3. Studiu Bibliografic

1. Introducere in Cloud Computing

În cadrul celei de-a patra revoluții industriale, cunoscută drept ,,4IR’’, tehnologiile informatice au evoluat semnificativ înspre descentralizare și minimizare în ceea ce privește latența [1], ceea ce a condus la apariția a trei paradigme fundamentale de calcul, și anume Cloud Computing, Fog Computing și Edge Computing. Aceste trei modalități de procesare a datelor se deosebesc prin poziționarea lor în arhitectura rețelei, proximitatea față de sursa datelor, respectiv prin capacitatea de a satisface cerințele specifice legate de latență, scalabilitate ori distribuție geografică [2].

Diagram of a cloud network

AI-generated content may be incorrect.



Figura 1. Domeniile de aplicare ale cloud computing-ului, fog computing-ului și edge computing-ului (conform Mahmud et al. [3]).

Piața a experimentat o expansiune rapidă ca urmare a integrării cloud computing-ului ca normă globală pentru procesarea și stocarea datelor. Această evoluție a fost facilitata de accesibilitatea crescândă a dispozitivelor ,,smart’’, fie că vorbim despre telefoane mobile, televizoare inteligente sau ceasuri inteligente [6]. Ca rezultat, s-a înregistrat o cerere semnificativă pentru resursele de rețea și servere dedicate procesării datelor, ceea ce a condus la aparitia unor paradigme inovatoare, precum Edge Computing și Fog Computing. Aceste concepte avansate propun o apropiere a procesării și capacității de stocare de utilizator, fenomen ce contribuie la reducerea latenței și la creșterea disponibilității datelor, formându-se astfel domeniul de cercetare cunoscut sub numele de Edge Computing prin intermediul unei game largi de caracteristici și tehnologii ce contribuie la procesarea și stocarea datelor de la marginea rețelei.

Astfel, paradigma Fog Computing constituie un model stratificat, având ca obiectiv primordial conservarea avantajelor oferite de cloud pentru desfășurarea aplicațiilor sau serviciilor capabile să proceseze și să stocheze date pentru utilizatorii finali în noduri distribuite. Această structură, corelată cu nodurile edge, facilitează diminuarea timpului de răspuns necesar pentru sarcinile sau procesele generate în timpul utilizării.

Dintre cele trei paradigme, cloud computing-ul s-a distins ca fiind modelul cel mai răspândit și cu un grad de maturitate superlativ, fiind utilizat atât în mediul academic, cât și în sectorul privat, în mod implicit instrumentele de simulare[4] dobândind o importanță tot mai mare de-a lungul ultimelor decenii în tot ceea ce înseamnă cercetările dedicate acestei tehnologii aflate într-o continuă dezvoltare. Continuând această idee, prin intermediul proceselor de simulare devine posibilă o evaluare prealabilă a configurațiilor de servicii, precum și a strategiilor de alocare și gestionare a resurselor, ceea ce facilitează îmbunătățirea performanței sistemelor, testarea riguroasă a ipotezelor tehnice și comerciale, respectiv obținerea de rezultate fiabile, ușor de reprodus, cu un timp de execuție bine optimizat, și cu un risc considerabil diminuat printr-un necesar de investiții minime.

În contextul cercetărilor actuale, simularea se prezintă ca fiind metoda predominantă la nivel global, având capacitatea de a configura medii virtuale complexe ce integrează aplicații, sarcini, dispozitive și condiții de rețea specifice. Aceste simulări produc date semnificative referitoare la performanța unei infrastructuri anume, incluzând metrici de tipul utlizării procesorului central (CPU), lățimea de bandă disponibilă și latența, facilităndu-se astfel procedura de luare a unor decizii fundamentate cu privire la implementarea în mediile reale.

