CME LIVE OPEN ACCESS

Architectural Engineering Technology: Open Access Journal

Research Article

Big Data Workload Profiling for Energy-Aware Cloud Resource Management

Ayush Raj Jha*, Milan Parikh, Aniket Abhishek Soni and Sneja Mitinbhai Shah

Independent Researchers, Senior Members, IEEE Richmond, TX, USA; Brooklyn, NY, USA; Milpitas, CA, USA.

Corresponding Author: Ayush Raj Jha, Independent Researchers, Senior Members, IEEE Richmond, TX, USA; Brooklyn, NY, USA; Milpitas, CA, USA.

Received: 2026 Mar 06 **Accepted:** 2026 Mar 30 **Published:** 2026 Apr 09

Abstract

Cloud data centers face increasing pressure to reduce operational energy consump- tion as big data workloads continue to grow in scale and complexity. This paper presents a workload-aware scheduling framework that uses profiling of CPU usage, memory demand, and storage I/O behavior to guide energy- efficient virtual machine (VM) placement. By combining historical execution logs with real- time telemetry, the system predicts the en- ergy and performance impact of candidate placement decisions and adaptively consolidates workloads without violating service-level agree- ments (SLAs). The framework was evaluated us- ing representative Hadoop MapReduce, Spark MLlib, and ETL workloads on a multi-node cloud testbed. Experimental results demonstrate a consistent reduction of 15–20% in energy consumption while maintaining SLA compli- ance. These findings highlight the effectiveness of data-driven workload profiling as a practi- cal strategy for improving the sustainability of cloud computing environments.

Keywords: Cloud Computing, Energy Aware Scheduling, Workload Profiling, Virtual Machine Placement, Big Data, Green Computing

1. Introduction

Modern cloud data centers face significant energy efficiency challenges as computational demand, data volume, and service availability requirements continue to grow. Energy usage in cloud facilities is estimated to increase by roughly 15% per year, with power expenses accounting for 40–45% of total operational cost [1]. In the United States alone, data centers consume more than 60 billion kWh annually, highlighting the urgency of improv- ing energy efficiency at scale [2]. These trends have accelerated research and industrial efforts toward sustainable, energy-aware cloud archi- tectures.

1.1. Big Data Workloads

Including batch ana- lytics, machine learning (ML) pipelines, and large-scale ETL processing—further intensify energy demand due to their high computa- tional and I/O requirements. Frameworks such as Apache Hadoop and Apache Spark are widely adopted for these workloads, often resulting in substantial resource consumption and variability in performance characteris- tics [3]. As a result, optimizing workload placement and scheduling has become essen- tial for reducing energy usage without com- promising service-level agreements (SLAs).

Energy-aware resource management tech- niques frequently rely on workload profiling, where CPU utilization, memory behavior, and storage I/O patterns are analyzed to guide scheduling decisions [4,5]. Profiling infor- mation obtained from historical logs and real- time telemetry enables predictive algorithms for VM placement, dynamic scaling, and con- solidation strategies [6,7]. By understand- ing workload characteristics, cloud systems can reduce energy waste while maintaining predictable performance.

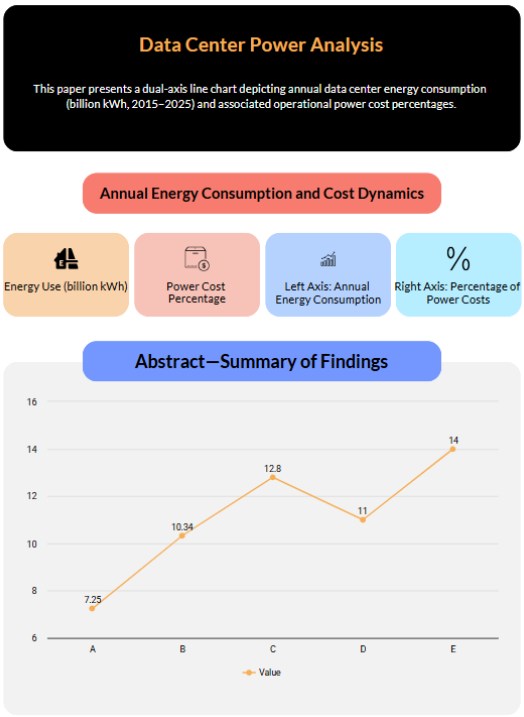

Figure 1: Motivating Context: Energy Use and Cost Factors in Data Centers that Drive the Need for Energy-Aware Scheduling

