3. Studiu Bibliografic

1. Introducere in Cloud Computing

În contextul celei de-a patra revoluții industriale (4IR), tehnologiile informatice au evoluat în direcția descentralizării și reducerii latenței[1], conducând la apariția a trei paradigme esențiale de calcul: Cloud Computing , Fog Computing și Edge Computing. Aceste trei moduri de procesare a datelor se diferențiază în funcție de poziționarea în rețea, de apropierea față de sursa datelor și de capacitatea de a răspunde cerințelor 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 crescut rapid ca urmare a adoptării cloud computing-ului ca standard pentru procesarea și stocarea datelor în întreaga lume. Acest lucru s-a datorat faptului că utilizatorii au avut acces la dispozitive inteligente, cum ar fi telefoane mobile, televizoare inteligente și ceasuri inteligente [6]. Acest lucru a dus la o creștere semnificativă a nevoii de resurse de rețea și servere pentru procesarea datelor, cee ace a determinat apariția unor paradigme noi precum Edge Computing și Fog Computing, care aduc procesarea și capacitatea de stocare mai aproape de utilizator, ceea ce reduce latența și crește disponibilitatea 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.

În cele din urmă, paradigma Fog Computing este un model stratificat cu scopul principal de a menține beneficiile cloud-ului pentru implementarea aplicațiilor sau serviciilor capabile să proceseze sau să stocheze date pentru utilizatorii finali în noduri distribuite. Acest lucru, împreună cu nodurile edge, permite reducerea timpului de răspuns necesar pentru sarcinile sau procesele care sunt create în timpul utilizării.

Dintre cele trei paradigme, cloud computing-ul s-a evidențiat ca fiind cel mai răspândit și matur model, utilizându-se atât în mediul academic, cât și în sectorul privat. În ultimii ani, instrumentele de simulare [4] au căpătat un rol tot mai important în cercetarea dedicată acestei tehnologii. Prin intermediul simulării, pot fi analizate în prealabil configurările de servicii și strategiile de alocare și gestionare a resurselor, ceea ce contribuie la îmbunătățirea performanței, testarea ipotezelor tehnice și comerciale și obținerea unor rezultate fiabile, reproductibile, cu investiții minime, riscuri reduse și timp de execuție scurt.

În cercetarea actuală, simularea este abordarea preferată la nivel internațional, permițând definirea unor medii virtuale care includ aplicații, sarcini, dispozitive și condiții de rețea specifice. Aceste simulări generează date valoroase privind performanța infrastructurii: precum utilizarea CPU-ului, lățimea de bandă sau latența, facilitând astfel deciziile fundamentate cu privire la implementarea în mediu real.

1. Platforme de simulare

Nu putem omite faptul ca există o varietate de simulatoare dedicate mediilor Cloud, Fog și Edge[4], fiecare cu propriile caracteristici și funcționalități. Ele permit configurarea parametrilor de sistem, cum ar fi arhitectura rețelei, tipurile de dispozitive, cerințele aplicațiilor, sau caracteristicile proceselor și oferă o platformă de testare flexibilă și accesibilă. Cu toate acestea, niciun simulator nu reușește să acopere complet toate cerințele pe care le poate presupune un scenariu complex. Această diversitate de instrumente ridică o provocare suplimentară: alegerea simulatorului potrivit în funcție de scopul cercetării. În literatura de specialitate au fost propuse mai multe clasificări și analize comparative ale simulatorilor, printre care se remarcă:

* CloudSim[8], cel mai utilizat simulator pentru infrastructuri cloud, axat pe alocarea resurselor și programarea sarcinilor (scheduling);
* iFogSim și YAFS[5], extensii axate pe mediile Fog și Edge, cu suport pentru distribuție geografică și topologii dinamice;
* LEAF și EdgeCloudSim[9], orientate spre simularea aplicațiilor IoT și testarea performanței în scenarii cu latență redusă;
* SimGrid[10], un instrument puternic pentru simulări distribuite, extins recent prin proiecte precum ENIGMA pentru a include mobilitate și vizualizare geografică

În acest context, comparația între aceste simulatoare devine crucială, nu doar din punct de vedere tehnic, ci și practice, în funcție de scenariile reale care se doresc a fi simulate.