1. Platforme de simulare

Nu poate fi omis faptul că există o plajă variată de simulatoare dedicate mediilor de tip Cloud, Fog sau Edge[4], fiecare având propriile trăsături distinctive și funcționalități specifice. Aceste instrumente permit configurarea de diverși parametri sistemici, inclusiv arhitectura rețelei, tipurile de dispozitive implicate, cerințele aplicațiilor ori caracteristicile proceselor în sine, oferindu-se astfel o platformă de testare care se dovedește a fi atât flexibilă, cât și accesibilă. Totuși, trebuie menționat faptul că niciun simulator nu reușește să satisfacă în mod amănunțit toate cerințele posibile ale unui scenariu complex, întrucât această diversitate de instrumente generează un obstacol suplimentar, și anume selecția simulatorului adecvat în funcție de obiectivele specifice pe care cercetarea le urmărește.

În literatura de specialitate, au fost formulate numeroase clasificări și analize comparative ale simulatoarelor, dintre care se remarcă, în primul rând, CloudSim[8], întrucât acesta este cel mai utilizat simulator specific infrastructurilor cloud, concetrându-se pe alocarea eficientă de resurse și pe programarea sarcinilor(scheduling), fiind facilitată analiza performanței în mediile virtualizate. În plus, avem și extensiile special concepute pentru abordarea particularităților mediilor Fog și Edge, și anume iFogSim și YAFS[5], acestea oferind suport pentru topologii dinamice și implicit pentru distribuții geografice, elemente esențiale unei gestionări eficiente a resurselor unor arhitecturi emergente. Alte categorii de simulatoare ce trebuie menționate sunt LEAF și EdgeCloudSim[9], care sunt specializate pe simularea de aplicații de tip Internet of Things sau ,,IoT’’, respectiv evaluării performanței în condiții de latență scăzută și având un rol important în optimizarea interacțiunilor unor dispozitive conectate. Nu în ultimul rând, SimGrid[10] reprezintă un instrument robust pentru simulările distribuite, fiind extins recent prin intermediul inițiativelor precum ENIGMA, având ca scop principal integrarea mobilității și vizualizării geografice, fiind urmărită îmbunătățirea capacității de modelare a sistemelor complexe. Așadar, perspectiva efectuării unor comparații între aceste simulatoare devine un factor de o importanță deosebită atât din punct de vedere tehnic, cât și din punct de vedere practic datorită perspectivei scenariilor reale ce urmează a fi replicate.

În continuarea aspectelor prezentate anterior, tabelul atașat mai jos oferă o sinteză comparativă a simulatoarelor utilizate cel mai frecvent în literatura de specialitate, subliniindu-se paradigma abordată, limbajul de programare aplicat, suportul pentru mobilitate și gestionarea energiei, precum și câteva observații esențiale.

|  |  |  |  |
| --- | --- | --- | --- |
| Simulator | Paradigma | Limbaj | Observatii principale |
| CloudSim[] | Cloud | Java | Simulator clasic, neavând mobilitate dinamică |
| iFogSim[] | Cloud + Fog | Java | Cu topologie fixă, utilizat pentru IoT |
| YAFS | Fog + Edge | Python | Mobilitate reală, având grafuri dinamice |
| LEAF | Edge | Java | Folosit pentru evenimente discrete, IoT distribuit |
| EdgeCloudSim | Cloud + Edge | Java | Simplu, folosit pentru aplicații de tip edge |
| SimGrid | Distribuit | C/C++ | Foarte greu de personalizat, folosit pentru arhitecturile distribuite |
| ENIGMA | Fog + Edge | C/C++ | Are mobilitate IoT, implicit vizualizare geospatială bazată pe SimGrid |

Tabel 1. Comparație între cele mai utilizate simulatoare [5].

În urma unei analize cuprinzătoare a simulatorilor disponibili, am optat pentru utilizarea CloudSim ca fundament pentru simulările prezentate în această lucrare, având în vedere popularitatea, maturitatea și flexibilitatea remarcabile pe care acesta le are. În literatura de specialitate, CloudSim se evidențiază printr-o serie de versiuni care au fost dezvoltate de-a lungul timpului, fiecare dintre acestea aducând îmbunătățiri semnificative și extensii funcționale, și anume: CloudSim, versiunea de bază, CloudSim Plus, care este orientat pe obiect și modular, și nu în ultimul rând, varianta cea mai recentă, CloudSim 7G, extinsă cu funcționalități moderne. //data release in table + paper.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versiune | Anul apariției | Limbaj | Extensibilitate | Funcționalitate |
| CloudSim | 2009 | Java | Bună | Simplu, oferă modelare pentru VMs implicit datacenters |
| CloudSimPlus | 2017 | Java | Ridicată | Imbunătățire la modularitate, ușor de utilizat, cod mai clar si mai bine structurat |
| CloudSim7G | 2023 | Java | Foarte bună | Oferă flexibilitate aplicațiilor distribuite, suportă simularea și managementul rețelelor în cloud, include funcționalități de management al energiei pentru simularea consumului de energie, integrarea IoT, etc. |