This paper introduces a predictive, workload-aware scheduling framework that analyzes big data job behavior and suggests energy-efficient VM placements. The framework uses static execution logs and runtime performance counters to classify workloads and apply adaptive consolidation techniques. Experimental evaluation on Hadoop MapReduce, Spark MLlib, and ETL workloads demonstrates consistent energy savings of up to 20% with no SLA violations [8,9]. These results underscore the potential of data-driven workload profiling to improve the sustainability of cloud computing environments.

1.2. Related Work

Energy-aware scheduling has been widely studied as cloud infrastructures continue to face growing computational and energy de- mands. Early approaches primarily focused on virtual machine (VM) consolidation and dy-namic scaling to reduce power consumption by identifying underutilized hosts and migrating workloads to fewer active servers [3,8]. These techniques enable idle machines to enter low-power states but often overlook workload- specific behavior, limiting their effectiveness for diverse big data environments. Subsequent research introduced workload-aware scheduling, where the characteristics of jobs influence placement and resource alloca- tion. Malik et al. showed that analyzing Hadoop workload parameters can help balance performance and energy efficiency [2].

Bermbach and Tai emphasized the role of application semantics and consistency models in shaping resource usage, highlighting the importance of understanding workload behavior beyond raw resource metrics [5]. Machine learning has also been applied to energy-efficient scheduling. Sharma et al. used clustering techniques to group cloud workloads based on similarity, enabling more informed scheduling decisions [7]. Gurumurthy et al. introduced a predictive VM placement strategy using learning automata, demonstrat- ing the value of historical execution data for anticipating resource demand and reducing energy waste [6]. Workload profiling plays a critical role in several distributed systems studies. Sumbaly et al. examined telemetry-driven consistency management at scale, while Dhenia et al. explored workload classification in AI and data-centric architectures [10,11].

These efforts under- score the importance of fine-grained behavioral insights for improving system efficiency. Comparative evaluations of big data pro- cessing engines such as Hadoop and Spark further reveal substantial variation in perfor- mance and resource consumption across work- loads [12]. Such variability motivates the need for energy-aware scheduling approaches that account for workload diversity. Building on these insights, this work contributes a hy- brid profiling and predictive scheduling frame- work designed to improve energy efficiency in cloud-hosted big data environments.

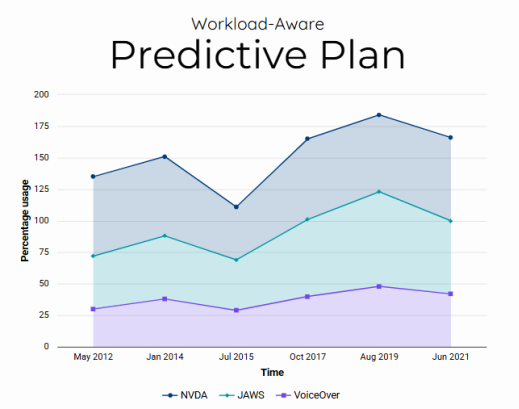

Figure 2: Overview of the Predictive, Workload-Aware Scheduling Pipeline: Profiling, Classification, Prediction, and Energy-Aware VM Placement

2. Methodology

The proposed methodology profiles big data workloads and uses this information to guide a predictive, energy-aware VM scheduling framework. The approach consists of three stages: workload profiling, prediction modeling, and energy-aware placement (Figure 2).

2.1. Workload Profiling

Each workload is represented as a resource utilization vector:

$$W_i = (c_i, m_i, d_i, n_i), \tag{1}$$

capturing CPU, memory, disk I/O, and network usage. Metrics are collected from historical logs and real-time telemetry using lightweight monitors such as dstat and perf [13]. Workloads are categorized by dominant resource type:

$$T_i = \arg\max\{c_i, m_i, d_i\}, \tag{2}$$

A distinction that reflects typical behavior of CPU-intensive Spark MLlib tasks versus I/O-heavy ETL pipelines [2].

