2019 IEEE 5th International Conference on Big Data Intelligence and Computing (DATACOM)

A Data-Driven Approach to Monitoring Colocation Data Centers

Brian Setz*, Subrahmanya VRK Rao†, Alexander Lazovik‡, Marco Aiello*
**Department of Service Computing, University of Stuttgart, Stuttgart, Germany*
{brian.setz, marco.aiello}@iaas.uni-stuttgart.de
†*Cognizant Technology Solutions, Chennai, India*
subrahmanyavrk.rao@cognizant.com
‡*Distributed Systems Group, University of Groningen, Groningen, the Netherlands*
a.lazovik@rug.nl

***Abstract*—A colocation data center rents out rack space and equipment to third parties, while providing core services such as power, cooling, and bandwidth. Energy consumption, power distribution, and cooling account for the majority of the operational costs. Reducing these operational costs is important in order to increase the profit margins of the data center. When implementing cost-saving measures, the Service Level Agreements and the high Quality of Service have to be maintained. Therefore, monitoring the effect of these measures is key. In a colocation data center however, monitoring the effect on the IT equipment owned by third parties is challenging, due to limited access to the equipment. This work addresses the question: can we monitor third party IT equipment in a colocation data center, without requiring access to the operating system? To answer this, we collect 2.5 billion data points from over 160 servers in a data center, monitoring the server state and the environmental parameters. We utilize this dataset to discover and train multiple models that allow colocation data centers to monitor third party servers without requiring access to the server's operating system. These models enable Data Center Operators to monitor the effect of cost-saving measures on the thermal state of the servers. As well as to monitor the computational load, in order to assist in the expansion planning process.**

***Index Terms*—data center monitoring, data-driven modeling, colocation data centers**

I. Introduction

Monitoring the IT infrastructure of a data center is essential to guarantee the high Quality of Service specified in Service-Level Agreements, and to determine the effect of energy efficiency increasing measures on the data center. There is a class of data centers where extensive monitoring is not trivial: colocation data centers. A colocation data center is a type of data center where rack space, network infrastructure, and IT equipment are rented out to third parties. The colocation data center provides only the power, cooling, bandwidth, and security services. While traditional data centers own and maintain all of the IT equipment, and therefore have full control over the data center, colocation data centers only concern themselves with the core infrastructure of the data center, while the IT equipment is owned by their customers. Therefore, a Data Center Operator (DCO) in a traditional data center has insights into the complete state of the data center, including the state of IT equipment. This is unlike a DCO working in a colocation data center, who does not have detailed information regarding the state of the IT equipment.

Both traditional and colocation data centers concern themselves with reducing the operational expenditures, while maintaining high Quality of Service, in order to increase profit margins. A significant portion of the operational costs incurred by data centers are related to energy consumption, power distribution, and cooling. These three categories account for 72% of a data centers monthly costs, that is excluding the amortization of servers [1]. Thus increasing the energy efficiency of a colocation data center can greatly reduce the monthly expenses. Common measures to improve the energy efficiency in a data center include: optimizing the air flow handling, increasing the temperature set point of the cold aisles, using free air cooling and air economizers, selecting higher efficiency power system equipment, as well as using machine learning for optimization [1]. Extensive monitoring is required to determine the effect of these energy efficiency measures on the IT equipment and data center environment. Furthermore, monitoring is also important for expansion planning, in terms of increasing available space and optimizing the compute load, but perhaps even more so in terms of the power and thermal capacities of the data center.

There are conflicting interests between the DCOs of the colocation data center and the third parties (i.e. the data center's customers) who rent rack space in the data center: the customers are able to monitor their own equipment, but there is no incentive to share this data with the DCOs as the customers do not directly benefit from increased profit margins. In fact, there are privacy and security concerns which customers have to consider when exposing data to the colocation data center. Conversely, for a colocation data center, server metrics such as CPU utilization and CPU temperature are of great interest as the CPUs of servers are the primary contributors to the overall electrical load and thermal load of a data center [2]. While the colocation data center cannot monitor the internal state (e.g. utilization) of the IT equipment, they can monitor other aspects of the data center, such as the global temperature, humidity and electrical load. There are also metrics which can be monitored by both the customers as well as the data center.

In this work, we analyse the server metrics that are available

978-1-7281-4117-6/19/$31.00 ©2019 IEEE
DOI 10.1109/DataCom.2019.00043

to colocation data centers and customers. We use these metrics to discover models that enables monitoring of the customers IT equipment. We are interested in models that describe the electrical load, computational load, as well as the thermal load of the data center. These models would; (1) give colocation data centers the tools required to monitor the server infrastructure despite not having access to the servers; (2) measure the effects of energy saving actions on the server equipment; and (3) assist in the expansion planning process to increase rack space, as well as thermal and power capacities.

We conduct a large scale experiment where we collect 2.5 billion data points from over 160 High Performance Computing (HPC) servers in a single data center over a period of 5 months. The dataset consists of a multitude of metrics, ranging from CPU usage and power consumption, to inlet and outlet air temperatures. We propose a method for *data-driven monitoring of colocation data centers*. The objective is to discover correlations between different metrics, and utilize them to develop models that assist in monitoring and controlling a colocation data center without requiring customers to expose their data. *In this work, we answer the question whether it is feasible to monitor the server infrastructure of a colocation data center without requiring customers to provide access to the operating system, in turn enabling extensive monitoring for colocation data centers despite conflicting interests*.