Tabel 2.Comparație între simulatoarele CloudSim [7], [11]

Dintre multiplele variante existente ale simulatorului CloudSim, CloudSim 7G a fost selectat pentru această cercetare, întrucât este cea mai sofisticată și cuprinzătoare iterație din această serie de instrumente. Spre deosebire de versiunile anterioare, caracterizate prin module disparate, adesea dificil de integrat și de întreținut, CloudSim 7G introduce o arhitectură reconfigurată și standardizată, capabilă să sprijine simulări complexe și multi-modulare într-o manieră coezivă și eficientă.

Unul dintre cele mai notabile beneficii ale acestei versiuni o reprezintă suportul său robust pentru modelarea consumului energetic, realizat printr-o refactorizare substanțială a modulului dedicat energiei, precum și a altor componente relevante, inclusiv cele asociate containerelor și rețelelor. În contextul în care eficiența energetică a centrelor de date constituie o preocupare centrală atât pentru cercetare, cât și pentru sectorul industrial [12], CloudSim 7G facilitează o evaluare realistă a strategiilor de conservare a energiei și a politicilor de scalare automată a consumului energetic. Acesta oferă oportunitatea simulării suprapunerilor de containere pe mașini virtuale, gestionarea supraîncărcării, migrarea sarcinilor de lucru și analiza penalizărilor rezultate din procesul de virtualizare.

În plus, CloudSim 7G integrează funcționalități derivate din modulele istorice, precum NetworkCloudSim, ContainerCloudSim și CloudSimSDN, reunite sub o arhitectură unificată, care se distinge prin compatibilitate și extensibilitate. Această sinteză complexă elimină dificultățile de compatibilitate care au fost întâmpinate anterior, facilitându-se conceperea unor scenarii hibride. În cadrul acestor scenarii, pot coexista mașini virtuale (VM-uri), containere, funcții serverless și servicii distribuite, toate acestea operând într-un mediu de simulare singular.

În ceea ce privește performanța, versiunea 7G realizează optimizări semnificative ale codului sursă, diminuându-l cu peste 13.000 de linii, concomitent cu o îmbunătățire considerabilă a timpului de execuție și a eficienței utilizării memoriei, reușind să aloce cu până la 25% mai puțină memorie heap, fără compromiterea funcționalității. Această versiune introduce, de asemenea, concepte avansate, inclusiv suport pentru virtualizare imbricată și modelarea penalizărilor de latență asociate interfețelor virtuale, aspecte esențiale pentru simularea realistă a infrastructurilor cloud de nouă generație.

Prin urmare, CloudSim 7G asigură un echilibru optim între realismul simulării, extensibilitate și eficiență computațională, reprezentând alegerea ideală pentru obiectivele prezentei lucrări, care se focalizează pe analiza performanței și consumului energetic în scenarii avansate de tip cloud.

1. Algoritmi de planificare a sarcinilor

În contextul sistemelor de cloud computing, gestionarea programării sarcinilor constituie o componentă esențială, având o influență semnificativă asupra calității serviciilor oferite utilizatorilor, eficienței economice și performanței la nivel global. În acest context, Al-Arasi și Saif[13] analizează o problemă esențială în domeniul programării sarcinilor în cloud computing, focalizându-se în mod special asupra eficienței tehnicilor meta-euristice. Cloud computing-ul, definit prin resurse abundente și dinamice, necesită implementarea unor metode optime de programare pentru alocarea sarcinilor către mașinile virtuale (VM) într-un mod care să îndeplinească o varietate de obiective de optimizare. Aceste obiective includ, printre altele, reducerea timpului total de execuție (makespan), minimizarea costurilor operaționale, asigurarea unei distribuții echilibrate a sarcinilor și îmbunătățirea eficienței energetice.