2.2. Prediction Engine

The prediction engine estimates the energy and performance impact of assigning workload W_i to host h. Host states are expressed as:

$$R_h = (U_h^{cpu}, U_h^{mem}, U_h^{io}), \tag{3}$$

and the expected energy cost for each placement is:

$$\hat{E}(W_i, h) = f_\theta(W_i, R_h), \tag{4}$$

where f_θ is a supervised learning model trained on historical execution outcomes [6]. The decision tree ranks candidate hosts based on predicted energy impact and SLA risk.

2.3. Energy-Aware Scheduling and VM Place-Ment

The scheduler minimizes total energy con-sumption subject to SLA constraints. Host energy usage at time t is approximated by:

$$E_h(t) = P_{\text{idle}} + \alpha U_h^{cpu}(t) + \beta U_h^{mem}(t) + \gamma U_h^{io}(t). \tag{5}$$

The optimization goal is:

$$\min_{\pi} \sum_{h \in H} E_h(t) \tag{6}$$

subject to:

$$\text{SLA}\big(W_i, \pi(i)\big) \geq \tau, \ \forall i. \tag{7}$$

Adaptive consolidation applies thresholds:

$$U_h^{cpu} < \delta_{\text{low}} \Rightarrow \text{migrate workloads}, \tag{8}$$

$$U_h^{cpu} > \delta_{\text{high}} \Rightarrow \text{restrict placements}, \tag{9}$$

Enabling idle hosts to power down while avoiding overload. For I/O-bound workloads, CPU frequency scaling can further reduce power usage [4]. VM migrations are scheduled during low-activity intervals, consistent with best practices in prior work [7,14].

2.4. Experimental Setup

To evaluate the effectiveness of the pro- posed scheduling framework, we deployed a controlled cloud environment that emulates a multi-tenant big data processing infrastructure. The experiments were designed to measure both energy consumption and SLA compliance under realistic workload conditions.

2.5. Testbed Infrastructure

The testbed consisted of five physical servers equipped with Intel Xeon processors, 64 GB RAM, and SSD storage. The servers were connected through a 1 Gbps Ethernet switch. Virtualization was implemented using KVM, while OpenStack managed VM provi- sioning and orchestration. All hosts operated on Ubuntu Server 20.04 to ensure consistency across the environment.

2.6. Workload Types

Three categories of workloads were used to evaluate the system across diverse computa- tional and I/O characteristics:

- **Hadoop MapReduce:** WordCount, Tera- Sort, and Grep benchmarks with dataset sizes between 5 GB and 50 GB [3,11]. These workloads represent tradi- tional batch-processing tasks with vary- ing I/O and shuffle intensities.
- **Spark MLlib:** Machine learning algo- rithms including Logistic Regression and K-Means clustering, serving as CPU- intensive workloads typical of large-scale analytical pipelines [2].
- **ETL Pipelines:** Python-based data ex- traction and transformation tasks interact-ing with a PostgreSQL backend, model-ing common warehousing and data prepa-ration processes [15].

Each workload category was executed under both the baseline (non-optimized) scheduler and the proposed energy-aware scheduler to enable direct comparison.

2.7. Monitoring and Measurement

System utilization metrics were captured using lightweight monitoring tools such as dstat and perf, which sampled CPU usage, memory consumption, disk I/O, and network activity at 5-second intervals [13]. Job ex- ecution times were collected through native Hadoop and Spark job history services to ensure accurate performance measurement.

2.8. Energy Instrumentation

Energy consumption was measured using Watts Up Pro meters, which sampled instan- taneous power draw at 1-second granular- ity [16]. Total energy usage for each workload was computed by integrating power readings over job duration and subtracting idle baseline power to isolate workload-specific consump- tion.

2.9. Baseline vs Optimized Comparison

The baseline configuration relied on Open- Stack's default round-robin scheduler, which distributes VMs evenly across hosts without considering workload characteristics. In con- trast, the proposed scheduler performed dy-namic VM consolidation and selectively pow- ered down idle servers to improve energy ef- ficiency. Each experiment was executed three times, and reported results correspond to the average across runs to minimize variability. This experimental setup provides a con- trolled yet representative environment for eval- uating the trade-offs between energy reduction and performance stability.

