi ușor alocate în mod aproape automat” (Grance & Mell, 2014). Definiția în sine naște din start anumite provocări și întrebări referitoare la securitatea acestor sisteme, provocări legate de ubicuitate, acces prin rețea, resurse computaționale partajate sau ușurința în alocarea resurselor cu minimă interacțiune între client și furnizorul de resurse.

Korshed et. al. identifică multiple provocări și lipsuri cu privire la securitatea actuală a sistemelor cloud utilizând o metodă de clasificare automată pentru a structura vulnerabilitățile identificate. Conform acestui studiu, principalele provocări legate de securitatea acestor sisteme sunt (Khorshed, Ali, & Wasimi, 2012):

1. Elementele de noutate prin care tehnologiile cloud se diferențiază de alte tipuri de sisteme distribuite: virtualizare, resurse partajate, lucrare la cerere și în mod automat sub formă abstractizată a unui serviciu;
2. Atribute cheie ale acestor sisteme cum ar fi: globalizarea, autonomia în funcționare și managementul resurselor și al proceselor interne, caracterul distribuit, diferitele modele de deployment;
3. Diversele roluri ale părților implicate: furnizorii de cloud, brokerii de servicii, furnizorii de tehnologie, consumatorii de servicii cloud și utilizatorii finali;
4. Transparența față de locație, localizarea, calitatea serviciilor, SLA-uri și reglementări legale referitoare la protecția datelor în astfel de infrastructuri.

Putem identifica un ”trio critic” format din trei elemente cheie referitoare la disponibilitatea și securitatea datelor în cloud: **securitatea datelor** afectează în mod direct **disponibilitatea datelor** care la rândul ei afectează în mod direct **disponibilitatea serviciilor** cu implicații serioase în continuitatea operațiilor de business din cadrul organizațiilor care utilizează aceste servicii de cloud. De exemplu, o corupere a unor date sau o ștergere neautorizată a unor date din cauza unui incident de securitate poate induce un comportament defectuos în funcționarea serviciului. În acest caz implicațiile la nivelul operațiilor de business care utilizează acel serviciu de

cloud pot fi multiple pornind de la întârzieri în activitățile uzuale până la implicații legate de creșterea costurilor.

Singh et. al. identifică patru dimensiuni ale securității în sistemele cloud: securitatea componentelor software, securitatea infrastructurii, securitatea mediilor de stocare și securitatea canalelor de transmisie a datelor (Singh & et. al., 2016). Efortul nostru de cercetare în direcția securității sistemelor cloud se focusează pe a treia dimensiune, cea a securității mediilor de stocare cu accent pe confidențialitatea datelor și pe problema insider-ului malițios. Aceiași autori (Singh & et. al., 2016) identifică mai multe vulnerabilități comune ale sistemelor cloud și le clasifică după domenii de interes:

1. Virtualizare: izolarea la nivel de mașină virtuală, monitorizarea la nivel de mașină virtuală;
2. Nivelul aplicație: *disponibilitatea serviciilor* și a aplicațiilor, licențierea, standardizarea;
3. Încrederea între părțile implicate: factorul uman, criminalitate cibernetică, guvernanța asupra datelor;
4. Securizarea clienților: autorizarea accesului, intimitatea, SLA-ul
5. Medii de stocare: anonimizarea datelor, *disponibilitatea datelor*, encriptarea datelor, confidențialitatea datelor, locația fizică a datelor și segregarea în funcție de roluri de acces;
6. Sistemele de operare din spatele serviciilor vin și ele cu anumite vulnerabilități: sistemul de operare instalat în mașina virtuală, sistemul de operare al serverului care găzduiește componentele virtualizate.

Putem observa că problema disponibilității datelor și serviciilor este prezentă atât la nivelul aplicație cât și la nivelul mediilor de stocare întrucât indisponibilitatea datelor poate afecta disponibilitatea serviciilor. Astfel contribuția noastră referitoare la disponibilitatea serviciilor în cloud are două dimensiuni:

1. O dimensiune pur funcțională – disponibilitatea serviciilor de cloud în contextul unor SLA-uri existente și angajate față de clienți, conform cărora serviciul trebuie să fie operațional ori de cate ori clientul are nevoie de el;
2. O dimensiune de securitate – disponibilitatea serviciilor e abordată și privită ca și un risc pentru continuitatea proceselor de business și pentru accesul la date și resurse;

Subashini și Kavitha introduc în 2011 ideea de ”lipsă de încredere” ca fiind principalul obstacol în adopția la scară largă a serviciilor de cloud la nivel organizațional. Datele confidențiale stocate în medii cloud vin cu multiple riscuri, cel mai mare fiind reprezentat de către factorul uman (Subashini & Kavitha, 2011). Această problemă este în general descrisă ca fiind problema ”atacului insider-ului malițios” și este din ce în ce mai pregnantă în contextul serviciilor de cloud computing. În 2015 compania Verizon

a raportat că 55% dintre atacurile asupra datelor confidențiale au fost de tip insider malițios, ridicând problema transformării administratorilor de cloud din persoane oneste în atacatori de tip insider malițios puternici datorită drepturilor privilegiate pe care le au în sistem (Mishra & et. al., 2016).

Practic problema lipsei de încredere și a acestei reticențe referitoare la stocarea și procesarea datelor în cloud din partea consumatorilor de cloud se rezumă la dificultatea garantării în procent de 100% din partea furnizorului de cloud, că va fi capabil 100% să păstreze acele date confidențiale în contextul în care factorul uman nu poate fi eliminat total din ecuație. Soluțiile se direcționează pe două paliere: definirea de SLA-uri specifice pe probleme de securitate – numite și SecSLAs sau abordarea problemei prin strategia security-by-design.

Securitatea sistemele cloud este un domeniu complex și divers de cercetare unde fiecare vulnerabilitate sau provocare trebuie tratată la un nivel detaliat pentru a-i înțelege toate implicațiile în întreg sistemul în scopul găsirii celor mai bune soluții. De asemenea necesită o analiză și la nivel macro pentru a face conexiunile necesare și a identifica potențialele implicații care pot apărea și la alte nivele. O abordare practică în construirea unui model unificat și standardizat în domeniul securității cloud este introdus de Cloud Security Alliance în forma unor specificații pentru definirea unui model de serviciu Security as a Service (SecaaS). Aceste specificații includ nouă categorii și sunt însoțite și de detalii legate de implementarea lor (Fielder & Smith, 2016): (1) Managementul identității și al accesului, (2) Prevenirea pierderilor de date, (3) Securitate Web, (4) Securitate E-mail, (5) Evaluarea securității, (6) Managementul intruziunilor, (7) Securitatea informațiilor și managementul incidentelor, (8) Implementarea encriptării, (9) Continuitatea activității și planul de recuperare în caz de dezastru.

Stocarea și procesarea datelor în cloud nu ridică numai provocări tehnice ci și non-tehnice precum cele de ordin legal referitoare la protecția proprietății asupra datelor sau jurisdicția sub incidența căreia se află datele stocate în funcție de locația lor fizică. Datorită evoluției tehnologice rapide în domeniu, instrumentele legale care să protejeze datele procesate cu aceste tehnologii în diverse colțuri ale lumii nu pot ține pasul. Centrele de date pot fi localizate oriunde în lume iar datele pot fi replicate în diferite locații în câteva secunde. Aceste date sunt purtătoare de drepturi de proprietate și de know-how iar protecția lor este esențială nu numai din punct de vedere tehnic și al securității ci și din punct de vedere legal. Astfel apare următoarea întrebare: ce legi protejează aceste drepturi asupra datelor când acestea sunt stocate și procesate de tehnologii cloud? Eforturi comune de regularizare în domeniu se întreprind la nivel internațional: (1) Data Protection Act 1998 – protecția indivizilor în relație cu orice date ce pot fi utilizate pentru identificarea lor, (2) Clauze de confidențialitate – rezultă de multe ori din relații de natură contractuală, (3) Drepturi referitoare la baze de date și copyright – datele pot fi de asemenea purtătoare de know-how.

Pentru a veni în întâmpinarea acestor probleme, în ultimii ani marii furnizori din domeniu fac eforturi constante în a oferi tot mai mult control utilizatorului în legătură cu alegerea jurisdicției/a locației în care datele sunt stocate și procesate.