Există o structurare complexă a sarcinilor în cadrul celor trei straturi fundamentale ale arhitecturii cloud[13]: stratul software, concentrat pe maximizarea experienței utilizatorului prin minimizarea timpului de procesare și a costurilor aferente, stratul platformă, dedicat virtualizării resurselor și optimizării gestionării sarcinilor, respectiv stratul infrastructură, care se focalizează pe implementarea coordonată a resurselor și facilitarea colaborării între diversele entități cloud. Procesul de planificare a sarcinilor este, de asemenea, segmentat în trei etape distincte[13]: identificarea resurselor disponibile, selecția adecvată a acestora și alocarea sarcinilor, de regulă, mediată de un broker de centre de date (DataCenter Broker), care este susținut de servicii de informații cloud.

Este bine documentat în literatura de specialitate faptul că planificarea sarcinilor în domeniul cloud computing este clasificată ca NP-hard (Not Polynomial hard), ceea ce indică faptul că, în contextul instanțelor de mari dimensiuni, obținerea unor soluții exacte devine o întreprindere impracticabilă. Urmărindu-se aproximarea soluțiilor optime într-un interval de timp acceptabil, cercetătorii din sfera academică au propus o varietate de abordări euristice și metaeuristice, incluzând metodele tradiționale de programare, cum ar fi „First Come First Serve” (FCFS), „Round Robin” (RR) și „Shortest Job First” (SJF), care s-au dovedit însă ineficiente în furnizarea unor performanțe optime în cadrul complexității NP-hard asociate sistemelor cloud. Deși abordările euristice, precum Min-Min și Max-Min, oferă anumite îmbunătățiri, acestea sunt adesea constrânse de capcana optimilor locali. În contrast, algoritmii meta-euristici, inspirați de fenomenele naturale și structurile biologice, propun soluții care manifestă o scalabilitate mai mare și se apropie de optimalitate. Astfel, aceste tehnici sunt catalogate în două categorii fundamentale, și anume cele bio-inspirate (de exemplu, algoritmii genetici și algoritmii memetici) și cele bazate pe inteligența colectivă a roiului (de exemplu, „PSO” - Particle Swarm Optimization și „ACO” - Ant Colony Optimization).

O contribuție deosebită a acestei analize[13] rezidă în efectuarea unei evaluări comparative riguroase a unui număr semnificativ de algoritmi meta-euristici, bazată pe o varietate diversificată de factori, printre care se numără mediul de simulare dominant (CloudSim), metricile de optimizare corespunzătoare (inclusiv durata de realizare, costurile, termenele limită, consumul de energie etc.), amploarea experimentală (volumul sarcinilor și resurselor implicate) și natura interdependenței sarcinilor, fie că sunt dependente sau independente. Printr-o clasificare sistematică și detaliată, studiul subliniază că durata de realizare se prezintă ca fiind criteriul cel mai frecvent vizat în procesele de optimizare, fiind menționată în 33% din cercetările analizate, urmată de costuri (18%), echilibrarea sarcinilor (16%) și alte dimensiuni relevante, precum eficiența energetică și debitul. Agoritmii de Optimizare a Roiului de Particule (PSO) și Algoritmul Genetic (GA) se evidențiază ca fiind cele mai frecvent aplicate metode metaeuristice în domeniile inteligenței în roi, respectiv al algoritmilor bio-inspirați.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algoritm | Tipul algoritmului | Optimizare principala | Avantaje | Dezavantaje |
| FCFS[] | Tradițional | Primul venit primul servit | Ușor de implementat | Timpul de execuție și costul ridicat |
| RoundRobin | Tradițional | Distribuție egală a resurselor | Potrivit pentru medii interactive | Creste overhead-ul, nu e bun pentru sarcini cu timp de execuție mare |
| Random Scheduler | Tradițional | Aleatoriu | Ușor de implementat | Lipsa predictibilității, controlului și foarte ineficient |
| LJF | Tradițional | Se prioritizează sarcinile lungi | Se imbunătățeste utilizarea resurselor | Sarcinile scurte sunt blocate de cele lungi |
| Min-Min | Euristic | Se reduce makespan-ul | Timp scurt la finalizarea sarcinilor | Sarcini distribuite inegal, având in vedere resursele |
| Max-Min | Euristic | Incărcare echilibrată | Utilizează foarte bine resursele | Sarcinile scurte sunt blocate de cele lungi |
| ACO | Meta-euristic | Se balansează incărcarea, are makespan | Adaptiv | Timp de convergență mare |
| Genetic Algorithm(GA) | Meta-euristic | Are makespan, cost | Foarte flexibil, eficient în privința optimizărilor | Sensibil la parametri |
| PSO | Meta-euristic | Se balansează incărcarea, are makespan | Optimizări globale bune | Poate rămâne blocat intr-o soluție locală, care pare a fi optimă, însă se nu este. |

