Beneath the SURFace

An MRI-like View into the Life of a 21st-Century Datacenter

ALEXANDRU UTA, KRISTIAN LAURSEN, ALEXANDRU IOSUP, PAUL MELIS, DAMIAN PODAREANU, AND VALERIU CODREANU

Alexandru Uta is an assistant professor in the computer systems group at LIACS, Leiden University. He received his PhD in 2017 from Vrije Universiteit Amsterdam on topics related to distributed storage systems for scientific workloads. His current research interests are in taming large-scale infrastructure—from designing reproducible experiments to understanding and evaluating performance, as well as designing efficient large-scale computer systems. a.uta@liacs.leidenuniv.nl

Kristian Laursen is pursuing a BSc in computer science at Vrije Universiteit Amsterdam and interns at the SURFsara offices. His professional interests span the field of computer science with a focus on datacenter technologies and distributed systems. kristianvalurlaursen@gmail.com

Alexandru Iosup is a full professor at Vrije Universiteit Amsterdam, chairing the research group on massivizing computer systems and SPEC-RG's cloud group. He is a member of the Netherlands Young Royal Academy of Arts and Sciences. He received a PhD in computer science from Technische Universiteit Delft. His distributed systems work received prestigious recognition: the 2016 Netherlands ICT Researcher of the Year, the 2015 Netherlands Teacher of the Year, and several SPEC community awards, including the SPECtacular Award. A.Iosup@vu.nl or @AIosup

Real-world data is crucial in understanding and improving our world, from health care to datacenters. To help the computer systems community with data-driven decisions, we open-source a collection of fine-grained, low-level operational logs from the largest public-sector datacenter in the Netherlands (SURFsara). In this article, we describe the infrastructure providing the data, give examples of some of this data, and perform thorough statistical analysis to indicate that this ongoing collection not only reflects the ground truth but will be useful to designers and maintainers of large clusters, and generally to computer systems practitioners.

Medical professionals employ MRI images to look inside our bodies, thus gaining a deeper understanding of the spread and effects of diseases. Open-source collections [1] of medical images enable building or improving analysis tools and training new professionals. In contrast, for computer systems, we do not yet fully benefit from MRI-like views on datacenters. Open source operational traces are scarce and bereft of low-level metrics. Absent such metrics, large-scale systems experts and infrastructure developers are currently forced to design, implement, and test their systems using unverified, sometimes even unrealistic, assumptions. The operational traces we propose help alleviate this problem. Moreover, low-level details of MRI images also offer clinicians predictive capabilities on the evolution of diseases. Similarly, our operational traces would allow for predictive analysis of systems behavior.

Real-world data can be instrumental in answering detailed questions: How do we know which assumptions regarding large-scale systems are realistic? How do we know that the systems we build are practical? How do we know which metrics are important to assess when analyzing performance? To answer such questions, we need to collect and share operational traces containing real-world, detailed data. The presence of low-level metrics is not only significant, but they also help researchers avoid biases through their variety. To address variety, there exist several types of archives, such as the Parallel Workloads Archive, the Grid Workloads Archive, and the Google or Microsoft logs (the Appendix gives a multi-decade overview). However, such traces mostly focus on higher-level scheduling decisions and high-level, job-based resource utilization (e.g., consumed CPU and memory). Thus, they do not provide vital information to system administrators or researchers analyzing the full-stack or the OS-level operation of datacenters.

The traces we are sharing have the finest granularity of all other open-source traces published so far. In addition to scheduler-level logs, they contain over *100 low-level, server-based metrics, going to the granularity of page faults or bytes transferred through a NIC*. The metrics presented in this article are gathered every *15 seconds* from a GPU cluster at SURFsara, totaling over 300 servers. This cluster includes high-speed networks and storage devices, and it is being used for scientific research in the Netherlands in areas such as physics, chemistry, weather prediction, machine learning, and computer systems.

