CloudSimPer: Simulating Geo-Distributed Datacenters Powered by Renewable Energy Mix

Jie Song, Peimeng Zhu, Yanfeng Zhang, and Ge Yu, *Senior Member, IEEE*

***Abstract*—Nowadays, studies on energy-efficient datacenters, especially the DataCenters powered by Renewable Energy mix (DCRE), have gained great attention. DCREs are large-scale, geo-distributed, and equipped with on-site renewable energy generators. For these features, it is expensive to perform empirical evaluations of proposed algorithms and solutions on the real-world DCREs, while the state-of-the-art datacenter simulators are not applicable for DCREs. In this paper, we present CloudSimPer (CLOUD SIMulator hybrid-Powered by rEnewable eneRgy), a general-purpose simulator that comprehensively supports the simulation of DCREs. Besides the functions such as renewable energy, geo-distribution, and long-term simulation, we also design evaluation metrics and an integrated simulation case for experimental studies in the future. The main challenge of CloudSimPer lies in designing a new model and software layer upon CloudSim, to solve the complexity of traceable and comparable simulations which connect renewable energies, datacenters, workloads, regions, and schedulers. We use the term schedulers broadly, encompassing any optimization approaches on DCREs for energy saving. We prove CloudSimPer and integrated case to be valid, so that simulation results are scientifically sound, by examining the expectation and the simulation results, and comparing the simulation results with selected competitors. CloudSimPer offers simulation services to datacenter designers, datacenter administrators, and academics.**

***Index Terms*—Battery, CloudSim, datacenters, energy saving, green computing, renewable energy, simulation.**

I. Introduction

With the age of Big Data and global informatization, datacenters have become the indispensable infrastructure for modern society [1]. Due to its high energy consumption, the large-scale adoption of fossil energy for electric power has led to the gradual reduction and exhaustion trends of brown energy stored for millions of years. It also brought severe pressure to the climate. Fortunately, more experts pay attention to renewable energy because it is inexhaustible and environment-friendly. DataCenters powered by Renewable Energy mix (hereinafter DCRE) appear worldwide [2].

Manuscript received 9 May 2022; revised 28 November 2023; accepted 19 January 2024. Date of publication 23 January 2024; date of current version 15 February 2024. This work was supported by the National Natural Science Foundation of China under Grant 62162050. Recommended for acceptance by J. Wang. *(Corresponding author: Jie Song.)*

Jie Song and Peimeng Zhu are with Software College, Northeastern University, Shenyang 110819, China (e-mail: songjie@mail.neu.edu.cn; pmzhu444@163.com).

Yanfeng Zhang and Ge Yu are with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: zhangyf@mail.neu.edu.cn; yuge@mail.neu.edu.cn).

Digital Object Identifier 10.1109/TPDS.2024.3357532

The emerged DCREs come with a brand-new and hot research area. Due to renewable energy's geographical diversification, periodicity, intermittency, and instability, researchers have begun to study how to make full use of renewable energy and reduce the brown energy supplement. The traditional studies on datacenters, such as virtual machine scheduling, task allocation, and resource provisioning, are adapted to the new goals of energy utilization, and novel approaches, such as energy-aware temporal and spatial load balancing, are impactful. This paper generally calls the above optimization approaches *schedulers*.

Studies on DCREs bring a new challenge to the experimental platform. The experimental studies on real-world DCREs are costly and inefficient because establishing hardware, software, and energy supply are trivial and complicated. Thus, a DCREs simulator undoubtedly provides researchers with a convenient experimental environment. However, the state-of-the-art DCRE simulators have the following drawbacks:

- No general-purpose simulator supports both renewable energies and geo-distributed features.
- It cannot expect a scheduler to make a significant effect in a short period because renewable energy has a long-term periodicity. But no simulator supports long-term simulation, such as months or years.
- No unified simulation case, purely defined with publicly available traces or functions, is for evaluating the schedulers.
- No evidence shows that a simulator is valid or not, i.e., whether their simulation results are scientifically sound has not been sufficiently studied. The same scheduler on different simulators may have distinguishing optimization effects.

We propose our DCREs simulator, CloudSimPer (CLOUD SIMulator hybrid-Powered by rEnewable eneRgy), to overcome the above drawbacks. To comprehensively support the experimental studies on DCREs, we design evaluation metrics and an integrated case for simulation. We prove CloudSimPer and integrated case to be valid by two approaches: 1) Examining whether the simulation results accord with the expectation under the six simple schedulers whose effects are apparent. 2) Comparing the simulation results of CloudSimPer with selected competitors. Fig. 1 shows how the CloudSimPer simulates the geo-distributed DCREs, and the main parts discussed in this paper (gray boxes).

The traceable and comparable simulations, which connect renewable energies, datacenters, workloads, regions, and schedulers, bring the technical complexity to CloudSimPer. Although CloudSimPer encapsulates cores of CloudSim [3], a well-known

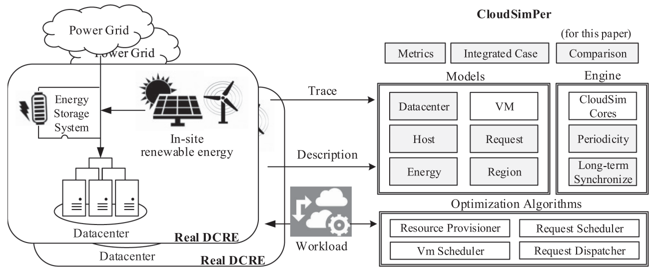


Fig. 1. CloudSimPer components and its simulation.

datacenter simulator, it cannot ease the technical complexity by simply adding complements to CloudSim. Instead, modeling a new mechanism and software layers upon CloudSim brings the design complexity to CloudSimPer. We prefer to reuse CloudSim's mature simulation mechanism in its bottom layers. We also have to design new layers to support new models, engines, metrics, and integrated cases shown as gray boxes in Fig. 1. Section II highlights the differences and challenges of CloudSimPer compared with CloudSim and other simulators.

The major contributions of this research are as follows:

- The paper concludes the required functions from the state-of-the-art simulators. Based on these requirements, we discuss the advantages and challenges of CloudSimPer.
- CloudSimPer provides the universal metrics, the integrated simulation case, and the simulation results of the case. The results show that the metrics and case are valid and act as a baseline for future studies. These offer essential services for the DCRE research.
- CloudSimPer defeats other simulators not only in their versatility but also in their validity. It brings new experiences to the DCRE studies.
- CloudSimPer's architecture and core simulating methods bring a reference to the research of datacenter simulation.

CloudSimPer offers services to both research and industry. *First*, datacenter designers could initially evaluate the designed DCRE before building it. For example, they can first examine the specifications such as the location, capacity, equipped renewable power plants, energy consumption, and cost. *Second*, datacenter administrators could simulate their planned strategies, to prove the performance, energy, or cost optimization before deploying them in DCREs. *Finally*, to the academic, CloudSimPer offers a simulation environment with an integrated case, and makes the empirical evalution and comparison fair, easily performed, fully repeatable, and configurable for arbitrary hypothetical scenarios.

The remainder of the paper is organized as follows: Section II shows the CloudSimPer's advantages by analyzing the state-of-the-art simulators. Section III describes CloudSimPer's architecture, and Section IV highlights core methods for simulation. Sections V and VI define CloudSimPer's metrics and the integrated simulation case. Section VII proves CloudSimPer and the case are valid by analyzing the simulation results. Section VIII compares CloudSimPer with four typical simulators. The last section discusses the conclusions and future works.

TABLE I
GENERAL FUNCTIONS OF DCRE SIMULATORS EXTRACTED FROM EXPERIMENT SETUP OF 85 PAPERS

1 Simulation tools
1.1 Self-made simulator: developing a simulator for the research.
1.2 Third-party simulator: adopting the public simulator for the research.
2 Workload
2.1 Traces-based: generate workload according to publicly available traces.
2.2 Functions-based: generate workload according to specific functions.
3 Renewable energy
3.1 Solar and wind: support solar and wind energy.
3.2 Weather traces: support weather traces to calculate power.
3.3 Power traces: support power traces directly.
3.4 Power stations: support power stations to generate power.
3.5 Mathematical simulation: support math functions to generate power.
3.6 Static price: the energy price is constant.
3.7 Dynamic price: the price model change with time or conditions.
3.8 Carbon emissions: the carbon emissions are estimated.
4 Datacenter
4.1 Server attributes: datacenter specification with various attributes of servers.
4.2 Server capacities: datacenter specification is the capacities of servers.
4.3 Power values: the power model is an enumeration of powers
4.4 Power functions: the power model is a mathematical function of powers and working status.
4.5 Virtual machine: support VM specification.
4.6 Location: support the datacenters' locations.
4.7 Bandwidth: support the bandwidth of servers.
5 Battery
5.1 Battery energy: support the battery as energy.
5.2 Battery optimization: support an optimization approach for batteries.
5.3 Battery cost: consider the cost of batteries.
6 Scheduler (Optimization approaches)
6.1 Capacity planning: planning datacenters' power capacity.
6.2 Power control: adjusting the power of servers to meet the requirements.
6.3 VM migration in a datacenter: temporal load balancing in resource layer
6.4 VM migration among datacenters: spatial load balancing in resource layer
6.5 Task schedule: temporal load balancing in computation layer
6.6 Request dispatch: spatial load balancing in the service layer

II. OUR ADVANTAGES

A. State-of-the-Art Simulators

By enumerating the search results of the scholar database, we have collected the papers relevant to "renewable energy" and "datacenter" in the recent decade. Among them, 100 papers contain interesting experimental studies on DCREs. Most studies conducted experiments to validate their propositions no matter the optimization approaches were energy management, workload immigration, or request scheduler. At the beginning, studies such as GreenSwitch [4], [5], GreenHadoop [6], GreenSlot [7], GreenCassandra [8], GreenPar [9], GreenSort [10] and GreenGear [11] performed their experiments on real-world testbeds. Among these testbeds, the Grid5000, as a geo-distributed datacenter in France, is shared mostly [9], [12], [13]. Other publicly available testbeds for the academic, such as the micro datacenter in [14] and GreenStar [15], which could "follow the wind, follow the sun," are never popular due to the operational and maintenance cost. After the last work [12] in 2017, researchers would conduct experiments on simulation platforms instead of practical testbeds.

About 85 papers addressed the known solutions of DCRE simulation. After a careful study, we conclude six groups of general functions and list them in Table I. CloudSimPer treats these functions as requirements since no simulator supports them all.