The remainder of this paper is organized as follows. Section 2 explores the state of the art. We describe the dataset of metrics, the acquisition method, and the metric selection process in Section 3. In Section 4, we propose numerous models for monitoring servers in a colocation data center and evaluate them. The results are discussed in Section 5. We highlight our future work in Section 6, and end with concluding remarks in Section 7.

II. State of the Art

The conflicting interests that exists between the DCOs of the colocation data center and the third party customers who are renting space in a colocation data center play a central role in this work. A data center can monitor metrics related to the global temperature, humidity, and electrical load with relative ease. They also have full control of the cooling systems. The customers of the colocation data center can monitor the internal state of the server, as they have direct access to the operating system. These metrics include memory usage, number of processes running, but more importantly CPU utilization and temperature. There are also server-related metrics that colocation data centers can monitor without requiring server access, such as inlet and outlet air temperature, and power consumption. Data centers often use metered Power Distribution Units (PDUs) that monitor the power consumption. The collection of air temperature data may require the deployment of additional sensors.

A colocation data center aims to reduce the operation expenditures, which can be achieved by optimizing the overall efficiency of the data center. An example of a measure that reduces the operational expenditures and increases the overall efficiency is an increased temperature set point. This reduces the load on the cooling equipment. However, this also has a significant effect on the IT equipment, as it will be operating at higher temperatures. This can have a major impact on the performance, due to thermal throttling and humidity issues. Therefore, the colocation data center has an interest in metrics that are only accessible by their customers, such as the CPU utilization (compute load) and the CPU temperature (thermal load). The customer on the other hand has no incentive to share the data, and instead has privacy and security considerations to take into account.

Taking an Internet of Things (IoT) approach to data center monitoring allows for the collection of a large and varied dataset. This dataset can also be used for evaluating the many key performance indicators that are applicable to data centers [3]. Furthermore, correlating the metrics may uncover models that allows colocation data centers to monitor the third party IT equipment, despite the lack of access. These model may also assist in increasing the data center's energy efficiency. For example, models that detect comatose or zombie servers can contribute to the energy efficiency of the infrastructure. Comatose servers are servers which are no longer in use and serve no useful purpose. A report from the Anthesis Consulting Group states that the percentage of comatose servers in data centers is around 25% [4]. Utilizing IoT for data collection and analysis leads to improvements in the data center's service levels by use of predictive maintenance [5].

The authors of [6] propose a real-time data center monitoring infrastructure using low-power wireless sensors. They monitor the following parameters: temperature, humidity, airflow, water, security, vibration, differential air pressure, and fire systems. They state that such a system can be rapidly deployed and would enable real-time predictive modeling. In [7], the authors also propose a wireless system architecture for monitoring a data center. They consider not only sensing, but also actuation. They propose to control the Power Distribution Units and the cooling systems using wirelessly networked sensors and actuators. A different approach is taken by [8], combining Big Data strategies and 3D gaming technology to monitor and visualize a HPC cluster. They collect 5,000 environmental data points (external metrics) and 3,500 server data points (internal metrics). Their environmental data points include temperature, humidity, air pressure, power consumption, voltages, and amperages. Server data points consist of software versions, CPU load, memory allocations, disk utilization, network and link utilization, storage health and the state of the job scheduler.

In our work, we focus on metrics related to IT equipment, we do not collect metrics regarding cooling systems, air pressure, and fire systems. However, the scale at which we collect our metrics is orders of magnitude larger than [8]. We rely on IoT sensors to collect billions of data points in order to enhance the monitoring capabilities of colocation data centers.

The power usage and energy efficiency of both servers and data centers are topics which are actively researched. One of the most diverse datasets is provided by the SPECpower_-

ssj2008 benchmark [9]. The benchmark measures performance and power of servers using gradual load levels. The benchmark focuses on collecting measurements, but does not focus on modeling. In [10], the authors propose a complete system model for modeling the power consumption of six subsystems: CPU, memory, chipset, I/O, disk, and GPU. They have observed an average error of less than 9% per subsystem when evaluating their model. The authors of [11] propose a different approach to modeling power consumption. In their model, they consider servers which are running virtual machines, and evaluate how the number of virtual machines influences the server's power consumption. They also include the idle power consumption and the overhead introduced by the hypervisor. Linear regression is applied to obtain the values of the model's parameters. The authors do not consider metrics other than power consumption and CPU usage. In [12], the authors model the CPU usage and the power consumption as random variables and exploit the monotonicity property to describe the relationship between these variables. The authors report mean errors between 2% to 5.2%, depending on the dataset. Only the power consumption and CPU usage are used. The authors of [13] propose a method to estimate the power consumption of individual CPU cores based on the measured CPU core temperature. They also develop a technique to optimize the throughput on CPU's that have thermal constraints. Their optimization method improved throughput by 4%, when compared to existing temperature-based methods.

We observe that the state of the art focuses primarily on estimating the power consumption based on internal metrics such as CPU utilization, core temperature, and sometimes even considers the hypervisor itself and the number of running virtual machines. In other words, they estimate the external metrics (often power consumption) based on internal metrics (often CPU usage). In our work, we approach this problem from the opposite direction because a data center can easily measure the power consumption of IT equipment. It is the internal server metrics that are difficult for the colocation data center to measure and monitor.