This archive is a valuable resource for many professionals: software developers, system designers, infrastructure developers, machine learning practitioners, and policy-makers. During 2020, we will release monthly on Zenodo the trace data gathered in the previous 30

Paul Melis is group leader of the visualization group at SURFsara, which supports users of its computing infrastructure with visualization expertise and software development, on topics such as data visualization, 3D modeling and rendering, and virtual reality. He has an MSc in computer science from the University of Twente in the Netherlands, and worked on topics in scientific visualization and VR at the University of Groningen and University of Amsterdam before joining SURFsara in 2009. Paul.Melis@surfsara.nl

Damian Podareanu is a senior consultant in the High Performance Machine Learning Group from SURFsara. He studied mathematics and computer science at the University of Bucharest and the Polytechnic University of Bucharest and artificial intelligence at the University of Groningen. He worked as a software developer and scientific programmer for multiple companies before joining SURFsara in 2016. He is currently involved in several classical machine learning projects: IPCC, Examode, ReaxPro. damian@surfsara.nl

Valeriu Codreanu studied electrical engineering at the Polytechnic University of Bucharest, following up with a PhD in computer architecture at the same institute. Valeriu continued as a researcher at Eindhoven University of Technology and University of Groningen, working on GPU computing, computer vision, and embedded systems. He joined SURFsara in 2014, and in 2016 became PI of an Intel Parallel Computing Center project on scaling up deep learning. Valeriu is currently leading the High Performance Machine Learning Group at SURFsara. valeriu.codreanu@surfsara.nl

days, as FAIR (see https://www.go-fair.org/fair-principles/) open data. In this article, we provide a high-level overview of the metrics and data we gather, and a high-level characterization of the first three months of operation in 2020.

Three Months in the Life of a Datacenter

The SURFsara datacenter is used mostly by researchers from the Netherlands, running workloads in areas such as physics, chemistry, weather forecasting, machine learning, and computer systems. Users run primarily HPC-like workloads and deep-learning training, using combinations of regular CPU-, and multi-GPU-servers. A minority of the workloads run on big-data-like systems.

Figure 1 and Table 1 present a summary of several metrics computed over all the GPU servers in the LISA cluster over three months. The individual data points in Figure 1 represent the maximum value for a given hour over all servers, normalized to the maximum value of that metric for the whole period. Table 1 presents the range of values we encountered. We depict here only 10 metrics out of the 100+ collected. Even this high-level summary can be useful to datacenter engineers. For example, the alternation of the five colors for the metric *GPU Fanspeed* shows that the maximum fan speed for a GPU in the LISA cluster varies significantly during the three months analyzed, suggesting that there are very different levels of load in the system over this period. Engineers have to be alert, especially when the load is extreme, either very high or very low.

Our logs register all the interactions of user workloads with the datacenter itself. They also register maintenance events (e.g., adding or replacing servers—these are events which can be derived from the metrics), and unusual events (e.g. job failures, server failures, reboots). Last,

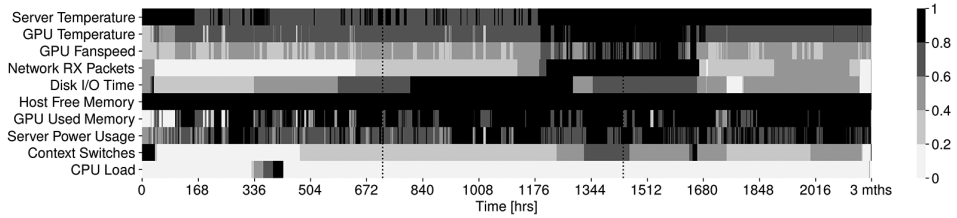

Figure 1: Metric variety and server load variability of the GPU-enabled servers in the LISA cluster over three months (January 1–March 31, 2020). Each data point represents the maximum value a server has encountered for that metric, normalized to the highest encountered value for that metric. The online version of this article shows this heat-map in color.

