A Data-driven Analysis of a Cloud Data Center: Statistical Characterization of Workload, Energy and Temperature

Shashikant Ilager[1], Adel N. Toosi[2], Mayank Raj Jha[3], Ivona Brandic[1], Rajkumar Buyya[4]
[1]*Vienna University of Technology, Austria*, [2]*Monash University, Australia*
[3]*Università degli Studi dell'Aquila, Italy*, [4]*University of Melbourne, Australia*

ABSTRACT

To efficiently manage large-scale cloud data centers, it is critical to understand data centers' workload, energy, and thermal characteristics and their impact on the system through data-driven analysis. However, most publicly available traces solely focuses on application workloads ignoring energy and thermal aspects, forcing existing studies to rely on inaccurate and unrealistic analytical or simulation models. In this paper, we present a comprehensive data-driven analysis of a production cloud data center. We monitor and collect the physical machine-level metrics such as resource utilization, energy, and temperature for up to nine months, with a system size of over 26000 CPU cores hosting, on average 1300 virtual machines. We perform a systematic statistical analysis to characterize the monitored data, and study their distributions, variations, trends, and inter-dependencies. We also develop data-driven models to predict resource usage and energy consumption of a physical machine, and demonstrate the usefulness of our dataset through this use case. Our study reveals interesting insights into the energy and thermal phenomena of a data center. The outcome of this analysis helps to increase infrastructure efficiency and long-term strategic planning and improve key business KPIs. The ope-sourced dataset and artefacts enables to investigate new optimization approaches and use cases by researchers.

KEYWORDS

Data centers, Workload characterization, Energy and thermal analysis, Workload, and energy forecasting

1 INTRODUCTION

Data centers (DCs) are the core infrastructure for modern computing systems such as cloud, edge, and High-Performance Computing (HPC) systems. They offer subscription-based services to a wide range of applications from different domains, such as scientific and business workloads, that demand high reliability and Quality of Service (QoS) [2, 7, 24]. However, DCs are complex cyber-physical systems that consume enormous amounts of energy [3, 5, 24]. They vary in their capacity and size from micro DCs at the edge to hyper-scale cloud DCs in remote locations with some hyper-scale DCs requiring up to 100 megawatts of power at peak. The high energy consumption and power density of DCs also generate thermal hot

spots that increase the risk of system failure, performance degradation, and thermal throttling, as well as the cost of cooling energy [15, 24]. Therefore, it is essential to understand the characteristics of DCs in terms of resource usage and their energy and thermal behaviors for efficient resource management.

Advantages of Statistical Characterization : A modern cloud DC contains many subsystems, including computing, storage, network equipment, cooling systems, and other facility-related subsystems, each interacting in a complex manner. Thus, statistical analysis of DC regarding resource usage levels, energy consumption, and thermal behavior is critical to understanding the difficult trade-offs between different subsystems [2, 29]. Such statistical analysis is non-trivial and has multiple benefits for cloud service providers and users [5, 21]. First, with a better knowledge of the system dynamics (e.g., average or peak usage), service providers can create a *optimized capacity planning*, and reduce their capital and operating expenditure [1, 15, 23]. Second, system characterization helps in *designing resource management techniques* such as scheduling, provisioning, and scaling to increase resource utilization and simultaneously reduce the infrastructure's energy consumption [5, 10, 15, 24]. Third, users can decide *optimal resource requirements* for their application workloads and reduce costs [2, 7]. Finally, the statistical characterization of relevant parameters helps *researchers to set up realistic experiments and simulations* by configuring the experimental parameters accurately [14, 21, 28].

Challenges: *The existing analysis and characterization of large-scale systems [7, 13, 17, 18, 22, 25, 29] consider only the traces collected at user, job, or Virtual Machine (VM) level, and do not consider Physical Machine (PM) level metrics.* For instance, the Google Cluster dataset [25] characterized task length, requested and provisioned resources, and usage metrics (CPU, RAM, Disk). Similarly, Microsoft Azure has published many traces characterizing their Azure services, such as VMs [13] and Serverless functions [28]. Furthermore, few studies have explored private Cloud DCs and their business workloads [29], and simulation models for characterizing large-scale cloud DCs [21]. One common aspect of all these datasets and analyses is that they do not consider PM-level metrics such as energy usage, CPU temperature, and inlet temperature and only analyze workloads at the task or VM level. The absence of PM-level public datasets can be attributed to the fact that revealing energy-related metrics of a public cloud would disclose business-critical information, *raising the privacy and confidentiality issues.* However, PM level metrics are critical for understanding the implications of dynamic workloads on DCs' energy and thermal behaviors.

Additionally, the lack of realistic traces representing the DC's physical environment compels researchers to use analytical [9] or static heuristic methods to model the DC power and temperature

phenomena [20]. Such modeling techniques are proven to be inaccurate in representing a dynamic and complex DCs [15, 23].

Contributions: This paper comprehensively analyzes a physical environment of a private cloud DC. We collect PM-level metrics focusing on resource usage, energy, and temperature-related readings. The data is collected from 144 rack-cabinet servers hosting, on average, 1300 VMs. The total infrastructure consists of over 26000 CPU cores and 25000 GB primary memory. The traces are collected over 9 months, with a ten-minute long interval, and the entire dataset contains 3.61 million tuples. Initially, we analyze the dataset with basic statistical measures such as min, max, median, and standard deviation. Then, we study the distributions and variations of monitored parameters to analyze the system state and its dynamicity in run-time. We characterize the dataset based on CPU, memory, network usage, energy, and CPU temperature. The comprehensive analysis and characterization of the dataset produce many interesting insights and observations. For example, while the majority of machines are highly underutilized (with a mean CPU load < 20% CPU), the system's energy consumption is still significant (with mean power consumption > 40%).

In summary, 80% of machines use less than 20% of their CPU while 40% of machines have more than 50 % of peak power usage, *reflecting the need for the efficient utilization of resources and energy spent.* Similarly, PMs in DC operate in above-average CPU temperatures, irrespective of their workload conditions. Furthermore, as a practical use case, we develop workload forecasting and power prediction models for DC servers using the dataset.

The **key contributions** of this work are:

- We **collect the PM-level traces for up to nine months and statistically characterize** the dataset.
- We **analyze the dataset and its different features to understand the distribution, variations, and correlations** and identify key insights.
- We provide a **use case study for predictive analysis** and develop CPU load and power consumption models of PMs.
- We **open source and publish the dataset** [1] for broader use by the research community.

2 BACKGROUND AND METHODOLOGY

This section introduces the cloud infrastructure under investigation, workload, and our data collection method. We also discuss the statistical tools used to characterize the datasets.

This work analyzes the traces of a private DC that serves as a research cloud (RC) and hosts various workloads. The RC is hosted by the University of Melbourne (UniMelb), which provides on-demand computing resources to researchers, similar to commercial cloud service providers. The RC consists of nearly 26,000 virtual cores and 25,000 GB of memory and supports private networking, load balancing, and DNS services. The RC uses OpenStack [27], a platform for unified resource management and virtualization. OpenStack's provisioning, scheduling, and networking mechanisms are employed to manage the computing resources of the RC.

Workloads: The typical workloads and use cases include data analytics, scientific experiments, web hosting, virtual desktop, and

many others. A resource request from a user project includes computing resources, volume storage, object storage, or database services. The computing resources are provisioned as VM instances in different flavors ranging from 1 vCPU core to 80 vCPU cores. The storage resources are provided in virtual volumes or object storage containers (similar to AWS S3). The database service includes engines such as MySQL and PostgreSQL. The users manage their resources through OpenStack's APIs or OpenStack dashboards.