Capitolul 4. Managementul autonom al penelor pentru servicii de tip cloud

Acest capitol prezintă contribuția științifică pe direcția de cercetare referitoare la asigurarea disponibilității serviciilor în cloud computing prin managementul autonom al stărilor de pană utilizând arbori de pene. Aplicăm concepte din teoria arborilor de pene într-un mediu care permite virtualizarea utilizând agenți autonomi.

Considerăm un sistem de tip cloud IaaS în care resursele pot fi virtualizate cu ușurință având un număr *n* de noduri și fiecare nod având o instanță a unui serviciu instalată și care necesită un număr minim de resurse de diferite tipuri pentru a funcționa la parametri normali. Considerăm că aceste servicii sunt livrate într-o manieră ”as-a-service” existând un acord SLA între furnizorul de servicii cloud și consumatorul serviciilor. Principalele proprietăți ale sistemului nostru sunt următoarele:

1. Disponibilitatea serviciilor este garantată prin SLA (de exemplu 99% uptime);
2. Fiecare instanță a unui serviciu are capacitatea de a fi replicat sau migrat către un alt nod atunci când e nevoie;

În acest context al replicării, instanța primară este instanța serviciului care deservește în mod normal toate cererile, iar instanța replicată este doar o clonă sincronizată a instanței primare cu scop de backup. Dacă la un moment dat o instanță este migrată pe un alt nod, acesteia i se alocă noi resurse virtualizate, noua instanță este instalată iar starea instanței inițiale este replicată și pentru noua instanță. Diferența dintre o instanță replicată și una migrată este momentul de timp în care fiecare este creată: replica se creează împreună cu instanța primară fiind o clonă sincronizată a acesteia, pe când cea migrată este alocată doar la nevoie în orice moment din ciclul de viață al instanței primare.

Fiecare instanță a unui serviciu în timpul rulării produce log-uri care pot notifica o eroare apărută în sistem, o avertizare sau o informație. Analizând și abstractizând aceste log-uri putem evalua în orice moment starea de bună/rea funcționare a instanței serviciului. Modelul nostru ia în considerare două categorii de log-uri și le abstractizează: erori și non-erori. O eroare survenită poate indica o viitoare stare de indisponibilitate iar o non-eroare poate indica buna funcționare a sistemului.

În acest context și scenariu descris mai sus, ne propunem construirea unui arbore de pene care să accepte ca și input abstractizări probabilistice ale diferitelor log-uri din sistem, și apoi pe baza regulilor de agregare uzuale din analiza arborilor de pene să producă estimări probabilistice ale unor posibile stări de pană la nivel de instanță sau la nivel de sistem. Astfel pentru fiecare instanță, fie ea primară sau replicată îi asociem un agent care folosește intern un arbore de pene asemănător cu cel reprezentat în Figura 4 Modelul - arbore de pene pentru o instață replicată cu posibilități de migrare pentru a raționa. Acest arbore are următoarea structură: nodul N_{pk} corespunde instanței primare, nodul N_{rk} corespunde replicii instanței primare iar nodul M_{rk} corespunde unei posibile

instanțe migrate la un moment dat, nodul R_k nu are corespondent ca și instanță de serviciu el fiind un nod de agregare care evaluează funcționarea strategiei de replicare în evitarea stării de indisponibilitate a serviciului, iar nodul S caracterizează disponibilitatea întregului sistem și la fel ca și nodul R_k nu are corespondent într-o instanță reală. Nodul R_k se numește nod al replicării, pe când M_{rk} se numește nod al migrării iar nodul S se numește nodul general. Nodul M_{rk} evaluează tot probabilistic capacitatea sistemului de a-și migra instanțele și capacitatea sistemului de a aloca resursele necesare unei noi instanțe migrate la un moment dat.

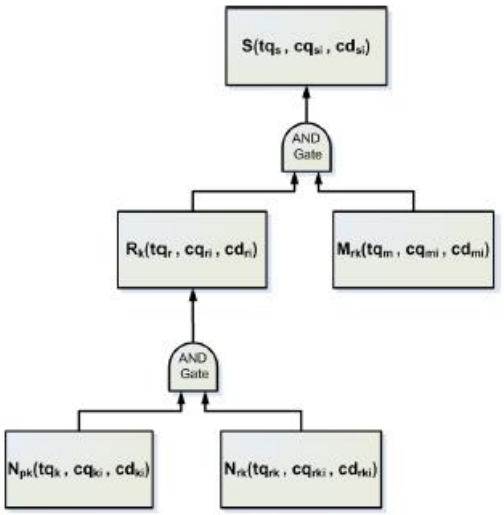

Figura 4 Modelul - arbore de pene pentru o instață replicată cu posibilități de migrare

Fiecare nod din arborele de pene este caracterizat de trei indicatori probabilistici: tq – probabilitatea teoretică de indisponibilitate a nodului, cq – probabilitatea calculată a nodului în urma apariției unui log in sistem la un moment t cu asumpția că evenimentul care a produs log-ul respectiv este independent de anterioarele, cd – probabilitatea calculată a nodului în urma apariției unui log în sistem la un moment t cu asumpția că evenimentul care a produs log-ul respectiv a fost influențat de către anterioarele.

Indicatorii probabilistici teoretici se calculează o singură dată în faza de inițializare a agentului pe baza valorilor de disponibilitate angajate în SLA și rămân constanți pe toată durata de viață a instanței, respectiv agentului. De exemplu dacă pentru un nod (instanță primară sau replica) avem angajat un SLA de 99% disponibilitate, probabilitatea teoretică așteptată ca nodul să devină indisponibil este de (100-99) / 100 = 0.01. Indicatorii teoretici la nivelul nodurilor agregate R și S se calculează utilizând operațiile uzuale impuse de arborii de pene. Probabilitatea teoretică aferentă nodului migrării M se calculează indirect cu ajutorului SLA-ului angajat și a probabilității teoretice aferente nodului replicării R: $ReplicationLevel^2(1 - Q_{SLA})$;

De fiecare dată când un log apare în sistem se creează o abstractizare probabilistică de forma E(ip, dp) a evenimentului, care actualizează indicatorii probabilistici calculați ai nodului corespunzător instanței care a notificat log-ul respectiv. ip este probabilitatea ca evenimentul notificat să inducă o stare de indisponibilitate la nivelul nodului pornind

de la asumpția că evenimentul este independent de cele anterioare. dp este probabilitatea ca evenimentul notificat să inducă o stare de indisponibilitate la nivelul nodului pornind de la asumpția că acesta este cauzat de evenimentele anterioare (evenimente în lanț). Odată actualizați indicatorii unui nod, se impune recalcularea întregului arbore de pene conform regulilor de agregare impuse de teoria arborilor de pene:

- Dacă la nivelul nodului primar sau al replicii apare un eveniment E(ip, dp) indicatorii nodului R_k se recalculează: $cq_{r_i} = cq_{k_i} \cdot cq_{r_{k_i}}$; $cd_{r_i} = cd_{k_i} \cdot cd_{r_{k_i}}$

- Dacă în sistem are loc un eveniment care poate afecta capacitatea sistemului de a migra instanțele sau de a aloca resursele necesare creării unei noi instanțe apare un eveniment E(ip, dp) la nivelul nodului migrării, iar indicatorii probabilistici se modifică. Totodată se recalculează și probabilitățile de la nivelul nodului S: $cq_{s_i} = cq_{r_i} \cdot cq_{m_i}$; $cd_{s_i} = cd_{r_i} \cdot cd_{m_i}$

De fiecare dată când un eveniment cauzează actualizarea arborelui de pene, agentul trebuie să ia o decizie cu privire la sistem prin compararea valorilor calculate cu cele teoretice:

- Dacă la nivelul nodului replicării R_k indicatorii calculați se apropie în proporție de 80% de indicatorul teoretic tq, atunci riscul ca strategia de replicare a sistemului să devină nefuncțională este mare, impunându-se decizia de migrare a instanței cu indicatorii calculați la momentul t la nivel minim, dacă migrarea este fezabilă (în funcție de starea sistemului și de posibilitatea de a aloca resurse suplimentare)

- Dacă la nivelul nodului general S indicatorii calculați se apropie în proporție de 80% de indicatorul teoretic tq, atunci riscul ca strategia de replicare și cea suplimentară de migrare să devină ineficiente, întreg sistemul fiind în pericol să devină indisponibil.