III. Methodology

To discover metric correlations and candidate models to enable data-driven monitoring of colocation data centers, a large dataset covering many metrics is required. The University of Groningen has two on-campus data centers that house the university's IT infrastructure, in addition to a number of HPC clusters and co-located servers. The data is gathered from one of these HPC clusters; the Peregrine cluster, consisting of 164 blade servers. The cluster is utilized by researchers and staff of the University of Groningen to assist in computationally expensive jobs. Over the years, the Peregrine cluster has assisted in more than 150 scientific publications. The type of jobs that run on the cluster vary greatly, resulting in a varied data set covering many different work loads.

Each of the 164 servers has two Intel Xeon E5-2680 v3 CPU's and 128 gigabytes of Random Access Memory (RAM), bringing the total to 3960 individual CPU cores and 21 terabytes of RAM respectively. Each server has 1 terabyte of internal disk space, and is connected to a storage area network consisting of 463 terabytes. The servers are locally connected by a 56 gigabit per second Infiniband network connection, and are also connected by a 10 gigabit per second ethernet connection to the Internet. Data collection started on the 1st of December 2016. Table I shows the metrics that are collected every 10 seconds, and their unit of measurement. Every month, around 560,000,000 individual data points are collected from 164 servers. For this work, a subset of approximately 2,500,000,000 data points was used. Of the 164 servers, 15 servers recorded unusable data due to faulty configurations, or missing data. Resulting in a usable dataset consisting of 149 servers in total.

TABLE I
The list of collected server metrics.

Metric	Name	Unit
Internal Metrics		
T_{cpu}	CPU Temperature	Degrees Celsius
CPU_{user}	CPU Util. (User)	Percentage
CPU_{sys}	CPU Util. (System)	Percentage
Mem_{free}	Unused RAM	Kilobytes
$Proc_{run}$	# Processes Running	Integer
$Proc_{total}$	# Processes Total	Integer
Eth_{in}	Ethernet In	Bytes / Second
Eth_{out}	Ethernet Out	Bytes / Second
Inf_{in}	Infiniband In	Bytes / Second
Inf_{out}	Infiniband Out	Bytes / Second
External Metrics		
P_{watts}	Power Consumption	Watts
T_{in}	Inlet Air Temperature	Degrees Celsius
T_{out}	Outlet Air Temperature	Degrees Celsius

We collect the dataset using a variety of hardware and software sensors. The software sensors collect internal server metrics such as RAM utilization as reported by the operating system. The hardware sensors collect external metrics, such as the air temperature. In order to prepare the dataset for analysis, we apply the following sanitation steps:

1) Combine CPU_{user} and CPU_{system} to obtain the CPU_{total} such that $0.0 \leq CPU_{tot} \leq 100.0$.
2) Subtract T_{in} from T_{out} to obtain T_{diff}, the temperature difference between inlet and outlet.
3) Detect and remove outliers / faults introduced by measurements errors.
4) Compute the correlation between every pair of metrics.
5) Determine the lag between selected metrics.

Step 1 and 2 are mutations of the dataset. In the first step, we combine the CPU usage in user space (CPU_{user}) and kernel space (CPU_{sys}), the combination of these metrics gives us the total CPU usage (CPU_{tot}). In the second step, we subtract the temperature of the cold air entering the server (T_{in}) from the temperature of the hot air exiting the server (T_{out}) in order to obtain the temperature difference (T_{diff}).

In the third step, we remove the faulty measurements from the dataset. If a row in the dataset does not conform to the following constraints, it is discarded:

TABLE II
KENDALL'S TAU CORRELATION, INCLUDING STANDARD DEVIATION.

	T_{cpu}	P_{watts}	$Proc_{run}$	CPU_{tot}
P_{watts}	0.76 ± 0.07	-		
$Proc_{run}$	0.54 ± 0.11	0.62 ± 0.13	-	
CPU_{tot}	0.73 ± 0.08	0.83 ± 0.08	0.70 ± 0.13	-
T_{diff}	0.78 ± 0.11	0.82 ± 0.09	0.57 ± 0.12	0.76 ± 0.07

1) $T_{cpu} > 0$
2) $P_{watts} > 0$
3) $\sum_{m \in M} m > 0$

Inspection of the dataset shows that P_{watts} and T_{cpu} are sometimes 0, while other metrics are not. This indicates a measurement error. Thus, the first and second constraints are required. The final constraint ensures that rows with all zeros are excluded, by verifying that the sum of all metrics is not zero. These all-zero rows appear when the monitoring system is unavailable.

For step 4, the goal is to identify which metrics are correlated, and to what extent. This allows us to perform feature selection and use the appropriate set of metrics for our models. We determine the correlation between metrics using Kendall's tau coefficient [15] for each individual node:

$$\tau = \frac{n_c - n_d}{n(n-1)/2} \tag{1}$$

Where n_c is the number of concordant pairs, n_d is the number of discordant pairs, and n is the number of pairs. When applied to the data we immediately notice that there are numerous metrics with very weak correlations ($-0.5 < \tau < 0.5$): T_{in}, $Avail_{ram}$, $Proc_{tot}$, Eth_{in}, Eth_{out}, Inf_{in} and Inf_{out}. We discard the metrics with a weak correlation, as they would have little to no contribution to our models. Table II provides the resulting mean correlation and standard deviation of the remaining metrics, where the mean is taken over the correlation results for each individual server. The remaining metrics consist of two external metrics (T_{diff} and P_{watts}) and three internal metrics (T_{cpu}, $Proc_{run}$, and CPU_{tot}). The characteristics of the metrics are described in Table III.