2.1 Data Collection

We collect the traces from two clusters of the RC; the first cluster has nodes with Dell R840 architecture, while the second one has DELL C6320 server nodes. All the machines are rack servers deployed in standard 42U rack cabinets. The R840 is a dual CPU machine (64 cores/CPU) with 64 GB primary memory, while the DELL C6320 is a dual CPU machine (80 cores/CPU) with 196 GB primary memory. The DELL C6320 are newer machines in our DC compared to the R840 machines. These are a subset of machines from our Cloud platform from which we collected the data. This is common in related work, such as characterizing Bitbrains and Google workloads [25, 29], where the anonymization is achieved through a selection of only a part of the infrastructure. However, our dataset is more specific and reports the physical machine (PM)-level traces, including power, CPU temperature, and usual resource utilization metrics. The traces collected from two clusters are called as *dataset1* (*D1*) and *dataset2* (*D2*), from here onwards.

A brief summary of the dataset is presented in Table 1. The D1 traces are collected for a duration of ∼ 9 months, while D2 contains ∼ 5 months. The reason for collecting data for an extended period is to capture all the dynamics and variations of system parameters, which is only possible when resources experience different usage levels. The D1 traces has 80 PMs, with an average of over 750 VMs running on it, while the D2 contains 64 PMs, with an average of 570 VMs running on it. Thus, in total, 144 serves and 1338 average number of VMs together. The data is recorded with a log interval of 10 minutes. The total count of resources includes over 26,000 CPU cores and 34,000 GB of memory. After data filtration and cleaning, the D1 dataset contains 2653869 tuples, and D2 contains 961158 tuples; thus, the dataset in total has 3.61 million tuples. In D1, each tuple contains 16 features, including utilization metrics, power, thermal, and fan speed sensor measurements, while a few features are not available in D2 (fan speeds and inlet). The details of these features are given in Table 2. Two CPU temperature measurements are reported as each host is has two CPUs in D1 and D2.

We run a `collectd2` daemon in the DC servers to collect the data, which is an open-source application that collects system and application performance counters periodically through system interfaces such as IPMI and sensors. The metrics are accessed through network APIs and stored in an SQL database. We used several bash and python scripts to pre-process the data and removed invalid measurements (e.g., NaN).

Note that our traces do not include data about VMs, like arrival time, deletion time, or VM level usage. Instead, we investigate resource consumption at a PM level. It also protects the anonymity of users. Moreover, in our DC, workloads often use the same VMs for long periods, typically over several months. Since we conduct our

	Architecture	Hosts (total= 144)	VM (total =1338)	CPU Cores (total = 26624)	Memory	Period	Interval	Tuples (total=3615027)
dataset1 (D1)	Dell R840	80	768	10240	10240	~9 months	10 minutes	2653869
dataset2 (D2)	DELL C6320	64	570	16384	24576	~5 months	10 minutes	961158

Table 1: Brief overview of the dataset

Table 2: Definition of features collected

Features	Definition
CPU	CPU Load (%)
R	RAM- Random Access Memory (mb)
R_x	RAM in usage (mb)
N_{CPU}	Number of CPU cores
N_{CPUx}	Number of CPU cores in use
N_{Rx}	Network inbound traffic (Kbps)
N_{Tx}	Network outbound traffic (Kbps)
P_c	Power consumed by host (Watts)
T_{CPU1}	CPU 1 temperature (℃)
T_{CPU2}	CPU 2 temperature (℃)
fs_1	fan1 speed (RPM)
fs_2	fan2 speed (RPM)
fs_3	fan3 speed (RPM)
fs_4	fan4 speed (RPM)
T_{in}	Inlet temperature (℃)
N_{vm}	Number of VMs running on host

Table 3: Feature set variations in the dataset

| Metrics	Min		Max		Mean		SD	
	D1	D2	D1	D2	D1	D2	D1	D2
CPU	0.0	0.0	68.36	86.31	10.35	22.29	14.21	18.51
R_x	0.49	1.44	64.42	191.41	35.60	46.77	14.12	35.40
N_{Rx}	0	0	10.65+e8	52.63+e8	28.75+e5	25.59+e6	17.71+e6	20.94+e7
N_{Tx}	0	0	11.24+e8	59.97+e8	22.86+e5	14.55+e6	14.66+e6	19.41+e7
N_{vm}	0	0	54	261	9.6	9.01	5.58	32.91
N_{CPUx}	0	0	128	320.00	55.93	39.81	20.27	40.56
$fs_1 - fs_4$	280	-	13941.66	-	8804	-	2687.12	-
T_{CPU1}	29.14	26.66	82.01	79.23	57.00	59.52	11.06	11.60
T_{CPU2}	25.46	25.58	77.95	81.40	48.29	59.76	8.79	12.85
P	55.86	260.00	448	806.00	205.58	546.44	65.55	122.07
T_{in}	4	-	25.75	-	17.73	-	3.66	-

(a) CPU (b) RAM
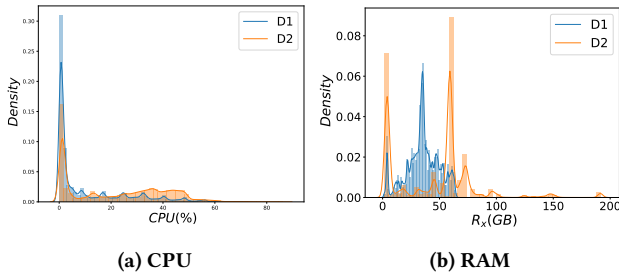
Figure 1: PDF of CPU and Memory

analysis over traces at the PM-level and are interested in studying the physical environment of a DC with PM-level usage, energy, and temperature phenomena, VM-level metrics fall out of scope in our study. In fact, our traces do not have data about them to report.

2.2 Statistical Tools for Characterization

In this work, we comprehensively characterize PMs using data corresponding to utilization in terms of CPU, memory, number of CPU cores, number of VMs, and network resources. In addition, we also characterize energy and CPU temperatures. We use commonly used statistical tools to analyze the traces, such as exploratory analysis, temporal analysis, and correlations [25, 29]. Initially, we provide an overview of the dataset using basic statistical tools and report the min, max, mean, mean, and standard deviation (SD) values. Then, we perform *Exploratory Analysis* to analyze distribution and variations. We use statistical instruments such as Probability Density Function (PDF), Cumulative Distributed Function (CDF), and unitless Coefficient of variation (CoV) (ratio of SD and mean) values observed for all PMs. We perform *Temporal Analysis* to identify trends and time patterns in our time series dataset, where we aggregate over time by summing the average value of observed parameters for all the PMs each hour. In addition, to understand the dynamicity, we report the peak-to-mean ratio (ratio of peak-to-mean values) computed for hourly aggregated intervals.

Furthermore, we perform *Correlation Analysis* to study the dependency between the different parameters. We use Pearson Correlation Coefficient (PCC), which measures the linear relationship between two variables. In a word, these statistical methods help systematically characterize the dataset and understand the intricacies of a complex DC environment. These statistical tools are commonly used to characterize datasets in other works [2, 29].

In the following sections, we analyze important variables relevant to resource utilization, power, and CPU temperature in detail and only provide basic statistics for fan speeds and inlet temperature. The analysis is organized in two parts: *Resource Utilization* and *Energy and Thermal* analysis.

