3. Studiu Bibliografic

1. Introducere in Cloud Computing

În cadrul celei de-a patra revoluții industriale (4IR), tehnologiile informatice au evoluat semnificativ în direcția descentralizării și minimizării latenței[1], ceea ce a condus la emergența a trei paradigme fundamentale de calcul: 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, precum și prin capacitatea de a satisface cerințele specifice legate de latență, scalabilitate și 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 inteligente, inclusiv telefoanele mobile, televizoarele inteligente și ceasurile 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.O gamă largă de caracteristici și tehnologii care contribuie la procesarea și stocarea datelor la marginea rețelei formează domeniul de cercetare cunoscut sub numele de Edge Computing.

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 sfera academică, cât și în sectorul privat. În ultimele decenii, instrumentele de simulare [4] au dobândit o importanță crescândă în cadrul cercetărilor dedicate acestei tehnologii emergente. Prin intermediul proceselor de simulare, este posibilă evaluarea prealabilă a configurațiilor de servicii, precum și a strategiilor de alocare și gestionare a resurselor, ceea ce facilitează îmbunătățirea performanței sistemelor, testarea riguros a ipotezelor tehnice și comerciale, precum și obținerea unor rezultate fiabile și reproductibile, cu un minimum de investiții, un risc considerabil diminuat și un timp de execuție optimizat.

În contextul cercetării contemporane, 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 infrastructurii, incluzând metrici precum utilizarea procesorului central (CPU), lățimea de bandă disponibilă și latența, ceea ce facilitează astfel luarea de decizii fundamentate în ceea ce privește implementarea în medii reale.

1. Platforme de simulare

Nu putem omite faptul că există o plajă variată de simulatoare dedicate mediilor Cloud, Fog și Edge[4], fiecare având propriile trăsături distinctive și funcționalități specifice. Aceste instrumente permit configurarea diverselor parametri sistemici, inclusiv arhitectura rețelei, tipurile de dispozitive implicate, cerințele aplicațiilor, precum și caracteristicile proceselor în sine, oferind astfel o platformă de testare care se dovedește a fi atât flexibilă, cât și accesibilă. Cu toate acestea, niciun simulator nu reușește să satisfacă în mod exhaustiv toate cerințele posibile ale unui scenariu complex. Această diversitate de instrumente generează o provocare suplimentară: selecția simulatorului adecvat în funcție de obiectivele specifice ale cercetării. În literatura de specialitate, au fost formulate numeroase clasificări și analize comparative ale simulatoarelor, dintre care se remarcă:

• CloudSim[8], cel mai utilizat simulator pentru infrastructuri cloud, se concentrează pe alocarea eficientă a resurselor și programarea sarcinilor (scheduling), facilitând astfel analiza performanței în medii virtualizate;

• iFogSim și YAFS[5], extensii special concepute pentru a aborda particularitățile mediilor Fog și Edge, oferind suport pentru distribuții geografice și topologii dinamice, esențiale pentru gestionarea eficientă a resurselor în aceste arhitecturi emergente;

• LEAF și EdgeCloudSim[9], dedicate simulării aplicațiilor Internet of Things (IoT) și evaluării performanței în condiții de latență scăzută, contribuind la optimizarea interacțiunilor dintre dispozitivele conectate;

• SimGrid[10], un instrument robust pentru simulările distribuite, care a fost recent extins prin inițiative precum ENIGMA, având ca scop integrarea mobilității și vizualizării geografice, îmbunătățind astfel capacitatea de modelare a sistemelor complexe.

În acest cadru, compararea acestor simulatoare capătă o importanță crucială, nu doar dintr-o perspectivă tehnică, ci și din una practică, având în vedere scenariile reale ce se intenționează a fi replicate.

Tabelul de mai jos oferă o sinteză comparativă a celor mai frecvent utilizate simulatoare în literatura de specialitate, subliniind 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 classic, fara mobilitate dinamica |
| iFogSim[] | Cloud + Fog | Java | Are topologie fixa, utilizat pentru IoT |
| YAFS | Fog + Edge | Python | Mobilitate reala, continue grafuri dinamice. |
| LEAF | Edge | Java | Folosit pentru evenimente discrete, IoT distribuit |
| EdgeCloudSim | Cloud + Edge | Java | Pentru aplicatii de tip edge, simplu |
| SimGrid | Distribuit | C/C++ | Foarte greu de personalizat, folosit pentru arhitecturi distribuite |
| ENIGMA | Fog + Edge | C/C++ | Are mobilitate IoT, implicit vizualizare geospatiala pe baza SimGrid |

Tabel 1. Comparație între cele mai utilizate simulatoare [5], [8], [9].