The metrics that describe the internal state of a server are CPU_{tot}, T_{cpu}, and $Proc_{run}$. However, we exclude $Proc_{run}$ as it is too specific to the tasks that a server is performing, and therefore does not generalize well in different environments. For example, a single process could have a CPU usage of 100%, while the same holds true for 100 processes with 1% CPU usage each. Furthermore, the correlations between $Proc_{run}$ and the external metrics are significantly lower than the other correlations we have observed in Table II. Thus we have the following metrics, CPU_{tot} and T_{cpu}, representing the internal state of the server. The external metrics are P_{watts} and T_{diff}, as these can be measured externally to the server. It has to be noted that T_{diff} in reality consist of two metrics: T_{in} and T_{out}, the in- and outlet air temperature. Based on the Kendall's Tau correlation we identify the following models to be explored and evaluated:

TABLE III
CHARACTERISTICS OF THE SELECTED SET OF METRICS.

	T_{cpu}	P_{watts}	$Proc_{run}$	CPU_{tot}	T_{diff}
Mean	56.06	263.73	18.02	62.36	13.96
S.D.	12.51	70.24	24.12	37.43	3.29
Min.	24.00	77.00	0.00	0.00	0.00
25%	34.00	196.00	12.00	29.22	11.00
50%	43.00	280.00	22.00	79.20	15.00
75%	60.00	336.00	24.00	100.00	17.00
Max.	100.00	448.00	2867.00	100.00	25.00

1) **CPU Load models** (CPU_{tot}), utilizing T_{diff} and / or P_{watts}.
2) **CPU Thermal models** (T_{cpu}), utilizing T_{diff} and / or P_{watts}.
3) **Air Flow Thermal models** (T_{diff}), utilizing P_{watts}.

The CPU Load models estimate the CPU_{tot} metric, with T_{diff} and / or P_{watts} as input features. This is an important metric for the colocation data center as comatose (unused) servers have a significant impact on the energy consumption of the data center, and it provides an indication of the overall load in the data center.

The CPU Thermal models estimate the T_{cpu} metric, also with T_{diff} and / or P_{watts} as input features. The CPU Thermals are another important metric from the data center point of view, as this relates to the cooling load. It also allows the colocation data center to monitor the effects of changes in the cooling system on the servers.

Finally, the Air Flow models estimate the T_{diff} metric, with P_{watts} as the input feature. These models also correlates strongly to the cooling load of the data center. The Air Flow Thermal model utilizes one external metric (P_{watts}) to estimate another external metric (T_{diff}). This model does not provide any insights in the internal state of a server, but it allows the colocation data center to describe the relationship between power consumption and the increase in air temperature which has to be mitigated by the cooling system.

In total we identify 7 models to evaluate; 3 different variations of CPU Load models, 3 different variations of CPU Thermal models, and 1 Air Flow Thermal model. All of the models relate directly to either the IT equipment or cooling system, the two categories responsible for the majority (75%) of a data center's energy consumption. These models also assist with measuring the effect of energy savings measures without requiring internal access to the IT equipment.

Finally, we consider the notion of lag between two discrete time-series (step 5). Inspection of the data set shows that there is a delay between the time-series. The cause of this lag is twofold: there is a delay between individual observations from sensors, and certain time-series have a natural tendency to lag behind. For example, an increase in T_{cpu} would eventually lead to an increase in T_{diff} (as exhaust temperatures increase). However, it takes a certain amount of time for the heat to dissipate, which causes lag between the two time-series. Intuitively, decreasing the lag between time-series increases the number of concordant pairs, and consequentially increases the correlation. A higher correlation between time-series is of benefit when modeling the data. To determine the exact amount of lag between two metrics, we utilize the cross-correlation. Given two discrete time-series $x[m]$ and $y[m]$, the cross-correlation is defined as [16]:

$$R_{xy}(k) = \sum_{m=-\infty}^{\infty} x[m]y[m-k] \tag{2}$$

Where $k \in \mathbb{Z}$, and $-\infty \leq k \leq \infty$. Parameter k is also known as the lag parameter. For each of the selected metrics we determine the lag that exists between them by performing cross-correlation for every pair of time-series and selecting the k which maximizes (since the time-series are positively correlated) $R_{xy}(k)$ such that:

$$lag_{units} = \arg\max_{k} \; R_{xy}(k) \tag{3}$$

We calculate the lag individually for every server and adjust the time-series accordingly. Shifting the time-series based on the lag further increases Kendall's tau correlation.

IV. Models and Evaluation

We distinguish between two types of models: the individual models and the universal model. Individual models are models which we train separately for each individual server, using only the subset of data that belongs to a specific server. For the universal model, we train and evaluate one global model using all available data, of every server. This verifies the ability of the models to generalize.