TABLE II
COMPARING CLOUDSIMPER WITH CLOUDSIM [3], PLUG4GREEN [16], PHILHARMONI [20], DCWORMS [21], MATPOWER [22], RENEWSIM [23], ECOMULTICLOUD [24], SIMGRID [25], AND IFOGSIM [26]

In support of simulation functions	**Our**	[3]	[16]	[20]	[21]	[22]	[23]	[24]	[25]	[26]
Geo-distributed datacenter	✓✓	✓	×	✓	×	×	×	✓	×	×
Power model	✓✓	✓	✓✓	✓	✓✓	✓✓	✓✓	✓	✓	✓
Workload generation	✓✓	×	✓✓	✓✓	✓✓	×	✓	✓✓	✓✓	✓
Consistent clocks in parallel executions	✓	×	×	?	×	×	×	✓	✓	×
Renewable energy	✓✓	×	×	×	×	×	✓	×	×	×
Dynamic energy pricing	✓✓	×	×	✓✓	×	✓✓	✓	×	×	×
Battery management	✓✓	×	×	×	×	×	✓	×	×	×
Quite Long-term simulation (monthly)	✓	×	×	×	✓	×	×	×	×	×
Resource management	✓✓	✓✓	×	×	✓	×	×	×	×	✓✓
Thermal management	×	×	×	✓	×	×	×	×	×	×
Network and SDN	×	✓	×	×	×	×	×	×	✓✓	✓✓
Virtual machine management	✓✓	✓✓	✓✓	✓✓	×	×	×	✓	×	?
Spatial load balancing	✓✓	×	×	✓✓	×	×	×	✓	×	×
Temporal load balancing	✓✓	✓✓	✓✓	×	✓✓	×	×	×	×	✓✓
Configurable cases building		×	×	✓	×	×	×	×	×	✓

✓ = supported, ✓✓ = supported with customization, ×= not supported, ? = unknown

TABLE III
COMPARING EVALUATION AND METRICS OF KNOWN SIMULATORS AND CLOUDSIMPER

			Third-party tools	**Self-made simulators**	**CloudSimPer**
Metrics for simulation results	Detailed renewable energy	Similarities:	Renewable Energy Generation (REG) with time.		
		Differences:	Only RenewSim [23] supports renewable energy and provides hourly REG data.	66/103 studies provide hourly REG data, and the others do not.	Hourly or minutely REG data.
	Statistical renewable energy	Similarities:	Renewable Energy Utilization (REU)		
		Differences:	None provides the REU data.	31/103 studies provide REU as the optimization goal, 20/103 studies provide metrics similar to REU, and the others do not.	REU is additionally for simulation accuracy.
	Detailed power	Similarities:	All provide power data because of power models for simulation, so that Energy Consumption (EC) with time can be calculated.		
	Statistical energy consumption	Similarities:	Renewable Consumption Proportion (REP).		
		Differences:	None provides the REP data.	19/103 studies provide REP as the optimization goal, 9/103 studies provide metrics similar to REP, and the others do not.	REP is additionally for simulation accuracy.
Evaluation on a simulator	Simulation Accuracy	Similarities:	Checking whether the simulated REG and EC data along a special duration match the expectation.		
		Differences:	2/9 studies provide the evaluation on the accuracy. DcWorms [21] compares results of simulator to measurements of real servers. Simgrid [25] is compared to other frameworks in terms of execution time and speed. They are not renewable energy related because none supports renewable energy.	Only 31/103 studies provide the evaluation on the accuracy, such as [27]–[29]. They compare the simulated results with the baselines whose results are apparent. Md Sabbir Hasan et al. [30] and Zichen Xu et al. [31] try to compare the results with the real-world data, but the simulation cases are simple, not larger scale and geo-distributed datacenters.	Comparison with the baselines, also the fullfledged works with real-world trace.
	Simulation Efficiency	Differences:	5/9 studies provide the evaluation on efficiency, such as [3], [16], [21], [24], [25] , in a short-term simulation. For example, CloudSim gives two real cases to analyze the simulation efficiency. EcoMultiCloud defines relational functions between metrics.	None provides an evaluation on efficiency.	Efficiency evaluation for quite long-term simulations on scalable datacenters.

Repeatedly developing self-made simulators is costly despite their common grounds. Therefore, several studies that emerged after 2018 leveraged the third-party tools for their simulation, such as Plug4Green in [16], CloudSim in [17], [18], [19], Philharmoni in [20], DCWorms in [21], MATpower in [22], RenewSim in [23], EcoMultiCloud in [24], SimGrid in [25], and iFogSim in [26]. However, their weakness has two folds: First, none supports renewable energy powered and geo-distributed datacenters together; Second, none supports a longer-term simulation. Table II compares the functions of general-purpose simulators with CloudSimPer and shows our versatility.

The evaluation on simulators is for two proposes: the evaluation on the simulator itself, and the evaluation on the simulation results for the clients. Both the third-party tools mentioned in Table II and the self-made simulators mentioned in Table I could apply them. Table III compares the evaluation and metrics of known simulators and CloudSimPer.

Besides the new items mentioned in Table III, CloudSimPer also proposes a simulation accuracy evaluation with the metrics for simulation results. It combines the REU and REP. Section V discusses these metrics.

B. Goals

CloudSimPer focuses on the goal to compensate for the missing functions of CloudSim and other simulators. The main objectives are as follows.

First, CloudSimPer introduces the renewable energy supplying mechanism for datacenters in a simulation. CloudSimPer provides the energy generators of periodical, intermittent and unstable renewable energies.

Second, CloudSimPer introduces the geo-distributed datacenter and request generation. The simulation is not only in the temporal dimension but also in the spatial dimension; namely, CloudSimPer supports both temporal load balancing (workload scheduling) and spatial load balancing (geographical load scheduling).

Third, CloudSimPer introduces the long-term simulation. Because of the intermittence and instability of renewable energy over an extended period, such as weeks, months, or even a year, the study on energy utilization optimization of DCRE, such as renewable energy-aware load balance, takes no immediate effects but accumulates the benefits over a long period.

Fourth, CloudSimPer provides metrics to evaluate the optimization effect of schedulers, and an integrated simulation case to unify the case-building of DCRE studies.

Besides, CloudSimPer also implements the customizable case-building process and pluggable extensions. Such user-friendly functions are abbreviated in this paper.

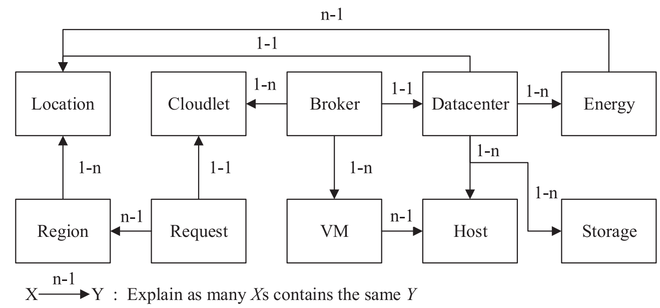


Fig. 2. CloudSimPer entities and relationships.

C. Challenges

CloudSimPer delegates the fundamental simulation functions to the encapsulated CloudSim [3]. However, to reach the goals mentioned previously, we need to solve two complexities. One is the technical complexity of connecting renewable energies, datacenters, workloads, regions, and schedulers. The other is the design complexity of encapsulating and reusing the CloudSim cores. In details. CloudSimPer faces the following challenges with corresponding solutions:

1) The entities of CloudSimPer, such as *datacenter, host, storage, broker, VM, energy, batteries, request, location, region*, and the relationships between entities, are complicated. We ravel out a complication through reasonable abstractions and the orderly relationships among them. We also designed the layered software architecture and agile modules to hold these entities. Section III gives the details.
2) CloudSimPer should provide a long-term simulation, such as days and months. On the contrary, CloudSim only supports one-time execution on predefined workloads in the minute-level simulation. Synchronizing the time of request generators, energy generators, batteries, and datacenters, also a cycling mechanism, are challenging. We propose the simulation timestamp, duration, periodicity of request and renewable energy. These concepts ensure long-term simulation can be properly modeled. Each generator runs in a thread, and their synchronization is implemented by inter-thread communication. Section IV gives the details.
3) To simulate the diversified and rich-featured renewable energy, CloudSimPer should provide generalized, customizable, and periodical renewable energy generators. It is challenging to abstract various renewable energy, including the charging and discharging of batteries, and design the energy generator, a geodistributed and time-dependent request generator. We address these challenges through delicate software design. We fully utilize the advance of object-oriented programming design. Some engineering details are abbreviated, but the diversified and rich-featured simulation case, as a result, is shown in Section VI.
4) The parameters for regions, datacenters, renewable energy, energy price, batteries, and requests should be practical, explainable, compatible, and verifiable. Moreover, it is challenging to design experiments to prove that the CloudSimPer is valid and better than other simulators, especially without the support of real-world DCREs. We first define a universal metric for the schedulers and a standard simulation case, then plan the simulation and comparison experiments to address the challenge. In the simulation, the metrics under different schedulers accords with our expectation; it proves that CloudSimPer is accurate. In the comparison, the same scheduler leads to different metric values in the various simulators under the simulator case; but CloudSimPer is the most reasonable and explainable one; it proves that CloudSimPer is precise.

III. Architecture

This section describes the system model and architecture of CloudSimPer. It explains the abstract entities and their relationships and then shows the software architecture of CloudSimPer, comparing that of CloudSim.

Entities: The entities in CloudSimPer are reasonable abstractions of the instantiated objects in a real-world DCRE. They are *datacenter, host, storage, broker, vm* (virtual machine), *cloudlet, energy, request, location, and region*. Among these ten entities, the former six are extended from CloudSim, and the latter four are proposed in CloudSimPer.

These entities are all connected and conceptually grouped into four categories: 1) energy entity such as *energy*; 2) resource entity such as *datacenter, host*, and *storage*; 3) geo-entity such as *location* and *region*; 4) workload entity such as *broker, vm, cloudlet* and *request*. Fig. 2 shows the relationships among these entities.

For the energy entity, a simulation case configures several *energie*s. Each *energy* represents a type of renewable energy, such as solar energy, wind energy, and rechargeable battery. The geography-relevant energies are associated with at least one *location* because different locations bring different energy-supplying regularities.

For the resources entity, a simulation case configures several *datacenter*s. Given a *datacenter*, CloudSimPer specifies its *location*, several *storage*s, *host*s, and *energie*s with the *location*s mapped to the *datacenter*. Typically, the *location*s of the *energie*s and the associated *datacenter* are the same. Namely, the *datacenter* has an on-site renewable energy power station. For the *datacenter*, the remotely supplied renewable energy, as a Public Power Supply System, is feasible but unconventional. As a stable and on-demand electric supply, the public electric supply is conventionally treated as brown energy and associated with the *datacenter* by default.

For the geo-entity, a simulation case configures several *region*s and *location*s. *Region*s are geographically none-overlapped. A *location* only belongs to one *region*. A *region* contains several *location*s if the *region* has *datacenter*s and *energie*s associated with these *location*s; otherwise, the *region* has no *location*s and is only for *request* generation.

For the workload entity, CloudSimPer starts a simulation case and instantiates them. *Request*s are generated in each *region* and dispatched to *broker*s. A *broker* acts as the proxy of the *datacenter* and takes the responsibilities of receiving *request*s, transferring *request*s to *cloudlet*s, creating *vm*s on hosts, and performing *cloudlet*s on *vm*s. Therefore, *datacenter*s and *broker*s are one-to-one mapped, and *request*s and *cloudlet*s are also one-to-one mapped. A *broker* deals with many *request*s, *cloudlet*s, and *vm*s. A *host* has many *vm*s, while a *vm* only exists on one host simultaneously.

The entities are complete. They cover all the possible concepts of simulating DCREs because we summarize them from 103 related studies, as discussed in Section II-A. The relationships among entities ensure flexibility to the best extent. We have examed all possible entity-to-entity mappings. We prefer 1-n relationships except for two conditions: first, it does not make sense, such as *datacenter* mapping multiple *locations*; second, it is over-designed and less efficient, such as datacenter possible but unnecessary mapping multiple *brokers*. Besides, we try to avoid the n-m mapping between entities. We decouple it with a new entity and two 1-n mappings. For example, a *location* may accept many kinds of *requests*, while a kind of request may be dispatched to different *locations*. So we introduce the *region* to decouple them. Last but not least, implementing these entities and relationships is also effective, but some software design techniques are not elaborated. We make CloudSimPer an open-source platform, and its source codes are available on GitHub.[1]

Fig. 3 shows the multi-layered software architecture and modules of CloudSimPer. The white, gray, and grid box modules are inherited from, extended from, and newly introduced to CloudSim. The architecture contains two parts, the bottom is the system part, and the top is the user part. The system part is explained from the bottom up.

The network layer simulates the network behavior of cloud datacenters. The CloudSimSDN-NFV supports the network function virtualization of CloudSim [32]. In CloudSimPer, it remains the modules of network topology simulation and message delay calculation unchanged.

The cloud resources layer is the primary layer for simulation. It simulates the datacenter hardware and the events within the datacenter. CloudSimPer extends the events handling mechanism for supporting the long-term simulation through cycling the requests. It also extends the datacenter and host simulation to be those powered by renewable energies mix.