Metric	Min	Max	Median	Mean	CoV
Server Temperature (Celsius)	24	35	26	26	0.08
GPU Temperature (Celsius)	23	91	31	36	0.38
GPU Fanspeed (Percentage)	0	100	0	8	1.93
Network RX Packets (# packets x 1^6)	0.000003	18	0.460	1	1.73
Disk I/O Time (ms x 1^6)	0.0008	82	9	12	1.01
Host Free Memory (GB)	0.602	268	256	222	0.31
GPU Used Memory (GB)	0	12	0	1	1.96
Server Power Usage (Watt)	0	1400	312	401	0.59
Context Switches (# switches x 1^9)	0.0000052	216	12	23	1.2
CPU Load (Run-queue length)	0	7000	1	12	18.1

Table 1: Value range, median, mean, and coefficient of variation for each metric depicted in Figure 1.

they capture phenomena, such as sudden drops in activity, or low system load. Figure 1 depicts such a phenomenon: at almost all times, the majority of the GPU-enabled servers have their host-CPU underutilized, but simultaneously their GPUs, their networking, and their I/O subsystems experience high utilization. System designers could leverage this empirical observation.

Performing the analysis exemplified in Figure 1 shows that there is ample load variability inside the LISA system. Yet, explaining it is complex. This load variability stems from load imbalances due to varying user-demand, occasionally poor scheduling decisions, and lacking load balancing over the entire set of servers. Only after understanding this complexity can we hope to tame the large design-space for resource management and scheduling decisions in modern datacenters. Moreover, Figure 1 depicts a recent trend we perceive in datacenter operations: GPU-servers are underutilized in terms of CPU load, so here it may be more cost-effective to equip the servers with less powerful (and cheaper) CPUs.

The SURF Archive

Datacenters already exhibit unprecedented scale and are becoming increasingly more complex. Moreover, such computer systems have begun having a significant impact on the environment: for example, training some machine learning models has sizable carbon footprints [2]. As our recent work on modern datacenter networks shows [3], low-level data is key to understanding full-stack operation, including high-level application behavior. We advocate it is time to start using such data more systematically, unlocking its potential in helping us understand how to make (datacenter) systems more efficient. We advocate that our data can contribute to a more holistic approach, looking at how the multitude of these systems work together in a large-scale datacenter.

SURFsara operates several systems inside their datacenter. In this archive, we release operational data from two of SURFsara's largest production clusters: LISA and Cartesius. The former is a 300+ server cluster containing more than 200 GPUs, interconnected with 40-Gbps and 10-Gbps networks. The latter is a 2000+ server cluster containing 132 GPUs and 18 Intel last-generation KNLs. The rest of the servers are a combination of thin (24 cores and 64 GB memory) and fat machines (32 cores and 256 GB memory). The total number of cores in Cartesius is roughly 47K, amounting to 1.8 PFLOPS double precision. Most servers are connected by an FDR InfiniBand network, ensuring 56 Gbps peak bandwidth, with a subset (18 Intel KNL and 177 Intel Broadwell) connected by an EDR InfiniBand network that enables 100 Gbps peak-bandwidth.

We gather metrics, at 15-second intervals, from several data sources:

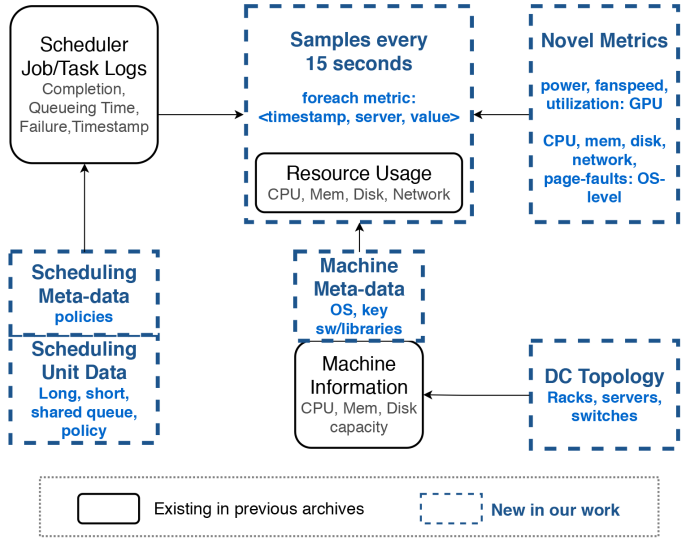

Figure 2: The schema for our collection of datacenter metrics. The figure highlights the novel components we propose, compared to state-of-the-art datacenter archives.