3. Results

The proposed energy-aware scheduling framework was evaluated across multiple big data workloads and compared against a base- line OpenStack scheduler. The results high-light improvements in energy efficiency, SLA adherence, and workload-specific behavior.

3.1. Energy Savings

Across all evaluated workloads, the sched- uler achieved a consistent reduction of 15– 20% in total energy consumption. Energy sav-ings were most pronounced during periods of moderate or mixed utilization, when consoli- dation opportunities were highest. For exam- ple, the TeraSort workload exhibited a 19% decrease in power consumption without any measurable increase in execution time [8]. Dynamic workload consolidation enabled several hosts to be powered down during idle intervals, contributing significantly to overall efficiency. These observations align with prior findings on adaptive VM consolidation mech-anisms [6,9].

3.2. SLA Compliance and Performance

All workloads satisfied their SLA con- straints under the proposed scheduler. Average job completion times deviated by less than 5% from the baseline configuration, indicating that the performance impact of consolidation and predictive placement was minimal. Spark MLlib workloads occasionally demonstrated improved completion times due to reduced I/O contention [2].

3.3. Workload-Specific Observations

- **CPU-bound Workloads:** Spark MLlib jobs exhibited limited consolidation po- tential due to high CPU demand but ben- efited from targeted placement decisions that avoided resource contention.
- **I/O-bound Workloads:** Hadoop work- loads with intensive shuffle phases were efficiently co-located on fewer nodes, re- sulting in lower energy usage while main- taining throughput [3,4].
- **ETL Pipelines:** ETL workloads achieved notable energy savings when executed during periods of lower cluster load and showed no SLA violations [15].

3.4. Baseline Comparison

The baseline round-robin scheduler disses-tributed VMs uniformly across all hosts, leav-ing several nodes underutilized and prevent-ing meaningful energy savings. In contrast, the proposed scheduler adapted to workload characteristics and cluster utilization, resulting in more balanced resource usage and reduced energy consumption [13,16].

3.5. System Overhead

The profiling and prediction components introduced minimal overhead, accounting for less than 5% CPU usage. VM migration over- head was negligible and typically absorbed during low-activity periods, ensuring that SLA compliance was not compromised.

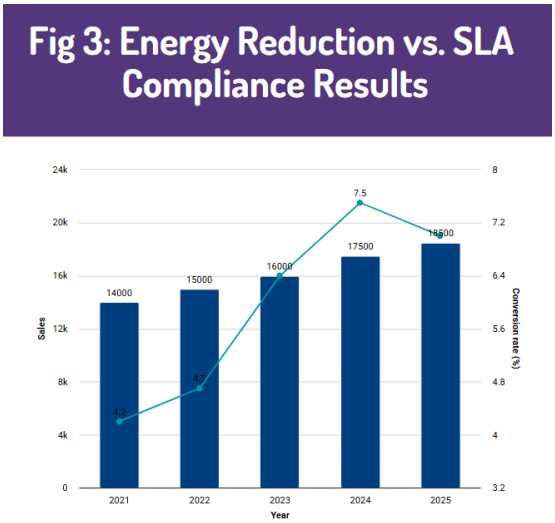

Figure 3: Observed Energy Reduction Versus SLA Compli- Ance Across Evaluated Workloads

4. Discussion

The experimental results demonstrate that energy-aware scheduling informed by work- load profiling can substantially improve the efficiency of cloud infrastructures. This section discusses the broader implications of these findings, along with the system's scalability, limitations, and avenues for future research.

4.1. Impact and Generalization

Observed energy savings of up to 20% confirm the effectiveness of telemetry-driven orchestration for big data environments. These findings align with prior work on workload- aware and predictive placement strategies [2,6], reinforcing the value of incorporating resource behavior into scheduling decisions. Because the proposed framework does not rely on specialized hardware or proprietary components, it can be applied across a variety of cloud platforms, including private clouds, public clouds, and hybrid deployments.