Tabelul de mai jos prezintă o comparație sintetică între cele mai utilizate simulatoare din literatura de specialitate, evidențiind paradigma vizată, limbajul folosit, suportul pentru mobilitate și energie, precum și câteva observații cheie.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Simulator** | **Paradigmă** | **Limbaj** | **Mobilitate** | **Energie** | **Observații principale** |
| CloudSim | Cloud | Java | Nu | Parțial | Simulator clasic, larg adoptat |
| iFogSim | Cloud + Fog | Java | Limitat | Da | IoT pe topologii fixe |
| YAFS | Fog + Edge | Python | Da | Da | Grafuri dinamice, mobilitate reală |
| LEAF | Edge | Java | Da | Da | Evenimente discrete, IoT distribuit |
| EdgeCloudSim | Cloud + Edge | Java | Limitat | Da | Simplu, dedicat aplicațiilor edge |
| SimGrid | Distribuit | C/C++ | Da | Da | Foarte performant, greu de personalizat |
| ENIGMA | Fog + Edge (SimGrid) | C/C++ | Da (IoT mobil) | Da | Vizualizare geospațială, suport mobilitate IoT |

**Tabel1**. Comparație între cele mai utilizate simulatoare [5], [8], [9].

După analiza generală a simulatorilor disponibili, am decis să utilizez CloudSim ca bază pentru simulările din această lucrare, având în vedere popularitatea, maturitatea și flexibilitatea sa. În literatura de specialitate, CloudSim are mai multe versiuni dezvoltate în timp, fiecare aducând îmbunătățiri și extensii funcționale semnificative: CloudSim, versiunea de bază, CloudSim Plus, orientat pe obiect și modular, implicit CloudSim 7G, versiune de ultimă generație, extinsă cu funcționalități moderne.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Versiune** | **Limbaj** | **Organizare** | **Suport energie** | **Extensibilitate** | **Funcționalități cheie** |
| CloudSim | Java | Clasică | Limitat | Bună | Modelare VM, datacenters, politici simple |
| CloudSim Plus | Java | OOP | Limitat | Excelentă | OOP, modularitate, cod refactorizat |
| CloudSim 7G | Java | Avansată | Complet | Ridicată | Microservicii, SDN, energie, IoT, modern UI |

**Tabel2.** Comparație între simulatoarele CloudSim [7], [11]

Dintre diversele versiuni disponibile ale simulatorului CloudSim, CloudSim 7G a fost ales pentru această cercetare, constituind cea mai avansată și cuprinzătoare generație din această serie de instrumente. Spre deosebire de versiunile anterioare, care includeau module disparate, adesea dificil de integrat și întreținut, CloudSim 7G propune o arhitectură reconfigurată și standardizată, capabilă să faciliteze simulări complexe, multi-modulare, într-o manieră coerentă și eficientă.

Unul dintre cele mai semnificative avantaje ale acestei versiuni constă în 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 reprezintă o preocupare centrală atât pentru cercetare, cât și pentru industrie [12], CloudSim 7G facilitează o evaluare realistă a strategiilor de economisire a energiei și a politicilor de scalare automată a consumului energetic, oferind oportunitatea de a simula suprapunerile de containere peste mașini virtuale, gestionarea supraîncărcării, migrarea sarcinilor de lucru și analiza penalizărilor rezultate din procesul de virtualizare.

În plus, CloudSim 7G integrează funcționalități provenite din modulele istorice, precum NetworkCloudSim, ContainerCloudSim și CloudSimSDN, reunită sub o arhitectură unificată, compatibilă și extensibilă. Această sinteză elimină dificultățile de compatibilitate întâmpinate anterior și facilitează conceperea de scenarii hibride, în cadrul cărora pot coexista mașini virtuale (VM-uri), containere, funcții serverless și servicii distribuite – toate operate într-un singur mediu de simulare.