Practic agentul atașat fiecărei instanțe de serviciu are rolul de a monitoriza log-urile generate de respectiva instanță, de a abstractiza probabilistic aceste evenimente și de a actualiza corespunzător nodurile arborelui de pene. Pe baza acestor indicatori probabilistici calculați la nivelul fiecărui nod din arbore, agentul va lua o decizie de transfer a responsabilității către replică (replicare) sau de migrare către o nouă instanță dacă e fezabil (în cazul în care strategia replicării a eșuat).

Noutatea abordării vine să acopere scenariul în care strategia replicării devine ineficientă și ambele instanțe replicate sunt în pericol să devină indisponibile ducând la indisponibilitatea totală a serviciului. În acest caz, ne folosim de elasticitatea alocării resurselor în medii cloud și în loc să pornim din start cu un nivel de replicare de 3 pornim cu un nivel de replicare de 2 iar a treia replică o alocăm prin migrare live doar dacă este nevoie, economisind astfel resurse și ținând costurile sub control.

Rezultate

Modelul a fost de aplicat practic în două scenarii: scenariul în care o instanță rulează independent de restul și scenariul în care o instanță așteaptă rezultatele altei instanțe pentru a fi folosite ca input pentru procesare. Modelul a fost dezvoltat incremental, în primă fază fiind doar un experiment ce primea ca și abstractizări de evenimente generate de un proces aleatoriu, continuând în a-l adapta pentru a accepta ca și input log-uri colectate dintr-un sistem de virtualizare XEN pentru a fi evaluat la nivel IaaS. Modelul a fost ulterior evaluat utilizând log-uri de disponibilitate a nodurilor într-un sistem HPC colectate dintr-un set de date din cunoscuta arhivă Failure Trace Archive.

Pentru a evalua modul de calcul și lucru al agenților de pene am reprezentat evoluția indicatorilor probabilistici împreună cu cei teoretici ca în Figura 5 Evoluția indicatorilor probabilistici la nivelul nodurilor. Linia constantă roșie reprezintă probabilitatea teoretică de indisponibilitatea a nodului pe când linia neagră descrie evoluția indicatorilor calculați de către agenți. De fiecare dată când linia indicatorilor calculați se apropie sau intersectează linia roșie, nodul este la risc de a deveni indisponibil. Observăm de asemenea că probabilitățile își micșorează valorile scăzând riscul de indisponibilitate a nodului. Acest lucru este posibil atunci când după o eroare notificată la nivel de nod, mai multe log-uri non-erori sunt notificate, însemnând că a fost doar o

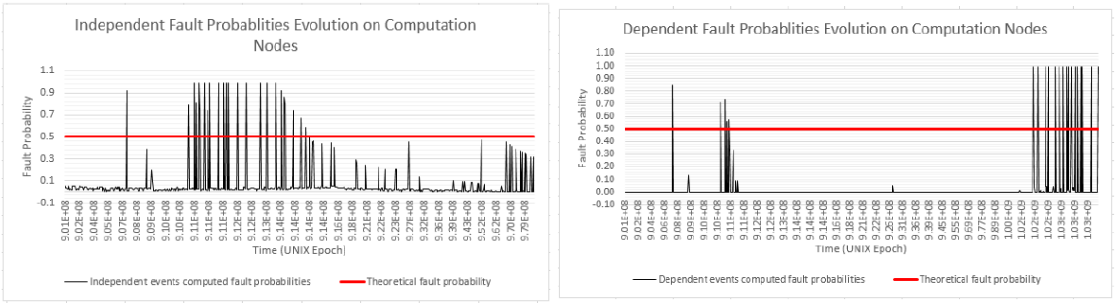

Figura 5 Evoluția indicatorilor probabilistici la nivelul nodurilor

Pentru a studia limitele modelului nostru de toleranță la pene, evaluarea valorilor probabilistice la nivelul nodului general este esențială. Nodul general este de asemenea un indicator foarte bun al capacității sistemului de a-și menține starea de disponibilitate în condițiile angajării strategiei de replicare și migrare descrise mai sus.

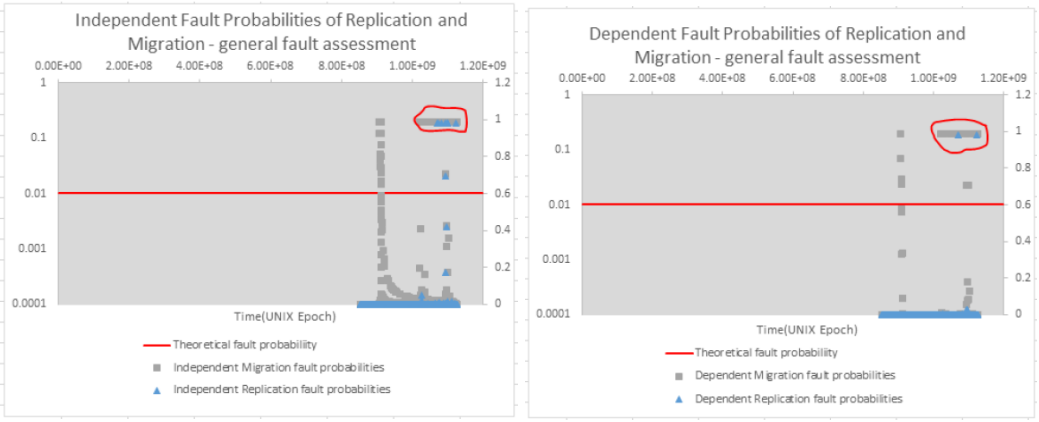

Figura 6 Evaluarea nodului general

Figura 6 Evaluarea nodului generalsuprapune valorile calculate ale nodului replicării și migrării (noduri din a căror agregare rezultă indicatorii nodului general) în care putem observa 4 respectiv 2 cazuri în care valorile indicatorilor probabilistici se suprapun și în același timp depășesc valoare teoretică de pană, în cazul nostru fiind de 0.01 (99% disponibilitate angajată în SLA). Acestea sunt cazurile în care modelul nostru bazat pe arbori de pene nu a fost capabil să mențină disponibilitatea sistemului, strategia replicării și a migrării eșuând. Practic agentul a declarat starea sistemului ca SYSTEM UNRELIABLE în 3 cazuri ca urmare a unor erori succesive, cazuri marcate cu roșu. Un sumar al deciziilor luate de către agent pe durata experimentului cu log-uri din Failure Trace Archive este prezentat în Tabel 1 Sumarul deciziilor agentului:

Decision	**Număr de cazuri**
OK	79 056
REPLICATED	605
REPLICATED TO OK	291
MIGRATED	8
SYSTEM UNRELIABLE	3

Tabel 1 Sumarul deciziilor agentului

În tabelul de mai sus putem observa că agentul a transferat controlul la replică de 605 ori, având și 291 de cazuri în care instanța primară a devenit din nou disponibilă. De asemenea au avut loc și 8 cazuri în care strategia migrării a fost necesară și 3 cazuri în care nici replicarea nici migrarea nu au putut salva situația, probabilitatea de pană calculată la nivelul nodului general fiind mare.

Calculând rata de indisponibilitate a sistemului avem:

TimpTotal = 179 414 800 500 UNIX EPOCHS

TimpTotal Indisponibilitate = 16 941 060 UNIX EPOCHS

Procent Indisponibilitate = 0.01%

Dacă am fi utilizat doar strategia replicării am fi obținut:

TimpTotal Indisponibilitate = 8 484 500 500 UNIX EPOCHS

Procent Indisponibilitate = 4.72%

Capitolul 5. Protecția datelor confidențiale în medii de stocare cloud utilizând arbori de pene

Prezentul capitol prezintă contribuția originală a lucrării pe direcția protecției datelor confidențiale stocate în servicii de tip public cloud. Urmând strategia ”security-by-design” propunem un protocol de securizare a datelor în medii de stocare cloud care să rezolve problema insider-ului malițios, în scopul protejării datelor confidențiale. Protocolul este aplicabil într-un scenariu hibrid care combină o infrastructură privată și una de tip public cloud. Acest protocol nu se bazează pe mecanismele clasice de encriptare a datelor ci pe o schemă de secret partajat formată din două componente: un algoritm pentru împărțirea secretului și un algoritm pentru distribuirea secretului într-un mod cât mai eficient din punct de vedere al securizării datelor. Secretul este considerat a fi sub forma unui fișier ce conține informații confidențiale care necesită protecție. Mecanismul de împărțire a fișierului în părți secrete (numite data chunk-uri) se face utilizând o metrică din teoria informației care asigură ca fiecare dintre chunk-uri conține minim de informație raportat la cantitatea totală de informație conținută în fișierul inițial (secretul). Chunk-urile astfel obținute sunt distribuite și stocate într-o infrastructură de tip public cloud astfel încât probabilitatea ca un atacator să reconstituie secretul să fie minimă.