- **Slurm**: all job, task, and scheduler-related data, such as running time, queueing time, failures, servers involved in the execution, organization in partitions, and scheduling policies.
- **NVIDIA NVML**: per GPU, data such as power metrics, temperature, fan speed, or used memory.
- **IPMI**: per server, data such as power metrics and temperature.
- **OS-level**: from either `procfs`, `sockstat`, or `netstat` data: low-level OS metrics, regarding the state of each server, including CPU, disk, memory, network utilization, context switches, and interrupts.

We also release other kinds of novel information, related to datacenter topology and organization.

The audience we envision using these metrics is composed of systems researchers, infrastructure developers and designers, system administrators, and software developers for large-scale infrastructure. The frequency of collecting data is uniquely high for open-source data, which could allow these experts unprecedented views into the operation of a real datacenter.

Our traces will benefit multidisciplinary teams in building better schedulers, better co-locating workloads to improve resource utilization and minimize interference. Recently, systems experts started teaming up also with machine-learning experts to produce AI-enhanced systems such as learned database indexes (work done by Tim Kraska et al.). All these stakeholders could benefit from our many low-level server metrics, which uniquely complement scheduler logs. Uniquely, our traces could help experts to understand how specific workloads interact with the

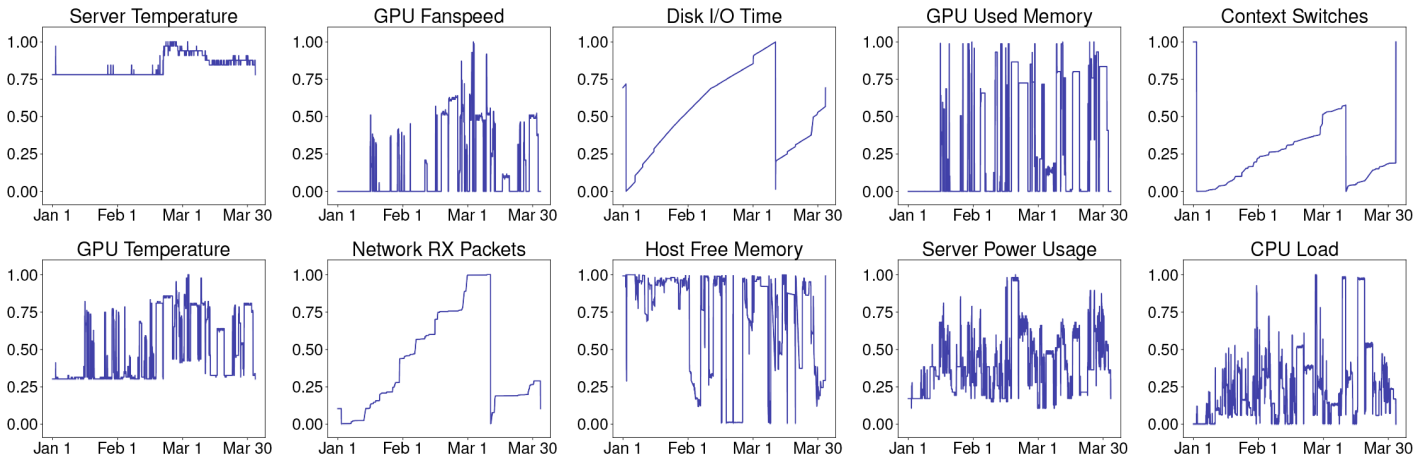

Figure 3: A dashboard to visualize 10 metrics for a single GPU-enabled server in the LISA datacenter. Each metric is normalized by the maximum value encountered during the three months recorded for this server. For all but the "Host Free Memory," higher means more loaded.

hardware, with each other, and where faults and performance issues originate.