3 RESOURCE UTILIZATION ANALYSIS

We analyze the dataset using the statistical tools mentioned in Section 2.2. The basic statistics (min, max, mean, SD) of the entire dataset is illustrated in Table 3.

3.1 Exploratory Analysis

CPU and Memory Usage: CPU and memory are dominant resource types in the cloud DCs. The VMs are provisioned based on resource units translated as CPU and memory requirements of users. CPU and memory utilization reflects user workloads' overall behaviors and governs the DC's power consumption and thermal characteristics [5, 24]. Therefore, we analyze them together here.

Figure 1 presents a PDF of CPU and RAM utilization of all the PMs for *dataset1* (D1) and *dataset2* (D2). In Figure 1a, the *x*-axis represents a quantized CPU load (%), and the *y*-axis represents the corresponding probability density value. As observed, most of the PMs utilize around ~ 10 % of their CPU capacity in D1 and D2,
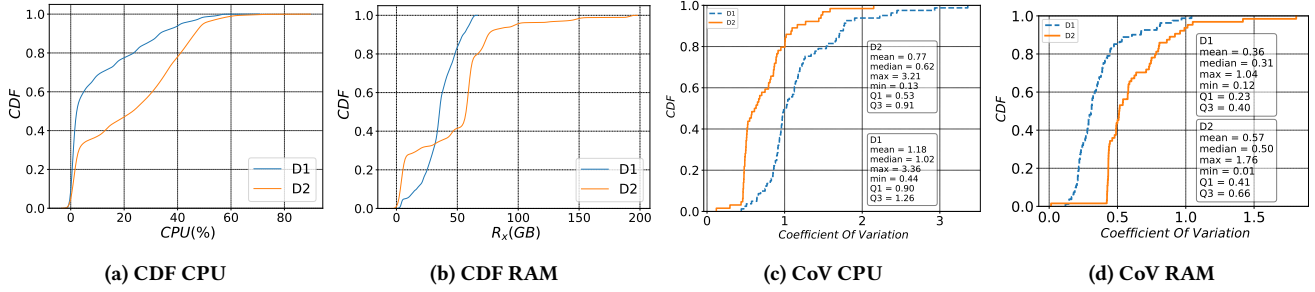
(a) CDF CPU　　(b) CDF RAM　　(c) CoV CPU　　(d) CoV RAM

Figure 2: CDF and CoV of CPU and RAM used

reflecting an extremely lower utilization of provisioned resources by the VMs. However, compared to D1, PMs in D2 have higher mean CPU usage $\sim 20\%$ (Table 3). Such a distribution is normal in private and public cloud DCs [13, 25, 29]. Because cloud platforms always over-provision the resources to meet peak demand, thus, resources are often underutilized. Therefore, increasing the utilization without impacting the workload QoS is challenging. Similarly, as seen in Figure 1b, the majority of PMs utilize more than 50% ($\mu = 35.60$) of available memory and 23.8% ($\mu = 46.67$) in D2. Since servers in D2 have larger available memory (196 GB/machine), this lower % memory utilization in D2 is justified compared to D1.

To understand the distributions and variations of CPU and memory utilization, we further characterize the CPU and memory utilization using CDF and CoV. Both the CDF and CoV behavior can be observed in Figure 2. The 80% of the machines have less than 20% CPU utilization for D1, and 40% of machines have less than 20% utilization in D2 (Figure 2a). Similarly, for memory (Figure 2b), 80% of PMs have less than 50 GB ($R = 64GB$) of memory usage in D1, while 90% of machines have less than 100 GB memory usage in D2 ($R = 196GB$), respectively. The higher CPU and memory utilization in D2 compared to D1 reflects a better bin-packing strategy of VMs in Cluster-2 compared to Cluster-1.

To understand the dynamicity of CPU and memory utilization, we report the CoV values. CoV values are unitless, allowing us to accurately compare the dynamicity across different parameters. A CoV value less than 1 represents lower dynamicity, i.e., utilization always hovers around mean utilization, and a CoV value more than 2 represents highly dynamic utilization behavior. As shown in Figure 2c, only 40% of PMs have CoV of less than 1 in D1 ($\mu = 0.77$), representing highly dynamic CPU usage in Cluster-1. However, 80% of PMs in D2 have CoV of less than 1 for CPU load ($\mu = 1.18$), indicating that most of the PMs in D2 have stable CPU loads. Nevertheless, significant PMs in D1 and D2 have high fluctuations, i.e., CoV values greater than 2, indicating CPU load is highly dynamic and unpredictable for these PMs. Similar distributions and behavior have been identified in large-scale cloud DCs [29]. Compared to CPU load, we find that memory utilization has less dynamicity. Almost all the PMs in D1 have less than 1 CoV value and more than 90% of PMs in D2 have CoV less than 1, as indicated in Figure 2d.

Number of CPU Cores Used and Number of VMs: Here, we perform an exploratory analysis of the number of CPU cores in use (N_{CPUx}) and the number of VMs running (N_{vm}). These two
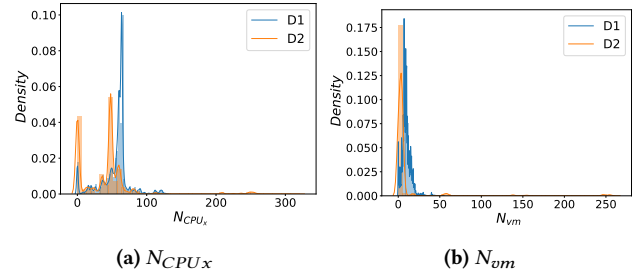

(a) N_{CPUx}　　(b) N_{vm}

Figure 3: PDF of N_{CPU} and N_{vm}

variables provide an overview of the resource utilization level of a DC infrastructure as a whole. DC's total and available capacity are usually measured in terms of free CPU cores [5], and the VM provisioning policies are designed based on the free CPU cores in PMs. Thus, a clear understanding of CPU cores usage would help DC operators with better capacity planning and help design optimized resource provisioning techniques [5, 24].

Figure 3 presents a PDF of N_{CPUx} and N_{vm} of all the PMs for D1 and D2. As observed in Figure 3a, most of the PMs in D1 utilize around 50 cores on average of their total core counts ($N_{CPU} = 128$) in D1, while 80 cores in D2 ($N_{CPU} = 196$), reflecting more than 50% of CPU cores are unused in DC. These insights give an opportunity for optimizations such as consolidation and flexible workload admission techniques to increase overall resource utilization. Similarly, the PDF of N_{vm} can be observed in Figure 3b. Most PMs host around 9-10 VMs on average in D1 and D2. The PMs in D2 host higher VMs than PMs in D1 (maximum= 261, see Table 3), as PMs in D2 are larger and host more VMs. However, it also decreases the overall mean of N_{vm} in D2 (as more PMs are idle).

