University of Salerno

Department of Computer Science

Master Degree in Computer Science

Thesis in Information Security

Exploring the correlation between PUE variation and cyberattacks: impact on Data Centers Security

Supervisors

Prof. **Michele Mastroianni**

Prof. **Francesco Palmieri**

Candidate

Vincenzo Emanuele Martone

Academic Year 2022-2023

Abstract

This research study investigates the correlation between Power Usage Effectiveness (PUE) and the detection of cyberattacks in data centers. The aim of this study is to explore how fluctuations in PUE values may affect the detection of cyberattacks, considering the potential impact on both energy efficiency and security. By analyzing real-world data and simulating two attack scenarios, namely a Denial of Service Attack and a cooling system attack, this research aims to provide valuable insights into the relationship between PUE and attack detection.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Overview and objectives

This research work aims to analyze and evaluate the utilization of the Data Center efficiency parameter *PUE (Power Usage Effectiveness)* for real world cyberattacks detection. In particular, two attack scenarios will be studied and simulated, namely *DoS (Denial of Service)* and cooling system attack, in order to establish if there is a significant variation in the *PUE* that leads to their detection.

1.1.1 PUE definition

PUE (Power Usage Effectiveness) is a Data Center efficiency parameter introduced by a non-profit consortium called *The Green Grid* in 2007 [1]. It is defined as the ratio of total facility energy to IT equipment energy (equation 1.1.1):

$$PUE = \frac{TotalFacilityEnergy}{ITEquipmentEnergy}$$

Total Facility Energy is defined as the energy consumed by the whole Data Center (including IT equipment, power delivery components, cooling system components and data center lighting), while *IT Equipment Energy* is defined as the energy consumed to run the facility's IT infrastructure. *PUE* value can range from 1 to infinity. A *PUE* value of 1 represents an ideal scenario where all the power consumed is used by the IT equipment, making it highly efficient. As the *PUE* value increases above 1, it indicates that a greater portion of the total power is consumed by non-IT equipment, which reduces the overall energy efficiency of the Data Center. Since this parameter provides an insight of the Data Center efficiency, describing how much energy is used by the IT equipment, it has been recognized globally as the industry's most used infrastructure efficiency metric for Data Centers. However, *PUE* depends on many attributes such as Data Center design and implementation, making it difficult to compare Data Centers based on public *PUE* reports. For this purpose, *The Green Grid* has provided a set of guidelines for organizations about making public claims regarding Data Centers *PUE* in order to make this parameter suitable for comparisons. Overall, *PUE* provides an insight about Data Centers energy efficiency helping designers to improve operational efficiency, potentially through comparisons with similar Data Centers.

1.1.2 PUE calculation

Avelar et al. [1] provides a three-level approach for measuring *PUE*. Each level provides a certain level of detail in *PUE* misuration, considering additional measurement points in order to provide further insight into Data Center infrastructure components' energy consumption. There are three main parameters that vary depending on the chosen level of measurement, namely: *IT Equipment Energy*, *Total Facility Energy* and *Measurement Interval*.

Level 1: Basic

With a level 1 measurement, the parameters are calculated as follows:

- **IT Equipment Energy**: it is measured at the output of the *UPS (Uninterruptible Power Supply)* equipment;
- **Total Facility Energy**: it is measured from the utility service entrance that supplies power to all the equipment within the Data Center;
- **Measurement Interval**: power measurements are performed once a month.

Level 2: Intermediate

With a level 2 measurement, the parameters are calculated as follows:

- **IT Equipment Energy**: it is measured at the output of the *PDUs (Power Distribution Unit)* equipment;
- **Total Facility Energy**: it is measured from the utility service entrance that supplies power to all the equipment within the Data Center;
- **Measurement Interval**: power measurements are performed once a day.

Level 3: Advanced

With a level 3 measurement, the parameters are calculated as follows:

- **IT Equipment Energy**: it is measured at each component of the Data Center excluding non-IT loads;
- **Total Facility Energy**: it is measured from the utility service entrance that supplies power to all the equipment within the Data Center;
- **Measurement Interval**: power measurements are performed at least every 15 minutes.

1.1.3 Denial of Service

DoS (Denial of Service) attack is a cyberattack that aims to make a resource unavailable to its users. Usually this kind of attack is performed either by flooding the hosts that provide a specific service until they are unable to respond to their legitimate users or by exploiting a vulnerability of the target. There are various *DoS* attack techniques that have been identified and categorized during the last years, such as *SYN Flood* and *Smurf* [2]. Nowadays, the most commonly used technique is the *Distributed Denial of Service (DDoS)* attack, which is a variation of *DoS* where the attack is carried out by multiple machines that are under the control of the attacker who is usually able to take control of several machines by exploiting security weaknesses [2]. Overall, *DoS* lead to damage in terms of time, money and reputation for an organization, so implementing network monitoring mechanism aimed at detection of *DoS* attacks is crucial.

1.1.4 Cooling system attack

Cooling system attacks pose severe security risks to Data Centers. Over the years, several authors have focused their attention on this specific kind of attacks, analyzing various aspects such as the threat model and the impact of thermal attacks. *Zhihui Shao et al.* [3] described the attacker's capabilities in a thermal attack scenario, indicating that the attacker runs power-intensive applications to increase server power consumption, resulting in a single server equipped with multiple CPUs and GPUs consuming a large amount of energy. These authors also divided the impact of thermal attacks in two main cathegories, namely:

- **Performance degradation**: when the server temperature exceeds a certain threshold a thermal emergency occurs. In this scenario the server runs in a low power state mode to avoid the temperature from rising further, which could lead to hardware damage. It is evident that this remediation strategy results in performance degradation;

- **System outage**: when a thermal emergency occurs and the server temperature keeps rising despite the load capping, the system automatically shuts down to prevent hardware damage. The consequences of a system outage depend on the nature of the application and may become catastrophic in the case of latency-critical applications.

The aforementioned work also explores thermal attack strategies and their feasibility aiming to describe a real-world attack scenario and highlight the motivations behind a thermal attack.

1.2 Motivations

There are two main reasons that led to the study of techniques based on *PUE* monitoring for cyberattacks detection. The first motivation concerns the ease of *PUE* calculation that does not require specific equipment apart from wattmeters that should be placed at various points within the Data Center, depending on the level of granularity desired in *PUE* calculation (as described in section 1.1.2). The second motivation is closely related to the cooling system attack scenario as it highlights the reasons why it is essential to be concerned about potential attacks on the Data Centers' management system. In recent years, several research studies have explored various methodologies to integrate Cloud and IoT solutions in the context of Data Centers monitoring and management. In 2016 *Q Liu et al.* [4] proposed an air conditioning system that includes both IoT sensors and cloud-based systems for Data Center management. Moreover, in 2020 *Ramphela et al.* [5] proposed an integrated monitoring system for Data Centers based on the development of various subsystems, such as an embedded system that includes sensors that gather monitoring data. In this context, the pervasive usage of IoT and Cloud technologies exposes Data Centers to additional risks. As reported in the research work by *Francesca Meneghello et al.* [6], the main risk associated with the IoT devices comes from the lack of the implementation of security mechanisms especially in cheaper devices that are widely spread. An unauthorized access to the Data Center's control system could lead to the modification of the cooling system's operational parameters. In cases where such violation may go unnoticed by the Intrusion Detection or Intrusion Prevention System, the proposed approach based on monitoring energy parameters can be a viable option for detecting such unauthorized access.