În ceea ce privește performanța, versiunea 7G realizează optimizări semnificative ale codului sursă, reducându-l cu peste 13.000 de linii, și îmbunătățește considerabil timpul de execuție, precum și eficiența utilizării memoriei, alocând cu până la 25% mai puțină memorie heap, fără a afecta negativ funcționalitatea. Această versiune introduce, de asemenea, concepte contemporane, inclusiv suport pentru virtualizare imbricată și modelarea penalizărilor de latență generate de interfețele virtuale, aspecte esențiale pentru simularea realistă a infrastructurilor cloud de nouă generație.

Prin urmare, CloudSim 7G oferă un echilibru optim între realismul simulării, extensibilitate și eficiență computațională, constituind alegerea ideală pentru obiectivele prezentei lucrări, care se concentrează pe analiza performanței și a consumului energetic în scenarii cloud avansate.

1. Algoritmi de planificare a sarcinilor

În cadrul sistemelor de cloud computing, managementul programării sarcinilor se dovedește a fi o componentă fundamentală, având un impact direct asupra calității serviciilor furnizate utilizatorilor, eficienței costurilor și performanței generale.

Astfel, Al-Arasi și Saif[13] examinează o problemă fundamentală în domeniul programării sarcinilor în cloud computing, concentrându-se în mod deosebit asupra eficienței tehnicilor meta-euristice. Cloud computing, caracterizat prin resurse ample și dinamic disponibile, impune implementarea unor metode optime de programare pentru alocarea sarcinilor către mașinile virtuale (VM) într-un mod care să satisfacă diverse 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 a sarcinilor în cadrul celor trei straturi fundamentale ale arhitecturii cloud[13]: stratul software, care se concentrează pe maximizarea satisfacției utilizatorului prin minimizarea timpului de procesare și a costurilor asociate; stratul platformă, dedicat virtualizării resurselor și optimizării echilibrului de sarcină; și stratul infrastructură, care se axează pe implementarea globală 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), susținut de servicii de informații cloud.

Este bine documentat că problema planificarii sarcinilor în cloud computing este clasificată ca fiind NP-hard (Not Polynomial hard), ceea ce sugerează că, în cazul instanțelor de dimensiuni mari, obținerea unor soluții exacte devine impracticabilă. Pentru a aproxima soluțiile optime într-un interval de timp rezonabil, cercetătorii din domeniul academic au prezentat[13] o varietate de abordări euristice și metaeuristice, incluzând metodele tradiționale de planificare, precum „First Come First Serve” (FCFS), „Round Robin” (RR) și „Shortest Job First” (SJF), care se dovedesc a fi ineficiente în furnizarea unor performanțe optime în cadrul complexității NP-hard a sistemelor cloud. În ciuda faptului că abordările euristice, precum Min-Min și Max-Min, prezintă anumite îmbunătățiri, acestea sunt frecvent limitate de capcana optimilor locali. În contrast, algoritmii meta-euristici, inspirați de fenomenele naturale și de structurile biologice, propun soluții care sunt mai scalabile și se apropie de optimalitate. Astfel, se clasifică aceste tehnici 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 semnificativă a acestei analize[13] constă în evaluarea comparativă a unui număr considerabil de algoritmi meta-euristici, fundamentată pe o diversitate de factori, precum mediul de simulare predominant (CloudSim), metricile de optimizare relevante (inclusiv durata de realizare, costurile, termenele limită, consumul de energie etc.), scala experimentală (volumul sarcinilor și resurselor implicate) și natura interdependenței sarcinilor (dependente sau independente). Printr-o clasificare riguroasă, studiul evidențiază faptul că durata de realizare se constituie în criteriul cel mai frecvent vizat pentru optimizare, fiind menționat în 33% din cercetările analizate, urmat de costuri (18%), echilibrarea sarcinilor (16%) și alte aspecte, cum ar fi eficiența energetică și debitul. Algoritmii PSO (Optimizarea Roiului de Particule) și GA (Algoritmul Genetic) se dovedesc a fi cei mai utilizați meta-euristici în domeniile inteligenței în roi, respectiv bio-inspirate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algoritm** | **Tip** | **Tipul sarcinilor** | **Optimizare principală** | **Avantaje** | **Limitări** |
| **FCFS** | Tradițional | Independente | Simplitate | Implementare facilă, execuție în ordinea sosirii | Nu optimizează timpul de execuție sau costurile |
| **Round Robin** | Tradițional | Independente | Distribuție echitabilă a timpului | Alocare egală a resurselor, potrivit pentru medii interactive | Ineficient pentru sarcini cu durată mare, crește overhead-ul |
| **Random Scheduler** | Tradițional | Independente | Aleator | Implementare rapidă și ușoară | Lipsă de control și predictibilitate, alocare ineficientă |
| **LJF** | Tradițional | Independente | Prioritizarea sarcinilor lungi | Poate îmbunătăți utilizarea resurselor | Întârzie executarea sarcinilor scurte sau urgente |
| **Min-Min** | Euristic | Independente | Reducerea makespan-ului | Timp total redus pentru finalizarea sarcinilor | Distribuirea sarcinilor poate fi inegală între resurse |
| **Max-Min** | Euristic | Independente | Echilibrarea încărcării | Utilizează mai bine resursele disponibile | Sarcinile scurte pot fi întârziate în favoarea celor lungi |
| **ACO** | Meta-euristic | Independente și dependente | Makespan, balansare încărcare | Adaptiv, capabil să evite soluții locale slabe | Necesită timp de convergență mai mare |
| **Genetic Algorithm (GA)** | Meta-euristic | Independente și dependente | Cost, Makespan, eficiență energetică | Eficient în optimizări multi-obiectiv, flexibil | Complex de configurat, sensibil la parametri |
| **PSO** | Meta-euristic | Independente și dependente | Makespan, balansare încărcare | Rapid, bun pentru optimizări globale | Poate converge prematur fără ajustări suplimentare |