To further analyze the distribution and dynamicity of N_{CPUx} and N_{vm}, we use CDF and CoV as indicated in Figure 4. The CDF of N_{CPUx} is illustrated in Figure 4a, where 80% of PMs have less than 50 CPU cores used in D1, and 90% of PMs have less than 100 CPU cores usage in D2. Similarly, for N_{vm}, almost all the machines host less than 50 VMs ($\mu = 9.6$, $max = 54$) in D1, while more than 10% of PMs in D2 host more than 50 VMs ($\mu = 9.1$, $max = 261$), respectively (Figure 4b). A similar justification stated in the PDF analysis can be given for this utilization distribution. These results indicate an opportunity to bin-pack more VMs on PMs, and need for techniques to manage the peak utilization scenarios (e.g., migration) [7, 15].
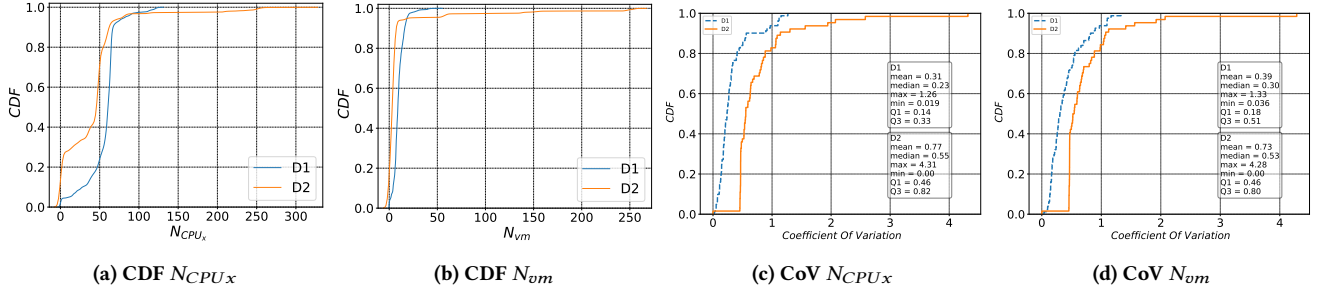
(a) CDF N_{CPUx} (b) CDF N_{vm} (c) CoV N_{CPUx} (d) CoV N_{vm}

Figure 4: CDF and CoV of N_{CPUx} and N_{VM}
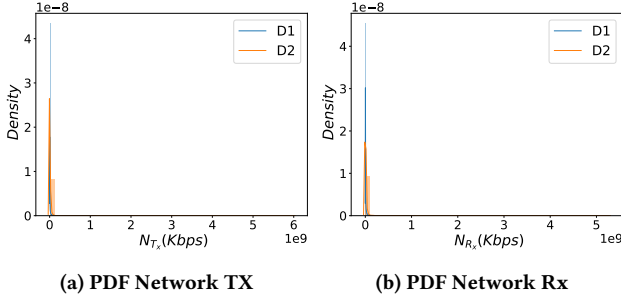

(a) PDF Network TX (b) PDF Network Rx

Figure 5: PDF of N_{Rx} and N_{Tx}

Figures 4c and 4d provide CoV of N_{CPUx} and N_{vm}, respectively. The CoV of both N_{CPUx} and N_{vm} follows identical trends. This is because the N_{CPUx} used directly depends on the N_{vm}, a PM hosts, and their configurations. VMs rarely change their CPU core capacity once provisioned and assigned to PMs. In addition, VMs are used for a long duration, indicating lower dynamicity. As seen in Figure 4c, 90% of PMs have CoV of lower than 1 in D1, while 80% ($\mu = 0.31$) of PMs have lower than 1 in D2 ($\mu = 0.77$), respectively. N_{vm} also follows similar CoV distribution for both D1 and D2 (Figure 4d). Therefore, we can infer that dynamcity of N_{CPUx} and N_{vm} is least significant problem to be addressed by DC operators.

Network usage: Our dataset has two network metrics for each PM, N_{Rx} (inbound traffic) and N_{Tx} (outbound traffic). Studying the network utilization behavior helps to analyze network I/O behaviors of workloads and design the necessary network infrastructure. Most of the VMs in our DC have minimal network communication, with the average mean of N_{Tx} being 2.26 Mbps and 14.56 Mbps in D1 and D2, respectively. Similarly, N_{Rx} has an average mean of 2.875 Mbps 25.55 Mbps for D1 and D2, respectively, as illustrated in Table 3. The PDF plot in Figure 5 shows that both N_{Tx} (Figure 5a) and N_{Rx} (Figure 5b) metrics have concentrated histograms with minimal distribution. It is expected as VMs in our DC host standalone VMs and is often used for compute-heavy jobs with minimal network communications. However, this may not be in the case of public cloud DCs where most VMs host applications accessed over the internet, thus showing higher network usage [13].

A similar trend, i.e., lower distribution is observed for the CDF as seen in Figures 6a and 6b. However, the CoV (Figure 6d, 6c) of N_{Tx} and N_{Rx} has higher value for both D1 and D2. The average CoV of 5.5 and 4.4 for N_{Tx} and 5.03 and 3.93 for N_{Rx} for D1 and D2,

respectively, showcasing a high deviation from the mean network usage, thus higher dynamicity in network usage traffic of PMs.

3.2 Temporal Analysis

In this subsection, we analyze resource usage trends over time. The knowledge of the temporal behavior of workloads and characterizing the time patterns of resource usage help to provision sufficient resources for user workloads. We study peak, mean, and minimum resource usage over time following [29]. We also report the peak-to-mean ratio as a measure of dynamicity over time. We report hourly intervals for the entire duration of the dataset. For all the resources analyzed, we find that workloads are more dynamic than most previously described DC workloads and more in line with cloud workloads' volatile resource usage patterns.

CPU and Memory: Figures 7a and 7b illustrate the usage over time of CPU resources, which provide hourly peak, mean, and min usage of CPU load. Since datasets belong to two clusters and have different periods of monitored data, we plot them separately. The average *peak-to-mean ratio* of CPU usage is 6.3 and 2.33 for D1 and D2, respectively, indicating a significant fluctuation in CPU usage. A similar ratio has been observed in other workload traces such as Google trace (peak-to-mean ratio of 1.3, daily intervals), Azure traces (peak-to-mean ratio of 1.7, 15-minute intervals), and the Microsoft Messenger trace (peak-to-mean ratio of 2.5-6.0, 30-second intervals). The higher peak-to-mean ratio in our DC reflects much more dynamicity than existing datasets.

Like CPU, Figure 8 indicates the memory usage over time for D1 and D2. The average *peak-to-mean ratio* of memory usage is 1.6 and 2.72 for D1 and D2, indicating PMs in D2 have much more fluctuation than in D1. Such variations between D1 and D2 might be due to a higher multi-tenancy in D2, where few machines in D2 host almost five times more VMs on a single host (see Table 3).

We also analyze the temporal behavior of other resources, including the N_{vm}, N_{CPUx}, N_{Tx} and N_{Rx} and find similar higher peak-to-mean ratios for both datasets (D1, D2). Due to page limitations, we only present numerical values here instead of plots. We find that the peak-to-mean ratio for N_{vm} is (2.43, 25.11), and for N_{CPUx} it is (1.55, 11.98). The significantly higher ratio for D2 is because larger PMs allow hosting many VMs into a few PMs. At the same time, other PMs are left unused, affecting the mean value of resource usage, thus reflecting it on the peak-to-mean ratio. The ratio for N_{Tx} is (40.15, 31.55), and N_{Rx} is (47.51, 31.93), indicating higher fluctuations in network usage in PMs.
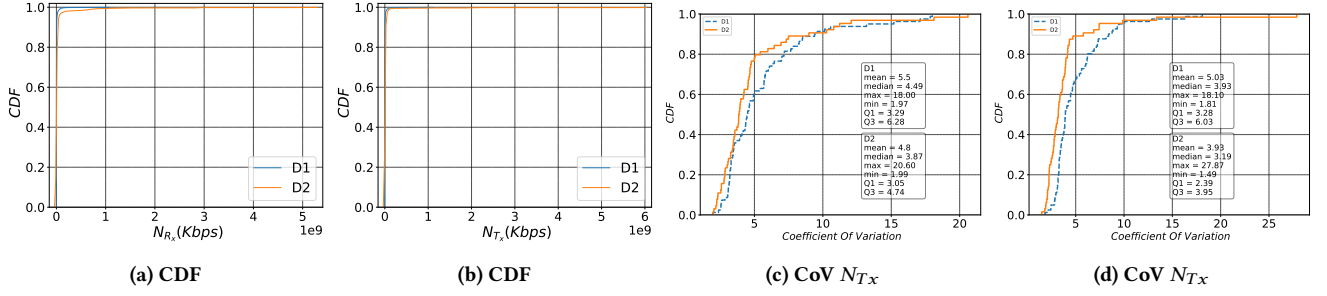
(a) CDF   (b) CDF   (c) CoV N_{Tx} (d) CoV N_{Tx}

