Survey on Energy Consumption Optimization Approach in Container Based Cloud Environments

Francesco Pace Napoleone, Marian-Catalin Mutu-Costan, Ryzhuk Ihor, Ewan Soueidan
Université de Toulouse

Abstract
Growing demand for cloud services has shifted data center design from performance-centric to energy-efficient architectures. Containerization, a lightweight alternative to virtual machine, enables flexible resource allocation and reduced overhead. This survey reviews energy optimization techniques in container-based cloud environments by systematically selecting studies from 2010–2020 and categorizing them by scheduling scope, optimization method, energy-saving strategy, and Docker maturity. Our analysis covers early VM consolidation methods that achieved up to 83 % energy savings, predictive scheduling, and advanced container orchestration. Overall, container-based optimizations—particularly those leveraging AI and hybrid heuristics—significantly reduce energy consumption while maintaining Quality of Service (QoS), pointing to promising future research directions.

Keywords

Cloud Computing; Containerization; Energy Optimization; Microservices; Resource Allocation

Contents

1 Introduction

Cloud computing evolved to support scalable platform usage, moving from single-file applications to client/server architectures with monolithic backends, then to microservices with containers, and now transitioning to micro-frontends. Containers run directly on the host using Linux CGroups, bypassing the hypervisor overhead inherent in VMs (e.g., Proxmox, VMWare), which allows for full utilization of host CPU resources and simplifies telemetry monitoring. This lightweight nature is evident in Docker files that simply copy files and run builds, making containers easier to integrate into CI/CD pipelines.

However, this increased reliance on cloud-based infrastructures has led to a significant rise in energy consumption. In 2006, the electricity costs for IT infrastructures in the United States alone were estimated at $4.5 billion, with projections to double by 2011[4]. Energy consumption optimization has since become a critical concern, especially as cloud data centers now account for approximately 1-1.5% of global electricity use[11]. Despite efficiency improvements, the demand for digital services continues to grow, pushing the need for more sustainable solutions.

Early research focused on heuristic-based approaches to optimize virtual machine (VM) placement, achieving energy savings of up to 83% while maintaining only a 1.1% service level agreement (SLA) violation rate[4]. More recently, research has shifted from VM-based allocation towards containerized environments, where energy efficiency is influenced by scheduling strategies, workload distribution, and infrastructure optimizations. Studies indicate that modern cloud providers, including Amazon, Microsoft, Google, and Meta, have doubled their energy consumption between 2017 and 2021, reaching 72 TWh in 2021[14;9;11].

The paper "Survey on Energy Consumption Optimization Approach in Container Based Cloud Environments" further highlights that containerization not only drives scalability and reproducibility but also plays a crucial role in optimizing energy consumption. It explores strategies for efficient resource allocation, reducing power overhead, and ensuring that the benefits of container-based deployments extend beyond performance to sustainability in cloud infrastructures.

In this work, we present a state-of-the-art review on Energy Consumption Optimization Approaches in Container-Based Cloud Environments. Our survey of the available literature—predominantly spanning from 2010 to 2020—reveals that foundational research primarily focused on energy measurement, basic optimization strategies, and energy visualization techniques[4].

Early contributions, such as those by Beloglazov and Buyya[4], as well as Piraghaj et al.[15], laid the groundwork for dynamic resource allocation and energy-efficient container consolidation. Later advancements introduced more sophisticated container scheduling mechanisms, including availability-aware scheduling[1], concurrent scheduling in heterogeneous clusters[10], and hybrid AI-driven resource allocation[17].

In parallel, energy-efficient resource management techniques gained prominence, incorporating renewable energy-aware scheduling[12], optimization-based consolidation methods[16;15], and brownout-based scheduling strategies[18]. Several studies further explored predictive optimization and SLA-aware provisioning frameworks to enhance energy efficiency[7;8;13;5;6].

Beyond optimization techniques, researchers have also examined broader energy consumption trends and policy implications[2], reflecting the increasing emphasis on sustainability in cloud computing. Additionally, efforts in DevOps-driven elastic container management have contributed to improving the adaptability and efficiency of containerized cloud applications[3].

Although these prior works have significantly advanced the field, our review highlights an evolving trend toward more integrated and user-centric energy management strategies, as reflected in recent data on energy consumption in cloud environments[14;9;11].