Tabel3.Comparație între Algoritmii de Planificare a sarcinilor

1. Alocarea resurselor pentru eficiența energetică

Utilizarea tehnologiilor de cloud computing a cunoscut o evoluție semnificativă, depășind limitele conceptelor tradiționale de tipul calculului paralel, grid computing-ului sau a calculului distribuit. În prezent, această paradigmă tehnologică integrează virtualizarea serverelor cu o diversitate de resurse și infrastructuri emergente. În calitate de sistem contemporan, cloud computing-ul se confruntă cu o gamă extinsă de provocări, în special în contextul optimizării alocării resurselor. Eficiența în alocarea acestor resurse este de o importanță deosebită, având în vedere impactul său asupra modului în care infrastructura virtualizată este distribuită pentru satisfacerea cerințelor utilizatorilor, conform acordurilor de nivel de serviciu (SLA), urmărind înn paralel minimizarea costurilor, precum și reducerea consumului de energie.

Alocarea resurselor constituie un proces complex ce implică distribuirea capacității de procesare, a memoriei și a lățimii de bandă între diverse aplicații și utilizatori în cadrul mediilor cloud. Obiectivul fundamental al acestui demers este de a asigura o alocare atât echitabilă, cât și eficientă a acestor resurse, garantând, în consecință, desfășurarea activităților într-un interval temporal adecvat, concomitent cu optimizarea consumului energetic și a cheltuielilor operaționale. Centrele de date se confruntă cu provocarea de a menține un echilibru delicat între disponibilizarea resurselor necesare pentru operațiuni cu prioritate înaltă și riscul de a consuma aceste resurse în mod ineficient, prin alocarea lor către procese de o prioritate inferioară. Conform analizei efectuate, se estimează că peste 20% din energia utilizată de centrele de date provine dintr-o utilizare ineficientă a resurselor, fiind astfel evidențiată necesitatea stringentă de a spori eficiența energetică în acest context[14].

Printre problemele fundamentale identificate se regăsesc dificultățile inerente prognozării cerințelor aplicațiilor utilizatorilor, asigurarea capacității echipamentelor fizice de a susține, în mod simultan, multiple instanțe de mașini virtuale, gestionarea eficientă a programării sarcinilor asociate proceselor de lungă durată, precum și atingerea unui nivel optim de eficiență energetică, în ciuda creșterii volumului de muncă. Eseul subliniază importanța primordială a armonizării perspectivelor și intereselor atât ale furnizorilor de servicii, cât și ale consumatorilor acestora, în scopul realizării unor servicii cloud cu un grad înalt de productivitate.

În vederea sistematizării concluziilor, cercetarea [14] îmbrățișează o taxonomie care facilitează clasificarea modalităților de alocare a resurselor în cinci categorii bine definite, și anume strategice, bazate pe resurse țintă, orientate spre optimizare, fundamentate pe programare, precum și tehnici axate pe consumul de energie.