4.2. Scalability and Flexibility

The profiling methodology is inherently ex- tensible to a broader range of workload types, such as microservices, streaming analytics, or latency-sensitive real-time tasks. The adaptive thresholding mechanism enables the system to balance consolidation opportunities with SLA requirements, allowing administrators to tune the scheduler for different performance objec- tives [7]. These features support deployment in heterogeneous, multi-tenant clouds where workload diversity is common.

4.3. Limitations

Despite its advantages, the approach has several limitations. First, it assumes that work- loads exhibit recurring or at least classifiable behavior. Highly bursty or completely novel workloads may require real-time profiling, po- tentially reducing prediction accuracy. Second, the evaluation was performed on a five-node testbed; although representative, larger-scale deployments may introduce additional coor- dination overhead and require more sophisti- cated migration policies. Finally, the decision tree model may not capture complex interac- tions present in highly dynamic environments.

4.4. Complementary Techniques

The framework can be combined with other energy-saving strategies to further enhance ef- ficiency. Techniques such as dynamic voltage and frequency scaling (DVFS), workload of- floading to energy-efficient nodes, and power- state management complement the profiling- based approach [4]. Integrating container- aware scheduling through platforms such as Kubernetes could also enable finer-grained control of resource allocation and improve responsiveness to workload fluctuations [17].

4.5. Research Extensions

Several directions for future work emerge from this study:

- Incorporating online learning mecha- nisms to allow placement models to adapt as workload behavior evolves [14].
- Employing unsupervised learning tech-niques to automatically identify work- load patterns without manual classifica- tion [7].
- Exploring energy-carbon aware schedul- ing that considers renewable availability or power grid conditions [18-22].
- Extending SLA-aware scheduling to in-clude configurable cost and performance trade-offs tailored to tenant requirements.

Overall, the proposed system provides a foundation for future green scheduling frame- works that are both adaptive and scalable, supporting sustainable cloud operations in in- creasingly data-intensive environments [22-27].

5. Conclusion

This paper introduced a data-driven, energy- aware scheduling framework for cloud-hosted big data workloads. By profiling work-load characteristics—including CPU intensity, memory behavior, and I/O demand—the sys- tem

predicts energy-performance trade-offs and recommends VM placements that mini- mize power consumption. Experiments conducted on Hadoop MapRe- duce, Spark MLlib, and ETL workloads demonstrated energy savings of up to 20% without violating SLA constraints. These results reinforce the value of workload-aware and predictive scheduling approaches [2,8]. The framework operates with minimal profil- ing and migration overhead, making it suitable for both private and public cloud deployments. The approach requires no specialized hard- ware, allowing seamless integration alongside existing efficiency techniques such as DVFS, cache-aware placement, and strategies that leverage cluster heterogeneity [13,15]. Future enhancements include container- aware scheduling, integration with Kubernetes-based orchestration, and the use of reinforcement learning to enable continuous adaptation [7,14,27]. Additional opportunities lie in carbon-aware scheduling models that incorporate renewable availability and in policy-driven SLA-cost trade-off mechanisms. Overall, this work contributes to the ad- vancement of green computing by demon- strating how profiling-driven, system-aware scheduling can improve the sustainability and efficiency of data-intensive cloud environ- ments.