Inspection of the dataset shows that there is a near-linear correlation between the external metrics (P_{watts}, T_{diff}) and the internal metrics (CPU_{tot}, T_{cpu}). Therefore, we define a linear regression model of at most two parameters (Eq. 4):

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 \tag{4}$$

Where x_1 and x_2 are the parameters of the model, w_0, w_1, w_2 are the weights, and $\hat{y}$ is the estimated value. Since the data is not fully linear, we transform the features into polynomial features. For every model, we generate all polynomial features from the 1st degree to the 10th degree. For example, given two features x_1 and x_2, the transformation to polynomial features of the 2nd degree is performed according to Equation 5.

$$z = [x_1, x_2, x_1 x_2, x_1^2, x_2^2] \tag{5}$$

The resulting model is a linear regression model in which the features are polynomial (Eq. 6). Thus we can apply any linear regression technique, such as ordinary least squares fitting, to obtain our polynomial model.

$$\hat{y} = w_0 + w_1 z_1 + \cdots + w_5 z_5 \tag{6}$$

To train our individual models we apply the Ridge Regression method, which is expressed as ordinary least squares, the first term in Equation 7, with an additional L2 regularization term, the second term in the equation. The α parameter controls the strength of the regularization.

$$\min_{w} \|Xw - y\|^2 + \|\alpha w\|^2 \tag{7}$$

Due to the sheer size of the complete dataset (2.5 billion data points) compared to the dataset for an individual server (15 million data points), we have to modify the approach for learning universal models, as we cannot fit all 2.5 billion data points simultaneously. Instead of Ridge Regression, we fit the polynomial models using Stochastic Gradient Descent with L2 regularization [17]. We partially fit the data of every individual server, until all the training data has been fitted.

Finally, to evaluate how well our models generalize, we apply k-fold cross-validation with $k = 10$ [18]. We evaluate both the Root Mean Squared Error (RMSE) and the Coefficient of Determination (R^2) obtained using cross-validation. In Section IV-A we evaluate the individual models, and in Section IV-B we evaluate the universal model. The results are discussed in Section V. The tables which summarize the results distinguish between $RMSE_i$ and R_i^2 for the individual models, and $RMSE_u$ and R_u^2 for the universal model.

A. Individual Models

We train models for each of the 149 servers. For each of the seven models, we train 10 distinct iterations where we vary the polynomial degree from 1 to 10. In total we train ($149 \times 7 \times 10 =$) 10,430 different polynomial models. When the results are summarized, we discard the results from polynomial degrees 7 till 10, as in nearly all cases the models start to overfit at the 5th or 6th degree. The exception to this observation are the models with only a single feature. For these models, as the number of degrees approaches the number of data points, the observed error approaches 0.

1) Individual CPU Load models (CPU_{tot}): First, we evaluate the results for modeling the CPU load (CPU_{tot}). The summary of the results can be found in Table IV. In this table we list the mean $RMSE_i$ and the mean R_i^2 metrics (and their standard deviations) for every selection of input features. The input features are the external parameters as selected in Section III. For feature P_{watts}, we observe a significant improvement in $RMSE_i$ and R_i^2 between the 1st and 3rd degrees, with $RMSE = 8.13$ and $R^2 = 0.95$. For feature T_{diff}, the same improvement is observed at the 3rd degree, where $RMSE_i = 12.45$ and $R_i^2 = 0.89$. Furthermore, we can clearly see that the models start to overfit from the 6th degree, as the standard deviation of $RMSE_i$ increases, and the R_i^2 significantly decreases. The last features we evaluate to model the CPU Load is the combination of P_{watts} and T_{diff}. As observed in previous cases, a notable improvement happens until the 3rd degree, where $RMSE_i = 7.68$ and $R_i^2 = 0.96$. Improvements are observed until the 5th degree, after which the models overfit as the increased standard deviation of the

TABLE IV
SUMMARY OF MODELING CPU LOAD (CPU_{tot}), i = INDIVIDUAL MODELS, AND u = UNIVERSAL MODEL.

Degree:	1st	2nd	3rd	4th	5th	6th
Features: P_{watts}						
$RMSE_i$	10.02 ± 0.84	9.95 ± 0.87	8.13 ± 0.85	8.08 ± 0.85	8.05 ± 0.86	8.02 ± 0.86
R_i^2	0.92 ± 0.03	0.93 ± 0.03	0.95 ± 0.03	0.95 ± 0.03	0.95 ± 0.03	0.95 ± 0.03
$RMSE_u$	10.68 ± 0.53	10.68 ± 0.53	8.71 ± 0.34	8.67 ± 0.32	8.67 ± 0.32	8.66 ± 0.32
R_u^2	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.00	0.97 ± 0.00	0.97 ± 0.00	0.97 ± 0.00
Features: T_{diff}						
$RMSE_i$	14.25 ± 1.06	14.16 ± 1.18	12.45 ± 1.16	12.37 ± 1.21	12.35 ± 1.26	13.13 ± 5.77
R_i^2	0.85 ± 0.04	0.85 ± 0.03	0.89 ± 0.03	0.89 ± 0.03	0.89 ± 0.03	0.63 ± 2.10
$RMSE_u$	15.68 ± 0.92	15.66 ± 0.88	13.68 ± 0.87	13.64 ± 0.89	13.67 ± 0.95	13.51 ± 0.85
R_u^2	0.91 ± 0.02	0.91 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.94 ± 0.01
Features: P_{watts} and T_{diff}						
$RMSE_i$	9.71 ± 0.83	9.5 ± 0.86	7.68 ± 0.8	7.51 ± 0.79	7.35 ± 0.81	8.14 ± 5.89
R_i^2	0.93 ± 0.02	0.93 ± 0.02	0.96 ± 0.02	0.96 ± 0.02	0.96 ± 0.02	0.68 ± 2.27
$RMSE_u$	10.43 ± 0.47	10.30 ± 0.46	8.38 ± 0.32	8.30 ± 0.32	8.14 ± 0.30	9.10 ± 2.97
R_u^2	0.96 ± 0.01	0.96 ± 0.01	0.98 ± 0.00	0.98 ± 0.00	0.98 ± 0.00	0.97 ± 0.03

$RMSE_i$ and the lower R_i^2 signify. The optimal result is obtained using both P_{watts} and T_{diff} as features to model the CPU Load with a polynomial of the 5th degree. This results in a mean $RMSE_i$ of 7.35%.