The cloud services layer provides simulated resources such as CPU, memory, storage, and bandwidth. The VM provisioning

[1]https://github.com/CloudLab-NEU/CloudSimPer

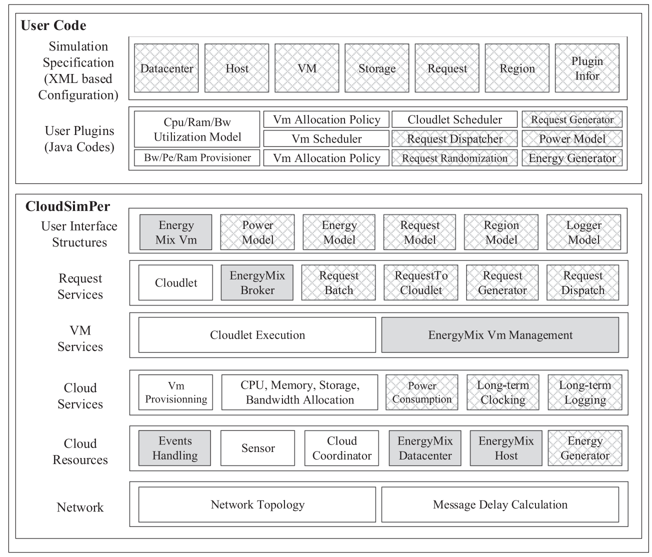


Fig. 3. CloudSimPer architecture.

algorithms simulate activities relevant to deploying and customizing virtual machines with these resources. The new power consumption module calculates the dynamic power of the hosts. The new long-term clocking module keeps the timestamp of the simulation case, i.e., from May 1st to September 30th. The new long-term logging modules track the states of entities during the simulation.

The VM services layer implements the *cloudlet* execution. CloudSimPer updates the VM management module of CloudSim with the integration of renewable-energy-aware schedulers. For example, VM allocation and VM migration algorithms consider the supplement and consumption of renewable energy.

The request services layer is a new layer of CloudSimPer. The four new modules in this layer generate sufficient *request*s batch by batch according to the given rules, dispatch *request*s to *datacenter*s, and transfer *request*s to *cloudlet*s. Meanwhile, the energy-mix broker is an update of the broker module for integrating with renewable energy.

The user interface structures layer gives six modules for entities. Five of them are new modules. These modules parse the users' configuration, such as plugin schedulers and case specifications, and integrate the configuration to the under layers of CloudSimPer.

The top part of CloudSimPer stack is user code layers for customizing entities and algorithms. The upper layer (layer of user plugins) contains all the extended algorithms, represented as plugins and implemented as Java code. The lower layer (layer of simulation specification) contains the specification of *datacenter*s, *host*s, *vm*s, *storage*s, *request*s, and *region*s in XML format. The plugin configuration defines the mapping relationship between plugin names appearing in the entities' specification in the upper layer and the plugin implementations as Java codes in the lower layer. Table IV explains these plugins.

In Table IV, the "owner" is the entity associated with a plugin. The "mapping from" are the input or source entities of plugins,

TABLE IV
PLUGINS IN CLOUDSIMPER

Name	Owner	Mapping From	Mapping To
RequestDispatcher	CloudSimPer	Requests	*Brokers*
VmAllocationPolicy	*Datacenter*	*Vm*s of a *broker*	*Host*s of a *datacenter*
BwProvisioner	*Host*	Bandwidths of the *host*	*Vm*s
PeProvisioner	*Host*	CPU of the *host*	*Vm*s
RamProvisioner	*Host*	Ram of the *host*	*Vm*s
PowerModel	*Host*	CPU utilization of the *host*	*Power*s of the *host*
VmScheduler	*Host*	*Vm*s of a *host*	*Vm* is running or not
CloudletScheduler	*Vm*	*Cloudlets* of a *vm*	*Cloudlet* is executing or not
CpuUtilizationModel	*Request/Cloudlet*	CPU utilization of a *vm*	The *Cloudlet*
RamUtilizationModel	*Request/Cloudlet*	Ram utilization of a *vm*	The *Cloudlet*
BwUtilizationModel	*Request/Cloudlet*	Bandwidth utilization of a *vm*	The *Cloudlet*

and the "mapping to" are output or target entities of plugins. For example, *VmAllocationPolicy* maps the *vm*s of a *broker* to the *host*s of a *datacenter*, *VmScheduler* maps the *vm*s of a *host* to the two states, running or suspending.

IV. CORE METHODS

This section describes core methods for simulation, such as duration, periodicity, request, and energy. In CloudSimPer, both the request and energy generator rely on simulation duration, periodicity, temporal and spatial information.

Simulation Timestamp and Duration: CloudSimPer does not define the concept of dates, such as a week, month, and year. The date is given as *n*-th-day-of-a-year, and time is given as *n*-th-second-of-a-day. Therefore, a timestamp in the simulation is represented as two integers: second-of-a-day and day-of-a-year. The simulation duration is the number of a second between the start timestamp and the end timestamp.

The time for executing a simulation case, named execution time, positively correlates to the scale and simulation duration of the case. Execution time is significantly less than simulation duration. For example, a one-month middle-scale case required five minutes of execution time on a commodity server. We do not study execution time in this paper. Therefore, the time hereinafter refers to the simulation time in simulation duration, not the execution time.

Periodicity: In CloudSimPer, *request* and *energy* are two entities relied on periodicity. The most fine-grained element of time is a *cycle* with predefined rules. Short Term Periodicity (STP) and Long Term Periodicity (LTP) are two forms of periodicity. A simulation case defines a *duration* overlaid with iterative LTPs, each LTP is overlaid with iterative STPs, and each STP is overlaid with iterative *cycle*s. For example, *duration* is a year, LTPs are four seasons, STPs are weeks, and *cycle*s are typical days.

CloudSimPer generates requests periodically. A request generator has three parameters in a simulation cycle: *time, amount*, and *size*. Namely, the generator determines at which time, how many requests are generated, and how large they are.

Energy Unit: CloudSimPer models renewable energy with energy units, the atomic element of energy in a fixed duration. CloudSimPer provides three methods to determine the values of renewable units: 1) the calculation method based on theoretical formulas; 2) the extraction method based on renewable energy traces or weather traces; 3) the periodic prediction method based on a pre-trained model.

By default, CloudSimPer supports solar, wind, and battery energies. Corresponding energy generators create energy supplements by iterating the given energy units.

CloudSimPer employs the static and dynamic models of renewable energy price. The former defines the customizable on-peak and off-peak prices. The latter calculates the price according to the production. Energy price exponentially decreases with the increase of solar radiation or wind speed. For example, the price of solar energy at the time t within a day is:

$$P_t = P_m \cdot e^{-\omega\delta_t} \cdot (t > t_p) \tag{1}$$

where δ_t is the normalized solar radiation factor at time t; t_p is the time slot; P_m is the minimum grid energy price when solar energy is unavailable ($t < t_p$); Let P_0 be the current price for the feed-in tariff, and then coefficient ω is calculated through the equation $P_0 = P_m \cdot e^{-\omega}$.

The battery gets charged only by renewable energies. It charges (discharges) when the renewable energy is adequate (inadequate) to the datacenter's demands. Renewable energy is over-produced when the battery is full, and then extra energy is abandoned. On the contrary, additional energy is taken from the power grid when renewable energy and the battery are insufficient to fulfill the datacenter's energy demands. Battery simulation is turned off by default.

V. METRICS

This section gives several metrics for evaluating the simulator (CloudSimPer or others are valid or not), the simulation case (parameters are applicable or not), and the scheduler (effectiveness or not). The original simulation results are two folds: the DCRE's energy consumption and renewable energy generation, represented as two curves along the simulation time. CloudSimPer names them c-curve and g-curve, respectively.

Generation Curve and Consumption Curve (g-curve and c-curve): The generation curve, short for g-curve, refers to the curve whose x-axis is simulation duration and y-axis is the amount of generated renewable energy at the time. The consumption curve, short for c-curve, refers to the curve whose x-axis is simulation time and y-axis is the amount of consumed energy, including renewable energy and brown energy, at the time.

The g-curve and c-curve show the time-series simulation results, and the "producer-consumer relationship" changed over time. Such changes represent whether the c-curve, which is adjustable through the schedulers, follows the g-curve, which is natural and un-adjustable in the runtime environment.

Fig. 4 shows examples of g-curve and c-curve where (1), (2), (3) represents stable phase, no wind phase and fluctuant phase,respectively. The green line is a c-curve, and the red line is a g-curve. Two lines are almost matching in the stable period (wind) due to the capacity design of the datacenter. Nevertheless, their gap is also evident in the fluctuated or deficient period (wind). For this example, the datacenter does not run any green-aware scheduler so that the c-curve is stable and does not follow the g-curve.

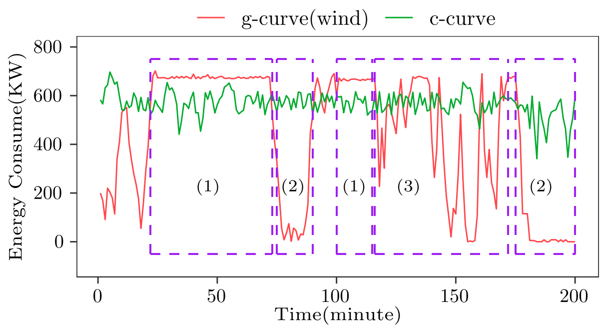


Fig. 4. Examples of g-curve and c-curve.

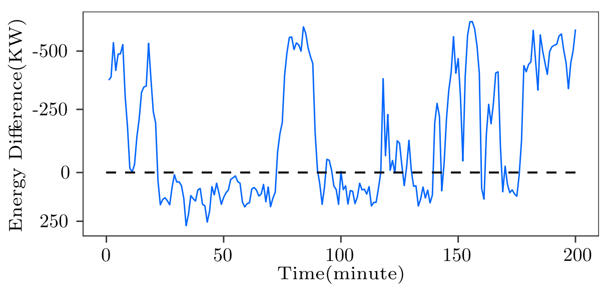


Fig. 5. Example of d-curve.

Difference Curve (d-curve): The d-curve refers to the difference between the c-curve and g-curve. The x-axis is simulation time, and the y-axis is a gap, which is the amount of generated renewable energy minus the amount of consumed energy per time. A positive gap means renewable energy is deficient, while a negative gap means it is sufficient.

Metric d-curve: First, supposing the simulator has functional drawbacks or the simulation case is designed carelessly, the datacenter may significantly lack/excess renewable energy if the d-curve is remarkably higher/lower than the x-axis. Second, supposing the scheduler is quite effective, the datacenter may primarily consume renewable energy rather than brown energy if the d-curve is stabilized and closed to the x-axis. Fig. 5 shows the d-curve of the traces in Fig. 4, an un-optimized case.

The three curves are for evaluating both the simulator and scheduler. For example, we could compare the simulated c-curves with expected c-curves which are apparent under the baseline schedulers. State-of-the-arts studies also propose timely renewable energy metrics like c-curve and g-curve. We refine them for two considerations: First, the third-party simulators do not integrate the c-curve and g-curve, while half of the self-made simulators prefer these curves. CloudSimPer, as the third-party simulator, should highlight the timely detail data with visualization; Second, CloudSimPer also suggests the d-curve to highlight the timely renewable utilization.

Renewable Utilization: Renewable utilization μ is the extent to which renewable energy powers the datacenter. It refers to the proportion of consumed renewable energy in all generated renewable energy.

$$\mu = \frac{renewable\_energy\_consumption}{renewable\_energy\_generate} \tag{2}$$

Renewable Proportion: Renewable proportion ρ is the extent to which renewable energy replaces brown energy to power the datacenter. It refers to the proportion of renewable energy in consumed energy (including brown energy).

$$\rho = \frac{renewable\_energy\_consumption}{energy\_consumption} \tag{3}$$

Metrics μ and ρ is for evaluating the scheduler. State-of-the-art schedulers have different purposes but may inevitably mention metrics like renewable utilization and proportion. Optimization works should follow uniform measurement; therefore, defining new metrics is useless and brings inconsistency, and CloudSimPer has to provide widely accepted metrics for schedulers. However, metrics for evaluating simulators, as discussed in Table III (see Section II-A), have no such consensus. CloudSimPers novel joint measure (μ, rho) is an effective metric for such evaluation. The metric is more holistic than the g- c- d-curves, supported by results in Fig. 16 (see Section VIII-E).

Metrics (μ, ρ): The utilization and proportion jointly measure the simulation case's validity and the optimization effects.