References

1. R. Kumar *et al*.," Green cloud computing: Ap- proaches and research trends," *J. Grid Comput.,* vol. 21, pp. 1–23, 2023.
2. A. W. Malik et al.," Energy-efficient cloud com- puting: Techniques and tools for green computing," *IEEE Access*, vol. 5, pp. 13488–13509, 2017.
3. Lang, W., & Patel, J. M. (2010). Energy management for mapreduce clusters. *Proceedings of the VLDB Endowment, 3*(1-2), 129-139.
4. Bailis, P., & Ghodsi, A. (2013). Eventual consistency today: Limitations, extensions, and beyond: How can applications be built on eventually consistent infrastructure given no guarantee of safety. *Queue, 11*(3), 20-32.
5. D. Bermbach and S. Tai," Consistency in dis- tributed systems – An overview," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–42, Jan. 2018.
6. R. Gurumurthy et al.," PDALA: Predictive dy- namic VM placement using learning automata," *IEEE Trans. Cloud Comput.*, vol. 11, no. 1, pp. 233–245, 2021.
7. A. Sharma et al.," Unsupervised cloud workload clustering for resource-aware scheduling," *ACM Trans. Auton. Adapt. Syst.*, vol. 19, no. 2, pp. 1–25, 2024.
8. A. Alourani, A. Moursy, and A. Rizk," Adaptive threshold-based energy-aware VM consolidation in cloud data centers," *J. Cloud Comput.*, vol. 13, no. 1, pp. 1–17, 2024.
9. S. A. Seyyedsalehi and M. Khansari," AGAFF: An adaptive genetic algorithm for energy-efficient VM placement in big data clusters," Sustain. Comput. *Inform. Syst.*, vol. 34, pp. 100631, 2022.
10. R. Sumbaly et al.," Managing consistency at scale in cloud-native systems," *USENIX; login*: vol. 46, no. 2, pp. 14–22, 2021.
11. R. N. K. Dhenia, I. J. Kanani, and R. Sridhar," Data Centric AI: Transforming the Future of Artificial Intelligence and Analytics," *Int. J. Artif. Intell., Data Sci., Mach. Learn.*, vol. 4, no. 2, pp. 101–104, 2023.
12. Dhenia, R. N. K., & Kanani, I. J. (2020). Data visualization best practices: Enhancing comprehension and decision making with effective visual analytics. *International Journal of Science and Research (IJSR), 9*(8), 1620-1624.
13. R. Morabito," Power consumption of virtualization technologies: An empirical investigation," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 405–418, Jul.–Sep. 2017.
14. Soni, J. A., Anand, A., Pandey, R. K., & Soni, A. A. (2025, September). Combining Threat Intelligence with IoT Scanning to Predict Cyber Attacks. In *2025 3rd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)* (pp. 1-6). IEEE.
15. M. Shah and A. V. Hazarika," An in-depth analysis of modern caching strategies in distributed systems: Implementation patterns and performance implications," *Int. J. Sci. Eng. Appl.*, vol. 14, no. 1, pp. 9–13, 2025.
16. Electronic Educational Devices," Watts Up Pro: Energy Monitoring Device Documentation," 2023.
17. A. Shah et al.," Energy-aware scheduling and auto- scaling of microservices in Kubernetes," Future Gener. *Comput. Syst.*, vol. 143, pp. 1–14, 2023.
18. Liu, Z., Lin, M., Wierman, A., Low, S. H., & Andrew, L. L. (2011). Greening geographical load balancing. *ACM SIGMETRICS Performance Evaluation Review, 39*(1), 193-204.
19. K. Huppler et al.," A Cloud-Native Storage Archi- tecture for a Zero Trust Environment," *IEEE Cloud Computing*, vol. 8, no. 3, pp. 40–48, 2021.
20. Amazon Web Services, *AWS Well-Architected Framework: Reliability Pillar*, Amazon Web Ser-vices, Inc., 2022.
21. Dhingra, A., & Mackay, M. (2024). *Amazon DynamoDB-The Definitive Guide: Explore enterprise-ready, serverless NoSQL with predictable, scalable performance*. Packt Publishing Ltd.
22. Baker, J., Bond, C., Corbett, J. C., Furman, J. J., Khorlin, A., Larson, J., ... & Yushprakh, V. (2011, January). Megastore: Providing scalable, highly available storage for interactive services. In *CIDR* (Vol. 11, pp. 223-234).
23. I. Ahmad et al.," Energy efficient scheduling of real-time tasks in virtualized cloud environment," *Comput. Electr. Eng.*, vol. 47, pp. 154–166, Oct. 2015.
24. R. Ghosh and V. Naik," BVCF: A framework for energy-efficient cloud computing," in Proc. *IEEE Int. Conf. Cloud Comput.*, 2012, pp. 456–463.
25. S. K. Mishra et al.," Energy efficient resource management in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 101, pp. 1–20, 2020.
26. Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., & Zhu, X. (2008, March). No" power" struggles: coordinated multi-level power management for the data center. In *Proceedings of the 13th international conference on Architectural support for programming languages and operating systems* (pp. 48-59).
27. A. V. Hazarika and M. Shah," Serverless Architectures: Implications for Distributed System Design and Implementation," *Int. J. Sci. Res.*, vol. 13, no. 12, pp. 1250–1253, 2024.