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 ale acestuia. În literatura de specialitate, CloudSim se evidențiază printr-o serie de versiuni care au fost dezvoltate de-a lungul timpului, fiecare dintre acestea introducând îmbunătățiri semnificative și extensii funcționale: CloudSim, versiunea de bază, CloudSim Plus, care este orientat pe obiect și modular, și, nu în ultimul rând, CloudSim 7G, cea mai recentă variantă, care este extinsă cu funcționalități moderne. //data release in table + paper

|  |  |  |  |
| --- | --- | --- | --- |
| Versiune | Limbaj | Extensibilitate | Functionalitate |
| CloudSim | Java | Buna | Simplu, ofera modelare pentru VMs implicit datacenters |
| CloudSimPlus | Java | Excelenta | Imbunatatire la modularitate, usor de utilizat, cod mai clar si bine structurat |
| CloudSim7G | Java | Ridicata | Ofera flexibilitate aplicatiilor distribuite, suporta simularea si managementul retelelor in cloud, include functionalitati 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, reprezentând cea mai sofisticată și cuprinzătoare iterație din această serie de instrumente. Spre deosebire de versiunile anterioare, care erau 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 este reprezentat de 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. Într-un context 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 economisire a energiei și a politicilor de scalare automată a consumului energetic. Acesta oferă oportunitatea de a simula suprapunerile 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, facilitând astfel 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 a compromite funcționalitatea. 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 globale.

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 ameliorarea eficienței energetice.

Există o structurare complexă a sarcinilor în cadrul celor trei straturi fundamentale ale arhitecturii cloud[13]: stratul software, care se concentrează pe maximizarea experienței utilizatorului prin minimizarea timpului de procesare și a costurilor aferente; stratul platformă, dedicat virtualizării resurselor și optimizării gestionării sarcinilor; și 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 că problema planificării 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. Cu scopul de a aproxima soluțiile 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, însă, s-au dovedit 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: 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[] | Traditional | Primul venit primul servit | Usor de implementat | Timpul de executie si costurile |
| RoundRobin | Traditional | Distributie egala a resurselor | Potrivit pentru medii interactive | Creste overhead-ul, nu e bun pentru sarcini cu timp de executie mare |
| Random Scheduler | Traditional | Random | Usor de implementat | Lipsa predictibilitatii, controlului si foarte ineficient |
| LJF | Traditional | Se prioritizeaza sarcinile lungi | Se imbunatateste utilizarea resurselor | Sarcinile scurte sunt blocate de cele lungi |
| Min-Min | Euristic | Se reduce makespan-ul | Timp scurt la finalizarea sarcinilor | Sarcini distribuite inegal, avand in vedere resursele |
| Max-Min | Euristic | Incarcare echilibrata | Utilizeaza foarte bine resursele | Sarcinile scurte sunt blocate de cele lungi |
| ACO | Meta-euristic | Se balanseaza incarcarea, are makespan | Adaptiv | Timp de convergenta mare |
| Genetic Algorithm(GA) | Meta-euristic | Are makespan, cost | Foarte flexibil, efficient in optimizari | Sensibil la parametrii |
| PSO | Meta-euristic | Se balanseaza incarcarea, are makespan | Optimizari globale bune | Poate ramane blocat intr-o solutie locala, care pare a fi optima, dar nu este,prea rapid. |

Tabel3.Comparație între Algoritmii de Planificare a sarcinilor

1. Alocarea resurselor pentru eficienta energetica

Utilizarea tehnologiilor de cloud computing a cunoscut o evoluție semnificativă, depășind limitele conceptelor tradiționale precum calculul paralel, grid computing-ul și calculul 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 esențială, având în vedere impactul său asupra modului în care infrastructura virtualizată este distribuită pentru a satisface cerințele utilizatorilor, conform acordurilor de nivel de serviciu (SLA), în timp ce urmărește, de asemenea, minimizarea costurilor 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 diverese 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, evidențiind astfel 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 crucială 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: 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: 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 pentru execuție. 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. Aceste strategii nu doar că contribuie la creșterea fiabilității sistemului, ci și la diminuarea necesităților de răcire. Totuși, este important de menționat că aceste tehnici adesea neglijează considerente esențiale, precum costurile asociate sau alinierea optimă a volumului de lucru.

Nu putem subestima 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 contribuie 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: alocarea dinamică a resurselor și abordările bazate pe inteligența artificială. Aceste metodologii permit furnizorilor de servicii cloud să efectueze analize riguroase și comparative între diversele resurse disponibile, să prezică cu acuratețe cerințele de resurse ale consumatorilor, în funcție de specificitatea sarcinii respective, și să determine infrastructura virtuală cea mai adecvată pentru a îndeplini eficient acea sarcină.

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. Această necesitate este fundamentată prin faptul că, printr-o administrare inteligentă a resurselor, se poate nu doar diminua consumul energetic, dar și reduce 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: tehnici conștiente de energie, care subliniază importanța minimizării utilizării energiei pentru a promova atât eficiența economică, cât și sustenabilitatea ecologică, și tehnici 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 spre 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). De exemplu, a fost dezvoltată o tehnică bazată pe principiile inteligenței colective, care facilitează migrarea dinamică a VM-urilor, având ca scop echilibrarea sarcinii sistemului și 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. În mod similar, s-a implementat optimizarea coloniilor de furnici în vederea abordării provocării alocării VM-urilor, având ca obiectiv reducerea atât a risipei energetice, cât și a consumului total de energie. Deși această metodă a reușit să îndeplinească aceste scopuri, nu a luat în considerare variabila timpului de răspuns.

Aceste tehnici orientate spre energie constituie un avans remarcabil în direcția unui cloud computing sustenabil, totuși, este demn de menționat că acestea tinde să privilegieze eficiența energetică, adesea în detrimentul altor parametri relevanți, precum timpul de execuție sau latența de răspuns.

Nu sunt ok

[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.