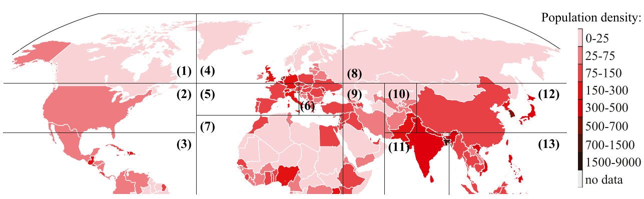
- *($\mu\uparrow$, $\rho\uparrow$):* "High Utilization and High Proportion" shows the simulator and case are good, and the scheduler is effective. First, renewable energy generation is less than the energy consumption of full-loaded power but close to the actual energy consumption as much as possible. Second, the scheduler is also effective.
- *($\mu\uparrow$, $\rho\downarrow$):* "High utilization and Low proportion" shows the poor simulator or defective simulation case. The datacenter consumes most renewable energy but still demands additional brown energy, for inadequate renewable energy caused by wrong case building or simulator's functional drawbacks.
- *($\mu\downarrow$, $\rho\uparrow$):* "Low Utilization and High Proportion" shows the poor simulator or defective simulation case. The datacenter consumes a few proportions of renewable energy while discarding the rest, for the excessive renewable energy caused by functional drawbacks, such as a defective simulation case and inaccurate energy models.
- *($\mu\downarrow$, $\rho\downarrow$):* "Low Utilization and Low Proportion" shows poor scheduler, while whether the simulator or simulation case is valid or not is unknown.

VI. Integrated Case

CloudSimPer package contains the evaluation metrics and a simulation case by default. The integrated case defines five aspects: 1) the features of selected regions; 2) the specification of the datacenter, including the number of datacenters, locations, capacities, memory, storage, virtual machines, and power models of servers; 3) the specification of the renewable energy associated with the centers; 4) the workload specification; 5) the default request schedulers as the baseline for future studies. Researchers can leverage both the simulator and the case for their experimental studies.

TABLE V
LOCATION AND CAPACITIES OF DATACENTERS

DataCenter	Location	Region	Hosts		VMs		Power		Served Regions	
			Name	Size	Name	Size	Idle(*KW*)	Full-load(*KW*)	ID	Factors
DC_CR	Cairo, Egypt	NA	HPXeon3104	50	Powerful	50	19740	45900	(5)	1.0
			HPDL580	100	Middle	100			(6)	1.2
			IBMP570	50	Weakness	200			(7)	2.7
									(9)	0.5
DC_MB	Mumbai, India	SA	HPXeon3104	100	Powerful	200	28440	61800	(10)	1.5
			HPDL580	200	Middle	200			(11)	4.5
			IBMX3850	200	Weakness	100			(12)	7.5
									(13)	1.0
DC_SH	Shanghai, China	EA	HPXeon3104	200	Powerful	100	24840	54600	(8)	1.125
			HPDL580	100	Middle	50			(12)	7.5
			IBMP570	100	Weakness	200			(13)	1.0
DC_LA	LosAngeles, United States	US	HPDL580	200	Powerful	200	21240	51300	(1)	1.2
			DELLR720	100	Middle	100			(2)	1.5
			IBMP570	100	Weakness	100			(3)	0.75
DC_OS	Oslo, Norway	NE	HPDL580	200	Powerful	200	22980	54000	(4)	0.1
			DELLT440	100	Middle	100			(5)	1.0
			HPXeon3104	100	Weakness	100			(6)	1.2
									(8)	1.125



ID	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Name	CA	US	LA	NE	WE	SE	NA	EE	ME	WA	SA	EA	SEA
Population-density	0.2	0.5	0.5	0.125	1	1	0.3	0.125	0.25	1.5	3	1.5	1.25
Region-size	6	3	1.5	0.8	1	1.2	9	9	2	1	1.5	5	0.8
Time-shift	-7	-8	-6	2	0	2	1	7	4	6	7	9	10

Fig. 6. Regions of the Northern Hemisphere.

A. Regions

The datacenters, energies, and requests are geo-distributed; therefore, they are explicitly associated with the regions. The case selects the northern hemisphere as a geographical range, and divides the range into 13 regions according to the time zone, population density, and administrative and cultural districts [33], [34], [35]. Fig. 6 shows these regions, short names, time zone, and population density factors.

In Fig. 6, the population-density, region-size, and time-shift dominate the number of requests issued in the region for a duration. The three elements are normalized values that take the region WE as the unit, i.e., population-density, region-size, and time-shift of WE are 1, 1, and 0, respectively. If a region has more population, the more workload is issued in the region, the larger capacity of the datacenter is designed for the region, and the datacenter consumes more energy.

B. Datacenters

According to our survey, more than half of multiple datacenters studies plan 3-5 datacenters. Following the state-of-the-art research, the case selects five datacenters distributed in the regions (2)US, (4)NE, (7)NA, (11)SA, and (12)EA, respectively. As shown in Table V, they are located in Los Angeles (United States), Oslo (Norway), Mumbai (India), Cairo (Egypt), and Shanghai (China). We choose these locations for the following consideration:

- They have different time zones, renewable energy features, and population densities.
- The locations are geographically close to the equator or monsoon belts so that excellent solar and wind resources are available all year round.
- Their time zones are complementary, i.e., their longitudes distribute almost evenly. Such distribution maximizes the aggregate amount of solar energy over 24 hours.
- Their locations are in the center of the densely inhabited district. They serve respective and adjacent regions and process the requests generated according to the regional population.

We design the datacenters' capacities for the following considerations:

- The types of hosts and virtual machines refer to the global datacenter layout of the global public cloud vendors published by the US Market Research Institute [36], also some papers [20], [37], [38], [39].
- The capacities of the datacenter proportionally scale with the population of regions they serve, respectively.
- The bandwidth is all set to be 1 GB.
- The VM capacities proportionally scale with the capacity of the corresponding datacenter.
- The power model of each host enumerates the discrete, exponent, linear, logarithm models supported by CloudSimPer.

The discrete power model for HPDL580 is between 1200 W and 1500W [40]. The exponent power model for IBMX3850 is as $P = 760 + c \times (u \times 10)^2$, where c is a constant is 10, and u is the utilization of the host [41]. The linear power model for HPXeon3104 is as $P = 220 - (1125 - 220) \times u$, where u is the utilization of the host [42]. The logarithm power model is as $P = i + c \times \log_a^{u \times 100}$, where i is the idle power of the server, c is a constant, a is the base of the logarithm, and u is the utilization of the host. (i, c, a) for IBMP570, HUAWEI2288H, DELLT440 and DELLR720 are (700, 50, 2), (550, 50, 2), (495, 40, 2), (460, 60, 2), respectively [37].

TABLE VI
SPECIFICATION OF SERVERS IN DATACENTERS

Host Name	Power Model	MIPS	CPU Numbers	RAM	Storage	BW
HPDL580	Discrete	8GHz	8	32G	500G	1000MB/s
IBMX3850	Exponent	7.4GHz	4	32G	1000G	500MB/s
HPXeon3104	Linear	6.8GHz	6	8G	1000G	500MB/s
IBMP570	Logarithm	16.8GHz	4	16G	500G	1000MB/s
HUAWEI2288H	Logarithm	12GHz	8	16G	500G	1000MB/s
HUAWEI2288H	Logarithm	8.8GHz	10	64G	2000G	1000MB/s
DELLR720	Logarithm	7.2GHz	4	4G	500G	500MB/s

TABLE VII
SPECIFICATION OF VIRTUAL MACHINES IN DATACENTERS

VM Type	Size	MIPS	CPU Numbers	RAM	BW
Powerful	2G	4000MHz	4	16G	500MB/s
Middle	1G	2000MHz	2	8G	200MB/s
Weakness	500M	1000MHz	1	4G	100MB/s

TABLE VIII
PARAMETERS OF RENEWABLE GENERATOR AND PRICE MODEL

DC Name	Maximum power(KW)	Renewable		Price	
		$\Upsilon_{pv}(KW)$	$S(m^2)$	P_0	P_m
DC_CR	70615	-	-	10	16
DC_MB	95076	-	-	6	9
DC_SH	84000	-	-	9	14
DC_LA	78923	-	-	7	15
DC_OS	83076	7	52	4	7

Tables VI and VII lists the detailed specification.

C. Renewable Energy

In this case, each datacenter has solar and wind energy. CloudSimPer provides two ways to simulate renewable energy. One is a calculation based on theoretical formulas; the other is a periodic prediction based on actual historical renewable energy traces. The datacenter OS adopts the former, and the other four datacenters, i.e., CR, MB, SH, and LA, adopt the latter.

For OS, a theoretical formula is applied to calculate solar panels. Equation (4) shows that solar radiation is very regular if the weather condition is ignorable.

$$P_r = \eta \Upsilon_{pv} \frac{I_t}{I_s} \tag{4}$$

where $\eta = 0.17$ represents the conversion coefficient of photovoltaic cells [43]; Υ_{pv} is the rated capacity of photovoltaic cells (see Table VIII); I_t is actual solar radiation at t o'clock of a day; $I_s = 1$ km/m^3 is solar radiation in the standard test condition.

Many factors such as geographic longitude and latitude, altitude, and atmospheric transparency, affect solar radiation. Equation (5) expresses the solar radiation outside the atmosphere.

$$I_0 = I_c \cdot (D_0/D)^2 = I_c \cdot \left[1 + 0.034 \cos\left(\frac{360n}{365.25}\right)\right] \tag{5}$$

where D is the distance between the sum and the earth, constant D_0 is the average distance between the sun and the earth, I_c is the solar constant, and n is the days counted from New Year's Day. Therefore, the solar radiation I_t at any place and at t time as the (6)–(8):

$$I_t = I_0 \cdot \sin h \cdot \cos\theta \tag{6}$$

$$\sin h = \sin\varphi \cdot \sin\delta + \cos\varphi \cdot \cos\delta \cdot \cos\omega \tag{7}$$

$$\cos\theta = \sin(\varphi - \beta) \cdot \sin\delta + \cos(\varphi - \beta) \cdot \cos\delta \cdot \cos\omega \tag{8}$$

where h is the angle between the sun's illumination and the ground, represents the angle between the solar ray and the normal line of the illuminated plane (solar panel); φ is local latitude, defined as the central angle between the sun-earth line and the equatorial plane of the earth at noon. $\theta = 0°$ when the sun shines vertically on the earth's equator. If the observed day is the nth day of the year, then:

$$\delta = 23.45° \times \sin\left(360 \times \frac{284 + n}{365}\right) \tag{9}$$

Moreover, the hour angle ω is as follows:

$$\omega = 15° \times (t - 12) \tag{10}$$

Meanwhile, another theoretical formula, as shown in (11), is applied to calculate the wind power model of OS.

$$P_t = \frac{1}{2}\eta\rho v^3 S \tag{11}$$

where P_t is wind power, ρ =1.293 is air density (kg/m^3), S is plane area perpendicular to the wind direction (see Table VIII), $\eta = 0.3$ is the ratio of actual output power to theoretical wind power, and v is wind speed. As explained in [44] and [45], wind speeds in a duration obey the Weibull distribution. For OS, the wind speed is sampled from a predefined Weibull distribution whose two parameters are randomly selected between boundary values, i.e., *scale* $\in$ [3.42, 8.56] and *shape* $\in$ [1.95, 3.38].

For the other four datacenters, both solar and wind energy traces are from the Measurement and Instrumentation Datacenter (MIDC) [46], [47]. The traces contain irradiance and wind speed for every minute of the corresponding month in 2011. We select the monthly traces because the workload is generated daily. CloudSimPer generates renewable energy through these traces according to (4) and (11). It adds the time offset of traces according to the time lags of the four datacenters. Besides, Υ_{pv} in (4) and S in (11) dominate the renewable generator's power for each datacenter. CloudSimPer sets them according to the power consumption of each datacenter, i.e., the average generator power is 65 percent of the maximum power of the datacenter [48]. Table VIII shows the details of these parameters.

D. Energy Prices

In this case, the energy price model of all datacenters follows (1), in which δ_t is the sum of normalized solar radiation and wind speed. Table VIII shows the details of the parameters for calculating δ_t.