Figure 6: CDF and CoV of N_{Rx} and N_{Tx}
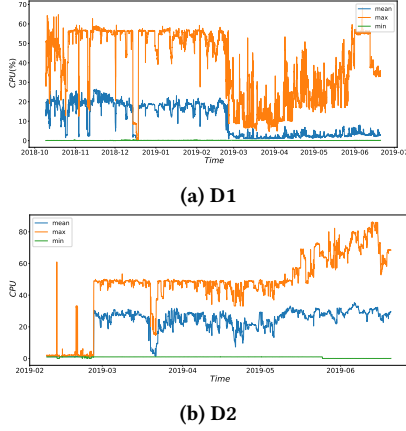

(a) D1

(b) D2

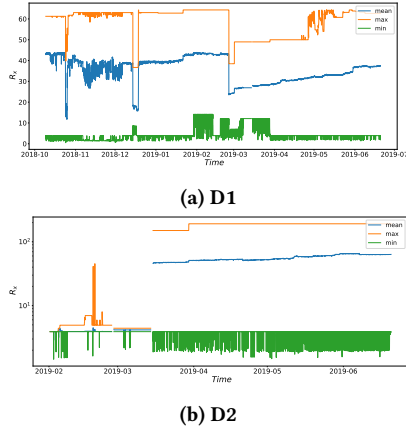Figure 7: CPU usage over time

(a) D1

(b) D2

Figure 8: RAM usage over time

Key insight: **The servers have lower average CPU and memory utilization, but they show high temporal variability and dynamicity, which indicates the need for efficient resource utilization policies that can handle bursty usage scenarios.**

4 ENERGY AND THERMAL ANALYSIS

It is important to note that, PMs in *dataset1* (D1) and *dataset2* (D2) have dual CPUs, so we have two separate CPU temperature

readings. The CPU2 temperature follows similar distribution as CPU1, thus, for the sake of brevity, we present the CPU temperature phenomenon of only CPU1.

4.1 Exploratory Analysis

Figure 10 presents the PDF of power (P) and CPU temperature. The PMs in D1 have a peak power consumption of up to 450 watts, and PMs in D2 have a peak power consumption of up to 800 watts. Most PMs in D1 consume power in the range of 200 watts, and many PMs experience peak power usage, as seen in Figure 10a. A similar trend is observed for D2. It is important to note that the average CPU utilization for the PMs in both D1 and D2 is between 10-20%, but the power consumption is in much higher percentile (μ = 45.7% in D1 and μ = 67.4% in D2). It is due to the idle node power consumption rate of PMs. Therefore, it becomes essential for DC operators to increase resource utilization of resources so that energy is spent on the computation cycle more effectively.

The high number of hosts operating at a peak CPU temperature can be observed from Figure 10b. The CPU temperature reaches up to 80 °C for D1 and D2. The high °C of CPU temperatures increases the chances of CPU throttling induced by thermal threshold. The peak temperatures in the DC environment can be due to multiple reasons, including workload level, cooling system settings, and physical phenomenon such as heat circulation effect in DC. [15, 20].

We further analyze the distribution and dynamicity of power and temperature using CDF and CoV similar to resource utilization (Figure 9). The CDF of Power P is illustrated in (Figure 9a). Here, more than 40% of PMs have more than 200 watts of power consumption (peak power up to 450 watts) in D1, and 40% of PMs have more than 600 watts of power consumption in D2 (peak power up to 800 watts). Similarly, for T_{CPU1}, more than 20 % of hosts exceed beyond 70 °C of CPU temperature (peak temperature of 80 °C). The high number of PMs experiencing above-average power consumption and CPU temperature is inconsistent with the observed CPU utilization, indicating a non-linear relationship between them.

4.2 Temporal Analysis

The Figures 9c and 9d provide CoV of P and T_{CPU1}. Figures 9c and 9d show lower fluctuations, i.e., deviating less from their mean values. The mean CoV of P is 0.24 and 0.16 for PMs in D1 and D2, while the mean CoV of T_{CPU1} is 0.13 and 0.14 for PMs in D1 and D2, respectively. The lower CoV of P and T_{CPU1} is because most of the PMs always operate around mean values (idle power contributing
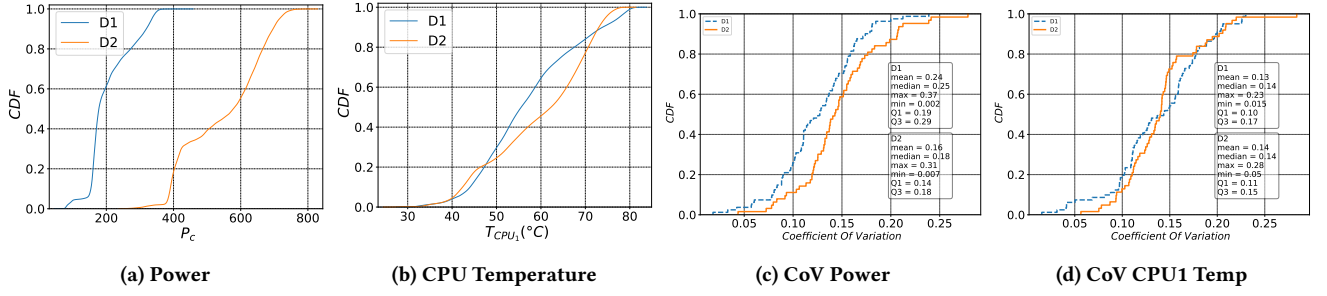
(a) Power   (b) CPU Temperature   (c) CoV Power   (d) CoV CPU1 Temp
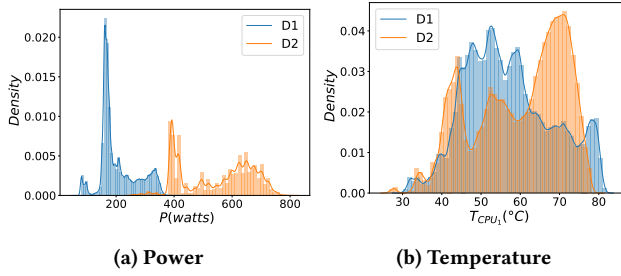
Figure 9: CDF and CoV of Power and CPU Temperature

(a) Power   (b) Temperature

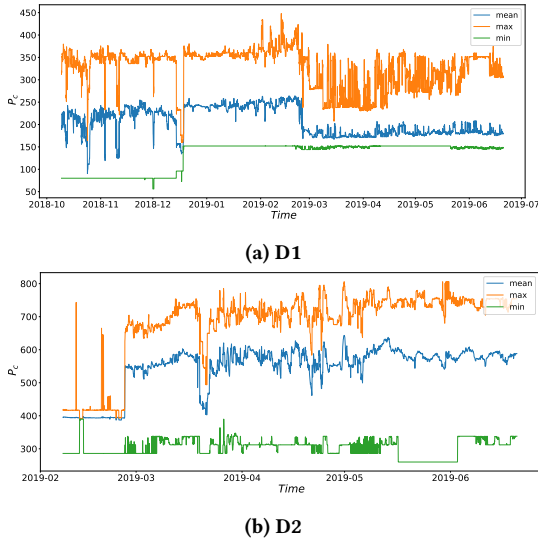**Figure 10: PDF of Energy and CPU Temperature**

(a) D1

(b) D2

Figure 11: Power consumption over time

to this large, and most machines are idle). This non-linearity of variations between resource utilization (e,g, CPU) and power and temperature behaviors makes it difficult to model power consumption or thermal response with simple analytical models [30].