2) Individual CPU Thermal models (T_{cpu}): Next, we analyze the results we obtained when modeling the CPU Thermal characteristics (T_{cpu}). A summary of these results is shown in Table V. Again we list the mean $RMSE_i$ and R_i^2 that was obtained individually for all 149 servers. First, we model the CPU Thermals using only P_{watts} as a feature. Again, we observe a significant improvement at the 3rd degree, where the $RMSE_i = 4.74$ and the $R_i^2 = 0.85$. A similar decrease in $RMSE_i$ is noted when only applying feature T_{diff}. At the 3rd degree we observe $RMSE_i = 3.51$ and $R^2 = 0.92$. At the 6th degree the models start to overfit; the $RMSE_i$ increases, as does the standard deviation. When using both P_{watts} and T_{diff} as features, we observe $RMSE_i = 3.23$ and $R_i^2 = 0.93$ for the 3rd degree. Significant overfitting occurs after the 5th degree, with R_i^2 becoming negative which indicates that the model fits the data extremely poor. The optimal results are obtained when using both P_{watts} and T_{diff} as features, a polynomial of the 5th degree models the CPU Thermal characteristics with a mean $RMSE_i$ of $3.17°C$.

3) Individual Air Flow Thermal models (T_{diff}): Finally, we assess the results of modeling the Air Flow Thermal characteristics (T_{diff}), the difference between air flow inlet and outlet temperature of a server. The results are summarized in Table VI In this case, there is only one feature that has a strong correlation, that is P_{watts}. We do not observe a significant increase at the 3rd degree, as we noted in previous models. The $RMSE_i$ and R_i^2 improve ever so slightly as the number of degrees increases. The optimal results are observed at the 10th degree, where the mean $RMSE_i$ of the Air Flow Thermal models is $1.07°C$.

B. Universal Model

After evaluating and inspecting all of the individual models in Section IV-A, it becomes apparent that the individual models of a given type appear to be very similar to one another. Therefore, the next step is to reduce the ten-thousand individual models to less than a hundred universal models, and determine how well these universal models generalize. Considering that large data centers can contain several hundreds of thousands of servers [19], it would be advantageous to develop a universal model that represents a significant subset of servers. For each of the 7 model types we train 10 distinct iterations where we vary the polynomial degree from 1 to 10. We train ($7 \times 10 =$) 70 different universal polynomial models.

1) Universal CPU Load models (CPU_{tot}): As before with the individual models, we first analyze the results (Table IV) for modeling the CPU Load with one universal model, using three different combinations features. When only using feature P_{watts}, we observe a significant decrease in $RMSE_u$ at the 3rd degree, where $RMSE_u = 8.71$ and $R_u^2 = 0.97$. For feature T_{diff} we record $RMSE_u = 13.68$ and $R_u^2 = 0.93$ at the 3rd degree, which shows the most significant improvement as well. And finally, when using both P_{watts} and T_{diff} as features, we obtain $RMSE_u = 8.38$ and $R^2 = 0.98$ at the 3rd degree. However in this case the the 5th degree shows slightly better results, whereas the 6th degree and higher show signs of overfitting. These results are in line with the results of the individual models for CPU Load. We observe the optimal results using a polynomial of the 5th degree while using both P_{watts} and T_{diff} as features. This yields a $RMSE_u$ of 8.14%.

2) Universal CPU Thermal models (T_{cpu}): We model the CPU Thermals using the same three sets of features as used when modeling the CPU Load. First we use P_{watts} as a feature, which yields significant improvements (Table V) up until the 3rd degree ($RMSE_u = 5.17$ and $R_u^2 = 0.92$), after which the $RMSE$ decreases much slower. Selecting T_{diff} as a feature yields better results, at the 3rd degree we obtain $RMSE_u = 4.22$ with $R_u^2 = 0.95$. After the 3rd degree we observe minor improvements in $RMSE$. When utilizing both P_{watts} and T_{diff} as features we get $RMSE_u = 3.88$ and $R_u^2 = 0.95$ at the 3rd degree, improving at the 4th degree

TABLE V
SUMMARY OF MODELING CPU THERMALS (T_{cpu}), i = INDIVIDUAL MODELS, AND u = UNIVERSAL MODEL.