Figure 2 presents a high-level view of the schema of the archive we propose. The structure resembles a snowflake schema, with the central *fact* table representing the low-level, high-resolution metrics we collect every 15 seconds from our datacenters. The *dimension* tables represent all other data that we can use to interpret and analyze the fact table. As SURFsara users run jobs, a data set of job-related metrics records scheduler logs (e.g., from Slurm). Simultaneously, many independent tools (e.g., Nvidia Management Layer (NVML)) gather data from each server and push them into the fact table. We keep a separate table containing the list of metrics we collect, enabling easy addition of metrics in the future. Moreover, we explicitly include in the data both server-level and topology information.

Our archive is online: https://doi.org/10.5281/zenodo.3878142.

What Our Archive Offers

There are many types of analyses one could do using the data we open source, such as the typical sysadmin dashboards exemplified by Figure 3. From utilization-level metrics, sysadmins can identify interesting points or correlations that could be examined in more detail, thus improving the daily operation of the datacenter. Using the data in this figure, one could easily correlate temperature increases with, for example, data received over networks, increase in I/O time, and context switches.

Other kinds of analyses are more complex, requiring data science techniques to delve deeper into possible meaning and correlations in our time-series data. Time series in datacenters often display sequential dependencies, meaning the value of a data point is statistically dependent on a previous one. One of the possible steps in analyzing time series is performing regression analysis, which assumes independence of observations. To ascertain the practical usefulness of our data, we perform some basic analytics.

We first investigate whether the time series is linearly correlated to a lagged version of itself, using the Pearson correlation for two independent variables, or, in time series terminology, *autocorrelation*. Figure 4 plots this autocorrelation to provide an insight into the possibility to reduce the amount of data [4]. We use the metric *Server Power Usage* averaged over the GPU-enabled nodes. The confidence interval, depicted in light gray/blue in the figure, lies between -0.2 and 0.2. The figure shows that high correlation values occur for small lags, which is reasonable considering the 15 second sampling frequency.

To further assess the usefulness of the collected metrics, we evaluate a first-order autoregression model, a parametric technique for fitting the observations. As Figure 5 depicts,

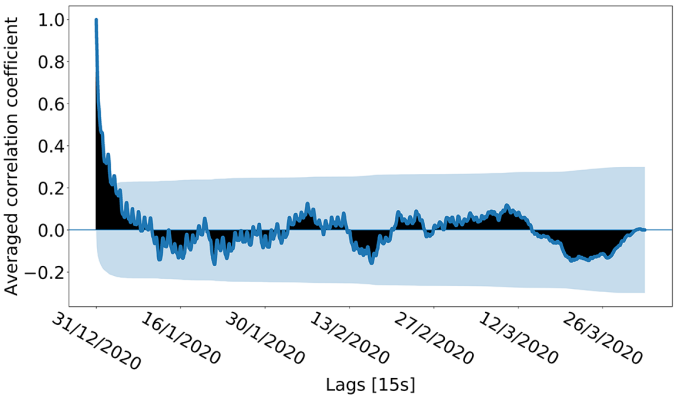

Figure 4: Pearson autocorrelation plot for the server power usage metric. Each point represents a period of 15 seconds. The light gray/blue shaded area represents confidence intervals. The horizontal axis shows 15 second lags, the vertical axis shows correlation values.