E. Batteries

In this case, each datacenter has a set of Flood Lead Acidic (FLA) batteries [49]. Each set has ten batteries. The battery capacity is adequate to be charged under sufficient solar and wind conditions. The parameters of FLA batteries are as follows: the capacity is 115 KW, the cost per discharge is 0.65$, the discharge rate is 5387.5 W, the discharge-to-charge ratio is 10, the efciency

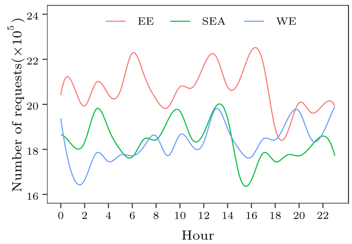


Fig. 7. Hourly requests for a day in Google trace.

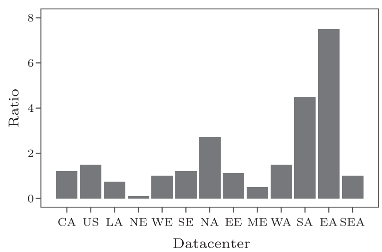


Fig. 8. Ratios of requests issued from each region for all days (WE = 1).

is 80%, the number of cells is 53, and the DoD is 0.8. Besides, the case uses an Ah-Throughput Model to evaluate the battery cycle life [50], [51]. The battery simulation is turned off by default.

F. Requests

The case adopts Google cluster usage traces [52] in February 2011 with utilization data to generate requests. Each request has resource requirements shown as the amount of requested CPU, memory, and storage. Meanwhile, the case applies the traces to every region and adjusts the timelines according to the region's local time to simulate the time lags across the world. It also scales the requests according to the region population, i.e., population-density multiples region-size. For example, Fig. 7 shows the scaled workload for WE, EE, and SEA along the time, in which the workload of region WE are from the original Google trace. Fig. 8 shows the ratios of requests issued from each region (WE = 1).

G. Schedulers

The case implements six typical request-schedulers as baselines. Their scheduling approaches are as follows:

- *Round Robin (RR):* The scheduler distributes a request to a datacenter in turns.
- *Nearest (NE):* The scheduler distributes a request to the datacenter whose location is the closest to the region where the request is issued.
- *Capacity (CA):* The scheduler distributes a request to the datacenter with the most available resources.
- *Max-Power (MP):* The scheduler distributes a request to the datacenter closest to full-load status.
- *Max-Green (MG):* The scheduler distributes a request to the datacenter with the most available renewable energy.

TABLE IX
EXPECT WORKLOAD BEING SCHEDULED TO DATACENTERS.

Schedulers	Number of requests
RR	The same workload for all datacenter
NE	The proportion of the served population (Table 5)
CA	The proportion of the datacenter's capability
MP	The proportion of the datacenter's renewable energy
MG	As much as a datacenter can process, in the alphabetical order of datacenter names
MC	The proportion of the datacenter's energy price

- *Min-Cost (MC):* The scheduler distributes a request to the datacenter with the lowest energy prices.

Table IX shows the expected workload (number of requests) for datacenters under six schedulers.

VII. SIMULATION

We simulate the integrated case defined in Section VI for a month and show the simulation results in g-curves, c-curves, and utilizations. We hourly select the 48 hours of g-curves, c-curves, and utilization curves for succinctness. We also monthly-aggregated utilizations in the bar plot. We prove that CloudSimPer and integrated case are valid because these results are scientifically sound, regular, and match the expectations.

A. Purposes

We perform the integrated case on CloudSimPer and show the results from different aspects following these Research Questions (RQs):

RQ1: Whether the CloudSimPer successfully simulates renewable energy generation?

RQ2: Whether the CloudSimPer successfully simulates datacenters' energy consumption?

RQ3: Is the integrated case well designed and ready to evaluate the proposed scheduler in future studies based on CloudSimPer?

RQ4: Whether a full-fledged scheduler on DCRE produced the results through CloudSimPer aligning with the published results.

B. RQ1: G-Curves

CloudSimPer successfully simulates renewable energy generation if the observations in g-curves of each datacenter accord with the expected phenomenon defined in the integrated case. Fig. 9 shows the solar g-curves, wind g-curves, and renewable (solar+wind) g-curves of five datacenters in 48 hours. The lines in each figure present the g-curve for a datacenter. Table X explains the observations in Fig. 9.

Overall, the trends of these curves are representative and prove that CloudSimPer successfully simulates DCREs' renewable energy generation.

C. RQ2: C-Curves

CloudSimPer successfully simulates energy consumption if the observations in c-curves under each scheduler accord with the expected phenomenon defined in the integrated case. For a

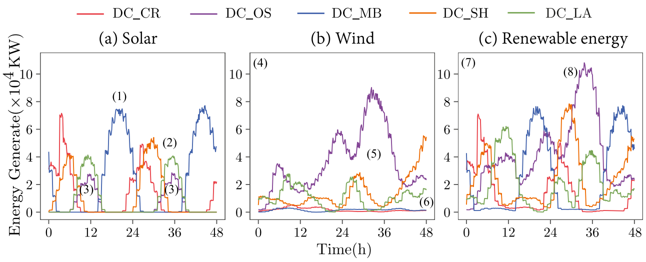


Fig. 9. Solar, wind, and renewable energy generated within five datacenters for two days.

TABLE X
EXPECTATION AND OBSERVATIONS IN FIG. 9

Expectation	Observation	ID.
MB is closest to the equator	Solar g-curve for MB has the highest crests	(1)
CR, SH, and LA are at the roughly the same latitudes but at different longitudes	Solar g-curves for CR, SH, and LA are similar in trend and shifted over the time	(2)
OS's solar energy is generated through the formulas	Solar g-curves for OS are very regular	(3)
We show the hourly average wind power.	All wind g-curves are relatively stable	(4)
OS, SH, and LA are in rich-wind coastal areas	Wind g-curves for OS, SH, and LA are on the top	(5)
CR and MB are in less-wind tropical areas.	Wind g-curves for CR and MB are at the bottom	(6)
Solar energy fluctuates enormously over time but not wind energy (#)	G-curves are similar to solar g-curves	(7)
Wind energy supplied the gap where solar energy is unavailable at night.	G-curves' crests and valleys are larger than solar g-curves.	(8)

#: The hourly variations of large-scale wind power stay 91%-94% of the time within ±5% of installed capacity [55].

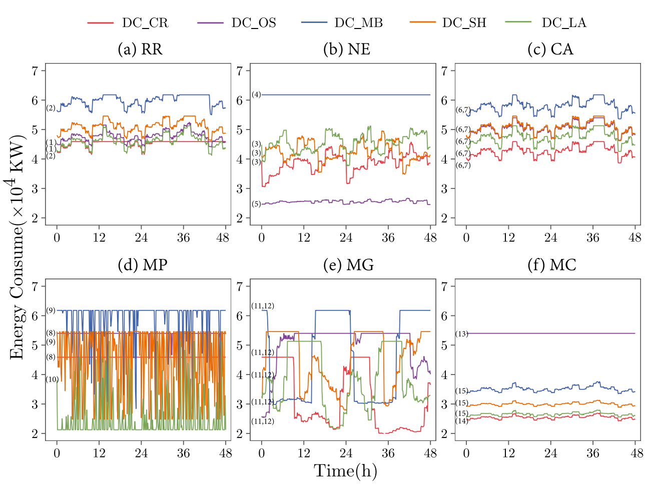


Fig. 10. Energy consumption (hourly) of five datacenters under six schedulers for two days.

datacenter with special features, a different scheduler brings a diverse workload (Table IX), and the workload dominates its energy consumption. The six schedulers in the integrated case are simple, and their effects are straightforward. Therefore, it is possible to infer the expected energy consumption. Besides, these c-curves would be baselines when studying more sophisticated schedulers in future.

TABLE XI
EXPECTATION AND OBSERVATIONS IN FIG. 10

Sch.	Expectation	Observation	DC	ID.
RR	Datacenters with high capability process the requests below the peak power	C-curves are similar to the workload curves	OS LA	(1)
	Datacenters with low capability process the workload crest with the continuous peak power	C-curves are similar to the workload curves chopped off crests.	CR MB	(2)
NE	Datacenters serve regions with similar and moderate populations.	C-curves are similar to RR ones	CR SH LA	(3)
	MB serves regions with larger populations.	C-curve is continuous in peak value	MB	(4)
	OS serves regions with small populations	C-curve is smooth in low values	OS	(5)
CA	Datacenters are neither idle nor full-load	C-curves have the same trends	ALL	(6)
	Datacenters have different capabilities	C-curves have diverse crests and valleys	ALL	(7)
MP	Datacenter whose name comes first has more chance to be full-loaded at peak power, and vice versa.	C-curves are nearly a straight line.	CR OS	(8)
		C-curves have valleys occasionally	MB SH	(9)
		C-curve has crests occasionally	LA	(10)
MG	More renewable energy, more requests, then g-curves shift with time zones	C-curves shift with time and are similar to g-curves	ALL	(11)
	Renewable reaches the peak values while the datacenter is already at peak power.	G-curves have sharp crests, but c-curves have flat crests	ALL	(12)
MC	Datacenter has the lowest renewable energy price.	C-curve is a straight line.	OS	(13)
	Datacenter has lower renewable energy price and small capability	C-curve is partly a straight line	CR	(14)
	Datacenters' workload is much less than their capability	C-curves are similar to the workload curve	MB SH LA	(15)

Fig. 10 shows the c-curves of five datacenters under the six schedulers in 48 hours. The lines in each figure present the c-curve for a datacenter under a scheduler. Referring to Tables IX and XI explains the observations in Fig. 10.

Overall, the trends of these curves are representative and prove that CloudSimPer successfully simulates DCREs' energy consumption.

D. RQ3: Utilizations

CloudSimPer designs the integrated case for evaluating the proposed scheduler in future studies, and defines renewable energy utilization (μ) as the metric for such evaluation. However, the integrated case should be valid first. For example, extraordinarily sufficient or inadequate renewable energy leads to resultless schedulers. The utilization under the scheduler should accord with the expectations in the well-designed case. Such expectations are apparent because the six schedulers in the case are straightforward.

Figs. 11 and 12 show the monthly-aggregated μ and hourly-aggregated μ of five datacenters under the six schedulers. Table XII explains the observations in Figs. 11 and 12. The details of Fig. 12 are abbreviated for succinctness.

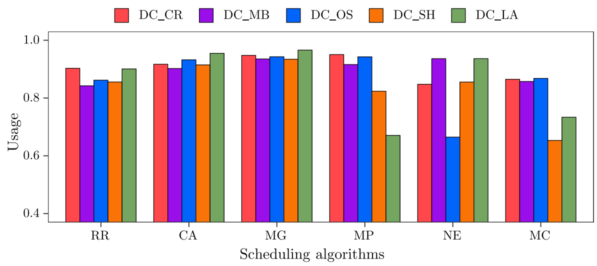


Fig. 11. Average renewable energy utilization of five datacenters under six schedulers for a month.

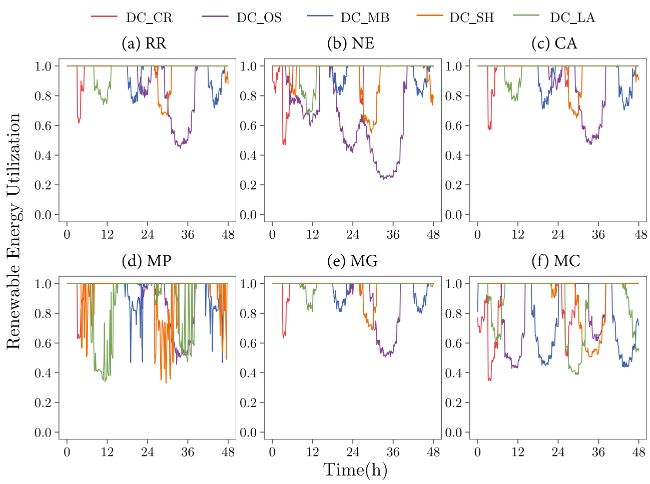


Fig. 12. Renewable energy utilization (hourly) of all datacenters under six schedulers for two days.