2 Methodology

Systematic Procedure for Paper Selection

The research process began by defining our scope together with our supervisor, Professor Jean-Marc Pierson. We focused on creating a state-of-the-art review addressing energy consumption optimization within Container and Cloud Computing.

Initially, we defined a structured "Search Pipeline" illustrated in Figure 1, ensuring systematic identification and evaluation of relevant papers.

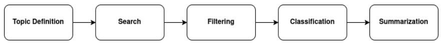

Figure 1: Search Pipeline Flowchart

We developed an initial search query: *((energy OR resource) AND container)*, including "resource" to account for better resource utilization directly influencing energy efficiency.

Clear exclusion criteria were established beforehand (Table 1), filtering out irrelevant work and focusing exclusively on impactful research.

Direct Exclusion of Paper If
Work is not in English
Work is not a scientific paper
Work has fewer than 10 citations
Work is not related to containers/cloud

Table 1: Exclusion Criteria

This approach yielded 34 relevant papers, sourced across various publishers, as depicted in Figure 2.

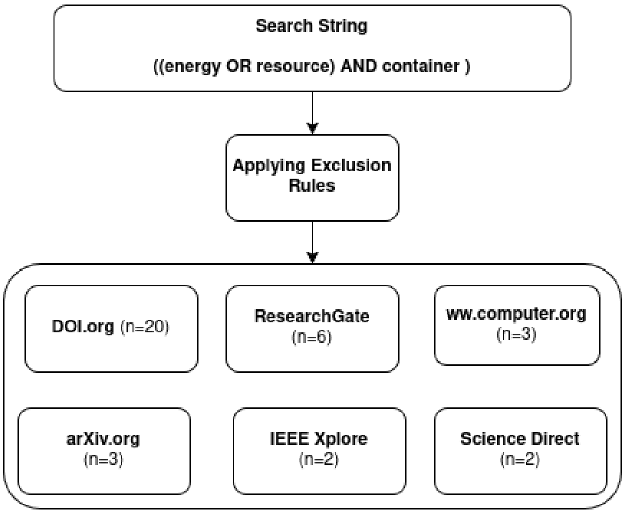

Figure 2: Distribution of Exported Papers by Publisher

The majority of selected studies were published between 2017 and 2019 (Figure 3), reflecting a peak period of research interest.

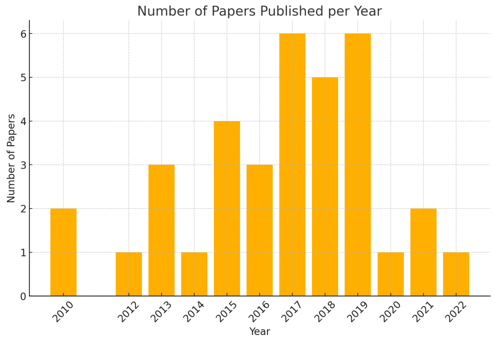

Figure 3: Papers per Year

After paper selection, we defined specific classification metrics (Table 2) to analyze the technical dimensions of energy-aware scheduling approaches.

Classification Categories
Optimization Technique
Target Level
Scheduling Scope
Energy Strategy
Docker Awareness

Table 2: Paper Classification Metrics

The classification categories are defined as follows:

- **Scheduling Scope**
  - **Online**: Decisions at runtime.
  - **Offline**: Known workload beforehand.
  - **Predictive**: Forecast-based decisions.
  - **Reactive**: Triggered by SLA violations or thresholds.
  - **Concurrent**: Simultaneous multiple-task scheduling.
- **Optimization Technique**
  - **Greedy**: Stepwise local decisions.
  - **Model-Based**: Predictive analytical models.

  - **PSO**: Bio-inspired global optimization.
  - **Evolutionary**: Genetic/evolution-based methods.
  - **Meta-Heuristic**: High-level heuristic strategies.
  - **MCMF**: Flow-network optimization.

- **Energy Strategy**
  - **Consolidation**: Migrate to reduce active hosts.
  - **Brownout**: Temporarily disable app components.
  - **QoS Maintenance**: Energy-saving without SLA violations.
  - **Renewable-aware**: Optimize for renewable energy availability.

- **Docker Awareness**
  - **Pre-Docker**: Pre-containerization (VM-only).
  - **Early Docker**: Containers used like VMs.
  - **Mature Docker**: Container-specific optimizations.