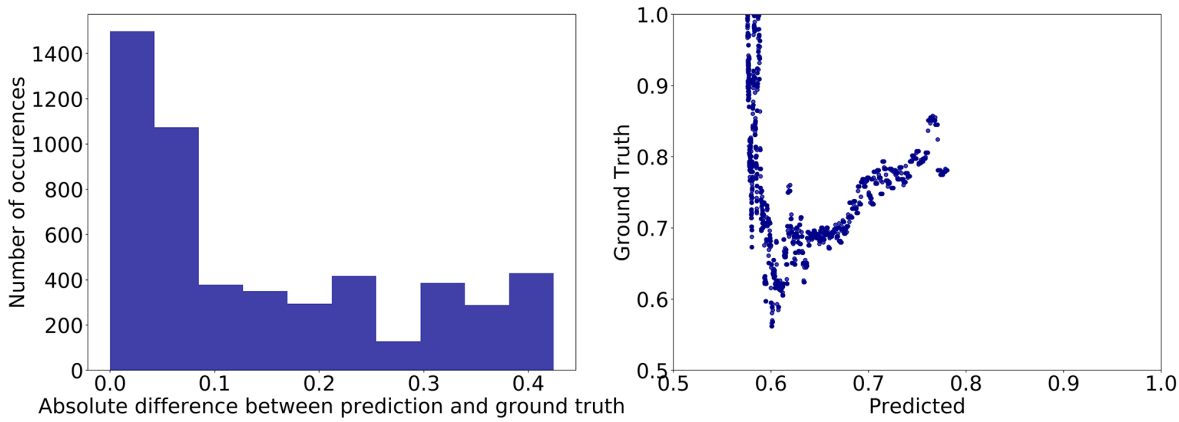

Figure 5: Autoregression histogram (left), predictions vs. ground truth values (right) for the Server Power Usage metric. On the right-hand plot we clip values to [0.5, 1.0], since there are no values below 0.5.

we measure the solution quality by computing the absolute distance between predictions and the ground truth. We chose 518,918 points for fitting the model and tested on 5,242 values, normalized between 0 and 1, by subtracting from each value the minimum and then dividing the result by the difference between maximum and minimum. We did no additional filtering. The autoregression histogram in Figure 5 (left) suggests a reasonable fit for the Server Power Usage metric. However, in Figure 5 (right), we see a possible overfitting behavior when scatterplotting the predictions against the ground truth. It seems that this simple technique is only capable of predicting the limited interval between 0.6 and 0.8, which is close to the normalized average (0.57), with the whole range being 0.17, 1 for this metric. This is an example of how data scientists could start analyzing our data. More in-depth analyses are certainly possible. We leave this for future work and invite others to run their analyses on the data we open-source.

Conclusion

Realistic assumptions are at the core of building and operating computer systems. Ideally, experts derive these assumptions from data gathered long-term from datacenters in the wild, with the finest of granularities and at the deepest levels of system information. Unfortunately for the computer systems community, only a few organizations currently have access to such data. Existing data sets and trace archives are bereft of such metrics, limiting their ability to support deeper insights.

We offer, as open-source and FAIR data, over 100 low-level metrics gathered at fine granularity from the largest public datacenter in the Netherlands, hosted by SURFsara. In this article, we gave examples and provided an initial analysis over a GPU-enabled cluster inside this datacenter. We showed there are large amounts of variability and imbalances, and correlations between several low-level metrics. Thus, there is value in performing data science analysis over our time-series data. We invite all researchers, practitioners, system designers, and datacenter operators to download and put to good use our open-source archive.

References

[1] R. A. Poldrack, D. M. Barch, J. Mitchell, T. Wager, A. D. Wagner, J. T. Devlin, C. Cumba, O. Koyejo, and M. Milham, "Toward Open Sharing of Task-Based fMRI Data: The OpenfMRI Project," *Frontiers in Neuroinformatics*, vol. 7, no. 12 (July 2013): https://dash.harvard.edu/bitstream/handle/1/11717560/3703526.pdf?sequence=1.

[2] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," arXiv, June 5, 2019: https://arxiv.org/pdf/1906.02243v1.pdf.