TABLE XII
EXPECTATION AND OBSERVATIONS IN FIG. 11 AND 12

Schedulers	Expectation	Observation
ALL	The generation powers are designed to be 65% of peak consumption powers.	All utilizations maintain relatively high values.
RR	Datacenters' workloads are the same; they are not all full-loaded; utilization is high if energy matches capability.	RR bars are relatively high.
CA	Datacenters' workloads scale with their capabilities; all datacenters are busy; utilization is high if energy matches capability.	CA bars are relatively high and higher than RR bars.
MG	Datacenters' workloads scale with their renewable energies; it is the best situation.	MG bars are higher than other groups
MP NE MC	Datacenters' workloads scale with other features; a datacenter may be idle when the renewable is sufficient	Some bars in MP, NE, and MC bars are significantly low.
ALL (curves in Figure 12)	Renewable energies and workloads are constants, not change with the schedulers.	Curves for different schedulers have a similar trend.
	Different schedulers vary the datacenters' real-time power	Curves' crests under six schedulers vary.

Overall, both monthly and hourly aggregated renewable energy utilization are representative, and prove that CloudSimPer successfully simulates the optimization effect of schedulers.

E. Validation

To validate CloudSimPer with real-world results, this subsection runs a full-fledged scheduler on CloudSimPer and produces results that align with the published ones. The selected scheduler would better satisfy the following four conditions:

1) It adopts renewable energies and publishes the energy data;
2) It adopts geo-distributed datacenters and publishes the data center's specifications;
3) It publishes the experimental results in detail, namely, time-serious curves rather than time-aggregated results.
4) It adopts the real-world data center as the experimental environment;

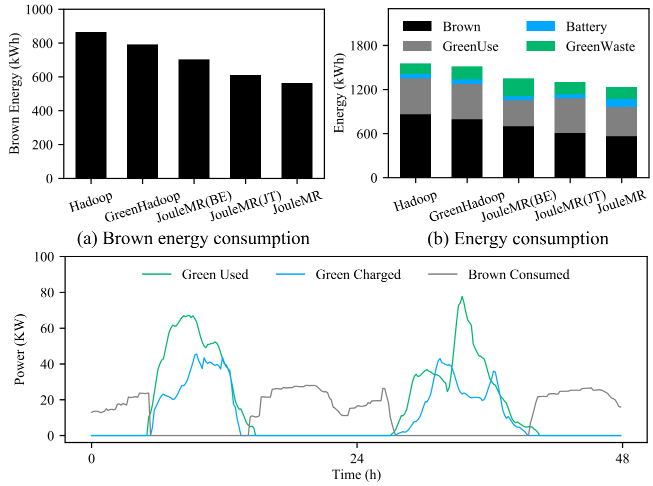
The former three conditions are mandatory for re-running the experiments. It is challenging to find a study that satisfies both four conditions, especially condition (4). If the selected scheduler adopts a simulator in its experiments, the simulator but not the scheduler dominates the difference when we compare the simulation results with CloudSimPer. In other words, we cannot use a simulation to validate our simulation; otherwise, the validation has no different from the previous section, namely whether the observations satisfy the expectations.

Fortunately, we manage to find one related work, JouleMR [54], nearly satisfying these conditions. However, we have to fill three gaps:

1) JouleMR runs not in a geo-distributed datacenter but in a cluster. So CloudSimPer configures a case quite smaller than the integrated one.
2) JouleMR truly runs MapReduce jobs on the cluster. *Loadgen* is a MapReduce job in the Gridmix benchmark in Hadoop whose length and execution time are configurable [54]. JouleMR uses *Loadgen* to generate Facebook workload. CloudSimPer can not execute MapReduce jobs but can directly use Facebook workload instead.
3) JouleMRs scheduler is a sophisticated MG scheduler with battery consideration. It runs on Hadoop middle-ware, where MapReduce jobs are scheduled in Java virtual machines. CloudSimPer does not support middle-ware simulation; analogously, it schedules workloads in virtual machines.

According to the description of Sections III, IV, and Table I in [54], we implement the five schedulers, namely Hadoop, GreenHadoop, JouleMR(BE), JouleMR(JT), and JouleMR, in CloudSimPer; then run a 48-hours-simulation. Fig. 13 shows CloudSimPers simulation result compared with the JouleMR paper [54].

Fig. 13(a) and (b) is corresponding to Fig. 10(a) and (b) in [54], respectively. The two sub-figures show the aggregated values of the brown-used, green-used, battery-used, and green-wasted. The JouleMR paper does not provide the exact values of these measurements, so a quantified comparison is difficult. However, the relative height difference of bars in these figures, which represents the differences in measurements among the five schedulers, are similar. Moreover, the JouleMR paper claims that JouleMR reduces 35% and 28% of brown-used compared with Hadoop and GreenHadoop, respectively. CloudSimPer



(a) Brown energy consumption

(b) Energy consumption

(c) Detailed energy consumption of JouleMR in CloudSimPer

Fig. 13. Validated results of JouleMR on CloudSimPer.

simulation backs these claims; the corresponding reduction ratios are 33.83% and 29.91%, calculated with data in Fig. 13. The optimization effects of JouleMR under a real-world cluster in [54] and CloudSimPer are almost the same, even though values of brown-used are different.

Fig. 13(c) shows hourly results compared with Fig. 11(d) in [54]. In Fig. 13(c), the sharps and values of green-use, greencharged, and brown-consumed energies are approximately the same as those of JouleMR [54]. Minor errors in simulation may be introduced due to the three gaps mentioned above. To this end, the simulated results through CloudSimPer align with the published ones.

VIII. Comparison

This section compares CloudSimPer with the typical simulators on the integrated case. After analyzing the metrics defined in Section V, we claim that the simulator affects the optimization results of the schedulers, and CloudSimPer with the integrated case is the most effective simulator among the competitors. For example, the same scheduler leads to different renewable utilization in the various simulators under the simulator case. But CloudSimPer is the most reasonable and explainable one.

A. Purposes

For the same case, schedulers and workload, we compare CloudSimPer with four typical simulators from different aspects following these Research Questions (RQs):

RQ1: Whether these simulators have the same simulated datacenters power? Which one is more acceptable?

RQ2: To what extent does the simulated power of each simulator conform to the available renewable energy? Which one is more acceptable?

RQ3: Do the schedulers have the same effect regarding renewable utilization and proportion as these simulators?

RQ4: How about the execution performance of CloudSimPer, especially in simulating larger-scale DCREs?

The simulation case and simulation duration, result aggregation, and visualization methods are the same as Section VII,

TABLE XIII
Functions of Five Competitors and CloudSimPer

Simulators	2.1	2.2	3.1	3.2	3.3	3.4	3.5	4.1	4.2	4.3	4.4	4.5	4.6
Trace	✓	×	✓	×	✓	×	×	✓	×	×	✓	✓	✓
Enum	✓	×	✓	×	✓	×	×	✓	×	✓	×	×	✓
Math	×	✓	✓	×	×	×	✓	×	✓	×	✓	×	✓
Native	×	✓	✓	×	×	×	✓	×	✓	×	×	×	✓
CloudSimPer	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓

except the simulator is not only CloudSimPer but also competitors. Among which, c-curves and d-curves are aggregated from all datacenters; utilizations and proportions are for each datacenter.

B. Competitors

This sub-section discusses the selection of competitors. We take two considerations about the selection. On the one hand, we cannot select the general-purpose simulators as mentioned in Section II and Table II for the following reasons: First, it is unfair to compare them for their implementation techniques vary. For example, they use different programming languages and run on various platforms. Second, none of these simulators can run the integrated case because they do not support renewable energy and geo-distributed datacenters together, and longer-term simulation.

On the other hand, we can neither select the experimental-purpose simulators. Section II-A shows various DCRE simulation works and concludes their functions in Table I. However, these functions are scattered over many experimental studies, but not collected on several simulators.

Since CloudSimPer encapsulates CloudSim, and CloudSim provides the essential capability for datacenter simulation, we extend the CloudSim in different ways to create the competitors. We cluster the existing experimental-purpose simulators into typical ones because the combinations of their functions (see Table I) in four groups are diversiform. For example, its workload and the generation mode of renewable energy are either trace-based or function-based; The power consumption model of the datacenter is either discrete or mathematically simulated.

Table XIII shows the typical simulators, their given names, and their functions. The functions supported by these simulators cover most state-of-the-art research. None-typical functions such as energy price, carbon emissions, bandwidth, and optimization approaches barely affect the simulator's validity, so Table XIII does not list them.

- *Trace:* the *Trace* simulator adopts the traces-based generators for workload and renewable energy. It supports the power functions, geo-distributed datacenter, and VM.
- *Enum:* the *Enum* simulator is the same as the *Trace* simulator, except it adopts enumerable values as powers and does not support VM.
- *Math:* the *Math* simulator adopts the mathematical function-based generators for workload and renewable energy. It supports power functions and geo-distributed datacenters, but not VM.
- *Native:* the *Native* simulator contains the minimal functions for DCRE simulation, namely the function-based

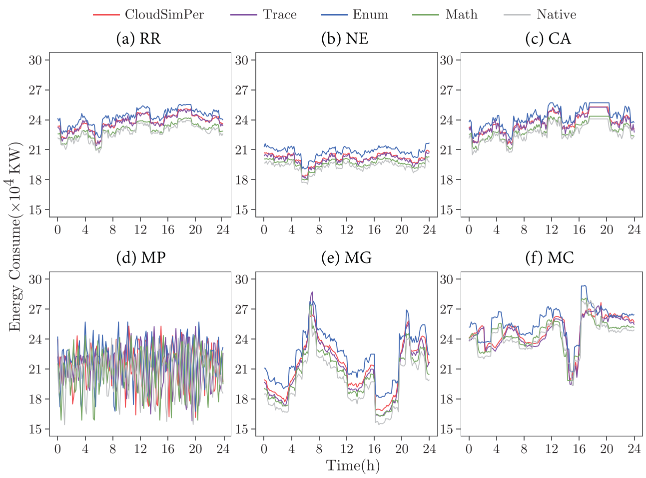


Fig. 14. C-curves of five simulators under six schedulers for a day, aggregated from all datacenters.

generators for workload, renewable energy, and geo-distributed datacenters.

To this end, the five simulators, named *Trace, Enum, Math, Native*, and CloudSimPer, are all extensions on CloudSim. Interestingly, *Native* comes first. It is the minimal extension on CloudSim to support DCRE simulation. And the other four are the extensions on *Native*. For example, item 4.6 is geographical support. It is an essential function for DCRE simulation, but only a few experimental works support it, neither CloudSim.

C. RQ1: C-Curves

Aggregated from all datacenters, Fig. 14 shows c-curves of the five simulators under the six schedulers within a day. First of all, the c-curves accord with those in Fig. 10, Section VII and Table XI explains the reasons for such accordance. The new four simulators also follow these explanations. This sub-section focus on the advantages of CloudSimPer proven by c-curves observations in Fig. 14, as explained in Table XIV.

According to the result comparison, the four selected simulators and CloudSimPer do not have the same simulated datacenters power. C-curves of CloudSimPer and *Trace* have moderate values, which are more reasonable than the other three simulators. CloudSimPer is better than *Trace* because it has less jitter in its c-curves, which indicates that datacenter power is smoother in the simulation.

D. RQ2: D-Curves

Aggregated from all datacenters, Fig. 15 shows d-curves of the five simulators under the six schedulers within a day. D-curve is the difference between c-curve and g-curve. Therefore, the observations on d-curves should accord with those on c-curves overlaying the differences between g-curves.