Scenariul nostru consideră două entități: consumatorul de servicii cloud și furnizorul de servicii cloud. Consumatorul de servicii cloud este entitatea care deține datele ce trebuie securizate și stocate pe infrastructura cloud pusă la dispoziție de către furnizorul de servicii cloud. Furnizorul de servicii cloud pune la dispoziția consumatorului resurse de stocare în forma unor multiple volume virtualizate (V_1, V_2, …, V_n). În scenariul nostru, consumatorul de servicii de stocare deține o infrastructură privată de servere (exemplu un intranet) asupra căruia are control absolut pe când asupra infrastructurii de public cloud are doar control limitat.

Protocolul are două faze, (1) împărțirea fișierului (a secretului) în chunk-uri și (2) distribuirea acestora în mediile de stocare, Figura 7 Protocolul de securizare a datelor confidențialeilustrând modul de funcționare al protocolului propus.

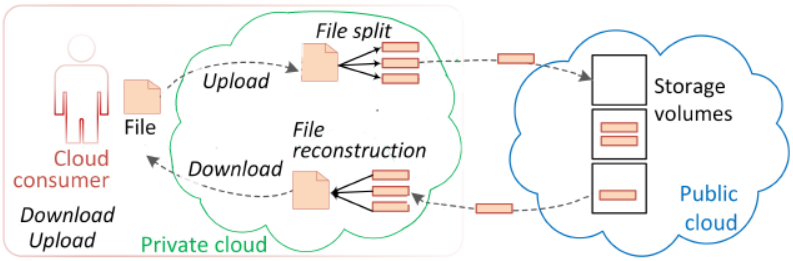

Figura 7 Protocolul de securizare a datelor confidențiale

1. **Faza de Upload** a fișierului are loc în momentul în care utilizatorul dorește stocarea securizată a unui fișier cu informații confidențiale în cloud-ul public.

Înainte de transferul efectiv al datelor către infrastructura externă, fișierul(secretul) este împărțit în segmente mai mici numite chunk-uri (părți secrete) care apoi sunt distribuite pe volumele de stocare în așa manieră încât probabilitatea de reconstrucție a fișierului de către un insider malițios să fie cât mai mică. Practic această fază de upload implementează schema noastră de partajare a secretului. Esențial aici este faptul că procesul de securizare a datelor are loc pe infrastructura privată a utilizatorului/consumatorului. Tot aici se păstrează și un așa-numit Chunk Distribution Dictionary (CDD) care ține evidența distribuirii și stocării fiecărui chunk din fișier pe volumele puse la dispoziție în infrastructura externă, precum și poziția fiecărui chunk în fișierul inițial. Foarte important este faptul că acest dicționar nu este transferat în exterior ci rămâne stocat pe infrastructura internă a consumatorului.

2. **Faza de download** are loc atunci când un utilizator dorește accesarea unui fișier transferat și securizat anterior utilizând strategia de mai sus. Transferul chunk-urilor de pe infrastructura externă în cea privată și reconstituirea fișierului se face pe baza informației stocate în Chunk Distribution Dictionary aflat în infrastructura privată. Astfel doar persoana care deține un CDD corect poate să reconstituie fișierul și să acceseze informația inițială securizată anterior.

Schema de partajare secretă a fișierului este formată din două componente:

a. Un algoritm de împărțire a fișierului în chunk-uri

b. Un algoritm de stocare distribuită a chunk-urilor în infrastructura de stocare externă

Împărțirea fișierului în chunk-uri

În cadrul strategiei de împărțire a fișierului în chunk-uri propunem doi algoritmi care utilizează concepte din teoria informației pentru a împărți fișierul în segmente de informație optime din punct de vedere al protecției datelor confidențiale conținute. Practic ne propunem să implementăm un mecanism de obținere a chunk-urilor astfel încât fiecare chunk să conțină o cantitate minimală de informație raportată la întreaga informație conținută în fișierul original. Dacă considerăm exemplul *John Smith - $200 000*, informația văzută în această formă este completă și putem asociat cu certitudine suma de 200 000 cu persoana John Smith; situația se schimbă atunci când informația este văzută partajat și aparent aleatoriu prezentată de exemplu astfel: *hn | J | o | $2| th |Smi| 00 | 000*.

Pentru a implementa acest mecanism am introdus doi algoritmi care folosesc entropia informației și distanța Kullback-Leibler ca și metrici pentru împărțirea unui fișier în chunk-uri optimale astfel încât fiecare astfel de chunk să stocheze o cantitate minimă de informație raportat la informația totală conținută în fișier. Dacă $I(f)$ este informația medie conținută în fișierul f iar $I(f, c_i)$ este distanța informațională de la chunk-ul c_i la f, algoritmii noștri caută chunk-uri optimale caracterizate de valori ale distanței $I(f, c_i)$ cât mai mari raportate la informația medie conținută în f: $I(f, c_i) >> I(f)$:

$$I(f) = -\sum_{i=1}^{s} P(x_i) log P(x_i); \qquad I(f, c_i) = \sum_{i=1}^{s} P(x_i) log(\frac{P(x_i)}{Q(x_i)});$$

$P(x_i)$ – probabilitatea ca un octet x_i să apară în fișierul f ; $Q(x_i)$ – probabilitatea ca un octet x_i să apară în chunk-ul c_i

Prima versiune a mecanismului de împărțire a fișierului în chunk-uri se numește *Algoritmul de căutare cu limită fixă* deoarece căutarea în spațiul soluțiilor este limitată de constanta STEPSAHEAD. Algoritmul este următorul iar căutarea unui chunk optimal este ilustrată în Figura 8 Căutarea unui chunk optimal:

Algorithm 1 File splitting algorithm with fixed limit search space.

```
1: compute I(f)
2: while f ≠ ∅ do
3:     repeat
4:         c_i = GetBytes(f, MINCHUNK)
5:         compute I_i(f, c_i)
6:     until I_i(f, c_i) > I(f)
7:     f = f − c_i
8:     compute I_i(f, c_i)
9:     maxKL = I_i(f, c_i), countdecline = 0
10:    candidatechunk = c_i
11:    while countdecline ≤ STEPSAHEAD do
12:        b_k = GetBytes(f, MINCHUNK)
13:        c_{i+1} = c_i || b_k
14:        f = f − b_k
15:        compute I_i(f, c_{i+1})
16:        if I_i(f, c_{i+1}) > maxKL then
17:            maxKL = I_i(f, c_{i+1})
18:            candidatechunk = c_{i+1}
19:            countdecline = 0
20:        else
21:            countdecline = countdecline + 1
22:        end if
23:    end while
24:    save candidatechunk candidatechunk = ∅
25: end while
```

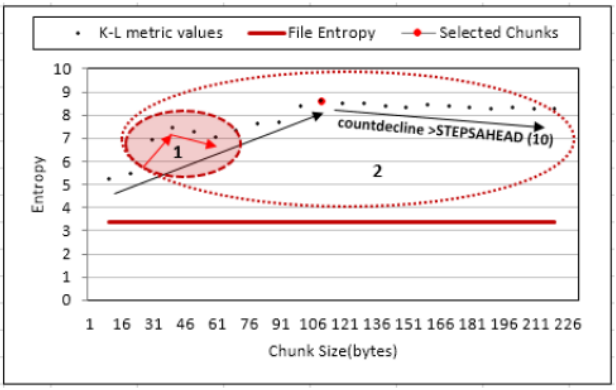

Figura 8 Căutarea unui chunk optimal

Practic lungimea unui chunk optimal din fișierul f este dată de punctul de inflexiune dintre un trend ascendent de valori Kullback-Leibler consecutive și un trend descendent de valori Kullback-Leibler consecutive. Complexitatea algoritmului este $O(n^2)$.