We incorporated **Docker Awareness** to highlight the technological transition from virtual machines to containers, essential for understanding changes in scheduling complexity. **Energy Awareness** is included to represent approaches' capability to balance energy efficiency with QoS constraints, recognizing the different trade-offs between performance and sustainability.

Finally, Tables 3 and 4 summarize our classified selection, and Figure 4 visually correlates scheduling approaches with their publication years.

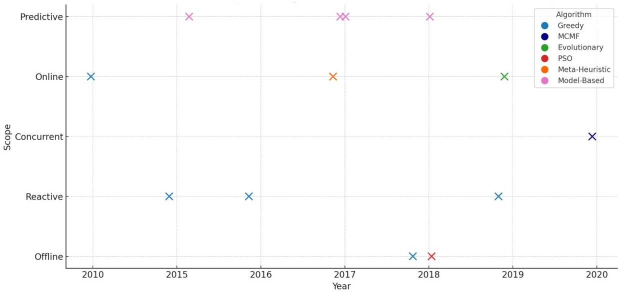

Figure 4: Trend Analysis

Paper Ref.	**Optimization Technique**	**Target Level**	**Scheduling Scope**	**Energy Strategy**	**Docker Awareness**
Availability-Aware Scheduler[1] (2018)	Greedy	Container, VM	Offline	Availability	Mature Docker
Renewable-Aware Scheduling[12] (2019)	Greedy	Container, VM	Reactive	Consolidation	Mature Docker
Concurrent Container Scheduling[10] (2020)	MCMF	Container	Concurrent	Consolidation	Mature Docker
GP Hyper-Heuristic for Containers[17] (2019)	Evolutionary	Container, VM	Online	Consolidation	Mature Docker
PSO-Based Container Consolidation[16] (2018)	PSO	Container	Offline	Consolidation	Mature Docker
Energy-Efficient Container Framework[15] (2015)	Greedy	Container, VM	Reactive	Consolidation	Early Docker
VM Consolidation Strategy[6] (2017)	Meta-Heuristic	VM	Online	Consolidation	Mature Docker
Predictive Resource Provisioning[7] (2015)	Model-based	VM	Predictive	Consolidation	Early-Docker
Dynamic VM Reallocation[4] (2010)	Greedy	VM	Online	Consolidation	Pre-Docker
Autonomic Container Management[3] (2017)	Model-based	Container, VM	Predictive	QoS	Mature Docker

Paper Ref.	Optimization Technique	Target Level	Scheduling Scope	Energy Strategy	Docker Awareness
Brownout-Based Scheduling[18] (2016)	Greedy	Application Layer	Reactive	Brownout	Mature Docker
GPR + Convex VM Planning[5] (2017)	Model-Based	VM, PM	Predictive	Consolidation	Mature Docker
SLA-Aware Consolidation[13] (2018)	Model-Based	VM, PM	Predictive	Consolidation	Mature Docker

Table 3: Classification of Technical Papers on Energy-Aware Scheduling

Paper Ref.	Type	Scope / Focus
Survey on Energy-Aware Scheduling[8]	Survey	Overview of VM and container-level techniques
Global Energy Estimates[14]	Analysis	Environmental impact of data centers
Cloud Impact on Energy[9]	Report	Energy use in European data centers
IEA Report on Data Centers[11]	Report	Global energy trends in ICT and data centers
EU Data Center Energy Trends[2]	Analysis	EU-level trends in ICT energy consumption

Table 4: Survey and Analysis Papers

3 Trend reconstruction

Chronological evolution of the trend

3.1 Early approaches

Early approaches date back to 2010, with Docker being released in 2013. In 2006, the cost of electricity consumed by IT infrastructures in the US was estimated at 4.5 billion dollars and was projected to double by 2011[4].

In the initial attempts described in[4], a simple Bin Packing variation with dynamic voltage and frequency scaling (DVFS) enabled was used to address this problem. The results, obtained using the CloudSim environment, showed an energy savings gain of 83% compared to no policy, with an SLA violation of 1.1%. This marked the beginning of research in this area.