Degree:	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**
Features: P_{watts}						
$RMSE_i$	4.96 ± 0.54	4.84 ± 0.56	4.74 ± 0.55	4.72 ± 0.55	4.67 ± 0.55	4.65 ± 0.55
R_i^2	0.84 ± 0.04	0.85 ± 0.04	0.85 ± 0.04	0.85 ± 0.04	0.86 ± 0.04	0.86 ± 0.04
$RMSE_u$	5.32 ± 0.23	5.23 ± 0.25	5.17 ± 0.25	5.17 ± 0.25	5.13 ± 0.23	5.13 ± 0.23
R_u^2	0.91 ± 0.01	0.91 ± 0.01	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01
Features: T_{diff}						
$RMSE_i$	4.04 ± 0.35	3.88 ± 0.39	3.51 ± 0.36	3.4 ± 0.35	3.39 ± 0.37	3.47 ± 0.71
R_i^2	0.89 ± 0.04	0.90 ± 0.03	0.92 ± 0.03	0.92 ± 0.02	0.92 ± 0.03	0.90 ± 0.17
$RMSE_u$	4.69 ± 0.30	4.61 ± 0.35	4.22 ± 0.31	4.15 ± 0.33	4.13 ± 0.34	4.13 ± 0.35
R_u^2	0.93 ± 0.01	0.93 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01
Features: P_{watts} and T_{diff}						
$RMSE_i$	3.90 ± 0.36	3.66 ± 0.39	3.23 ± 0.34	3.17 ± 0.36	3.21 ± 0.78	4.79 ± 10.17
R_i^2	0.90 ± 0.03	0.91 ± 0.02	0.93 ± 0.02	0.93 ± 0.03	0.89 ± 0.33	-6.58 ± 60.01
$RMSE_u$	4.42 ± 0.23	4.26 ± 0.28	3.88 ± 0.24	3.84 ± 0.25	3.85 ± 0.35	3.86 ± 0.34
R_u^2	0.94 ± 0.01	0.94 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01

after which the error slowly increases. The best results are obtained when using both features, a 4th degree polynomial gives a $RMSE_u$ of $3.85°C$.

3) Universal Air Flow Thermal models (T_{diff}): The last universal model we train and evaluate is for modeling the Air Flow Thermals. The best result is observed when using a polynomial between the 3rd and 10th degree. This yields a $RMSE_u = 1.25$ and $R_u^2 = 0.93$ (Table VI). The error remains nearly constant after the 3rd degree, which results in an optimal $RMSE_u$ of $1.25°C$.

V. DISCUSSION

Based on the evaluation, we conclude that the individual models are, when compared to the universal model, better at estimating the CPU Load, CPU Thermals, and the Air Flow Thermals. This is expected, as each server will have unique characteristics that are included in its own individual model. These variations can, for example, be caused by the positioning of the server in the rack. However, the universal model is nearly as accurate as the individual models in terms of $RMSE$. Both the individual and the universal models demonstrate similar behaviour: as the number of polynomial degrees increases beyond the 6th degree, the models tend to overfit. After the 3rd and 4th degree there is little to no improvement in terms of $RMSE$ and R^2. Therefore the 3rd and 4th degree strike a good balance between computational complexity, which grows exponentially as number of polynomial degrees increases, and the $RMSE$ and R^2 scores.

Without knowledge of the internal state of a server, we prove that we can accurately estimate the CPU Load using only external metrics. The CPU Load ranges from 0% to 100%. Exclusively utilizing the power consumption (P_{watts}) of the server as a feature for the models yields an error of 8.02% (individual) or 8.66% (universal) when estimating the CPU Load. When we estimate the CPU Load based on the temperature difference (T_{diff}) between inlet air and outlet air, we obtain less accurate results with an error of 12.35% (individual) or 13.51% (universal). When considering both power consumption and temperature difference as features, the error is 7.35% (individual) or 8.14% (universal).

We can also estimate the CPU Thermals based on external parameters. The CPU Thermals range from $24°C$ to $100°C$. When determining the CPU Thermals using the server's power consumption as a feature, we obtain an error of $4.65°C$ (individual) or $5.13°C$ (universal). Using the temperature difference between inlet and outlet air to estimate the CPU Thermals yields a better result, with an error of $3.39°C$ (individual) or $4.13°C$ (universal). Estimating the CPU Thermals using both power consumption and temperature difference results in an error of $3.17°C$ (individual) or $3.84°C$ (universal).

In addition, we demonstrate that we can estimate the Air Flow Thermals without requiring access to the server. While this is not used for determining a server's internal state, it is useful from the data center perspective as it relates directly to the cooling load of a data center. In our dataset, the Air Flow Thermal characteristics range from $0°$ to $25°$. Given the power consumption of a server, our model yields an error of $1.07°C$ (individual) or $1.25°C$ (universal) when estimating the Air Flow Thermals.

VI. CONCLUSION

We investigated the issue of monitoring third party IT equipment in a colocation data center from the Data Center Operator (DCO) perspective, with the goal of uncovering models that can be used to monitor and control colocation data centers more efficiently. In this work, we have provided a positive answer to the question whether it is feasible to monitor the server infrastructure of colocation data centers without requiring direct access to the customer's servers, avoiding the conflicting interests of the DCO and customer.

Our answer is based on 2.5 billion data points collected from 164 individual servers, covering 13 distinct metrics. We determined the correlation between each of the metrics, and identified external metrics that are good candidates for the estimation of the internal state of servers. We presented our approach to modeling the relationship between internal

TABLE VI
SUMMARY OF MODELING AIR FLOW THERMALS (T_{diff}), i = INDIVIDUAL MODELS, AND u = UNIVERSAL MODEL.