1.3 Results

The achieved results suggest that a correlation between *PUE* fluctuations and the presence of cyberattacks exists in both simulated scenarios. In particular, in the *DoS* scenario, there is a substantial variation in *PUE* up to a certain load threshold (approximately 80%), while in the cooling system attack scenario, the *PUE* variation is independent of the Data Center load, making the proposed approach particularly suitable for the detection of this specific attack. A detailed analysis of the results achieved is presented in Chapter 5, where the entire research work process is comprehensively discussed.

1.4 Structure of the Thesis

The Thesis is structured as follows:

- **Chapter 2 (State of the Art):** this chapter provides an overview about the currently available techniques of *DoS* and cooling system attack detection. Furthermore, in order to choose a viable platform to perform Data Center simulations, several Cloud simulators will be analyzed and compared;
- **Chapter 3 (GreenCloud Simulator Overview):** this chapter explores in depth the *GreenCloud* simulator as it has been chosen as the reference tool for the simulations;
- **Chapter 4 (Data Center Design):** this chapter describes the architecture, the energy model, the IT capacity and the power and cooling parameters of the virtual Data Center used for the performed simulations;
- **Chapter 5 (Research Study):** this chapter illustrates the research study process, starting from the changes implemented in *GreenCloud* in order to make it suitable for this work and then describing the performed simulations and the *PUE* calculation. The chapter

concludes with a discussion on the achieved results, providing an idea of how the variation of PUE can be a suitable parameter for attacks detection;

- **Chapter 6 (Conclusions and Future Work):** this chapter summarizes the research study and discusses potential future works.

CHAPTER 2

State of the art

2.1 Energy-related security threats

2.2 Simulation tools comparison

Simulation tools play a critical role in various domains of cloud computing. They offer researchers and infrastructure designers a virtual environment to work in, eliminating the expenses associated with physical infrastructure. Over time, several authors have conducted comparative studies to assess different simulators and highlight their unique characteristics, assisting users in selecting the most suitable option for their specific context. In this section, several cloud simulators that have been studied by the researchers will be discussed, focusing on their main strengths, drawbacks and the problems they aim to address. Since the main focus is to identify a simulator that is well-suited for energy consumption metrics extraction, features that ensure the accuracy of the simulations will be prioritized in order to make this work relevant in real scenarios. The following argument is based on several previous surveys, namely: *N. Mansouri, et al.* [7], *P. Suryateja* [8], *D. Perez Abreu, et al.* [9], *Khaled M. Khalil, et al.* [10], *Nimisha Patel & Hiren Patel* [11].

CloudSim

When it comes to cloud simulators it is impossible to overlook *CloudSim* [12], as it is one of the most widely used *event-driven* tools among researchers. A search for the keyword "*CloudSim*" on the *Scopus* platform yields 2020 results (this search was performed in May 2023), indicating its widespread adoption in the research community. Over the years, several simulators based on *CloudSim* have been proposed. This tool, written in *Java*, is comprehensive and highly extensible. One notable feature is its virtualization engine, which allows the creation and management of virtualization services on a network node. Additionally, *CloudSim* provides the capability to allocate machine cores in two different ways: space-shared and time-shared. In space-shared allocation, each machine is divided into a set of cores, and each core is assigned to a single job until it is completed. On the other hand, in time-shared allocation multiple jobs can be assigned to a single core and each of them is executed for a certain amount of time before another job is chosen. Overall, *CloudSim* offers researchers a comprehensive and flexible platform to simulate and study virtualization and resource allocation techniques [7]. Architecture of *CloudSim* is shown in figure 2.1. It is composed by three layers, described below:

- **CloudSim core simulation engine**: initially implemented through the discrete event simulation engine *SimJava* that supports functionalities such as queuing, processing of events, creation of Cloud system entities, communication between components and management of the simulation clock ([12]), it has been replaced with an engine that provides some advanced operations;

- **CloudSim simulation level**: it provides various interfaces and services that allow to model and simulate Cloud-based data center environments;
- **User code**: it allows to set up various simulation parameters such as number of machines, their specification, number of tasks and broker scheduling policies.

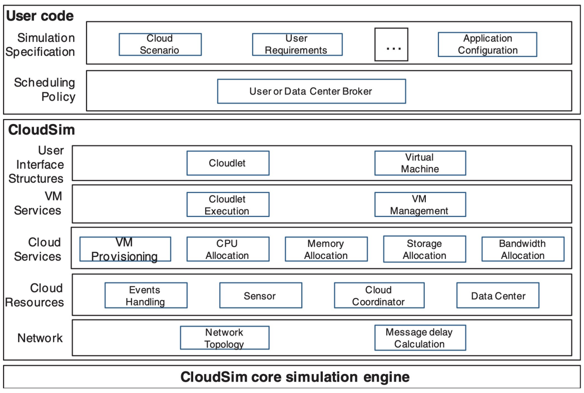

Figure 2.1: CloudSim architecture

NetworkCloudSim

NetworkCloudSim [13] implements a network layer and provides several communication models, including message-based, packet-based and flow-based models. It also offers an accurate evaluation of the scheduling of machines in the data center. Finally, it provides a basic energy model of the data center, although it does not extensively focus on energy efficiency and it does not provide a packet-level network model [7] [11].

CloudAnalyst

CloudAnalyst [14] makes simulation work easier as it provides a *GUI*. Its main features is the ability to gather information about the geographical location of users and data centers. The simulator also offers a set of metrics based on response time and request processing. However, it does not implement a complete communication model based on the *TCP/IP* protocol and it provides a poor energy model [7].

EMUSim

EMUSim [15] includes an emulation and a simulation environment. The simulator is able to get profiling data about the running application behavior through the emulation environment; these data are used to build a simulation environment. However, the emulation environment strongly limits scalability and makes the simulator not suitable for large workload scenarios [7].

CDOSim

CDOSim [16] integrates *CloudSim* simulator and *CloudMIG* framework. This simulator adopts a scaling technique that assigns a new virtual machine when CPU usage exceeds a specific threshold. It provides a set of client-centric metrics, allowing developers to compare different solutions based on various deployment parameters. Another noteworthy feature is the presence of a benchmark module that detects the impact of choosing a specific architecture on application performance. However, the communication model employed by CDOSim is overly simplistic and strongly limits large-scale applications [7].