By 2015, researchers shifted toward predictive modeling to overcome the limitations of static thresholds. In[7], a Wiener filter-based predictor was introduced to estimate cluster workloads, combined with Best Fit Decreasing for physical machine allocation. This approach improved energy efficiency by up to 33% compared to heuristic methods. Shortly after, in 2017, further improvements were made to predictive models[5]. This work extended previous research by incorporating Gaussian Process Regression (GPR) for non-stationary workload prediction, accelerated using the Fast Fourier Transform to mitigate GPR's computational complexity. Additionally, a convex optimization-based migration strategy was employed. These enhancements resulted in a 35% energy efficiency improvement over heuristic methods, with only a 15% latency tradeoff.

3.2 From Virtual Machines to Containers

As cloud infrastructures evolve, the shift from VM-based to container-based deployments has become central to achieving energy efficiency and maintaining SLA compliance. Containers offer a lightweight alternative to VMs, reducing overhead and enabling more flexible resource allocation[1]. Early container-based systems focused on reducing active VM counts through migration techniques—employing modular watchdogs and Pearson correlation checks to optimize resource usage[1].

Subsequent research advanced container scheduling by using flow network models to frame the allocation problem as a minimum cost flow problem. This approach effectively balances multi-resource constraints (e.g., CPU and memory) and has proven scalable in simulations involving up to 5000 machines[10].

More recent hybrid methods, which combine evolved rules from genetic programming with human-designed heuristics, address the two-level allocation challenge (assigning containers to VMs and VMs to physical machines). These techniques not only enhance energy efficiency but also improve fault tolerance by incorporating reliability metrics such as Mean Time To Failure (MTTF) and Mean Time To Repair (MTTR)[17;1]. Complementary studies, such as those by Kumar et al.[12] and Shi et al.[16], further demonstrate that container-based approaches can significantly reduce energy consumption and improve overall system performance.

This evolution—from basic migration techniques to advanced, hybrid scheduling—forms a critical foundation for the energy optimization and resilience strategies.

4 Method Analysis

Hardware analysis

4.1 Methodology

Over time, different researchers have utilized various testing environments and hardware configurations to evaluate their energy-aware scheduling algorithms. To examine the evolution of their experimental approaches, we have collected and structured data from the selected papers, focusing on the simulation environment, the number of CPU Cores, memory capacity, and power consumption of the experimentation hardware. Specifically, we define:

- **P idle (W)** as the power consumption at low CPU utilization (generally below 50% load).
- **P max (W)** as the power consumption at maximum CPU load.

In some studies, multiple hardware configurations were reported—either due to hardware availability or to assess performance on more powerful setups. In these cases, multiple entries are provided in the table. Note that when power consumption figures were not experimentally reported but instead obtained from hardware specification (by searching for the make and model), those values are marked with an asterisk ($*$). Whenever certain data fields were omitted by the authors, we use “Nm” (not mentioned).

Paper Ref.	Simulation Environment	CPU Cores	Memory (GB)	Power P idle (W)	Power P max (W)
Availability-Aware Scheduler[1] (2018)	CloudSim	Nm	Nm	Nm	Nm
Renewable-Aware Scheduling[12] (2019)	Nm	4	64	100	150
Renewable-Aware Scheduling[12] (2019)	Nm	8	128	120	200
Renewable-Aware Scheduling[12] (2019)	Nm	16	256	150	250
Concurrent Container Scheduling[10] (2020) Homogeneous Cluster	ExoGENI	2	6	Nm	Nm
Concurrent Container Scheduling[10] (2020)	ExoGENI	1	3	Nm	Nm
Concurrent Container Scheduling[10] (2020)	ExoGENI	2	6	Nm	Nm
Concurrent Container Scheduling[10] (2020)	ExoGENI	4	12	Nm	Nm
GP Hyper-Heuristic for Containers[17] (2019)	Nm	Nm	0.8	Nm	Nm
PSO-Based Container Consolidation[16] (2018)	Nm	4	16	Nm	Nm
Energy-Efficient Container Framework[15]	CloudSim	4	64	86	117
Energy-Efficient Container Framework[15]	CloudSim	8	128	93	135
Energy-Efficient Container Framework[15]	CloudSim	16	256	66	247
VM Consolidation Strategy[6] (2017)	Nm	2	Nm	170	220
Predictive Resource Provisioning[7] (2015)	Nm	Nm	Nm	107	300.81
Dynamic VM Reallocation[4] (2010)	Nm	1	8	Nm	Nm
Autonomic Container Management[3] (2017)	LEGIS	Nm	16	Nm	Nm
Brownout-Based Scheduling[18]	CloudSim	2	4	156$*$	247$*$
GPR + Convex VM Planning[5] (2017)	Nm	4	12	225$*$	375$*$
SLA-Aware Consolidation[13] (2018)	CloudSim	2	4	102	117
SLA-Aware Consolidation[13] (2018)	CloudSim	2	4	116	135