**Tabel3.** Comparație între Algoritmii de Planificare a sarcinilor

1. Alocarea resurselor

Utilizarea tehnologiilor de cloud computing a evoluat dincolo de conceptele tradiționale, precum calculul paralel, grid computing-ul și calculul distribuit. În prezent, această tehnologie integrează virtualizarea serverelor cu o diversitate de resurse și infrastructuri emergente. Ca sistem contemporan, cloud computing-ul se confruntă cu multiple provocări, în special în ceea ce privește optimizarea distribuției resurselor. Alocarea eficientă a acestor resurse este extrem de importantă, deoarece aceasta determină modalitățile prin care infrastructura virtualizată este distribuită pentru a răspunde cerințelor utilizatorilor, în conformitate cu acordurile de nivel de serviciu (SLA), concomitent cu minimizarea costurilor și reducerea consumului energetic.

Alocarea resurselor se referă la distribuirea capacității de procesare, a memoriei și a lățimii de bandă între diverse aplicații și utilizatori în mediile cloud. Obiectivul principal constă în asigurarea unei alocări echitabile și eficiente a acestor resurse, garantând astfel desfășurarea activităților într-un interval de timp adecvat, concomitent cu optimizarea consumului de energie și a cheltuielilor operaționale. Centrele de date se confruntă cu provocarea de a stabili un echilibru între menținerea resurselor necesare pentru operațiunile de înaltă prioritate și riscul de a irosi aceste resurse pe procese de prioritate inferioară sau ineficiente. Conform analizelor efectuate, se estimează că peste 20% din energia consumată de centrele de date este rezultatul unei utilizări ineficiente a resurselor, ceea ce subliniază importanța sporirii eficienței energetice în acest context[14].

Printre problemele fundamentale identificate se numără dificultățile asociate cu prognozarea cerințelor aplicațiilor utilizatorilor, asigurarea capacității echipamentelor fizice de a susține simultan multiple mașini virtuale, gestionarea eficientă a programării sarcinilor pentru procese de lungă durată, precum și atingerea unui nivel optim de eficiență energetică, în ciuda expansiunii volumului de muncă. Eseul evidențiază semnificația esențială a armonizării perspectivelor și intereselor atât ale furnizorilor de servicii, cât și ale consumatorilor acestora, în vederea realizării unor servicii cloud de înaltă productivitate.