TeachCloud

TeachCloud [17] is designed to be suitable for students who want to have some practical experience with Cloud Computing simulation, covering aspects such as networking, Service Level Agreement contraints, web-based applications, Service Oriented Architecture, virtualization, and so on. Through a user-friendly *GUI* it allows to design several network architectures in addition to the pre-existing ones. This simulator lacks realism in various aspects since it is mainly designed for academic purposes. For example, the simulator does not consider the possibility of faults in the data center, preventing developers from studying the impact of faults on application usage [7] [11].

DartCSim

DartCSim [18] provides a simple *GUI* that allows users to set various simulation parameters, such as the characteristics of the data center and the network topology. These parameters can be imported and exported at any level of the simulation and can be set for individual CPUs or the entire data center. However, this tool lacks a comprehensive energy model, which prevents developers from implementing strategies aimed at improving data center efficiency [7].

DartCSim+

DartCSim+ [19] aims to enhance *CloudSim* by introducing an energy model and a network model, allowing developers to design power-aware scheduling algorithms. However, this simulator does not include a cost model and lacks security features, which prevents developers from analyzing the security aspects of the data center [7].

ElasticSim

ElasticSim's [20] main feature is the resources runtime auto-scaling based on stochastic modeling that allows developers to design efficient scheduling algorithms. However, this simulator provides a poor energy consumption model and it cannot simulate security-related experiments. [7]

FederatedCloudSim

FederatedCloudSim [21] provides developers with the capability to test several types of cloud federations; this simulator is built upon the functionalities of *CloudSim* and expands them by incorporating Service Level Agreement management, workload generation, event logging, scheduling, and brokering. While the simulator offers comprehensive functionality for modeling and simulating cloud federations, it does not provide specific insights into the energy consumption of each data center in the federation [7].

FTCloudSim

FTCloudSim [22] is primarily designed to simulate reliability mechanisms in cloud services. It offers implementations of various reliability mechanisms, allowing developers to analyze and evaluate the performance of these mechanisms. The simulator also includes fault generation services, which enable the generation of faults based on specific probability distributions. However, one drawback of FTCloudSim is its simplified energy consumption model, which can impact the assessment of energy-efficient strategies or algorithms [7].

WorkflowSim

WorkflowSim [23] provides a platform for studying the performance impact of different job clustering strategies in a data center. It achieves this by implementing various workflow scheduling methods. However, WorkflowSim has limitations when it comes to simulating data-intensive applications. It does not consider the delays caused by input-output operations, which are crucial in such scenarios. Additionally, the supported fault model in WorkflowSim is limited, which can affect the realism of simulations. As a result, the accuracy and realism of simulations involving data-intensive applications may be compromised [7].

CloudReports

CloudReports [24] provides developers with a user-friendly *GUI* that allows them to manage various aspects of the data center and access detailed reports on resource utilization, virtual machine allocation, and energy consumption. Since the simulator offers valuable insights into these metrics, it enables developers to optimize resource usage and energy efficiency. On the other hand, one limitation of *CloudReports* is the absence of a security layer: this means that developers cannot explore and evaluate the security characteristics of the data center using this simulator [7].

CEPSim

CEPSim [25] models various *Complex Event Processing (CEP)* environments [26] where users can define queries using different proprietary languages and model the execution flow using a directed acyclic graph. The simulator implements various load scheduling algorithms, allowing developers to evaluate queries under different load conditions. However, one limitation of *CEPSim* is that it does not consider network consumption. As a result, the simulator may not provide a comprehensive analysis of energy consumption from a network perspective. Other factors such as network transmission impact on energy consumption are not explicitly taken into account [7].

DynamicCloudSim

DynamicCloudSim [27] is primarily concerned with addressing the instability of computing center parameters that can change during runtime. It specifically introduces failure models for task execution, allowing developers to define the failure rate when conducting experiments in a simulated environment. However, one limitation is that developers do not have the capability to calculate the energy consumption of the experiments accurately due to the restricted nature of the provided energy model; moreover, its failure model is strongly limited. [7] [8]

CloudExp

CloudExp [28] provides developers with a simple *GUI* that allows easy configuration of environment parameters and monitoring of their behavior. Specifically, CloudExp enables the definition of a Service Level Agreement (SLA) based on parameters such as the number of users, service availability and cost, network performance, and security measures. Additionally, it establishes a specialized framework for cloud computing in mobile device scenarios. However, the energy model used by the simulator is simplistic and not suitable for analyzing energy-aware strategies [7] [10].

CM Cloud

CM Cloud [29] is able to estimate overall energy simulation expenses through a comparison between several providers such as *Google*, *Microsoft* and *Amazon*. However, this simulator lacks a complete communication model, not allowing developers to define specific traffic patterns or investigate the overall impact of the traffic generated by the hosts of the network. Finally, there is no task failure model [7].

MR-CloudSim

MR-CloudSim [30] is based on the *MapReduce* computational model [31] for big data computation, allowing developers to work with data-intensive applications. One limitation is that it does not allow developers to accurately calculate service usage expenses as it does not consider file processing time and cost [7].

UCloud

UCloud [32] is designed to address educational purposes as it simulates cloud for universities, allowing developers to evaluate several policies in the public and private clouds. However, this simulator lacks a support for security policies and a cost model. [7]

CloudSimSDN

CloudSimSDN [33] is built for cloud environments based on *Software Defined Networking*, namely a programmatic approach to network management [34]. It is a scalable simulator that allows to manage energy consumption and resource policies as well as investigate several metrics such as performance. Its major drawback is that it models applications through a set of tasks and assumes long packet transmission between VMs, which can impact the granularity of application communication, making it not well-suited for works based on applications. [10] [9]

MDCSim

MDCSim [35] is a simulation platform for multilayer data centers analysis that allows developers to investigate applications performance under different loads and tier configurations as it has low simulation overhead. *MDCSim* is able to estimate several parameters, such as throughput, response time and power consumption, so developers can compare different energy policies. However, it lacks simulation realism as it does not provide a complete network model [7] [11].

GDCSim

GDCSim [36] is an open source, event-based simulator written in *C, C++ and Shell*, as part of the *BlueTool* computer infrastructure project funded by *NSF*. It is designed to analyze green data centers as it provides a simulation environment where it is possible to analyze the energy consumption of the data center in a simple and accurate way. This tool also takes into account the thermal impact of the data center, giving developers the ability to design cooling policies and energy management strategies with a particular attention to the Computational Fluid Dynamics (CFD) that allows to characterize the thermal effects and airflow patterns. However, this simulator does not consider aspects related to the security of the data center.