The case adopts four renewable energy traces and one renewable energy function. CloudSimPer perfectly simulates them because it supports trace-based and function-based energy generators. *Trace* and *Enum* only support trace-based energy generators, so they simulate more energy than CloudSimPer because their trace-based generators bring errors when simulating the

TABLE XIV
OBSERVATIONS (OBS.) AND ADVANTAGES (ADS.) OF CLOUDSIMPER

Obs: (1)	C-curves for each simulator are similar in their sharps, but they do not overlap. The gaps between them are apparent.
Ads. (1)	Simulators have the same results under the same case, workload, and schedulers. This observation proves CloudSimPer's correctness. The gaps indicate that the different implementations of CloudSimPer and other simulators affect c-curves.
Obs: (2)	Values of c-curves: $Enum > Trace \approx$ CloudSimPer $> Math > Native$
Ads. (2)	First, workloads dominate energy consumption. The case adopts Google workload traces. So the Enum, CloudSimPer, and Trace with trace-based workload generator are accurate. In contrast, the Math and Native with function-based workload generators (simulation function over time) have less workload. Second, Enum does not support VM, so it cannot adopt the temporal load balancing to save energy, but CloudSimPer and Trace do. Therefore, Enum has higher energy consumption than the latter two. To this end, CloudSimPer and Trace are better than the other three.
Obs: (3)	Jitter of c-curves: $Native > Enum > Trace >$ CloudSimPer $> Math$
Ads. (3)	Rough power generators cause the jitters. CloudSimPer and Math with function-based power generators can simulate multifarious powers smoothly. In contrast, Trace with traced power values, Enum with enumerable power values, and Native with a constant power value can not. CloudSimPer is better than Trace for fewer jitters.
Obs: (4)	CloudSimPer's c-curves under RR and CA are almost similar, except the crest of the former fluctuate while that of the latter is flat.
Ads. (4)	With the integrated case, CloudSimPer correctly simulated the two schedulers under which a datacenter's capacity dominates its power except that it is full-load. Under RR scheduler, not all datacenter may be full-load because datacenters' capacities vary; otherwise, the designed workload is overburdened. Under CA scheduler, all datacenter being full-load is a target, so the flat crest occurs; otherwise, the designed workload is insufficient.
Obs: (5)	CloudSimPer's c-curve under NE is lower (y-axis) than those under RR, CA, MG, MP, and MC.
Ads. (5)	CloudSimPer's results match the design purpose. Under the NE scheduler, datacenter MB, designed for the region with a larger population, is always full-load. Even so, many requests allocated to it are waiting for execution. Therefore, the lower values do not indicate the energy-saving but many unexecuted requests.

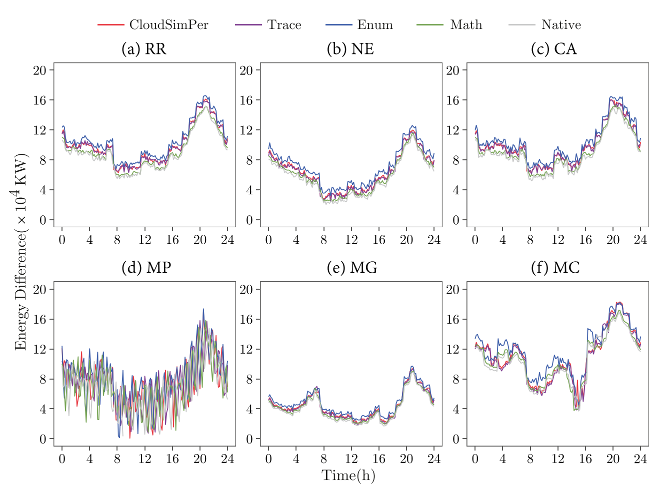


Fig. 15. D-curves of five simulators under six schedulers for a day, aggregated from all datacenters.

function-based energy in the case. In contrast, *Math* and *Native* only support function-based energy generators, so they simulate less energy than CloudSimPer because their function-based generators bring errors when simulating the four traces-based energy in the case. Overall, the expected values of g-curves are:

$$Enum = Trace > CloudSimPer > Math = Native \tag{12}$$

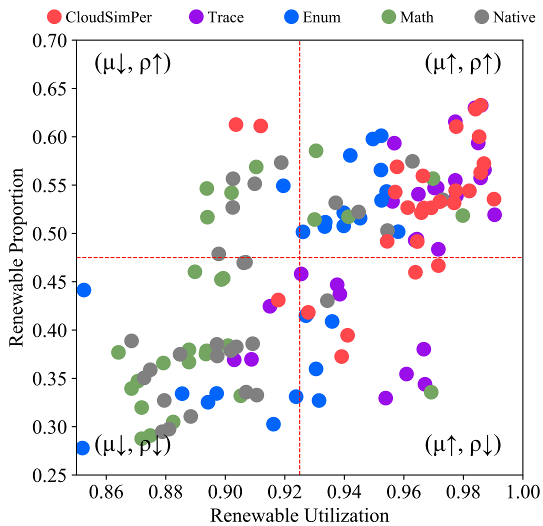


Fig. 16. 150 points of metrics (μ, ρ) for integrated case.

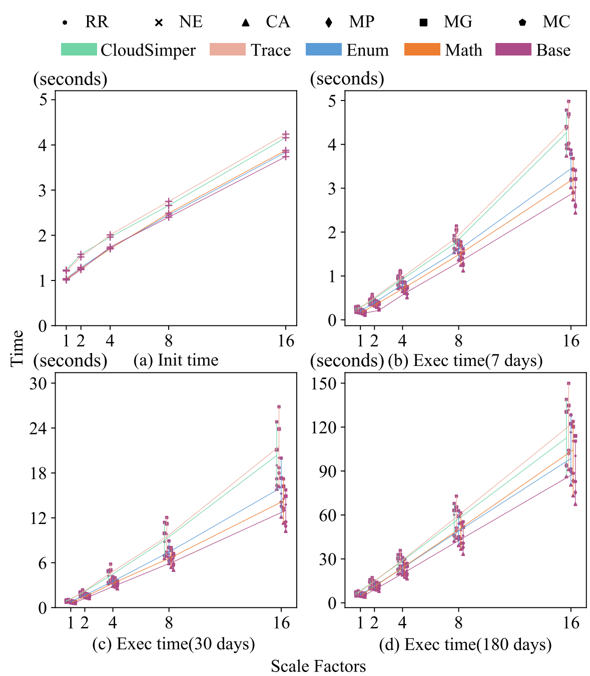


Fig. 17. Initialization time and execution time of the five simulators under the six schedulers in various SFs and simulation durations

With the expectation shown in (12), observations in Table XIV are all applicative to d-curves in Fig. 15 expect the Obs. (1) and (2). As shown in Fig. 15, the gaps between d-curves still exist but are not as apparent as Obs. (1); the values of d-curves still follow Obs. (2) but are not significant. These indicate that the g-curves' gaps offset the c-curves' gaps, but the latter is still dominated.

In the integrated case, the designed renewable energy is not enough, and renewable energy proportions (ρ) of datacenters are not as high as utilizations (μ). $\mu \in [0.85, 1)$ and $\rho \in [0.25, 0.7$. Therefore, the d-curves are always above $y = 0$. D-curves of all simulators follow the above facts and show similar effects under each scheduler. CloudSimPer may not be the best simulator whose d-curves are the closest to $y = 0$; however, according to the analysis of its c- and g- curves, it is the simulator that comforts the designed case best.

E. RQ3: Utilizations and Proportions

The renewable utilization and proportion (μ, ρ) jointly measure the validity of the simulator and simulation case. We estimate 150 groups of metrics (μ, ρ) to see the relationship between μ and ρ. (5 simulators $\times$ 6 schedulers $\times$ 5 datacenters), noticed that the integrated case is nominally divided into 5 sub-cases because each datacenter has its (μ, ρ), that is to say, we evaluate the parts of the case. We draw them on a scatter plot whose the x-axis is μ, and the x-axis is ρ, as shown in Fig. 16, in which symbols $\uparrow$ and $\downarrow$ represent relatively large and small values, respectively.

According to Section IV, we expect to see μ increase with ρ because $(\mu \uparrow, \rho \uparrow)$ is the condition of that "both simulator and simulation case are valid." The observation proves μ and ρ are positively related, confirming our expectations.

In Fig. 16, $\mu \in [0.85, 1)$ and $\rho \in [0.25, 0, 7]$. The (μ, ρ) pairs of CloudSimPer are closer to the top-right corner, while those of the *Native* and *Math* are closer to the bottom-left corner, comparing with other pairs representing the *Enum* and *Trace*. The reasons for these observations have been explained in the previous experiments and abbreviated here. To this end, the schedulers have a different effect on renewable utilization and proportion under these simulators and datacenters. The observations meet the expectation and prove that the CloudSimPer is effective, and the integrated case designer for CloudSimPer is also valid.

F. Efficiency

This section verifies the efficiency of CloudSimPer. As mentioned in Section IV, the time for executing a simulation case, named execution time, positively correlates to but is significantly less than the simulation time. We scale the integrated case multiple times, defined as SF (Scale Factors). For example, when SF $= 16$, the datacenter size would be scaled up 16 s times, 30400 servers in 80 datacenters, the same as sizes of requests and renewable energy. In the experiment, the SF is 1, 2, 4, 8, and 16; the simulation duration is 7, 30, and 180 days.

In the experiments, we measure the initialization time with various SF because CloudSimPer prepares traces and configuration. The initialization time relates to data center scales only. We also measure the execution time of various SF and simulation duration. Fig. 17 shows the initialization time and execution time of the five simulators under the six schedulers in various SFs and simulation durations. Table XV explains the corresponding observations of CloudSimPer on efficiency.

According to the result comparison, the execution time of four selected simulators and CloudSimPer are close under small-scale datacenters, and increases with datacenter scale. *Native*, *Math*, and *Enum* are slightly more efficient than CloudSimPer because CloudSimPer requires extra computation for new functions. Nevertheless, CloudSimPer's execution time is linear with

TABLE XV
OBSERVATIONS (OBS.) AND ADVANTAGES (ADS.) OF CLOUDSIMPER ON EFFICIENCY

Obs: (1)	Initialization time: $Native \approx Math \approx Enum$ ¡ CloudSimPer $\approx Trace$
Ads. (1)	Initialization time is relatively small compared with execution time, and linear to the datacenter scales. The initialization time of CloudSimPer and *Trace* are a little longer than other because they need to initialize the vms.
Obs: (2)	Execution time: CA ¡ MP $\approx$ MC ¡ RR ¡ MG ¡ NE (scheduler)
Ads. (2)	From the aspect of schedulers, whether requests are executed with or without delay dominates execution time. CA avoids delays because it allocates requests according to the datacenters capabilities. MP and MC may lead the busy and idle datacenters and possible delays (the datacenters order and energy price is unrelated to its capabilities). RR, MG, and NE bring more delays than other schedulers because they take no consideration about the datacenters capabilities.
Obs: (3)	Execution time: $Native < Math < Enum <$ CloudSimPer $< Trace$ (simulator)
Ads. (3)	From the aspect of simulators, their execution time is close in any cases because they are all based on CloudSim. Two factors dominate the execution time: 1) Whether support vm is the main factor. Simulation on vm management requires extra computation. It is why CloudSimPer and *Trace* take more time in execution; 2) the way to generate workloads, renewable energy, and server power is the inapparent factor. For example, $trace > rule > value \approx maxmin$. The extra cost, such as I/O and iteration, leads to different execution time.

both datacenter scale and simulation duration. The 180-days-simulation could be performed within 2 minutes. It is efficient enough for DCRE simulation.

IX. CONCLUSION AND FUTURE WORK

In this paper, we propose CloudSimPer, a general-purpose DCRE simulator. It introduces geo-distributed datacenter modules, establishes a periodical and intermittent renewable energy supply model and customizable power consumption model, and integrates the evaluation metric and universal simulation case. The main technical challenges are an architectural design for complex entities, geo-distributed and time-dependent renewable energy and requests generation, universal metrics for schedulers, and a standard simulation case. Our work offers essential services for the datacenter designers to test their designing plans before carrying them out, for the datacenter administrators to examine their planned strategies before deploying them, and for researchers to evaluate their proposition and compare them with others experimentally.

Our work can be extended in two directions. First, CloudSimPer can further support the network cost between datacenters, and then schedulers would prioritize such cost. Second, CloudSim manages the computing resource through the heavyweight virtual machine, and so does the CloudSimPer. How to support the simulation on lightweight containers is also an open challenge.

REFERENCES

[1] A. K. Singh, D. Saxena, J. Kumar, and V. Gupta, "A quantum approach towards the adaptive prediction of cloud workloads," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 12, pp. 2893–2905, Dec. 2021.

[2] J. Song, P. Zhu, Y. Zhang, and G. Yu, "Versatility or validity: A comprehensive review on simulation of datacenters powered by renewable energy mix," *Future Gener. Comput. Syst.*, vol. 136, pp. 326–341, Nov. 2022.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Pract. Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[4] R. Bianchini, "Leveraging renewable energy in data centers: Present and future," in *Proc. 21st Int. Symp. High-Perform. Parallel Distrib. Comput.*, 2012, pp. 135–136.