În scopul de a elimina această limitare a spațiului de căutare, am introdus o nouă versiune a algoritmului care folosește limita matematică a metricii Kullback-Leibler pentru limitarea spațiului de căutare a soluțiilor. La fel, algoritmul caută un punct de maxim local în curba de evoluție a valorilor Kullback-Leibler și l-am denumit *Algoritmul de căutare cu limită variabilă* având o complexitate de $O(n^2)$:

Algorithm 2 File splitting algorithm with variable search space.

1: compute $I(f)$
2: $kl_{up} = 0, kl_{down} = 0$
3: **while** $f \neq \emptyset$ **do**
4: **repeat**
5: $c_i = \text{GetBytes}(f, MINCHUNK)$
6: compute $I_i(f, c_i)$
7: **until** $I_i(f, c_i) > I(f)$
8: $f = f - c_i$
9: compute $I_i(f, c_i)$
10: compute $L_{KL}(f, c_i)$
11: $maxKL = I_i(f, c_i)$
12: $limitKL = L_{KL}(f, c_i)$
13: $candidatechunk = c_i$
14: **while** $kl_{down} \leq kl_{up} \cdot \frac{limitKL}{I(f)} AND kl_{down} \geq L_{KL}$ **do**
15: $b_k = \text{GetBytes}(f, 1)$
16: $c_{i+1} = c_i \cup b_k$
17: $f = f - b_k$
18: compute $I_i(f, c_{i+1})$
19: compute $L_{KL}(f, c_{i+1})$
20: **if** $I_i(f, c_{i+1}) \geq maxKL$ **then**
21: $kl_{up} = kl_{up} + (I_i(f, c_{i+1} - maxKL)$
22: $maxKL = I_i(f, c_{i+1})$
23: **else**
24: $kl_{down} = kl_{down} + (maxKL - I_i(f, c_{i+1})$
25: **end if**
26: **end while**
27: save *candidatechunk*
28: $kl_{up} = 0, kl_{down} = 0$
29: $candidatechunk = \emptyset$
30: **end while**

$$L_{KL}(f, c_i) = \sum_{i=1}^{s} \frac{s}{s} \cdot log(\frac{s}{s_{c_i}}) = \sum_{i=1}^{s} log(\frac{s}{s_{c_i}})$$

Distribuirea securizată a fișierului

Având la dispoziție un număr *n* de volume virtuale de stocare în cloud-ul public, și un fișier care conține informații confidențiale, putem proteja această informație împotriva accesului neautorizat prin împărțirea fișierului în *m* chunk-uri și utilizarea unei scheme secrete de partajare a secretului *F(n, m)* pentru a asigura stocarea securizată a fișierului *f* pe infrastructura externă. În cazul nostru, secretul este reprezentat de către fișierul *f*, părțile secretului sunt cele *m* chunk-uri iar părțile care împart secretul sunt reprezentate de volumele de stocare puse la dispoziție de către furnizorul de servicii cloud.

Abordăm problema în două moduri diferite dar urmărind același scop: *distribuirea chunk-urilor într-un mod în care să minimizeze probabilitatea ca un insider malițios să reconstruiască informația din fișierul inițial printr-un efort de brute-force attack.* Prima abordare este una pur probabilistică iar a doua abordare este cea bazată pe utilizarea arborilor de pene.

Abordarea probabilistică de distribuire a chunk-urilor se bazează pe formula experimentelor Bernoulli care în cazul nostru furnizează valoarea probabilității ca un atacator să obțină un singur succes în procesul de reconstrucție a fișierului dintr-un set de chunk-uri dat, t reprezentând numărul de încercări din partea atacatorului:

$$p(1) = \frac{t!}{(t-1)!} P(1-P)^{t-1} = t \cdot P(1-P)^{t-1}$$

Practic algoritmul nostru de distribuție a chunk-urilor va căuta să minimizeze valoarea P(1-P), când t este număr natural t > 0. P reprezintă probabilitatea evenimentului ca atacatorul să găsească setul corect de volume utilizat în stocarea fișierului țintă și să

ghicească toate chunk-urile componente ale fișierului țintă, împreună cu ordinea corectă a acestora în fișier. Algoritmul se rezolvă în timp polinomial $O(n^3)$ și are ca obiective atât minimizarea probabilității de reconstrucție a fișierului original, cât și utilizarea uniformă a resurselor de stocare:

Algorithm 5 Probabilistic distribution approach

```
1: minp = 1 kv = 0
2: ANTESELECTED = ∅
3: VUTILIZATION = ∅
4: while C ≠ ∅ do
5:     Randomly select c_i from C
6:     for each volume V_k ∈ V do
7:         compute P_F(n_c + 1, k_v)
8:         compute P = P_F · (1 − P_F)
9:         if P < minp AND V_k ∉ ANTESELECTED then
10:            minp = P
11:            candidateV = V_k
12:        end if
13:    end for
14:    ENQUEUE(candidateV, ANTESELECTED)
15:    if VUTILIZATION − ANTESELECTED = ∅ then
16:        ANTESELECTED = ∅
17:    end if
18:    if candidateV ∉ VUTILIZATION then
19:        ENQUEUE(candidateV, VUTILIZATION)
20:        kv = kv + 1
21:    end if
22:    store c_i to candidateV
23:    log transaction to local Chunk Distribution Dictionary
24: end while
```

Abordarea distribuției chunk-urilor bazată pe arbori de pene abordează problema prin reprezentarea structurii de chunk-uri extrase din fișier ca un arbore de pene (denumit arborele chunk-urilor), alături de setul de volume de destinație reprezentat sub aceiași formă (arborele volumelor). Practic considerăm că fiecare dintre chunk-urile unui fișier reprezintă un nod în arborele chunk-urilor, iar evenimentul de pană la nivelul fiecărui nod este definit ca evenimentul în care chunk-ul corespondent a fost ghicit și reintegrat corect în fișierul inițial de către un atacator. În mod asemănător, evenimentul de pană asociat fiecărui nod din arborele volumelor va fi evenimentul în care atacatorul a ghicit volumul corect care conține chunk-urile necesare reconstituirii fișierului țintă.

Așa cum Figura 9 Arborele de pene pentru distribuția chunk-urilor ilustrează, strategia de distribuție a chunk-urilor pe volume utilizând arbori de pene se bazează pe interclasarea celor doi arbori astfel încât probabilitatea nodului de pană general în arborele rezultat să fie minimă. Prin încercări succesive repetate, fiecare nod din arborele chunk-urilor va fi transferat ca și nod copil al unui nod din arborele volumelor astfel încât probabilitatea nodului general T_v să fie minimizată. Când unui nod chunk C_i i se găsește un nod părinte V_k care să respecte constrângerea impusă la nivelul nodului rădăcină, chunk-ul corespunzător nodului C_i va fi stocat pe volumul corespunzător nodului V_k.

Nodul rădăcină T_v descrie practic probabilitatea evenimentul ca atacatorul să fi găsit toate chunk-urile fișierului țintă. Arborele folosește ca și porți logice cele de tipul

AND, deoarece reconstrucția completă a fișierului se poate face doar dacă se cunosc toate volumele care stochează toate chunk-urile fișierului *f* iar fișierul nu poate fi reconstruit în totalitate dacă nu se cunosc toate chunk-urile componente.