Moreover it does not allow parallel execution of defined experiments [7]. Architecture of *GDCSim* is shown in figure 2.2. It is composed by four modules:

1. **BlueSim Tool**: a simulation package which integrates various software for HRMs (heat recirculation matrix) array generation;

2. **Input/Output Management**: the interface between the user and the simulator. It takes the following inputs: job trace, Service Level Agreement, management schemes and queuing model;

3. **Resource Management**: module that implements the following algorithms: workload management, power management, cooling management and coordinated workload, power and cooling management;

4. **Simulator**: it provides several modules such as the queuing module, the power module, the thermodynamic module and the cooling module.

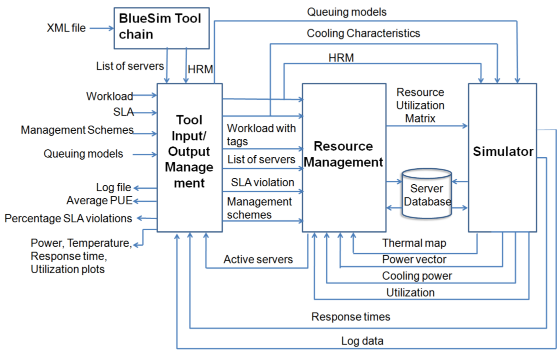

Figure 2.2: Architecture of the GDCSim simulation environment

CloudNetSim

CloudNetSim [37] models end-to-end communication between clients and servers. Its extensibility and modularity make it possible to integrate different modules and evaluate different VM deployment algorithms. This simulator provides a platform that allows developers to investigate resource management for realistic cloud applications. However, *CloudNetSim* implements a poor energy and thermal model that makes energy management algorithms implementation difficult. [7]

CloudNetSim++

CloudNetSim++ [38] is an open source simulator built on the top of *OMNET++*, written in C++. It introduces the concept of distributed data centers connected with physical network through various topologies. This simulation tool has a modular architecture that allows researchers to explore different aspects of data centers and to extend network topology by adding switches at the aggregation and core levels. *CloudNetSim++* provides a platform to analyze energy consumption during the simulation and an energy-aware scheduler which supplies several techniques such as Dynamic Voltage and Frequency Scaling. *CloudNetSim++* architecture is shown in figure 2.3 and consists of five modules:

- **Pricing Policy Manager**: computes the billing cost for each user request based on the agreement;
- **Cloud Usage Monitor**: analyzes usage patterns;
- **Task Scheduling Selection Module**: determines the scheduling policy;
- **VM Manager**: Determines VM assignments according to the received SLA requests;
- **User Task Scheduler**: receives all incoming user requests and distributes them to the appropriate VMs.

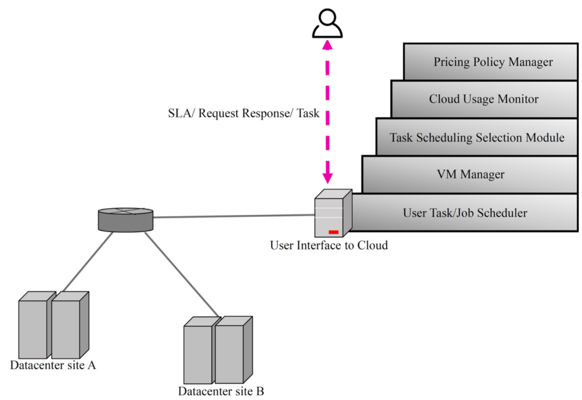

Figure 2.3: Architecture of the CloudNetSim++ simulation environment

GreenCloud

GreenCloud [39] is an open source simulator, written in *C++* and *OTcl* and built on the top of *NS2* that enables researchers to study the energy consumption of data centers. *GreenCloud* operates at packet level, as it simulates the behavior of individual network packets and their interactions within the TCP/IP protocol suite that is fully implemented by this simulator. *GreenCloud* aims to provide insights into the energy usage of various components within a data center, including servers, switches, and network links and allows users to evaluate the effectiveness of different energy-saving techniques and algorithms, by accurately modeling the energy consumption models. However, despite its utility in energy monitoring, *GreenCloud* has encountered challenges related to scalability because simulation times in *GreenCloud* tend to be relatively long. This simulator also requires significant memory resources, which can pose constraints on the size and complexity of the simulated scenarios.

These scalability limitations can make it difficult to analyze large-scale data center networks or evaluate energy-efficient protocols and algorithms in a timely manner. [7] The structure of *GreenCloud* simulation environment mapped onto the three-tier data center architecture is shown in figure 2.4.

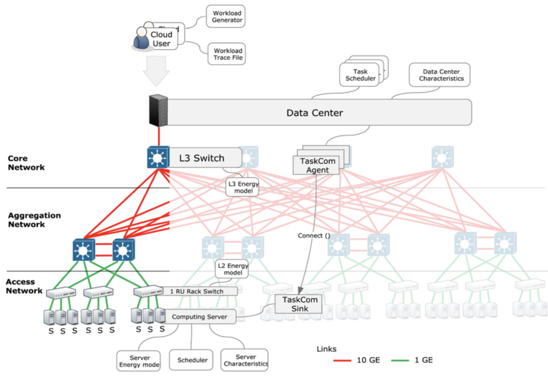

Figure 2.4: Architecture of the GreenCloud simulation environment

iCanCloud

iCanCloud [40] offers a dedicated *GUI* that allows users to configure the computing center and obtain graphical reports. *iCanCloud* supports the implementation of various brokering strategies, providing the flexibility to define different brokers connecting users and the computing center. However, this simulator does not address aspects related to the energy model and security [7].

secCloudSim

secCloudSim [41] is an extension of *iCanCloud* that implements simple security mechanisms lacking in its predecessor. Among the various security features, it includes an authentication protocol and *Access Control List (ACL)* that allows associating specific privileges with each authenticated user. *secCloudSim* provides a framework where researchers can develop security characteristics such as encryption, decryption, encapsulation, authentication, and privacy. However, the security mechanisms provided by *secCloudSim* are not very advanced, limiting researchers from studying infrastructure vulnerabilities [7].

GroudSim

GroudSim [42] is a *Java-based* simulator primarily used in the context of scientific applications. Developers can import *ASKALON* experiments [43] into this simulator to conduct simulations of real applications. However, this simulator lacks realism as it does not allow the configuration of a realistic network topology and does not scale effectively [7].

CloudSched

CloudSched [44] implements various energy-efficient algorithms and resource scheduling strategies for physical and virtual machines to avoid bottlenecks. It allows developers to define custom resource scheduling algorithms as needed. However, *CloudSched* does not consider task failures, making it unable to implement fault-tolerance strategies [7].

SimIC

SimIC [45] focuses mainly on the heterogeneity of environments in which experiments are executed. Developers can define inter-cloud scheduling algorithms based on various distributed parameters. However, the main limitations of this simulator are that it does not allow investigation of energy consumption, traffic controls, and congestions [7].