În scopul structurării concluziilor, studiul [14] adoptă o taxonomie care permite clasificarea abordărilor de alocare a resurselor în cinci categorii distincte: strategice, bazate pe resurse țintă, orientate spre optimizare, fundamentate pe programare și tehnici axate pe consumul de energie.

Astfel, strategiile fundamentate pe consumul de energie[14] își concentrează eforturile asupra diminuării utilizării energetice în centrele de date. Acestea pot fi ulterior clasificate în două categorii distincte: tehnici orientate pe eficiența energetică și tehnici centrate pe managementul temperaturii. Tehnicile eficiente din punct de vedere energetic își propun să optimizeze amplasarea mașinilor virtuale și să coordoneze distribuția sarcinilor de lucru, având ca scop principal reducerea consumului energetic. În acest context, se recurge la metaeuristici, precum algoritmii de optimizare inspirati din coloniile de furnici sau din coloniile artificiale de albine, pentru a spori eficiența energetică, deși această abordare poate conduce la o creștere a timpului necesar pentru execuție. Pe de altă parte, tehnicile bazate pe temperatură se concentrează pe evaluarea impactului distribuției volumului de lucru asupra temperaturii componentelor hardware, aspirând să prevină supraîncălzirea prin echilibrarea încărcărilor termice. Aceste strategii nu doar că contribuie la îmbunătățirea fiabilității sistemului, dar, de asemenea, reduc necesitățile de răcire. Totuși, acestea adesea neglijează considerente foarte importante, cum ar fi costurile implicate sau alinierea optimă a volumului de lucru.

Nu putem omite nici componenta esențiala a analizei tehnicilor de alocare strategică a resurselor care constă în rolul său fundamental de a facilita adaptabilitatea 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 realiza decizii inteligente și anticipative referitoare la distribuția resurselor. Ele contribuie la alinierea cerințelor consumatorilor cu resursele disponibile într-un mod care nu doar că optimizează performanța, dar și reduce costurile, sporind astfel satisfacția generală a utilizatorilor.

Tehnicile de alocare strategică a resurselor sunt, în general, clasificate în două categorii fundamentale: alocarea dinamică a resurselor și abordările fundamentate pe inteligența artificială. Aceste metodologii permit furnizorilor de servicii cloud să efectueze comparații riguroase între diversele resurse disponibile, să anticipeze cerințele de resurse ale consumatorilor în funcție de specificul sarcinii respective și să identifice 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 de cercetare să elaboreze strategii mai eficiente pentru distribuția resurselor. Această necesitate se justifică prin faptul că, printr-o gestionare inteligentă a resurselor, se poate nu doar diminua consumul energetic, ci și reduce costurile operaționale globale asociate infrastructurii cloud. Astfel, lucrarea [14] clasifică strategiile de alocare a resurselor orientate spre eficiență energetică în două categorii fundamentale: 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 analizează efectele termice asociate implementării resurselor și caută să atenueze acumularea de căldură în interiorul unităților fizice.

Alocarea resurselor pe baza consumului energetic vizează atingerea unor standarde de performanță, în conformitate cu constrângerile asociate consumului și distribuției energiei. Aceste strategii au devenit tot mai esențiale pentru a asigura că operațiunile centrelor de date rămân atât eficiente din punct de vedere operațional, cât și ecologice. O diversitate de cercetători a propus metode inovatoare pentru a diminua consumul energetic prin optimizarea plasării mașinilor virtuale (VM). De exemplu, s-a dezvoltat o tehnică fundamentată pe principiile inteligenței colective, care permite migrarea dinamică a VM-urilor, în scopul echilibrării sarcinii sistemului și reducerii consumului energetic. Această abordare s-a dovedit eficientă în minimizarea timpului de răspuns, totuși, nu a abordat în mod specific aspectul timpului de execuție. În mod similar, s-a mai implementat optimizarea coloniilor de furnici pentru a rezolva problema alocării VM-urilor, având ca obiectiv reducerea atât a risipei de energie, 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.

[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.