Table 5: Hardware and Power Specifications

4.2 Plots and Visualization

We visualized the simulation environments used in each paper over time to observe how they have evolved (Figure 5). Each point corresponds to a paper, placed according to its publication year and the simulation platform it utilized (CloudSim, ExoGENI, LEGIS, or “Nm” if not mentioned) Each point denotes a paper, positioned by its publication year and the simulation platform it utilized.

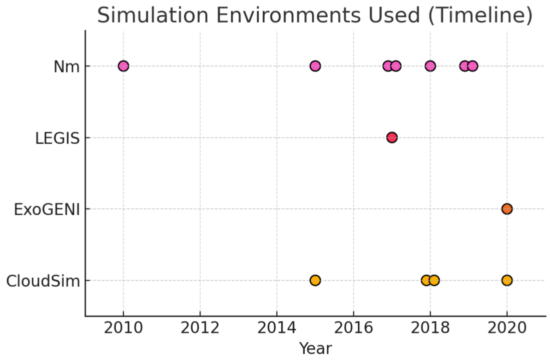

Figure 5: Timeline of simulation environments used in each paper (2010–2020).

Additionally, we plotted the relationship between the number of CPU cores and the *energy delta* (the difference $P_{\max} - P_{idle}$), shown in Figure 6. This second figure highlights how dynamic power consumption scales with increased core counts and is further distinguished by algorithm type (color) and simulation environment (marker shape).

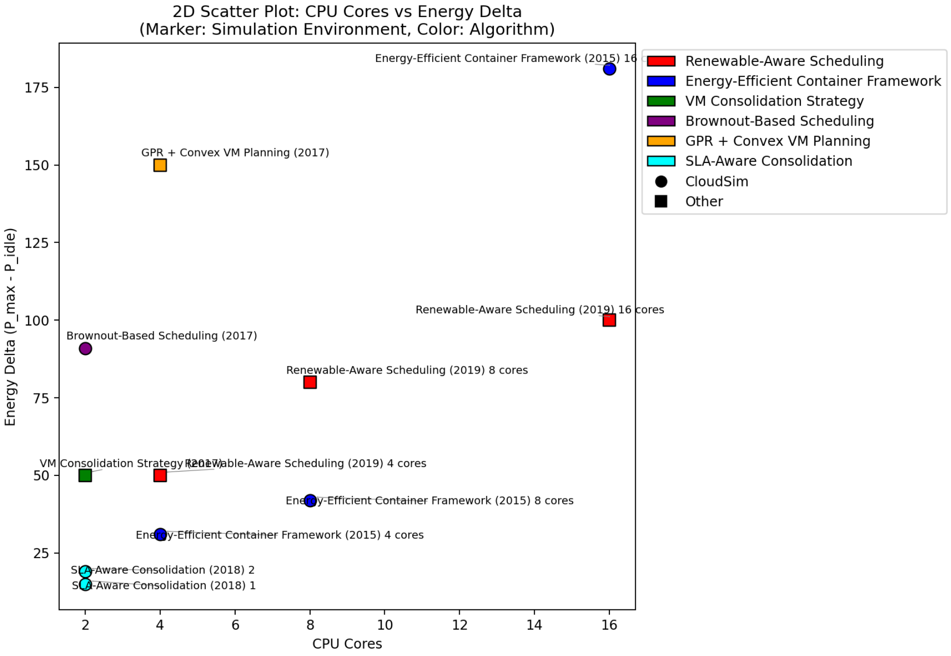

Figure 6: 2D Scatter Plot: CPU cores vs. Energy Delta ($P_{\max} - P_{idle}$).

5 Conclusions

Energy consumption optimization in container-based cloud environments has evolved from early heuristic methods to sophisticated, predictive, and container-aware scheduling approaches. Early consolidation techniques demonstrated significant energy savings through aggressive host shutdown and DVFS, but their reliance on static thresholds often limited adaptability under dynamic workloads[6]. In contrast, predictive algorithms that incorporate workload forecasting and dynamic resource allocation have achieved further energy reductions (typically 30–35%) while balancing the trade-off between energy savings and service quality[7;5].