Figures 11a and 11b illustrate the power consumption over time for *dataset1* (D1) and *dataset2* (D2). They provide hourly peak, mean, and min usage of P. The average *peak-to-mean ratio* of P usage is 1.59 and 1.23 for D1 and D2, indicating a sight fluctuation over time.
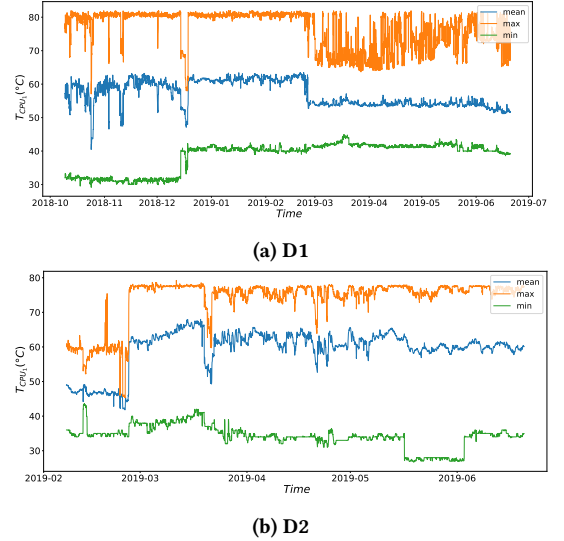

(a) D1

(b) D2

Figure 12: CPU Temperature over time

Figure 12 provides the min, max, and mean temperature of all PMs in D1 and D2 for their monitored period. For T_{CPU1}, the average *peak-to-mean ratio* is 1.34 (1.44 for T_{CPU2}) and 1.24 (1.25 for T_{CPU2}) for D1 and D2, indicating the close similarity of variations between power consumption and CPU temperatures of PMs.

Key insight: **The servers have high average power consumption and temperature, but low temporal variations, which means that most of the servers operate at their mean values (due to idle power consumption). Power and temperature are also not linearly related to the CPU utilization, which makes it hard to model with simple analytical models.**

5 CORRELATION ANALYSIS

Figure 13 shows the correlation between all the parameters. The left side in Figure 13 is for dataset (D1), while the right is for dataset2 (D2). The D1 has additional information about fan speeds ($fs1 - 4$) and inlet temperature (T_{in}) compared to D2. We do not include constant PM parameters that represent PM's configuration (N_{CPU} and R), which do not have correlations with other parameters in runtime. The correlation plots are based on the standard pairwise Pearson Correlation Coefficient (PCC) represented as a heat map. For illustration, the values are represented as color shades. Here,
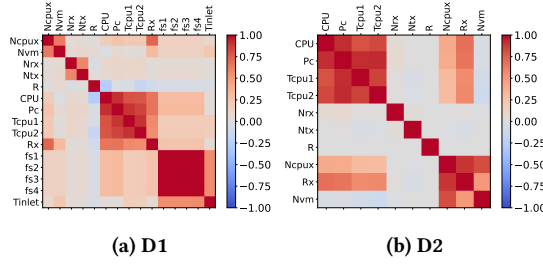
Figure 13: Correlation matrix of D1 and D2

the correlation value ranges from -1 to 1, close to 1 for highly correlated features, 0 for no correlation, and -1 for negative correlation. In addition, the correlation matrix is clustered based on the pairwise Euclidean distance to enhance interpretability. We observe that few parameters are highly correlated while others correlate negatively. For instance, power consumption represents high interdependencies with CPU load and temperatures. It is also important to note that inlet temperature (in D1) positively correlates with fan speeds and the number of VMs, indicating influence of workload level on cooling requirements. Moreover, factors like memory usage and machine fan speeds also have some degree of interdependence. In both D1 and D2, network parameters have the least correlation with other features in general, especially indicating its little influence on power consumption and CPU temperature metrics.

Key insight: **Power and CPU temperature are positively correlated with CPU, memory, and fan speeds, while network parameters have the lowest correlation. In D1, inlet temperature is also related to fan speeds and number of VMs, which reflects the impact of workload level on cooling settings.**

6 A CASE STUDY: WORKLOAD FORECASTING AND POWER CONSUMPTION PREDICTION

Evaluating traces with a potential use case enables researchers to identify the practical sensitivities of dataset and prove its generality [2]. We demonstrate this through a case study by developing workload forecasting and energy prediction models. PMs whose run-time CPU usage is not known a priori are commonly allocated with a significantly lower number of VMs, which leads to poor resource utilization and energy inefficiency. Consequently, predicting CPU load helps operators to plan and utilize resources efficiently. However, accurate workload estimation and power consumption is a non-trivial problem [5]. The simple linear-estimation models fail to capture complex cloud DCs' dynamics accurately. Consequently, the advancement of Machine Learning (ML) algorithms and data availability could address this problem effectively [12]. Therefore, we study how well our dataset can be used to develop new predictive models for estimating CPU usage and power consumption.

Since we predict *numerical target values,* i.e., CPU load (%) and power consumption (watts), we explore several commonly used multi-variate regression algorithms. Our selected candidate algorithms include Linear Regression (LR), Polynomial Regression (PR), Lasso Regression (LR), Ridge Regression (RR), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) with LSTM, XGBoost (XGB), LightGBM (LGBM), Random Forest (RF), and Decision Tree (DT). We used entire D1 and D2 datasets and followed
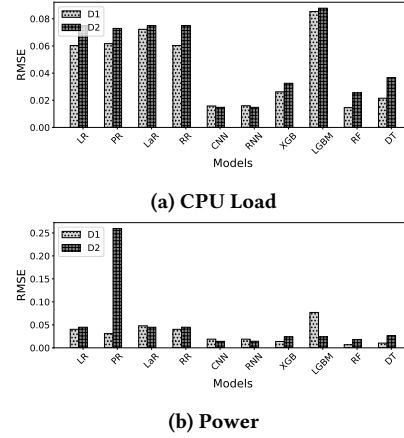

(a) CPU Load

(b) Power

Figure 14: Predictive modelling

standard ML practices to train the model, including dividing train and test datasets (70% and 30%). The models are trained to predict the next interval (10-minute intervals similar to dataset collection) CPU load and power consumption of any random server. The input features include variables mentioned in Table 2 except the target variables (we also exclude N_{CPU} and R, which are constant). We trained separate models for D1 and D2 as models must be built using data from the same architectures and environment since they significantly affect the performance [5]. The models are trained using ML algorithms from scikit-learn framework. We *set default model-specific hyper-parameters and did not perform any further optimization* to analyze the results among multiple models reasonably. The experiment results have shown that the models trained using our dataset can accurately estimate the CPU load and power consumption of servers, as shown in Figure 14. For both the CPU and power, D2 has a slightly higher error rate than D1. This could be attributed to the fact that D1 has more tuples and higher distribution, as it is collected for longer period. The CNN model achieves the best accuracy for CPU load prediction with an RMSE value of 0.015 and 0.014 for D1 and D2, respectively. Similarly, the RF model achieves the best results for energy prediction with an average RMSE of 0.0016 and 0.018 for D1 and D2, respectively.