[5] I. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and GreenSwitch: Managing datacenters powered by renewable energy," *ACM SIGPLAN Notices*, vol. 48, no. 4, 2013, pp. 51–64.

[6] Í. Goiri, T. D. Le, J. Nguyen, J. G. Torres, and R. Bianchini, "GreenHadoop: Leveraging green energy in data-processing frameworks," in *Proc. 7th ACM Eur. Conf. Comput. Syst.*, 2012, pp. 57–70.

[7] Í. Goiri et al., "Matching renewable energy supply and demand in green datacenters," *Ad Hoc Netw.*, vol. 25, pp. 520–534, Feb. 2015.

[8] W. Katsak, I. Goiri, R. Bianchini, and T. D. Nguyen, "GreenCassandra: Using renewable energy in distributed structured storage systems," in *Proc. 6th Int. Green Sustain. Comput. Conf.*, 2015, pp. 1–8.

[9] M. E. Haque, I. Goiri, R. Bianchini, and T. D. Nguyen, "GreenPar: Scheduling parallel high performance applications in green datacenters," in *Proc. 29th ACM Int. Conf. Supercomputing*, 2015, pp. 217–227.

[10] S. Subramanya, Z. Mustafa, D. Irwin, and P. Shenoy, "Beyond energy-efficiency: Evaluating green datacenter applications for energy-agility," in *Proc. 7th ACM/SPEC Int. Conf. Perform. Eng.*, 2016, pp. 185–196.

[11] X. Zhou, H. Cai, Q. Cao, H. Jiang, L. Tian, and C. Xie, "GreenGear: Leveraging and managing server heterogeneity for improving energy efficiency in green data centers," in *Proc. Int. Conf. Supercomputing*, 2016, pp. 1–14.

[12] A. Nadjaran Toosi, C. Qu, M. D. de Assuno, and R. Buyya, "Renewable-aware geographical load balancing of web applications for sustainable data centers," *J. Netw. Comput. Appl.*, vol. 83, pp. 155–168, Apr. 2017.

[13] L. Grange, G. Da Costa, and P. Stolf, "Green IT scheduling for data center powered with renewable energy," *Future Gener. Comput. Syst.*, vol. 86, pp. 99–120, Sep. 2018.

[14] A. Oleksiak, W. Piatek, K. Kuczynski, and F. Sidorski, "Reducing energy costs in data centres using renewable energy sources and energy storage," in *Proc. 5th Int. Workshop Energy Efficient Data Centres*, 2016, pp. 1–8.

[15] K.-K. Nguyen, M. Cheriet, M. Lemay, M. Savoie, and B. Ho, "Powering a data center network via renewable energy: A green testbed," *IEEE Internet Comput.*, vol. 17, no. 1, pp. 40–49, Jan./Feb. 2013.

[16] C.T. Dupont, G. Schulze, A. GiulianiSomov, and F. Hermenier, "An energy aware framework for virtual machine placement in cloud federated data centres," in *Proc. 3rd Int. Conf. Future Energy Syst. Where Energy Comput. Commun. Meet - e-Energy*, 2012, pp. 1–10. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2208828.2208832

[17] A. Khosravi, A. Nadjaran Toosi, and R. Buyya, "Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud data centers," *Concurrency Comput.: Pract. Experience*, vol. 29, no. 18, 2017, Art. no. e4125.

[18] M. Xu and R. Buyya, "Managing renewable energy and carbon footprint in multi-cloud computing environments," *J. Parallel Distrib. Comput.*, vol. 135, pp. 191–202, Jan. 2020.

[19] N. Gholipour, E. Arianyan, and R. Buyya, "A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers," *Simul. Modelling Pract. Theory*, vol. 104, 2020, Art. no. 102127.

[20] E. Sheme, S. Holmbacka, S. Lafond, D. Luanin, and N. Frashriri, "Feasibility of using renewable energy to supply data centers in 60 north latitude," *Sustain. Comput., Inform. Syst.*, vol. 17, pp. 96–106, Mar. 2018.

[21] K. Kurowski et al., "DCworms A tool for simulation of energy efficiency in distributed computing infrastructures," *Simul. Modelling Pract. Theory*, vol. 39, pp. 135–151, Dec. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1569190X1300124X

[22] R. D. C. E. ZimmermanMurillo-Sanchez and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011. [Online]. Available: https://ieeexplore.ieee.org/document/5491276/

[23] I. D.Y. Courchelle, T. LabitMonteil, and T. Guerout, "A data model for supplying a data center with several energy sources," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput. Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet of People Smart World Congr.*, 2016, pp. 1033–1038. [Online]. Available: https://ieeexplore.ieee.org/document/7816957/

[24] A.C. Forestiero, M. Mastroianni, G. MeoPapuzzo, and M. Sheikhalishahi, "Hierarchical approach for efficient workload management in geo-distributed data centers," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 1, pp. 97–111, Mar. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7557052/

[25] H.A. Casanova, A. Giersch, M. LegrandQuinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms," *J. Parallel Distrib. Comput.*, vol. 74, no. 10, pp. 2899–2917, Oct. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0743731514001105

[26] H.A. Gupta, A. VahidDastjerdi, S. K. DastjerdiGhosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments: IFogSim: A toolkit for modeling and simulation of Internet of Things," *Softw.: Pract. Experience*, vol. 47, no. 9, pp. 1275–1296, Sep. 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/spe.2509
[27] A. Zahra, M. Pore, and S. K. S. Gupta, "Impact of workload and renewable prediction on the value of geographical workload management," in *Proc. Energy-Efficient Data Centers - 2nd Int. Workshop, E^2DC*, Berkeley, CA, USA, 2013, pp. 1–15. doi: 10.1007/978-3-642-55149-9_1.
[28] M. Xu, A. Toosi, B. Bahrani, R. Razzaghi, and M. Singh, "Optimized renewable energy use in green cloud data centers," in *Proc. Int. Conf. Serv.-Oriented Comput.*, 2019, pp. 314–330.
[29] C. Xu, P. Li, R. Xia, S. Guo, and M. Guo, "Renewable energy-aware Big Data analytics in Geo-distributed data centers with reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 205–215, Jan.-Mar., 2020.
[30] M. S.Y. Hasan, T. K. Ledoux, and J. Pazat, "Exploiting renewable sources: When green SLA becomes a possible reality in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 5, no. 2, pp. 249–262, Apr.-Jun., 2017, doi: 10.1109/TCC.2015.2459710.
[31] Z. Xu, N. Deng, C. Stewart, and X. Wang, "CADRE: Carbon-aware data replication for GEO-diverse services," in *Proc. IEEE Int. Conf. Autonomic Comput.*, 2015, pp. 177–186.
[32] J. Son, T. He, and R. Buyya, "Cloudsimsdn-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments," *Software: Pract. Experience*, vol. 49, no. 12, pp. 1748–1764, Dec. 2019.
[33] Wikipedia, "Population_Density," 2022. [Online]. Available: https://en.wikipedia.org/wiki/Population_density
[34] Wikipedia, "Time_zone," 2022. [Online]. Available: https://en.wikipedia.org/wiki/Time_zone
[35] Center for Spatially Integrated Social Science, "Friedrich Ratzel, Clark Wissler, and Carl Sauer: Culture area research and mapping by Nina Brown," 2022. [Online] Available: https://web.archive.org/web/20050309084859/http://www.csiss.org/classics/content/15
[36] Datacenterknowledge. [Online]. Available: https://www.datacenterknowledge.com/cloud/telegeography-maps-world-s-cloud-data-centers
[37] N. Kumar, G. S. Aujla, S. Garg, K. Kaur, R. Ranjan, and S. K. Garg, "Renewable energy-based multi-indexed job classification and container management scheme for sustainability of cloud data centers," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2947–2957, May 2019.
[38] I. De Courchelle, T. Gurout, G. Da Costa, T. Monteil, and Y. Labit, "Green energy efficient scheduling management," *Simul. Modelling Pract. Theory*, vol. 93, pp. 208–232, May 2019.
[39] Z. Zhou et al., "Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms," *Future Gener. Comput. Syst.*, vol. 86, pp. 836–850, Sep. 2018.
[40] C. A. Stewart, M. R. Link, R. Henschel, D. Hancock, and H. Li, "Technical Report: Benchmarking an HP DL580 cluster at Indiana University (Mason)," Art. no. 11.
[41] P. Singh and K. Barr, "Analysis of IBM system x3850 M2 performance and scalability with VMware vSphere 4 and SAP solutions," Art. no. 15.
[42] C. Ma, X. Sun, C. Yao, M. Tian, and L. Li, "Research on sheep recognition algorithm based on deep learning in animal husbandry," *J. Phys.: Conf. Ser.*, vol. 1651, no. 1, 2020, Art. no. 012129.
[43] E.STAR, "Market and industry scoping report: Solar PV inverters," 2022. [Online]. Available: https://www.energystar.gov/ia/products/downloads/Solar_PV_Inverters_Scoping_Report.pdf
[44] A. Azad, M. Rasul, M. Alam, S. A. Uddin, and S. K. Mondal, "Analysis of wind energy conversion system using weibull distribution," *Procedia Eng.*, vol. 90, pp. 725–732, 2014.
[45] E. Dokur and M. Kurban, "Wind speed potential analysis based on Weibull distribution," *Balkan J. Elect. Comput. Eng.*, vol. 3, no. 4, pp. 231–235, 2015.
[46] National Renewable Energy Laboratory, "NREL Government," 2022. [Online]. Available: https://midcdmz.nrel.gov/apps/sitehome.1212pl?site=BMS
[47] National Renewable Energy Laboratory, "Wind research," 2022. [Online]. Available: https://www.nrel.gov/wind/
[48] X. Sun, N. Ansari, and R. Wang, "Optimizing resource utilization of a data center," *IEEE Commun. Surv. Tut.*, vol. 18, no. 4, pp. 2822–2846, Fourth Quarter, 2016.
[49] M. Eteiba, S. Barakat, M. Samy, and W. I. Wahba, "Optimization of an off-grid PV/Biomass hybrid system with different battery technologies," *Sustain. Cities Soc.*, vol. 40, pp. 713–727, Jul. 2018.
[50] R. Dufo-Lpez, J. M. Lujano-Rojas, and J. L. Bernal-Agust, "Comparison of different leadacid battery lifetime prediction models for use in simulation of stand-alone photovoltaic systems," *Appl. Energy*, vol. 115, pp. 242–253, Feb. 2014.
[51] C. Dupont, "Building application profiles to allow a better usage of the renewable energies in data centres," in *Energy EfficientData Centers*, ser. Title: Lecture Notes in Computer Science, vol. 8945, S. Klingert, M. Chinnici, and M. Rey Porto, Eds., Berlin, Germany: Springer, 2015, pp. 120–131.
[52] J. Wilkes, "More Google cluster data," 2011. [Online]. Available: http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html
[53] H. Holttinen, "Hourly wind power variations in the Nordic countries," *Wind Energy*, vol. 8, no. 2, pp. 173–195, Apr. 2005.
[54] Z.B. NiuHe and F. Liu, "JouleMR: Towards cost-effective and green-aware data processing frameworks," *IEEE Trans. Big Data*, vol. 4, no. 2, pp. 258–272, Jun. 2018, doi: 10.1109/TBDATA.2017.2655037.

Jie Song received the PhD degree in computer science from Northeastern University, in 2008. He is a professor with Software College, Northeastern University. His research interest includes big data management, green computing, and machine learning.

Peimeng Zhu received the bachelor degree in software engineering from Northeastern University, in 2020. He is a master student at Software College, Northeastern University. His research interest is green datacenters.

Yanfeng Zhang received the PhD degree in computer science from Northeastern University, China, in 2012. He is a professor with computer science and engineering, Northeastern University. His research interest includes Big Data processing and distributed system.

Ge Yu (Senior Member, IEEE) received the PhD degree in computer science from Kyushu University of Japan, in 1996. He is a professor with computer science and engineering, Northeastern University. His research interest includes database theory and technology, distributed and parallel systems.