Degree:	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**
Features: P_{watts}						
$RMSE_i$	1.10 ± 0.10	1.10 ± 0.10	1.09 ± 0.10	1.09 ± 0.10	1.09 ± 0.10	1.08 ± 0.10
R_i^2	0.88 ± 0.04	0.88 ± 0.04	0.88 ± 0.04	0.89 ± 0.04	0.89 ± 0.04	0.89 ± 0.04
$RMSE_u$	1.26 ± 0.10	1.26 ± 0.10	1.25 ± 0.10	1.25 ± 0.10	1.25 ± 0.10	1.25 ± 0.10
R_u^2	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01

and external metrics, discovering CPU Load models, CPU Thermal models, and Air Flow Thermal models. We distinguished between individual models and universal models, and determined that the universal models generalize well. In total, we trained and evaluated 10,430 individual models and 70 universal models. The CPU Load models allow the DCO to monitor the overall load in the data center and to use this information for expansion planning. The CPU Thermal models and the Air Flow Thermal models strongly relate to estimating the required cooling capacity, and can be used to optimize the cooling systems or to monitor the effect of cost-saving measures on the IT equipment.

The results are evidence that using external metrics as measured by the data center to estimate the internal state of a server is feasible. This enables data-driven monitoring for colocation data centers, as we demonstrate that access to the server's operating system or chassis is no longer needed to monitor the server's internal state. Instead, we can employ IoT-enabled sensors to measure the inlet and outlet air temperature, and the power consumption of a server. Using these inexpensive sensors, the DCOs can monitor the server infrastructures of their customers, as our models enable able them to monitor the CPU Load, CPU Thermals, and Air Flow Thermals using exclusively external parameters. In turn, this allows DCOs to ensure a high Quality of Service (QoS) while maintaining the existing Service Level Agreements (SLAs), while at the same time implementing cost-saving measures and monitoring the effect of these measures on the IT equipment.

In future work, we would like to collect data from servers with different characteristics (e.g. variations in chassis size, CPU models, GPU models) and introduce additional features to our models. We also would like to explore the role of AI planning and scheduling in work load placement, server placement, and cooling optimizations, utilizing the models we have uncovered.

ACKNOWLEDGMENT

The authors thank Bob Dröge, high-performance computing expert at the University of Groningen, and the staff of the Center of Information Technology for their support. The authors thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The work is supported by The Netherlands Organisation for Scientific Research NextGenSmart DC project, contract no. 629.002.102.

REFERENCES

[1] L. A. Barroso, J. Clidaras, and U. Hlzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*, 2013. [Online]. Available: http://dx.doi.org/10.2200/S00516ED2V01Y201306CAC024

[2] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 732–794, Firstquarter 2016.

[3] D. Vemula, B. Setz, G. S. V. Rao, G. R. Gangadharan, and M. Aiello, "Metrics for sustainable data centers," *IEEE Transactions on Sustainable Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[4] J. Koomey and J. Taylor, "Zombie / comatose servers redux," *Koomey Analytics and Anthesis*, 2017.

[5] V. K. Singh and J. Guo, "Improving service levels using internet of things infrastructure in data centers," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, May 2016, pp. 1–3.

[6] M. Levy and J. O. Hallstrom, "A new approach to data center infrastructure monitoring and management (dcimm)," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2017, pp. 1–6.

[7] M. Wiboonrat, "Data center infrastructure management wlan networks for monitoring and controlling systems," in *The International Conference on Information Networking 2014 (ICOIN2014)*, Feb 2014, pp. 226–231.

[8] M. Hubbell, A. Moran, W. Arcand, D. Bestor, B. Bergeron, C. Byun, . . . , and J. Kepner, "Big data strategies for data center infrastructure management using a 3d gaming platform," in *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, Sept 2015, pp. 1–6.

[9] L. D. Gray, A. Kumar, and H. H. Li, "Workload characterization of the specpower_ssj2008 benchmark," *SPEC International Performance Evaluation Workshop*, pp. 262–282, 2008.

[10] W. L. Bircher and L. K. John, "Complete system power estimation using processor performance events," *IEEE Transactions on Computers*, vol. 61, no. 4, pp. 563–577, April 2012.

[11] G. Warkozek, E. Drayer, V. Debusschere, and S. Bacha, "A new approach to model energy consumption of servers in data centers," in *2012 IEEE International Conference on Industrial Technology*, March 2012, pp. 211–216.

[12] W. Dargie, "A stochastic model for estimating the power consumption of a processor," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1311–1322, May 2015.

[13] D. Oh, N. S. Kim, C. C. P. Chen, A. Davoodi, and Y. H. Hu, "Runtime temperature-based power estimation for optimizing throughput of thermal-constrained multi-core processors," in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2010, pp. 593–599.

[14] A. Justice, T. Winkler, M. Feitosa, M. Graff, V. Fisher, K. Young, . . . , and L. Cupples, "Genome-wide meta-analysis of 241,258 adults accounting for smoking behaviour identifies novel loci for obesity traits," *Nature Communications*, vol. 8, p. 14977, 4 2017.

[15] Y. Dodge, *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, ch. Kendall Rank Correlation Coefficient, pp. 278–281.

[16] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, 1st ed. Prentice Hall, 6 1975.

[17] L. Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent*. Heidelberg: Physica-Verlag HD, 2010, pp. 177–186.

[18] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2, 1995, pp. 1137–1145.

[19] H. Geng, *Data Center Handbook*. John Wiley & Sons, 2014.