Key insight: **The results show that a comprehensive dataset can enable accurate workload and power prediction models, assisting data-driven resource optimizations. Additionally, our dataset can support other predictive models, such as CPU and inlet temperature.**

7 THREATS AND APPLICABILITY

In this section, we discuss the potential threats and applicability of our research and dataset.

7.1 Threats to Validity

- *Is the DC in this study representative of a large-scale public cloud DC and its workload?* The DC infrastructure used in this study may not be representative of a hyper-scale DC and its workloads. However, it provides insights into the understanding of rack-arranged servers and their complex phenomena in regard to energy and temperature under changing workload conditions using the specific test data (workloads of a research cloud).

Author	Organization	Infrastructure type	Trace types	Granularity of Traces	Characteristics	Utilization Metrics	Energy & Temperature
Reiss et al. [25]	Google	Public cloud	Scheduler logs	Jobs	Reporting the google cluster's compute job traces	✓	✗
Cortez et al. [7] & Hadary et al. [13]	Microsoft	Public cloud	VM traces	VM	Characterization of VM workloads & development of VM allocation and scheduling mechanisms	✓	✗
Shahrad et al. [28]	Microsoft	Public cloud	Serverless application traces	Functions	Characterization of FaaS workloads and corresponding resource optimization methods	✓	✗
Jiang et al. [18]	Alibaba	Public cloud	Batch and online jobs	Jobs	Analyzed effect of collocation of batch and online jobs on performance	✓	✗
Shen et al. [29]	BitBrain	Private cloud	VM level	VMs	Statistical characterization of VM level requested and used resources for business workloads	✓	✗
Jia et al. [17]	-	Private cluster	Data analysis jobs	Jobs	Bench marking and analyzing data analysis job on Hadoop cluster	✓	✗
Amvrosiadis et al. [2]	LANL & TwoSigma	HPC cluster	HPC cluster	Jobs	Study of HPC and on-premise financial data center traces of jobs	✓	✗
Our work	**UniMelb**	**Private cloud**	**PM traces**	**PM**	**Statistical characterization and analysis of PM level metrics**	✓	✓

Table 4: Related works and comparison with our study

- *Does this study capture the heterogeneity of modern cloud DC?* Modern DCs host vast heterogeneous servers with different CPU architectures, and GPGPUs, whereas our dataset consists of two types of servers. However, VM workloads on CPU architecture are still widely used in cloud DCs, and our dataset comprehensively represents such infrastructure.
- *Does the lack of VM or application level features affect the accurate characterization?* Our traces only contain PM-level metrics, and we do not have VM-level application telemetry data. It is worth noting that access to application-level data is often restricted in hyper-scale DCs due to privacy concerns. PM-level metrics can efficiently characterize and optimize the infrastructure [5, 24]. Researchers can use public application traces and simulate PM-level utilization [4, 6, 15], while our dataset can model PM's energy and temperature.

7.2 Applicability

Some of the important implications of our study and dataset are:

- *Predictive resource management in clouds:* By studying resource utilization patterns and the corresponding power and temperature behavior in a DC, operators can plan effective utilization of PMs and configure the knobs to reduce energy costs. Our dataset can be used as the basis for such analysis. It aids development of predictive ML models to forecast the resource and energy requirements and develop new resource management techniques, addressing the sustainability of cloud DC effectively.
- *Synthetic data generation:* Our dataset can be used to generate extensive synthetic datasets. Data-driven methods for synthetic data generation, such as Generative Adversarial Networks (GANs) [31], can use our dataset as representational training data for their learning processes.
- *Simulation experiments:* The key findings in this study can also help researchers to accurately configure their simulation parameters in their experiments. For instance, researchers can model their simulation scenarios using our distribution and variations values, accurately representing a real-world DC infrastructure.

8 RELATED WORK

The researchers and industry practitioners have explored methods to systematically analyze and characterize large-scale computing systems and develop better resource management techniques. In 2011, Google released a crucial set of scheduler logs [25], i.e.,

Google cluster-usage traces, which has been studied by hundreds of researchers to develop new resource management techniques based on it. The dataset provided information about how resources and computing jobs are handled within hyper-scale data centers with the massive scale of Google's workload demands. The traces are organized into three-layer, i.e., machines, jobs, and tasks. It has information about machine events (e.g., add, remove, update), job events (e.g., ids, submission and completion time, schedule time), and task-level resource utilizations (CPU, memory, and disk space). However, it has been identified that many works have been overfitting to Google's dataset characteristics [2].

Similarly, researchers at Microsoft Azure published multiple traces of their DC workloads.The *Azure VM traces* published in 2017 [7] and in 2019 provides logs of over 2 million VMs belonging to over 6000 Azure user subscriptions. It has logs for a 5-minute interval containing CPU utilization, VM creation, and deletion time. In addition, Microsoft also published a dataset for their FaaS (Function-as-a-Service) cloud service, representing serverless workloads. It contains hashed ids of function owners and applications and information such as the number of function executions, the number of executions per day, and corresponding trigger types [26, 28]. Furthermore, there are traces from Alibaba cloud likewise [18], analyzing characteristics of their DC's co-allocated online and batch jobs. They identified crucial observations regarding job failure patterns of batch jobs and resource consumption patterns of online services, which helps to develop strategies for efficient co-location of online and batch jobs. Nevertheless, all the available traces from public clouds only focus on application or VM-level metrics, ignoring the PM-level traces.

Similar to public clouds, there are also several datasets and analyses from many private cloud DCs. Jia et al. [17] studied data analysis workloads such as page views and daily visitors for web pages. The goal was to understand the impacts and implications of data analysis workloads on modern DC servers and recommend optimizations and architectural changes needed. Closest to our study is Shen et al. [29], who performed a comprehensive study of business-critical workloads hosted in private cloud DCs of Bitbrain. They collected long-term workload traces from two systems and analyzed requested and used resources in the CPU, memory, network, and disk of 1750 VMs for 1 and 3 months duration. Based on their findings, the authors also discussed the possible resource management technique to increase the cloud DC efficiency. These studies as well

focus on specific application types or VM-level metrics, ignoring the PM-level sensor metrics.

Table 4 compares most relevant available traces. As observed, all the traces provide workload or VM-level metrics and do not explicitly represent PM-level utilization metrics and servers' corresponding energy and temperature sensor readings. In addition, our traces are collected for a more extended period. It is important to note that numerous studies and traces are available representing the Grids and standard HPC clusters [2, 11, 19], and a wide variety of cloud applications such as failure analysis traces [8] and modeling spot instance prices [16]. All of these traces also represent only application-level traces without PM-level data. We do not review them in detail in this study, as we focus on virtualized and shared multi-tenant cloud infrastructures.