SPECI

SPECI [46] models aspects related to the scalability and performance of computing centers, allowing developers to monitor the system's behavior by varying its architecture. The simulator also enables investigation of inconsistencies that may arise in the event of failures. However, it does not model changes in the performance of virtual machines during execution, despite this factor being particularly relevant in real-world scenarios [7].

SCORE

SCORE [47] is well-suited for defining energy-aware scheduling algorithms, such as mechanisms for shutting down and powering on resources. However, it lacks a security module that would allow developers to investigate fundamental aspects of the computing center related to security [7].

GAME-SCORE

The GAME-SCORE simulator [48] is an extension of the SCORE simulator written in *Scala*, which uses a combination of discrete-event and multi-agent simulation approaches. Its primary purpose is to simulate energy-efficient IaaS in cloud environments. This simulation tool provides the flexibility to dynamically select energy-efficiency policies from a range of options, allowing for the shutdown of idle machines during runtime. As a practical example, it introduces an algorithm based on the Stackelberg Game that utilizes this capability. However, it's important to note that this simulation tool can accommodate other strategies as well. These strategies can involve the dynamic switching between various energy-efficiency policies and scheduling algorithms. The versatility of this tool enables the implementation of different approaches to optimize energy consumption in simulated environments. Architecture of GAME-SCORE is shown in figure 2.5. It is composed of two modules, described below:

- **Core Simulator Module**: the module responsible for executing the experiments composed of a workload generator, a core engine and a scheduling module;

- **Energy-Efficiency Module**: the module responsible for the implementation of the energy-efficiency policies.

It is additionally composed of a special module that implements the Stackelberg Game in order to dynamically switch between energy-efficiency policies.

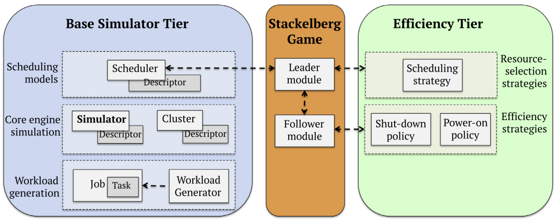

Figure 2.5: GAME-SCORE architecture

DISSECT-CF

DISSECT-CF [49] is an event-based open source simulator written in Java which aims to offer energy-aware scheduling for infrastructure clouds. In contrast to other recently developed simulators, *DISSECT-CF* takes a unique approach by separating energy modeling from resource simulation. This enables the inclusion of energy consumption that may not be directly related to the utilization of data center resources. By decoupling energy modeling, *DISSECT-CF* allows for more comprehensive energy and power modeling, facilitating the analysis of sophisticated energy-aware algorithms in areas such as virtual machine placement and task scheduling. Architecture of *DISSECT-CF* is shown in figure 2.6 and it is composed of five modules decribed below:

- **Event system**: this component serves as the time reference for simulations;

- **Unified resource sharing**: this subsystem establishes a flexible and lightweight foundation for sharing low-level computing resources, such as CPU and I/O;

- **Energy modeling**: *DISSECT-CF* includes components that allow simulator developers to monitor and analyze energy usage patterns of specific simulated resources, such as network links and disks;

- **Infrastructure simulation**: these components govern the behavior and interactions of various elements within the infrastructure, for example the virtual machines;

- **Infrastructure management**: this subsystem offers a user interface, and encompasses higher-level functionalities like virtual machine schedulers of infrastructure clouds.

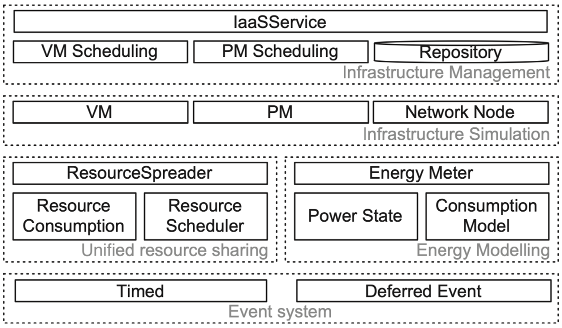

Figure 2.6: Architecture of the DISSECT-CF simulation environment

ICARO

ICARO [50] aims to analyze changes in the load of a computing center when the load structure dynamically varies at runtime [10].

SmartSim

SmartSim [51] simulates the behavior of mobile devices and resource-intensive applications [10]. With *SmartSim*, it is possible to model both system components and their behavior in terms of resource allocation and management [8].

PICS

PICS [52] is a simulator designed to evaluate the cost and performance of public IaaS (Infrastructure as a Service) while considering factors related to resource management and job scheduling. However, this simulator lacks a model for communication costs [8].

2.3 Choice of simulation tool

As discussed earlier, this research work requires accurate insights about the energy consumption of the Data Center components. This need led to the selection of the five tools between the ones analyzed in the previous section, that are most focused on analyzing energy consumption, namely: *CloudSim*, *DISSECT-CF*, *GreenCloud*, *GDCSim* and *GAME-SCORE*. Table 2.1 shows a summary of the selected simulators. As it can be observed, *GDCSim* is stated to be not available. According to the original paper about *GDCSim* [36], it was mentioned that the simulator would have been available on the *BlueTool* platform once it was ready, however, this platform is no longer accessible. The rest of the information has been gathered from the original papers related to the simulators, namely [33], [49], [39], [36], [53] and [48].

Table 2.1: Simulation tools summary

Simulator	Simulation type	Language	Availability
CloudSim	Event-based	Java	Open source
DISSECT-CF	Event-based	Java	Open source
GreenCloud	Packet-level	C++, oTcl	Open source
GDCSim	Event-based	C, C++, Shell	Not available
GAME-SCORE	Event-based and multi-agent	Scala, Java, Python	Open source

2.3.1 Comparison parameters

The selected simulators have been evaluated based on several parameters, including:

- **Last update**: it provides an indication of how well-supported and updated the simulator is;
- **Popularity**: it gives an idea of how popular the simulator is;
- **Availability**: it indicates the availability of the simulator;
- **Granularity**: it suggests the level of detail at which the elements of the data center and simulation can be defined;
- **Performance profile**: it provides an indication of the performance of the simulator;

2.3.2 Simulators evaluation

The following sections describe the simulators from the perspective of the evaluation parameters. Each feature of the simulators has been assigned a score on a scale from 1 to 5, in order to highlight strengths and weaknesses. The evaluation results are presented in the radar chart shown in figure 2.7.

Last update

CloudSim's, *GAME-SCORE*'s and *DISSECT-CF*'s last update (2020, 2018 and 2023 respectively) were gathered from the latest commits on their GitHub repositories[1]. *GreenCloud*'s latest files modification year is 2016 (the project has been download from the simulator platform[2]). Since *GDCSim* is not available, we can assume that its last update corresponds to the year of the last paper published about it [53], which is 2014.

