EDITOR **DEJAN MILOJICIC**
Hewlett Packard Labs;
dejan.milojicic@hpe.com

The Distribution Is the Performance

Eitan Frachtenberg, Hewlett Packard Labs
Viyom Mittal, University of California, Riverside
Pedro Bruel, Hewlett Packard Labs
Michalis Faloutsos, University of California, Riverside
Dejan Milojicic, Hewlett Packard Labs

In this column, we suggest how to sharpen our understanding of computer performance evaluation in light of variability and heterogeneity.

Since the dawn of computing, the established practice for performance evaluation has been to represent performance as a variable for some metric and measure or estimate it as accurately as possible. However, computer hardware and software have evolved in ways that render their performance a constantly varying value, which means that any single performance summary may no longer be representative of the system. Instead, performance is now often behaving like a complex random variable, and should therefore be handled as such, with the appropriate statistical tools.

Obviously, there is nothing novel about the requirement to handle complex distributions with appropriate statistical tools. For example, biologists, psychologists, clinical researchers, and social scientists have been managing uncertainty in their experiments and observations for decades. What's changed is that computer performance has evolved from quantities that were relatively easy to measure, model, and reproduce, to complex random variables with ostensibly nondeterministic or irreproducible behaviors. In this column, we review the confluence of recent factors that are changing performance evaluation, illustrate why the "old ways" of performance evaluation are no longer good enough, describe the evolution of performance evaluation in reaction to these changes, and predict some possible directions for the future of performance evaluation.

SOURCES OF PERFORMANCE VARIABILITY

There are many reasons why the performance of modern computer systems is growing ever more variable. Let us

Digital Object Identifier 10.1109/MC.2024.3362448
Date of current version: 5 April 2024

briefly review some of them, going from the bottom up.

First, modern computer hardware has become extremely complex,[3] with optimizations such as multicore processors, speculative execution, caching, dynamic voltage and frequency control, symmetric multithreading, and branch prediction, itself often unpredictable and proprietary.[5] Hardware is also growing more heterogeneous, both within chips and across chips, using accelerators such as GPUs. This heterogeneity combines different architectures, leading to multiple performance profiles that depend on the dynamic balance between the components.

System software and middleware also contribute to performance variability. The BIOS includes idiosyncratic optimizations and power-savings tradeoffs; the operating system can make seemingly arbitrary scheduling decisions in space and time with significant performance effects; and even the language runtime can appear spurious when making adaptive decisions, such as garbage collection and just-in-time optimizations.[4,12]

Similarly, the application software's performance can vary by factors such as changes in inputs, configuration parameters, and randomized heuristics and simulations.[14] In particular, many applications now rely on concurrency to satisfy their growing computational needs. The increasing use of parallel, distributed, and more recently, cloud and serverless design,[1] means that applications now include components interacting across multiple networks, which leads to an explosion in the number of possible execution orderings.[6] Such race conditions can be benign from a correctness perspective, but have a large impact on performance variability.

Even external factors affect performance. Environmental elements, which may be outside of a user's control, can still have a significant impact on observed performance, such as transient load on the system from interfering applications and daemons, network congestion, and even datacenter temperature.

THE PROBLEM WITH PERFORMANCE SUMMARIES

Nuance is always lost when we attempt to associate a single number with a complex phenomenon, such as the performance of a computer experiment. If we are not careful, we might fall into one of the many statistical traps involved in analyzing data and end up with a meaningless number, or we may misrepresent the underlying variance of what is measured. We might even end up reaching the opposite conclusion we would if, for example, we looked at experimental results using a histogram instead of relying on a summary.