9 CONCLUSIONS

This work presents a comprehensive analysis of PM-level metrics of a cloud DC, covering resource utilization, energy consumption, and thermal behavior. We used statistical tools to explore and examine the data center resources over time, revealing their distribution, variation, and usage patterns. Our analysis not only confirms some common assumptions, such as low utilization of computing resources in the cloud, but also challenges others, such as the relationship between utilization, energy, and heat. Our study and open source dataset provide a valuable resource for researchers and DC operators to conduct further studies and design innovative resource management techniques for cloud data centers.

ACKNOWLEDGMENTS

REFERENCES

[1] Anup Agarwal, Shadi Noghabi, Íñigo Goiri, Srinivasan Seshan, and Anirudh Badam. 2023. Unlocking unallocated cloud capacity for long, uninterruptible workloads. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 457–478.

[2] George Amvrosiadis, Jun Woo Park, Gregory R Ganger, Garth A Gibson, Elisabeth Baseman, and Nathan DeBardeleben. 2018. On the diversity of cluster workloads and its impact on research results. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. 533–546.

[3] Anders S G Andrae and Tomas Edler. 2015. On global electricity usage of communication technology: trends to 2030. *Challenges* 6, 1 (2015), 117–157. Publisher: Multidisciplinary Digital Publishing Institute.

[4] Özalp Babaoglu and Alina Sîrbu. 2018. Cognified Distributed Computing. *International Conference on Distributed Computing Systems (ICDCS)* (2018), 1741–1743.

[5] Ricardo Bianchini, Marcus Fontoura, Eli Cortez, Anand Bonde, Alexandre Muzio, Ana-Maria Constantin, Thomas Moscibroda, Gabriel Magalhaes, Girish Bablani, and Mark Russinovich. 2020. Toward ML-centric cloud platforms. *Commun. ACM* 63, 2 (2020), 50–59. Publisher: ACM New York, NY, USA.

[6] Walid Budgaga, Matthew Malensek, Sangmi Pallickara, Neil Harvey, F. Jay Breidt, and Shrideep Pallickara. 2016. Predictive analytics using statistical, learning, and ensemble methods to support real-time exploration of discrete event simulations. *Future Generation Computer Systems* 56 (2016).

[7] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. 2017. Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, Shanghai China, 153–167.

[8] Domenico Cotroneo, Luigi De Simone, Pietro Liguori, and Roberto Natella. 2020. Fault injection analytics: A novel approach to discover failure modes in cloud-computing systems. *IEEE Transactions on Dependable and Secure Computing* (2020).

[9] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. 2016. Data Center Energy Consumption Modeling: A Survey. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 732–794. ISBN: 1553-877X VO - 18.

[10] Jeff Dean. 2017. Machine learning for systems and systems for machine learning. In *Presentation at 2017 Conference on Neural Information Processing Systems*.

[11] Dror G. Feitelson, Dan Tsafrir, and David Krakov. 2014. Experience with using the Parallel Workloads Archive. *J. Parallel and Distrib. Comput.* 74, 10 (Oct. 2014).

[12] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM* 64, 12 (Dec. 2021), 86–92.

[13] Ori Hadary, Luke Marshall, Ishai Menache, Abhisek Pan, Esaias E Greeff, David Dion, Star Dorminey, Shailesh Joshi, Yang Chen, Mark Russinovich, and others. 2020. Protean: VM allocation service at scale. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. 845–861.

[14] Shashikant Ilager, Rajeev Muralidhar, Kotagiri Ramamohanarao, and Rajkumar Buyya. 2020. A Data-Driven Frequency Scaling Approach for Deadline-aware Energy Efficient Scheduling on Graphics Processing Units (GPUs). In *IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGRID 2020*.

[15] Shashikant Ilager, Kotagiri Ramamohanarao, and Rajkumar Buyya. 2021. Thermal Prediction for Efficient Energy Management of Clouds Using Machine Learnings. *IEEE Transactions on Parallel and Distributed Systems* 32, 5 (May 2021), 1044–1056.

[16] Bahman Javadi, Ruppa K Thulasiramy, and Rajkumar Buyya. 2011. Statistical modeling of spot instance prices in public cloud environments. In *2011 fourth IEEE international conference on utility and cloud computing*. IEEE.

[17] Zhen Jia, Lei Wang, Jianfeng Zhan, Lixin Zhang, and Chunjie Luo. 2013. Characterizing data analysis workloads in data centers. In *IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, Portland, OR, USA, 66–76.

[18] Congfeng Jiang, Guangjie Han, Jiangbin Lin, Gangyong Jia, Weisong Shi, and Jian Wan. 2019. Characteristics of Co-Allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study From Alibaba Cloud. *IEEE Access* 7 (2019).

[19] Jie Li, George Michelogiannakis, Brandon Cook, Dulanya Cooray, and Yong Chen. 2023. Analyzing Resource Utilization in an HPC System: A Case Study of NERSC's Perlmutter. In *International Conference on High Performance Computing*. Springer, 297–316.

[20] Xiang Li, Peter Garraghan, Xiaohong JIANG, Zhaohui Wu, and Jie Xu. 2017. Holistic Virtual Machine Scheduling in Cloud Datacenters towards Minimizing Total Energy. *IEEE Transactions on Parallel and Distributed Systems* (2017).

[21] Ismael Solis Moreno, Peter Garraghan, Paul Townend, and Jie Xu. 2014. Analysis, Modeling and Simulation of Workload Patterns in a Large-Scale Utility Cloud. *IEEE Transactions on Cloud Computing* 2 (April 2014).

[22] Tirthak Patel, Zhengchun Liu, Raj Kettimuthu, Paul Rich, William Allcock, and Devesh Tiwari. 2020. Job Characteristics on Large-Scale Systems: Long-Term Analysis, Quantification, and Implications. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*.

[23] Ana Radovanovic, Bokan Chen, Saurav Talukdar, Binz Roy, Alexandre Duarte, and Mahya Shahbazi. 2021. Power modeling for effective datacenter planning and compute management. *IEEE Transactions on Smart Grid* 13, 2 (2021).

[24] Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, et al. 2022. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems* 38, 2 (2022), 1270–1280.

[25] Charles Reiss, John Wilkes, and Joseph L Hellerstein. 2011. *Google cluster-usage traces*. Technical Report. Google Inc., Mountain View, CA, USA.

[26] Francisco Romero, Gohar Irfan Chaudhry, Íñigo Goiri, Pragna Gopa, Paul Batum, Neeraja J. Yadwadkar, Rodrigo Fonseca, Christos Kozyrakis, and Ricardo Bianchini. 2021. Faa$T: A Transparent Auto-Scaling Cache for Serverless Applications. In *Proceedings of the ACM Symposium on Cloud Computing*. ACM, USA.

[27] Omar Sefraoui, Mohammed Aissaoui, Mohsine Eleuldj, et al. 2012. OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications* 55, 3 (2012), 38–42.

[28] Mohammad Shahrad, Rodrigo Fonseca, Íñigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, and Ricardo Bianchini. 2020. Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. 205–218.

[29] Siqi Shen, Vincent Van Beek, and Alexandru Iosup. 2015. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*.

[30] Hongyang Sun, Patricia Stolf, and Jean-marc Pierson. 2017. Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters. *Future Generation Computer Systems* 71 (2017), 157–170. Publisher: Elsevier B.V..

[31] Chi Zhang, Sanmukh R. Kuppannagari, Rajgopal Kannan, and Viktor K. Prasanna. 2018. Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids. *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids* (2018).