[1]CloudSim: https://github.com/Cloudslab/cloudsim,
GAME-SCORE: https://github.com/DamianUS/game-score
DISSECT-CF: https://github.com/kecskemeti/dissect-cf

[2]http://greencloud.gforge.uni.lu/ftp/greencloud-v2.1.2.tar.gz

Popularity

Popularity has been evaluated through the citation numbers that have been obtained from the *SCOPUS* platform:

- CloudSim: 2043 citations;
- DISSECT-CF: 23 citations;
- GreenCloud: 31 citations;
- GDCSim: 3 citations;
- GAME-SCORE: 1 citation;

Availability

Tools availability is reported in table 2.1.

Granularity

According to the original papers about the analyzed simulators:

- *CloudSim* allows to create energy-conscious provisioning policies by overriding the method *getPower()* of the abstract class *PowerModel* whose input parameter is the utilization metric for Cloud host and return parameter is the power consumption value. *CloudSim* also provides a VM Allocation controller component (*VmAllocationPolicy*) that exposes some custom methods which can be used by developers in order to implement new policies based on several optimization goals, and a VM Scheduler component which can be extended to test several allocation policies;
- *DISSECT-CF* allows developers to define various energy consumption models based on simulation entities' power state. In order to compute energy consumption accurately, this simulator uses several meters and allows developers to define an aggregation function that addresses the dependency between two metered components (e.g., a virtual machine and the physical one where it is hosted);
- *GreenCloud* fully implements the *TCP/IP* protocol. This simulator is built on top of the *NS2* simulator, allowing customization of network topology and enabling work at the packet-level to implement specific traffic patterns and model real-world scenarios. Furthermore, thanks to its open-source nature, it allows the implementation of various energy management and workload scheduling algorithms;
- *GDCSim* architecture was designed to be modular and extensible in order to easily plug new components into the simulator and to perform various analysis under different

physical configurations. For example *GDCSim* allows to replace the cooling model with a user-defined one and to add new power consumption and resource management models;

- *GAME-SCORE*'s main aim is to dynamically apply energy policies during simulations by enabling us to dynamically choose between a catalog of energy-efficiency policies that shut-down idle machines in runtime. This simulator allows to implement strategies to dynamically switch between a set of energy efficiency policies and scheduling algorithms during simulations.

Performance profile

- *CloudSim*'s authors conducted various tests in order to analyze the performance. They found out that the time to instantiate an experiment setup with 1 million hosts is around 12 s. Moreover, they observed that the total memory usage never grew beyond 320 MB even for larger system sizes;
- *DISSECT-CF*'s authors performed several tests in order to evaluate the performance. In the original paper it was stated that *DISSECT-CF* scales comparably to other state-of-the-art simulators, such as *CloudSim*, since it never drops below linear scaling;
- Since *GreenCloud* is a packet-level simulator, its simulation time is significantly higher compared to the other state-of-the-art simulators. Various simulators comparative analysis reported the simulation time as a drawback of *GreenCloud*. In particular [10] states that *GreenCloud* simulation time is on the order of minutes;
- The authors of *GDCSim* conducted several large-scale experiments to validate this simulator. They discovered that each *GDCSim* simulation, which involved *HRM* generation taking 775 minutes, required less than 1 minute to complete. In comparison, corresponding *CFD* simulations took 30 minutes. Therefore, despite the initial cost of generating the *HRM*, *GDCSim* outperforms *CFD* simulations in terms of runtime in the long run [54].
- The authors of *GAME-SCORE* did not provide an insight about their simulator's performance in the original paper [48]. Moreover, none of the previously mentioned works about simulators comparison investigate *GAME-SCORE* performance, so the evaluation of this parameter is not possible.

2.3.3 Selection of GreenCloud

Based on the comparisons made among the different analyzed simulators, each of them has its strengths and weaknesses, however, for the needs required in the study addressed

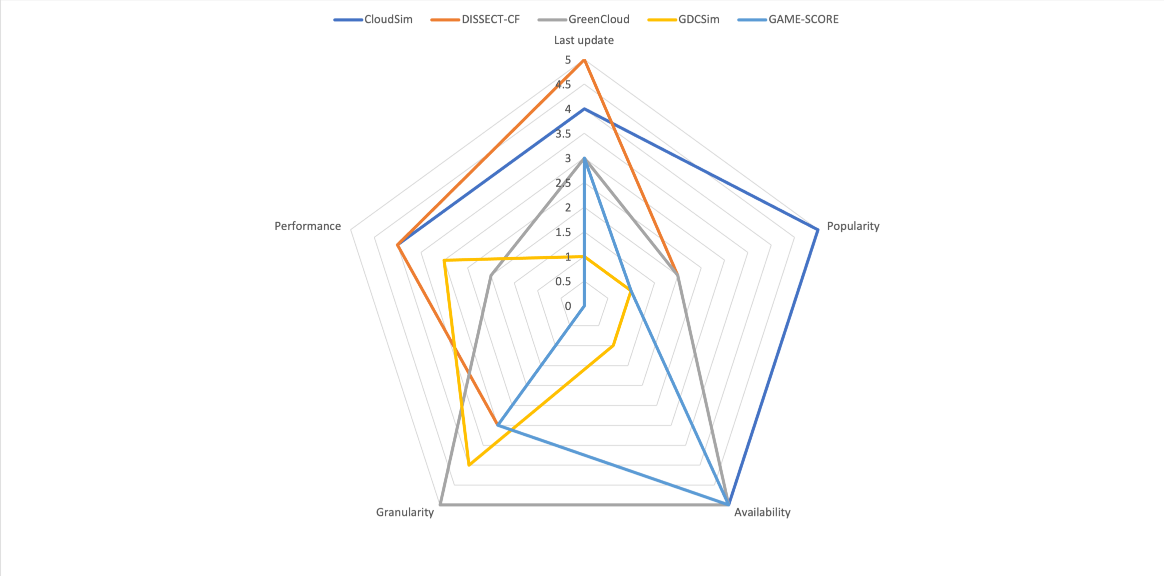

Figure 2.7: Simulators scoring

by this thesis work, it is essential to prioritize aspects related to the energy consumption of the computing center and the accuracy of simulations. From the conducted comparisons, it is evident that *GreenCloud* is the simulator that accurately considers these aspects as it is built on top of *NS2* simulator and fully implements the *TCP/IP* protocol. Despite *GreenCloud* simulation times tend to be high, for the purposes of this study it is reasonable to prioritize granularity over performance. Therefore, the study will continue using *GreenCloud* as the reference tool for the simulations to be conducted.