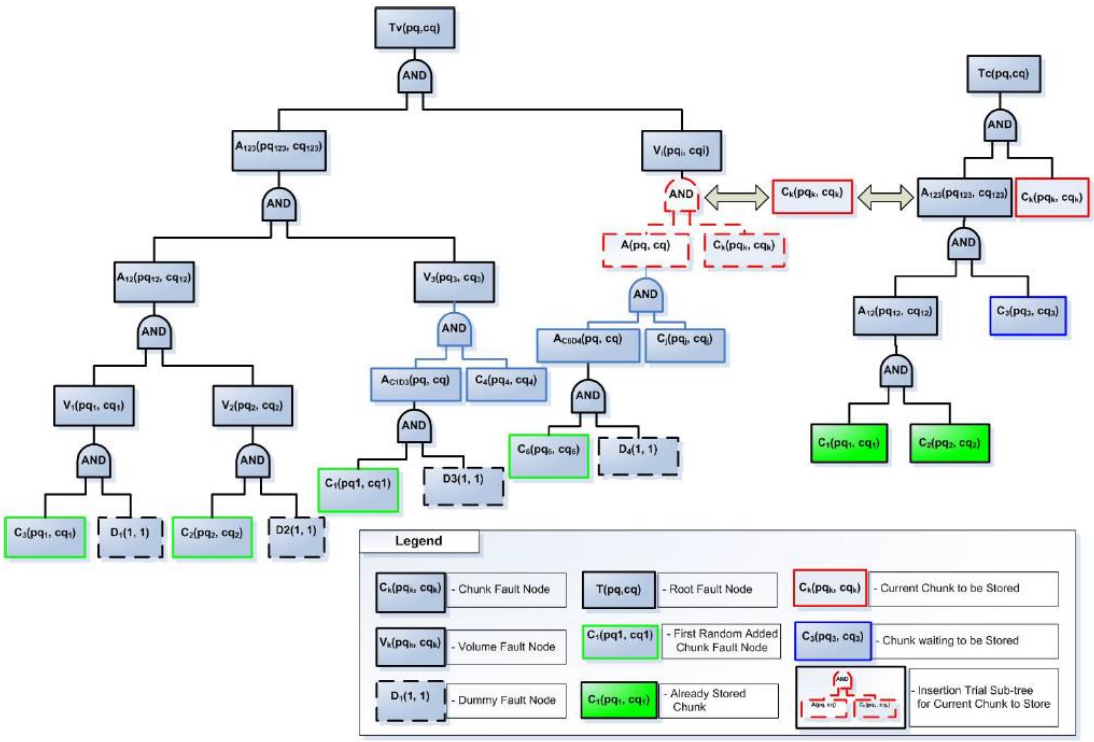

Figura 9 Arborele de pene pentru distribuția *chunk-urilor*

Algoritmul propus este unul mai complex decât în cazul distribuției probabilistice, și se rulează în timp polinomial $O(n^3)$.

Rezultate relevante

Modelul propus a fost implementat în simulatorul CloudSim unde am rulat diferite simulări analizând apoi datele obținute și urmărind patru strategii principale:

1. Evaluarea comparativă a celor doi algoritmi propuși pentru împărțirea fișierului în chunk-uri prin analiza valorilor Kullback-Leibler ale chunk-urilor optimale:

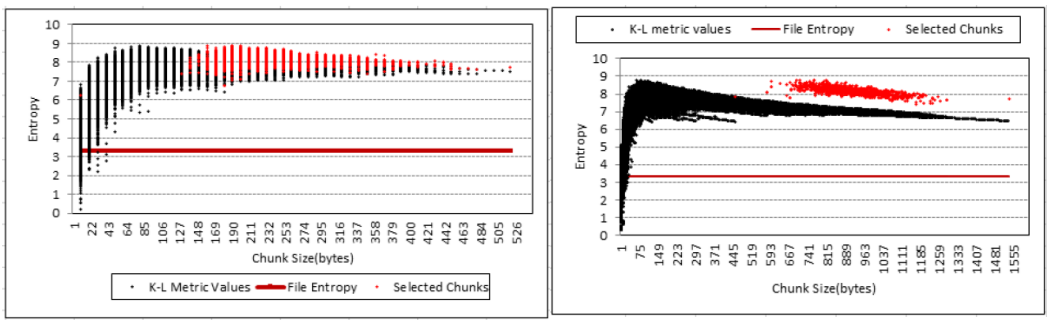

Figura 10 Selecția chunk-urilor optimale - algoritmul de căutare cu limită fixă vs. algoritmul de căutare cu limită variabilă

In Figura 10 Selecția chunk-urilor optimale - algoritmul de căutare cu limită fixă vs. algoritmul de căutare cu limită variabilă primul grafic prezintă valorile Kullback-Leibler obținute în cazul primului

algoritm de împărțire în chunk-uri pe când al doilea grafic prezintă valorile Kullback-Leibler obținute cu cel de-al doilea algoritm pentru cazul unui fișier cu lungimea de 1MB. Comparând cele două grafice putem observa că algoritmul cu limită variabilă oferă valori Kullback mai mari decât în primul caz, iar spațiul de selecție este mai compact decât în cazul celui cu limită fixă. Mai mult, strategia cu limită variabilă produce chunk-uri de dimensiuni mai mari.

Pentru a demonstra utilitatea abordării noastre am implementat un proces aleatoriu de împărțire a fișierului și distribuire a acestora, într-un scenariu identic, în scopul comparării cu rezultatele obținute în cazul celor doi algoritmi. Rezultatele procesului aleatoriu sunt prezentate în Figura 11 Împărțire aleatorie - chunk-uri egale vs. chunk-uri de lungime aleatorie. Primul grafic corespunde cazului când fișierul este împărțit într-un număr aleatoriu de chunk-uri egale, pe când al doilea grafic corespunde cazului în care fiecare chunk are dimensiune aleatorie.

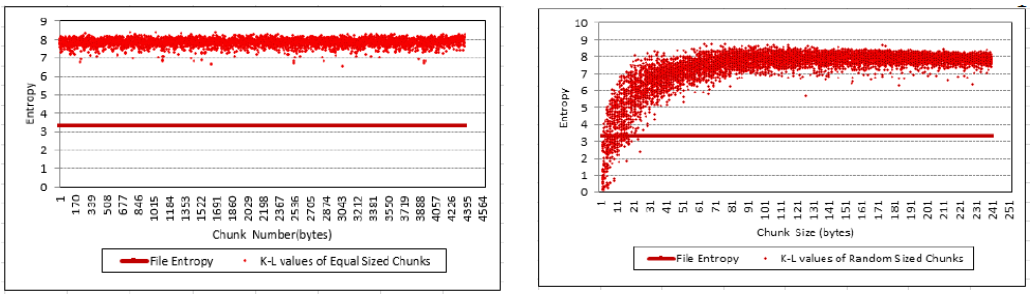

Figura 11 Împărțire aleatorie - chunk-uri egale vs. chunk-uri de lungime aleatorie

Comparând experimentul aleatoriu cu experimentul algoritmilor noștri putem concluziona că algoritmii propuși de noi oferă rezultate mai bune cu valori Kullback puternic grupate și maximale, pe când abordarea aleatorie nu oferă nicio selecție maximală a chunk-urilor.

2. Pentru evaluarea algoritmilor de împărțire în chunk-uri am utilizat distanța Levenshtein calculată între chunk-urile obținute și fișierul original, astfel obținând distanțe Levenshtein mari aflate pe un trend ascendent așa cum se poate observa din Figura 12 Distanța Levenshtein - algoritm cu limită fixă vs. algoritm cu limită variabilăpentru același experiment al fișierului de 1MB.

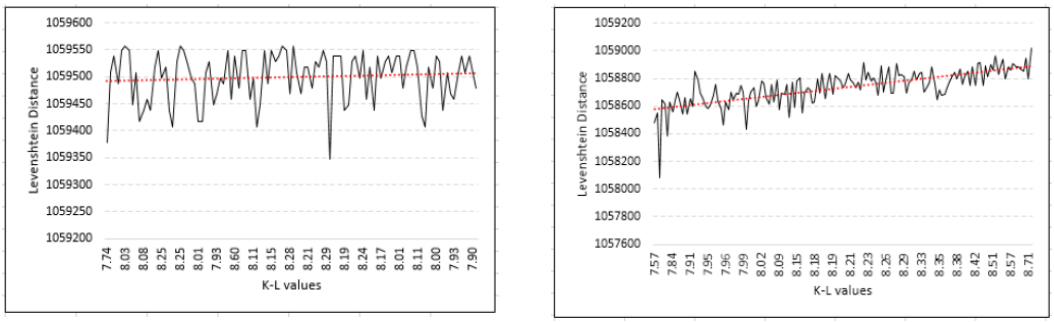

Figura 12 Distanța Levenshtein - algoritm cu limită fixă vs. algoritm cu limită variabilă

3. Evaluarea strategiilor de distribuție a chunk-urilor prin studiul evoluției probabilităților calculate în faza de distribuire a chunk-urilor a demonstrat minimizarea constantă a probabilităților de reconstrucție și o utilizare uniformă a resurselor. Totodată comparat cu procesul aleatoriu de distribuție a chunk-urilor abordările noastre minimizează constant probabilitățile de reconstrucție, pe când abordarea aleatorie nu impune nicio evoluție în acest sens. Figura 13 Evoluția probabilităților - distribuire chunk-uri probabilistică vs. distribuire chunk-uri bazată pe arbori de pene ilustrează evoluția probabilităților în fiecare dintre cele două abordări.