[3] A. Uta, A. Custura, D. Duplyakin, I. Jimenez, J. Rellermeyer, C. Maltzahn, R. Ricci, and A. Iosup, "Is Big Data Performance Reproducible in Modern Cloud Networks?" in *Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI '20)*, pp. 513–527: https://www.usenix.org/system/files/nsdi20-paper-uta.pdf.

[4] P. M. Broersen, *Automatic Autocorrelation and Spectral Analysis* (Springer Science & Business Media, 2006).

Appendix—On the Elusive Pursuit of Sharing Trace Archives

This work follows in the footsteps of major achievements. The importance of tracing was becoming apparent to the broad systems community at least since the mid-1960s, when instrumentation for collecting operational traces was made available as part of OS/360. By the early 1970s, the systems community was already discussing the importance of using real traces in performance engineering, and by the beginning of the 1990s this practice had already become commonplace.

Until the advent of the Internet, the sharing of traces seemed at best haphazard. The mid-1990s have witnessed the birth of trace archives, with the most prominent being the Internet Trace Archive (ITA, 1995). Focusing on the operation of the Internet, the ITA exhibits many modern features such as data collection and processing tools, and, most importantly, data shared with policies that today would be labeled as FAIR.

Established in the late 1990s, the Parallel Workloads Archive (PWA) [1] is perhaps the most successful example of how shared traces can help shape a community. The PWA started with just a few traces but a good format for sharing, and today it shares traces collected from about 35 environments, mostly from parallel production supercomputers and clusters, but also from research and production grids. Since the mid-2000s, sustained efforts have led to the creation of the Grid Workloads Archive [2] (2006), the Failure Trace Archive [3] (FTA, 2010), the Peer-to-Peer Trace Archive (2010), the Workflow Trace Archive [7] (2019), and the Computer Failure Data Repository, hosted by USENIX.

In the 2010s, the computing industry was transformed by the move to cloud services and by the advent of big data. Unsurprisingly, studies of how such systems operate have led to sharing of characteristics (notably, from Facebook, Yahoo, IBM, Taobao) and, rarely, of traces such as the multi-day trace from a large cluster at Google [4] or Microsoft [6]. Although sharing traces has been very useful for the community, the presence of only one or a few traces cannot account for the tremendous diversity of traces present "in the wild" as reported periodically by analytical studies [5].

References

[1] D. G. Feitelson, D. Tsafrir, and D. Krakov, "Experience with Using the Parallel Workloads Archive," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10 (October 2014), pp. 2967–2982: https://www.cse.huji.ac.il/~feit/papers/PWA12TR.pdf.

[2] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. Epema, "The Grid Workloads Archive," *Future Generation Computer Systems*, vol. 24, no. 7 (July 2008), pp. 672–686: https://doi.org/10.1016/j.future.2008.02.003.

[3] B. Javadi, D. Kondo, A. Iosup, and D. Epema, "The Failure Trace Archive: Enabling the Comparison of Failure Measurements and Models of Distributed Systems," *Journal of Parallel and Distributed Computing*, vol. 73, no. 8 (August 2013), pp. 1208–1223: https://doi.org/10.1016/j.jpdc.2013.04.002.

[4] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, "Towards Characterizing Cloud Backend Workloads: Insights from Google Compute Clusters," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 4 (March 2010), pp. 34–41: http://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/Appworkload.pdf.

[5] G. Amvrosiadis, J. W. Park, G. R. Ganger, G. A. Gibson, E. Baseman, and N. DeBardeleben, "On the Diversity of Cluster Workloads and Its Impact on Research Results," in *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC '18)*, pp. 533–546: https://www.usenix.org/conference/atc18/presentation/amvrosiadis.

[6] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms," in *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17)*, pp. 153–167: https://dl.acm.org/doi/pdf/10.1145/3132747.3132772.

[7] L. Versluis, R. Matha, S. Talluri, T. Hegeman, R. Prodan, E. Deelman, and A. Iosup, "The Workflow Trace Archive: Open-Access Data from Public and Private Computing Infrastructures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 9 (May 2020), pp. 2170–2184: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9066946.