CHAPTER 3

GreenCloud Simulator Overview

3.1 Main features

3.2 Energy model of switches, CPU, memory and disks

3.3 Available Data Center architectures

3.4 Workload scheduling algorithms comparison

CHAPTER 4

Data Center Design

4.1 Architecture

4.2 CPU energy model

4.3 IT capacity calculation

4.4 Power and cooling parameters

CHAPTER 5

Research Study

5.1 Changes implemented in GreenCloud

5.2 Executed simulations

5.2.1 Normal load simulation

5.2.2 High load simulation

5.2.3 DDoS scenario simulation

5.2.4 Cooling system attack scenario simulation

5.3 PUE formula approximation

5.3.1 Approximation under normal Data Center configuration

5.3.2 Approximation under altered Data Center configuration

5.4 PUE calculation in the executed simulations

5.5 Results

5.5.1 PUE utilization in DDoS scenarios

5.5.2 PUE utilization in cooling system attack scenarios

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

6.2 Future Work

Acknowledgements

Bibliography

[1] V. Avelar, D. Azevedo, A. French, and E. N. Power, "Pue: a comprehensive examination of the metric," *White paper*, vol. 49, 2012. (Citato a pagina 2)

[2] "Understanding denial-of-service attacks," Feb 2021. (Citato alle pagine 3 e 4)

[3] Z. Shao, M. A. Islam, and S. Ren, "Heat behind the meter: A hidden threat of thermal attacks in edge colocation data centers," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 318–331, 2021. (Citato a pagina 4)

[4] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with iot sensing and cloud-assisted smart temperature control system," *Computer Networks*, vol. 101, pp. 104–112, 2016. (Citato a pagina 4)

[5] M. K. J. Ramphela, P. A. Owolawi, T. Mapayi, and G. Aiyetoro, "Internet of things (iot) integrated data center infrastructure monitoring system," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–6, IEEE, 2020. (Citato a pagina 4)

[6] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019. (Citato a pagina 5)

[7] N. Mansouri, R. Ghafari, and B. M. H. Zade, "Cloud computing simulators: A comprehensive review," *Simulation Modelling Practice and Theory*, vol. 104, p. 102144, 2020. (Citato alle pagine 8, 9, 10, 11, 12, 13, 14, 16 e 17)

[8] P. S. Suryateja, "A comparative analysis of cloud simulators.," *International Journal of Modern Education & Computer Science*, vol. 8, no. 4, 2016. (Citato alle pagine 8, 12 e 19)

[9] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A comparative analysis of simulators for the cloud to fog continuum," *Simulation Modelling Practice and Theory*, vol. 101, p. 102029, 2020. (Citato alle pagine 8 e 13)

[10] K. M. Khalil, M. Abdel-Aziz, T. T. Nazmy, and A.-B. M. Salem, "Cloud simulators–an evaluation study," *International Journal Information Models and Analyses*, vol. 6, no. 1, 2017. (Citato alle pagine 8, 12, 13, 19 e 22)

[11] N. Patel and H. Patel, "A comprehensive assessment and comparative analysis of simulations tools for cloud computing," *Int J Eng Comput Sci*, vol. 5, no. 11, pp. 18972–18978, 2016. (Citato alle pagine 8, 9, 10 e 13)

[12] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011. (Citato a pagina 8)

[13] S. K. Garg and R. Buyya, "Networkcloudsim: Modelling parallel applications in cloud simulations," in *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pp. 105–113, IEEE, 2011. (Citato a pagina 9)

[14] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in *2010 24th IEEE international conference on advanced information networking and applications*, pp. 446–452, IEEE, 2010. (Citato a pagina 9)

[15] R. N. Calheiros, M. A. Netto, C. A. De Rose, and R. Buyya, "Emusim: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications," *Software: Practice and Experience*, vol. 43, no. 5, pp. 595–612, 2013. (Citato a pagina 9)

[16] F. Fittkau, S. Frey, and W. Hasselbring, "Cdosim: Simulating cloud deployment options for software migration support," in *2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)*, pp. 37–46, IEEE, 2012. (Citato a pagina 10)

[17] Y. Jararweh, Z. Alshara, M. Jarrah, M. Kharbutli, and M. N. Alsaleh, "Teachcloud: a cloud computing educational toolkit," *International Journal of Cloud Computing 1*, vol. 2, no. 2-3, pp. 237–257, 2013. (Citato a pagina 10)

[18] X. Li, X. Jiang, P. Huang, and K. Ye, "Dartcsim: An enhanced user-friendly cloud simulation system based on cloudsim with better performance," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, vol. 1, pp. 392–396, IEEE, 2012. (Citato a pagina 10)

[19] X. Li, X. Jiang, K. Ye, and P. Huang, "Dartcsim+: Enhanced cloudsim with the power and network models integrated," in *2013 IEEE Sixth International Conference on Cloud Computing*, pp. 644–651, IEEE, 2013. (Citato a pagina 10)

[20] Z. Cai, Q. Li, and X. Li, "Elasticsim: A toolkit for simulating workflows with cloud resource runtime auto-scaling and stochastic task execution times," *Journal of Grid Computing*, vol. 15, pp. 257–272, 2017. (Citato a pagina 10)

[21] A. Kohne, M. Spohr, L. Nagel, and O. Spinczyk, "Federatedcloudsim: a sla-aware federated cloud simulation framework," in *Proceedings of the 2nd International Workshop on CrossCloud Systems*, pp. 1–5, 2014. (Citato a pagina 11)

[22] A. Zhou, S. Wang, Q. Sun, H. Zou, and F. Yang, "Ftcloudsim: A simulation tool for cloud service reliability enhancement mechanisms," in *Proceedings Demo & Poster Track of ACM/IFIP/USENIX International Middleware Conference*, pp. 1–2, 2013. (Citato a pagina 11)

[23] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," in *2012 IEEE 8th international conference on E-science*, pp. 1–8, IEEE, 2012. (Citato a pagina 11)

[24] T. Teixeira Sá, R. N. Calheiros, and D. G. Gomes, "Cloudreports: an extensible simulation tool for energy-aware cloud computing environments," *Cloud computing: Challenges, limitations and R&D solutions*, pp. 127–142, 2014. (Citato a pagina 11)

[25] W. A. Higashino, M. A. Capretz, and L. F. Bittencourt, "Cepsim: a simulator for cloud-based complex event processing," in *2015 IEEE International Congress on Big Data*, pp. 182–190, IEEE, 2015. (Citato a pagina 12)

[26] D. C. Luckham and B. Frasca, "Complex event processing in distributed systems," *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford*, vol. 28, p. 16, 1998. (Citato a pagina 12)

[27] M. Bux and U. Leser, "Dynamiccloudsim: Simulating heterogeneity in computational clouds," in *Proceedings of the 2nd acm sigmod workshop on scalable workflow execution engines and technologies*, pp. 1–12, 2013. (Citato a pagina 12)

[28] Y. Jararweh, M. Jarrah, Z. Alshara, M. N. Alsaleh, M. Al-Ayyoub, *et al.*, "Cloudexp: A comprehensive cloud computing experimental framework," *Simulation Modelling Practice and Theory*, vol. 49, pp. 180–192, 2014. (Citato a pagina 12)