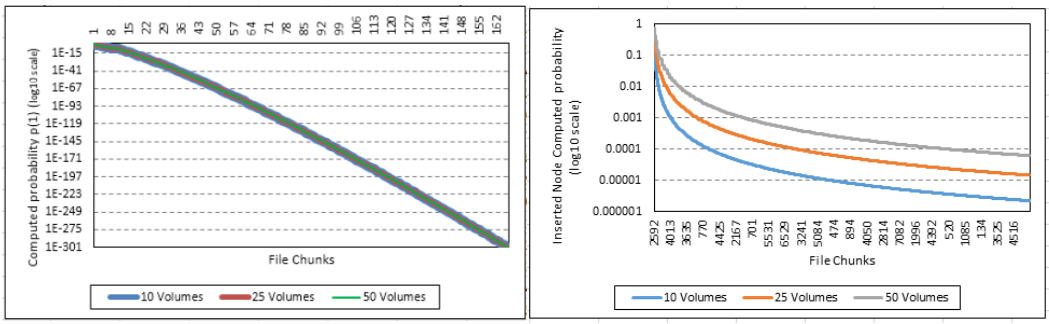

Figura 13 Evoluția probabilităților - distribuire chunk-uri probabilistică vs. distribuire chunk-uri bazată pe arbori de pene

De asemenea Figura 14 Utilizarea resurselor - distribuire chunk-uri probabilistică vs. distribuire chunk-uri bazată pe arbori de pene arată utilizarea uniformă a resurselor în cele două strategii de distribuție.

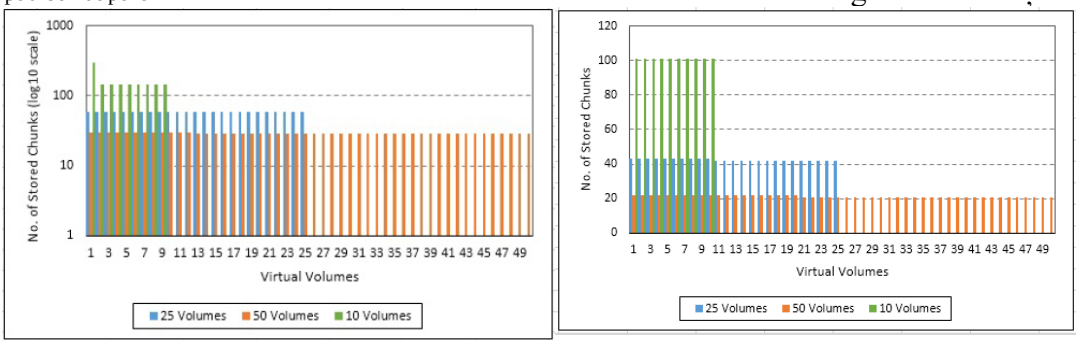

Figura 14 Utilizarea resurselor - distribuire chunk-uri probabilistică vs. distribuire chunk-uri bazată pe arbori de pene

4. Am evaluat tăria protocolului din punctul de vedere al securității aduse printr-o simulare de tip brute-force attack, observând numărul de încercări necesare pentru a reconstitui un fișier pe această cale. Astfel în ambele cazuri am obținut creșteri exponențiale pe măsură ce numărul chunk-urilor crește:

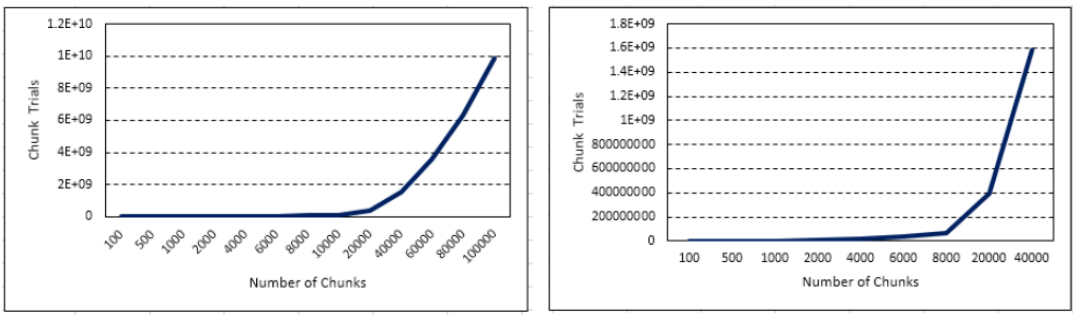

Figura 15 Evoluția încercărilor de reconstrucție prin brute-force - distribuție chunk-uri probabilistică vs. distribuție chunk-uri bazată pe arbori de pene

De asemenea în cazul strategiei de distribuție a chunk-urilor bazată pe arbori de pene (al doilea grafic), în Figura 15 Evoluția încercărilor de reconstrucție prin brute-force - distribuție chunk-uri probabilistică vs. distribuție chunk-uri bazată pe arbori de pene observăm o tendință de creștere mai abruptă a evoluției numărului de încercări ceea ce indică o distribuție mai bună din punct de vedere al securității oferită de strategia bazată pe arbori de pene.

Capitolul 6. Concluzii și dezvoltări ulterioare

Efortul nostru de cercetare a fost concentrat pe două direcții principale, și anume: capitolul 2 și 4 – disponibilitatea serviciilor în cloud, respectiv capitolele 3 și 5 – securitatea datelor în cloud cu accent pe protecția datelor confidențiale. Capitolul 1 s-a concentrat pe introducerea principalelor concepte și metode utilizate din teoria arborilor de pene. Analiza Arborilor de Pene poate fi un instrument puternic de estimare și descriere a riscurilor de pană în sisteme complexe, instrument care în opinia noastră nu a fost utilizat la scară largă în știința calculatoarelor pentru a rezolva probleme complexe precum apariția stărilor de pană sau evaluarea și abordarea riscului de securitate al sistemelor. Una dintre contribuțiile originale ale cercetării noastre este acela al utilizării Analizei Arborilor de Pene în construirea unui model de management autonom al penelor în sisteme de cloud computing, dar și introducerea unui model de securitate ce are la bază o schemă de secret partajat în care părțile secrete sunt distribuite utilizând un mecanism construit pe baza unui arbore de pene, în scopul minimizării probabilității de reconstrucție neautorizată a secretului.

Din perspectiva cerinței de continuitate a proceselor de business relația dintre disponibilitate și securitate este una de incluziune iar unele lucrări de specialitate tratează disponibilitatea serviciilor ca o problemă de securitate. Relația poate fi una reciprocă, deoarece și o slabă securitate a datelor poate reduce disponibilitatea lor în cazul producerii unui incident. Încă de la începuturile tehnologiilor cloud, una dintre barierele în calea adopției pe scară largă a acestora a fost lipsa de încredere dintre consumatori și furnizori, iar soluția propusă a fost o mai bună transparență între cele două părți. În opinia noastră această cerință de transparență este abordată mai mult prin audit și mai puțin prin angajamente SLA. Urmând aceiași idee, modelele propuse de disponibilitate și confidențialitate introduc o abordare ”by-design” a problemei ghidată de o puternică orientare către SLA-uri prin următoarele aspecte:

1. Modelul de management al penelor în cloud utilizează valoarea procentuală de disponibilitate a serviciilor angajată în SLA ca și valoare de referință și utilizează mecanisme de replicare și migrare pentru a se asigura că sistemul va atinge ținta valorilor de disponibilitate impusă prin SLA.

2. Modelul de securizare a datelor confidențiale stocate în cloud este bazat pe strategia ”security-by-design” și este capabil să furnizeze un nivel de securitate a datelor demonstrabil și ușor cuantificabil, nivel care poate fi angajat printr-un SLA care să însoțească integrarea modelului în interacțiunea cu un furnizor de cloud public.