În consecință, strategiile fundamentate pe consumul de energie își concentrează eforturile asupra reducerii utilizării energetice în centrele de date. Aceste strategii pot fi ulterior clasificate în două categorii distincte, respetiv tehnici axate pe eficiența energetică și tehnici orientate spre managementul temperaturii. Tehnicile eficiente din perspectiva consumului energetic urmăresc optimizarea plasării mașinilor virtuale și coordonarea distribuției sarcinilor de lucru, având ca obiectiv principal diminuarea consumului energetic. În acest context, se recurge la metaeuristici, cum ar fi algoritmii de optimizare inspirați de comportamentul coloniilor de furnici sau de coloniile artificiale de albine, cu scopul de a spori eficiența energetică, deși această abordare poate conduce la o prelungire a timpului necesar execuției. Pe de altă parte, tehnicile bazate pe temperatură se concentrează pe evaluarea influenței distribuției volumului de lucru asupra temperaturii componentelor hardware, având ca ambiție prevenirea supraîncălzirii prin echilibrarea încărcărilor termice. Cu alte cuvinte, aceste strategii nu contribuie doar la creșterea fiabilității sistemului, ci și la diminuarea necesităților de răcire, însă trebuie menționat faptul că aceste tehnici neglijează adesea considerente esențiale,cum ar fi costurile asociate sau alinierea optimă a volumului de lucru.

Totdată, nu putem subestima nici importanța componentelor esențiale ale analizei tehnicilor de alocare strategică a resurselor, care joacă un rol fundamental în facilitarea adaptabilității atât pentru furnizorii de servicii cloud, cât și pentru consumatorii acestora, în contextul caracterului dinamic și evolutiv al mediilor de cloud computing. Aceste strategii sunt elaborate cu scopul de a facilita luarea de decizii informate și anticipative în ceea ce privește distribuția resurselor, ele contribuind la armonizarea cerințelor consumatorilor cu resursele disponibile, într-un mod care nu doar optimizează performanța, ci și contribuie la reducerea costurilor, sporind astfel satisfacția generală a utilizatorilor.

Tehnicile de alocare strategică a resurselor sunt, în general, divizate în două categorii fundamentale, și anume alocarea dinamică a resurselor și abordările bazate pe inteligența artificială. Aceste metodologii permit furnizorilor de servicii cloud efectuarea de analize riguroase și comparative între diversele resurse disponibile, prezicerea cu acuratețe a cerințelor de resurse ale consumatorilor, în funcție de specificitatea sarcinii respective, respectiv determinarea infrastructurii virtuale care este cea mai adecvată pentru ducerea la bun sfârșit a respectivei sarcini în cel mai eficient mod posibil.

În ultimii ani, centrele de date au evoluat în calitate de consumatori semnificativi de energie electrică[14], ceea ce a determinat comunitatea academică să dezvolte strategii mai eficiente pentru gestionarea distribuției resurselor, necesitate fundamentată prin faptul că în baza unei administrări inteligente a resurselor, se poate obține nu doar diminuarea consumul energetic, dar și reducerea costurile operaționale globale asociate infrastructurii cloud. În acest context, lucrarea [14] propune o clasificare a strategiilor de alocare a resurselor, având ca obiectiv eficiența energetică, în două categorii esențiale, respectiv tehnicile conștiente de energie, care subliniază importanța minimizării utilizării energiei pentru a promova atât eficiența economică, cât și sustenabilitatea ecologică, și tehnicile conștiente de temperatură, care examinează efectele termice asociate implementării resurselor și caută să atenueze acumularea de căldură în interiorul unităților fizice.

Alocarea resurselor pe baza consumului energetic este orientată înspre atingerea unor standarde de excelență operațională, în conformitate cu constrângerile inerente consumului și distribuției energiei. Aceste strategii au devenit tot mai cruciale în asigurarea eficienței operațiunilor centrelor de date, atât din perspectiva performanței, cât și a sustenabilității ecologice. O varietate de cercetători au avansat metode inovatoare pentru a diminua consumul energetic, prin optimizarea plasării mașinilor virtuale (VM). Putem lua drept exemplu dezvoltarea unei tehnici bazate pe principiile inteligenței colective, care facilitează migrarea dinamică a VM-urilor, urmărind echilibrarea sarcinii sistemului concomitent cu reducerea consumului energetic. Această abordare a demonstrat o eficiență notabilă în minimizarea timpului de răspuns, însă nu a abordat în mod explicit problema timpului de execuție. Similar, a fost implementată optimizarea coloniilor de furnici în vederea abordării provocării alocării VM-urilor, având ca obiectiv reducerea risipei energetice și implicit a consumului total de energie. Deși această metodă a reușit să îndeplinească obiectivele urmărite, nu a fost luată în considerare variabila timpului de răspuns.