[29] D. C. Alves, B. G. Batista, D. M. Leite Filho, M. L. Peixoto, S. Reiff-Marganiec, and B. T. Kuehne, "Cm cloud simulator: a cost model simulator module for cloudsim," in *2016 IEEE World Congress on Services (SERVICES)*, pp. 99–102, IEEE, 2016. (Citato a pagina 12)

[30] J. Jung and H. Kim, "Mr-cloudsim: Designing and implementing mapreduce computing model on cloudsim," in *2012 International Conference on ICT Convergence (ICTC)*, pp. 504–509, IEEE, 2012. (Citato a pagina 13)

[31] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008. (Citato a pagina 13)

[32] M. H. Sqalli, M. Al-Saeedi, F. Binbeshr, and M. Siddiqui, "Ucloud: A simulated hybrid cloud for a university environment," in *2012 IEEE 1ST International Conference on Cloud Networking (CLOUDNET)*, pp. 170–172, IEEE, 2012. (Citato a pagina 13)

[33] J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon, and R. Buyya, "Cloudsimsdn: Modeling and simulation of software-defined cloud data centers," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 475–484, IEEE, 2015. (Citato alle pagine 13 e 19)

[34] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (sdn): a survey," *Security and communication networks*, vol. 9, no. 18, pp. 5803–5833, 2016. (Citato a pagina 13)

[35] S.-H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "Mdcsim: A multi-tier data center simulation, platform," in *2009 IEEE International Conference on Cluster Computing and Workshops*, pp. 1–9, IEEE, 2009. (Citato a pagina 13)

[36] S. K. Gupta, R. R. Gilbert, A. Banerjee, Z. Abbasi, T. Mukherjee, and G. Varsamopoulos, "Gdcsim: A tool for analyzing green data center design and resource management techniques," in *2011 International Green Computing Conference and Workshops*, pp. 1–8, IEEE, 2011. (Citato alle pagine 13 e 19)

[37] T. Cucinotta and A. Santogidis, "Cloudnetsim-simulation of real-time cloud computing applications," in *Proceedings of the 4th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems*, 2013. (Citato a pagina 14)

[38] A. W. Malik, K. Bilal, K. Aziz, D. Kliazovich, N. Ghani, S. U. Khan, and R. Buyya, "Cloudnetsim++: A toolkit for data center simulations in omnet++," in *2014 11th Annual High Capacity Optical Networks and Emerging/Enabling Technologies (Photonics for Energy)*, pp. 104–108, IEEE, 2014. (Citato a pagina 14)

[39] D. Kliazovich, P. Bouvry, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, vol. 62, pp. 1263–1283, 2012. (Citato alle pagine 15 e 19)

[40] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "icancloud: A flexible and scalable cloud infrastructure simulator," *Journal of Grid Computing*, vol. 10, pp. 185–209, 2012. (Citato a pagina 16)

[41] U. U. Rehman, A. Ali, and Z. Anwar, "seccloudsim: secure cloud simulator," in *2014 12th International Conference on Frontiers of Information Technology*, pp. 208–213, IEEE, 2014. (Citato a pagina 16)

[42] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, "Groudsim: An event-based simulation framework for computational grids and clouds," in *Euro-Par 2010 Parallel Processing Workshops: HeteroPar, HPCC, HiBB, CoreGrid, UCHPC, HPCF, PROPER, CCPI, VHPC, Ischia, Italy, August 31–September 3, 2010, Revised Selected Papers 16*, pp. 305–313, Springer, 2011. (Citato a pagina 16)

[43] T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. Seragiotto Jr, and H.-L. Truong, "Askalon: a tool set for cluster and grid computing," *Concurrency and Computation: Practice and Experience*, vol. 17, no. 2-4, pp. 143–169, 2005. (Citato a pagina 16)

[44] W. Tian, Y. Zhao, M. Xu, Y. Zhong, and X. Sun, "A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 153–161, 2013. (Citato a pagina 17)

[45] S. Sotiriadis, N. Bessis, N. Antonopoulos, and A. Anjum, "Simic: Designing a new inter-cloud simulation platform for integrating large-scale resource management," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 90–97, IEEE, 2013. (Citato a pagina 17)

[46] I. Sriram, "Speci, a simulation tool exploring cloud-scale data centres," in *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings 1*, pp. 381–392, Springer, 2009. (Citato a pagina 17)

[47] D. Fernández-Cerero, A. Fernández-Montes, A. Jakóbik, J. Kołodziej, and M. Toro, "Score: Simulator for cloud optimization of resources and energy consumption," *Simulation Modelling Practice and Theory*, vol. 82, pp. 160–173, 2018. (Citato a pagina 17)

[48] D. Fernández-Cerero, A. Jakóbik, A. Fernández-Montes, and J. Kołodziej, "Game-score: Game-based energy-aware cloud scheduler and simulator for computational clouds," *Simulation Modelling Practice and Theory*, vol. 93, pp. 3–20, 2019. (Citato alle pagine 17, 19 e 22)

[49] G. Kecskemeti, "Dissect-cf: a simulator to foster energy-aware scheduling in infrastructure clouds," *Simulation Modelling Practice and Theory*, vol. 58, pp. 188–218, 2015. (Citato alle pagine 18 e 19)

[50] C. Badii, P. Bellini, I. Bruno, D. Cenni, R. Mariucci, and P. Nesi, "Icaro cloud simulator exploiting knowledge base," *Simulation Modelling Practice and Theory*, vol. 62, pp. 1–13, 2016. (Citato a pagina 19)

[51] M. Shiraz, A. Gani, R. H. Khokhar, and E. Ahmed, "An extendable simulation framework for modeling application processing potentials of smart mobile devices for mobile cloud computing," in *2012 10th International Conference on Frontiers of Information Technology*, pp. 331–336, IEEE, 2012. (Citato a pagina 19)

[52] I. K. Kim, W. Wang, and M. Humphrey, "Pics: A public iaas cloud simulator," in *2015 IEEE 8th International Conference on Cloud Computing*, pp. 211–220, IEEE, 2015. (Citato a pagina 19)

[53] S. K. S. Gupta, A. Banerjee, Z. Abbasi, G. Varsamopoulos, M. Jonas, J. Ferguson, R. R. Gilbert, and T. Mukherjee, "Gdcsim: A simulator for green data center design and analysis," *ACM Trans. Model. Comput. Simul.*, vol. 24, jan 2014. (Citato alle pagine 19 e 20)

[54] S. K. Gupta, A. Banerjee, Z. Abbasi, G. Varsamopoulos, M. Jonas, J. Ferguson, R. R. Gilbert, and T. Mukherjee, "Gdcsim: A simulator for green data center design and analysis," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 24, no. 1, pp. 1–27, 2014. (Citato a pagina 22)