Principalele contribuții originale ale lucrării de față sunt:

1. Am propus un model de toleranță la defecte bazat pe concepte elementare din Analiza Arborilor de Pene aplicabil în sisteme de tip cloud și sisteme distribuite, model care utilizând un arbore de pene implementat computațional are rolul de

a previziona producerea unor viitoare stări de indisponibilitate în sistem. Modelul este implementat de un agent autonom care pe baza evenimentelor produse în sistem, face această predicție și este capabil să ia decizii de replicare sau migrare pentru evitarea evenimentului de indisponibilitate a sistemului. Modelul a fost implementat și evaluat în CloudSim iar agentul folosește ca strategie de evitare a penei totale în sistem utilizarea migrării live atunci când clasica strategie a replicării nu mai este eficientă. Experimentele conduse au demonstrat că modelul nostru este potrivit pentru asigurarea disponibilității serviciilor în cloud și sisteme distribuite.

2. Am introdus un model de protecție a datelor confidențiale pornind de la premisa ”security-by-design” aplicabil într-un scenariu hibrid private-public storage cloud. Modelul este bazat pe o schemă secretă de partajare a secretului având două componente:

    a. O strategie de împărțire a secretului: compusă de doi algoritmi ce au la bază entropia Shannon Fano și distanța Kullback-Leibler iar scopul lor este împărțirea secretului (fișierul) în așa fel încât fiecare parte (chunk) să conțină o cantitate minimală de informație relativ la întreaga informație conținută în secretul inițial;

    b. O strategie de distribuire a secretului: compusă din doi algoritmi, unul pur probabilistic bazat pe experimente Bernoulli, iar al doilea bazat pe arbori de pene care au ca scop distribuirea datelor pe volumele virtualizate de cloud storage astfel încât probabilitatea de reconstrucție neautorizată a secretului să fie minimă.

3. Contribuții relevante la Analiza Arborilor de Pene:

    a. Analiza Arborilor de Pene este în principal utilizată ca un instrument analitic în studii privind robustețea sistemelor pe când noi introducem utilizarea arborilor de pene în construirea unor mecanisme autonome de predicție și luare a deciziilor cu directă aplicabilitate în sisteme distribuite;

    b. În abordarea noastră spațiul stărilor de risc este unul dinamic, stările fiind abstractizate în forma unor indicatori probabilistici ce descriu potențialul unui eveniment apărut în sistem de a induce o stare de pană, evitând astfel problema exploziei spațiului de stări;

    c. În Analiza Arborilor de Pene clasică, tendința este de a lua în considerare doar evenimentele cu impact negativ asupra sistemului pe când modelul nostru ia în considerare și evenimentele cu impact favorabil sistemului, făcând abordarea mai dinamică și mai versatilă.

    d. Am adaptat un arbore clasic de pene pentru a putea fi utilizat în procesul de raționare și luare de decizii autonome al unui agent

e. Având în vedere că abordările bazate pe arbori de pene în știința calculatoarelor sunt relativ răslețe și izolate, am studiat domeniul de referință al securității și disponibilității în sisteme distribuite și de tip cloud în relație directă cu teoria arborilor de pene și aplicabilitatea acesteia în fiecare din domenii amintite într-o manieră cât mai unificată.

Direcții viitoare ale cercetării ar putea fi reprezentate de:

1. O analiză a compromisului de performanță computațională cu impact în procesarea și transferului datelor în cloud atunci când modelele propuse sunt utilizate în astfel de sisteme.
2. Dezvoltarea unui model de cost asociat integrării acestor modele într-un sistem de cloud real pentru a cuantifica mai ușor costul securității și al disponibilității datelor în cloud precum și economia de costuri adusă de implementarea celor două modele.

Bibliografia rezumatului

1. Ortmeier , F., & Schellhorn, G. (2007). Formal fault tree analysis - practical experiences. *Electronic Notes in Theoretical Computer Science*, 139-151.
2. Batista, G. B., & et., a. (2017). A qos-driven approach for cloud computing addressing attributes of performance and security. *Future Generation Computer Systems*, 260-274.
3. Bauer, E., & Adams, R. (2012). *Service Reliability and Service Availability.* Wiley - IEEE Press.
4. Broke, P., & Paige, R. (2003). Fault trees for security system design and analysis. *Computers and Security*, 256 - 264.
5. Cha, S., & Yoo, J. (2012). A safety-focused verification using fault trees. *Future Generation Computer Systems*, 1272 - 1282.
6. Chang, K.-H., & Cheng, C.-H. (2009). novel general approach to evaluating the {PCBA} for components with different membership function. *Applied Soft Computing*, 1044 - 1056.
7. Cheraghlou, M., & et. al. (2016). A survey of fault tolerance architecture in cloud computing. *Journal of Network and Computer Applications*, 81 - 92.
8. Codetta-Raiteri, D. (2005). The conversion of dynamic fault trees to stochastic petri nets, as a case of graph transformation. *Electronic Notes in Theoretical Computer Science*, 45-60.
9. Dorey, P., & A., L. (2011). *Cloud computing a security problem or solution?*
10. Fielder, K., & Smith, C. (2016, Mai 25). *Security as a service working group.* Retrieved from https://cloudsecurityalliance.org/group/security-as-a-service/
11. Fourneau, J.-M., & Pekergin, N. (2016). Dynamic fault trees with rejuvenation: Numerical. *Electronic Notes in Theoretical Computer Science*, 27 - 47.
12. Grance, T., & Mell, P. (2014, February 21). *The nist definition of cloud computing.* Retrieved from NIST: http://dx.doi.org/10.6028/NIST.SP.800-145
13. Grunske, L., & Joyce, D. (2008). Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles. *Journal of Systems and Software*, 1327 - 1345.
14. Haymes Y. (2005). In Haymes Y., *Risk Modeling, Assessment, and Management.* Wiley Series in Systems Engineering and Management.
15. Kabir, S., Walker, M., & Papadopoulos, Y. (2015). Quantitative evaluation of pandora temporal fault trees via petri nets. *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, (pp. 20-37). Paris.
16. Kabir, S., Walker, M., & Papadopoulos, Y. (2016). Fuzzy temporal fault tree analysis of dynamic systems. *International Journal of Approximate*, 20-37.
17. Khorshed, T., Ali, S., & Wasimi, S. (2012). A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. *Future Generation Computer Systems*, 833 - 851.
18. Labib, A., & Read, M. (2015). A hybrid model for learning from failures: The hurricane Katrina disaster. *Expert Systems with Applications*, 7869 - 7881.
19. Li, J. Z., Lu, Q., & et. al. (2013). Improving availability of cloud-based applications through deployment choices. *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference*, (pp. 43-50).
20. Mishra, P., & et. al. (2016). Intrusion detection techniques in cloud environment: A survey. *Journal of Network and Computer*, 18 - 47.

21. Nabi, M., Toeroe, M., & Khendek, F. (2016). Availability in the cloud: State of the art. *Journal of Network and Computer Applications*, 54-67.

22. Rajkumar Buyya, C. S. (2009). Cloud computing and emerging fITg platforms: Vision, hype, and reality for delivering. *Future Generation Computer Systems*, 599 - 616.

23. Rushdi , A., & Ba-Rukab, O. (2005). Fault-tree modelling of computer system security. *International Journal of Computer Mathematics*, 806-809.

24. Singh, S., & et. al. (2016). A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, 200 - 222.

25. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud. *Journal of Network and Computer Applications*, 1-11.

26. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 1-11.

27. Tekinerdogan, B., Sozer, H., & Aksit, M. (2008). Software architecture reliability analysis using failure scenarios. *Journal of Systems and Software*, 558 - 575.

28. Toeroe, M., & Tam, F. (2012). *Service availability: principles and practice.* John Wiley & Sons.

29. Tu, M., Xu, D., Xia, Z., & et. al. (2011). Reach availability modeling of replicated services. *Computer Software and Applications Conference (COMPSAC)* (pp. 688 - 693). IEEE.

30. Undheim, A., Chilwan, A., & Heegaard, P. (2011). Differentiated availability in cloud computing SLAs. *Grid Computing (GRID), 2011 12th IEEE/ACM International Conference* (pp. 126-136). IEEE/ACM.

31. Vesely, B. (2016, February 21). *Fault tree analysis (fta): Concepts and applications*. Retrieved from https://www.hq.nasa.gov/office/codeq/risk/docs/ftacourse.pdf