Aceste tehnici orientate spre energie constituie un avans remarcabil în direcția unui cloud computing sustenabil, însă este impetuos necesar de a fi meționat faptul că acesta tinde de cele mai multe ori să privilegieze eficiența energetică în detrimentul altor parametri relevanți, cum ar fi latența sau timpul de execuție.

[1] Van Dijk, J. *The Network Society*; Sage Publications: Thousand Oaks, CA, USA, 2012.

[2] S. Yi, C. Li, and Q. Li, "Fog computing: A taxonomy, survey and future directions," in *Proc. 2015 Workshop on Mobile Big Data (Mobidata '15)*, pp. 1–6, ACM, 2015.

[3] Mahmud, R.; Kotagiri, R.; Buyya, R. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything*; Springer: Berlin, Germany, 2018; pp. 103–130.

[4] E. Del-Pozo-Puñal, F. García-Carballeira, and D. Camarmas-Alonso, "A scalable simulator for cloud, fog and edge computing platforms with mobility support," *Future Generation Computer Systems*, vol. 144, pp. 117–130, 2023.

[5] M. Fahimullah, G. Philippe, S. Ahvar, and M. Trocan, "Simulation tools for fog computing: A comparative analysis," *Sensors*, vol. 23, no. 7, p. 3492, 2023.

[6] K. Kok, S. Karnouskos, D. Nestle, A. Dimeas, A. Weidlich, C. Warmer, *et al*., "Smart houses for a smart grid," in *Proc. 20th Int. Conf. and Exhibition on Electricity Distribution (CIRED 2009)*, Stevenage, UK: IET, Jun. 2009, p. 0751.

[7] R. Andreoli, J. Zhao, T. Cucinotta, and R. Buyya, "CloudSim 7G: An integrated toolkit for modeling and simulation of future generation cloud computing environments," *Software: Practice and Experience*, 2025. (în curs de apariție / in press, dacă nu are volum și pagini)

[8] M. A. Shahid, M. M. Alam, and M. M. Su’ud, "A systematic parameter analysis of cloud simulation tools in cloud computing environments," *Applied Sciences*, vol. 13, no. 15, p. 8785, 2023.

[9] M. Ashouri, F. Lorig, P. Davidsson, and R. Spalazzese, "Edge computing simulators for IoT system design: An analysis of qualities and metrics," *Future Internet*, vol. 11, no. 11, p. 235, 2019.

[10] P. Velho and A. Legrand, "Accuracy study and improvement of network simulation in the SimGrid framework," in *Proc. 2nd Int. Conf. on Simulation Tools and Techniques (SIMUTools '09)*, Mar. 2009.

[11] S. N. A. Jawaddi and A. Ismail, "Integrating OpenAI Gym and CloudSim Plus: A simulation environment for DRL agent training in energy-driven cloud scaling," *Simulation Modelling Practice and Theory*, vol. 130, p. 102858, 2024.

[12] R. Schmidt, D. Beaty, and J. Dietrich, "Increasing energy efficiency in data centers," *ASHRAE Journal*, vol. 49, no. 12, p. 18, 2007.

[13] Al-Arasi, R. A., & Saif, A. (2020). "Task Scheduling in Cloud Computing Based on Meta-Heuristic Techniques: A Review Paper." *EAI Endorsed Transactions on Cloud Systems*, vol. 6, no. 17, Jan. 2020.

[14] Abid, A., Manzoor, M. F., Farooq, M. S., Farooq, U., & Hussain, M. (2020). "Challenges and Issues of Resource Allocation Techniques in Cloud Computing." *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 7, pp. 2815–2839.