The transition from VM-based to container-based solutions has enabled finer-grained resource management, with approaches such as flow-network optimization for container placement proving effective even in large-scale simulations[10]. Additionally, strategies like brownout scheduling[18] and meta-heuristic or AI-driven methods[17;16] have shown promise in achieving global optimization over complex infrastructures, albeit with increased computational overhead.

A key insight from the surveyed literature is that no single method outperforms all others under every scenario; instead, each approach excels in specific operational contexts. For instance, consolidation-focused strategies yield high energy savings but require careful management to avoid SLA violations, whereas QoS-aware methods maintain performance at the expense of some energy efficiency[13]. Moreover, the emerging integration of renewable energy awareness into scheduling algorithms[12] points toward a future where energy management is not only efficient but also environmentally sustainable.

Looking ahead, future research should aim to integrate these diverse techniques into hybrid frameworks that combine predictive scheduling, dynamic consolidation, and real-time QoS monitoring. This integration, alongside the continued evolution of container orchestration platforms, is essential for achieving the dual goals of operational efficiency and sustainable cloud computing.

References

[1] Yanal Alahmad, Tariq Daradkeh, and Anjali Agarwal. *Availability-Aware Container Scheduler for Application Services in Cloud.*

[2] Maria Avgerinou, Paolo Bertoldi, and Luca Castellazzi. Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency. 10(10):1470.

[3] Cornel Barna, Hamzeh Khazaei, Marios Fokaefs, and Marin Litoiu. *Delivering Elastic Containerized Cloud Applications to Enable DevOps.*

[4] Anton Beloglazov and Rajkumar Buyya. Energy efficient allocation of virtual machines in cloud data centers. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pages 577–578. IEEE.

[5] Dinh-Mao Bui, YongIk Yoon, Eui-Nam Huh, SungIk Jun, and Sungyoung Lee. Energy efficiency for cloud computing system based on predictive optimization. 102:103–114.

[6] Alessandro Carrega and Matteo Repetto. Energy-aware consolidation scheme for data center cloud applications. In *2017 29th International Teletraffic Congress (ITC 29)*, pages 24–29. IEEE.

[7] Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, and Ammar Rayes. Energy-efficient resource allocation and provisioning framework for cloud data centers. 12(3):377–391.

[8] Abdul Hameed, Alireza Khoshkbarforoushha, Rajiv Ranjan, Prem Prakash Jayaraman, Joanna Kolodziej, Pavan Balaji, Sherali Zeadally, Qutaibah Marwan Malluhi, Nikos Tziritas, Abhinav Vishnu, Samee U. Khan, and Albert Zomaya. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. 98(7):751–774.

[9] Ralph Hintemann and Simon Hinterholzer. Cloud computing drives the growth of the data center industry and its energy consumption. *Data centers*, 2022.

[10] Yang Hu, Huan Zhou, Cees de Laat, and Zhiming Zhao. Concurrent container scheduling on heterogeneous clusters with multi-resource constraints. 102:562–573.

[11] IEA. Data centres & networks. `https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks`. Accessed: 2025-03-25.

[12] Neeraj Kumar, Gagangeet Singh Aujla, Sahil Garg, Kuljeet Kaur, Rajiv Ranjan, and Saurabh Kumar Garg. Renewable energy-based multi-indexed job classification and container management scheme. 15(5):2947–2957.

[13] Lianpeng Li, Jian Dong, Decheng Zuo, and JIaxi Liu. SLA-aware and energy-efficient VM consolidation in cloud data centers using host states naive bayesian prediction model. pages 80–87. IEEE Computer Society.

[14] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986, 2020.

[15] Sareh Fotuhi Piraghaj, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, and Rajkumar Buyya. A framework and algorithm for energy efficient container consolidation in cloud data centers. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 368–375. IEEE.

[16] Tao Shi, Hui Ma, and Gang Chen. *Energy-Aware Container Consolidation Based on PSO in Cloud Data Centers.*

[17] Boxiong Tan, Hui Ma, and Yi Mei. *A Hybrid Genetic Programming Hyper-Heuristic Approach for Online Two-level Resource Allocation in Container-based Clouds.*

[18] Minxian Xu, Amir Vahid Dastjerdi, and Rajkumar Buyya. Energy efficient scheduling of cloud application components with brownout. 1(2):40–53.