Take, for example, the famous performance report on the fastest computers in the world, the biannual TOP500 list.[13] The list reports and ranks performance on a single benchmark using just a handful of peak metrics. Aside from the fact that the list ignores many aspects relevant to usability—such as availability, reliability, and cooling requirements—the list only presents the performance characteristics of Linpack and ignores all of the other applications for which these computers were purpose-built. The report also fails to share details on expected performance, performance scalability, the number of runs it took to measure the optimal performance, or the variability of performance across hardware, middleware, and software parameters. Like the parable of the six blind men and the elephant, trying to describe a supercomputer's performance by concentrating on Linpack's Rpeak and Rmax alone offers a very incomplete picture. Such problems in performance evaluation description are not new and there exists a rich literature on the topic.[8,9,10]

Even when the experimenter is extremely careful, it's not easy to achieve complete isolation and independence between measurements. This is as true for physics, biology, and psychology as it is for computer experiments. A host of mischievously subtle effects can muddle measurements in each field of experimentation. These effects can cause a sequence of measurements to deviate from an independent and identically distributed (*iid*) sample, which is a formal requirement in some of the most used statistical methods. These deviations from the *iid* hypothesis are often small and don't interfere with analyses, but it is always good practice to check how far results deviate before committing to a statistical method.

THE EVOLUTION OF PERFORMANCE EVALUATION

Our key thesis is that we need to adopt: 1) a reproducibility-first and 2) distribution-focused approach to performance evaluation. To elaborate, we next discuss the challenges stemming from relying on point-metrics and performance summaries and the design principles for a modern approach that will enable reproducible and reliable performance evaluations.

Our prediction and position is that performance evaluation needs to adopt a reproducibility-first approach based on distributions rather than summaries. Performance summaries still have important uses, such as describing the expected (mean) performance when it is the most germane property of a system, but as we have just discussed, they can be opaque and even misleading. Since performance summaries, such as "average response time" or "maximum bandwidth," are increasingly less informative, we need to redefine performance in a way that captures the rich behavior of complex computer systems. Performance variability now becomes an object of interest, in addition to the summary.

As such, we need to shift our focus from a single number or two to the entire distribution. In other words, the distribution is the performance. The goal of a performance evaluation should be to fully characterize the performance distribution. This goal in turn spawns four new subgoals: 1) capturing the complete distribution efficiently and accurately, 2) communicating it effectively, 3) reproducing performance distributions, and 4) modeling and predicting

performance variability as an integral part of performance prediction.

1) Capturing the complete distribution may sound like a simple enough problem. Just repeat the experiment or run it long enough to identify all of the salient properties of the distribution: its modes, tails, outliers, and of course, stable summary statistics. But in practice, we are always subject to resource and time constraints and may not be able to run an experiment until we are certain to characterize the distribution correctly. And even if we could afford to, who's to say what is the correct duration or sample size for this characterization? Some real-world performance distributions are regular enough to capture with a few samples, while others may require hundreds (see, for example, Figure 1). This is a complex research question that remains largely open in the general case, although there are several strategies to stop an experiment when the properties of the distribution are fairly well understood.[11]

2) Communicating a distribution effectively is also harder than communicating a single number or two. There are three aspects to this challenge. First, humans are better equipped to visualize a distribution in two or three dimensions than as a table of numbers, which requires appropriate and informative graphics (as well as more "bandwidth" on the communicating document, such as page space or disk space). Examples of such distribution visualizations include density or violin plots, box plots, scatter plots, or combinations, such as raincloud plots (as depicted in Figure 2). Second, communicating distributions effectively also puts the onus on the writer to apply correct and relevant statistical analyses to the distribution, such as hypothesis tests, uncertainty quantification, Bayesian inference, and others, as appropriate. Such analyses sometimes require more than the basic statistical tools taught in undergraduate computer science, which in turn may percolate new requirements to future curricula. And lastly, performance distributions should also be communicated as complete datasets in digital form, so that readers may apply their own visualizations, statistical analyses, or conclusions.

The data availability aspect is inexorably connected to the increasing role of openness in science. There is already a large and growing trend of requiring the sharing of research artifacts, including software and its full configuration/environment. If we start treating performance as a distribution, then

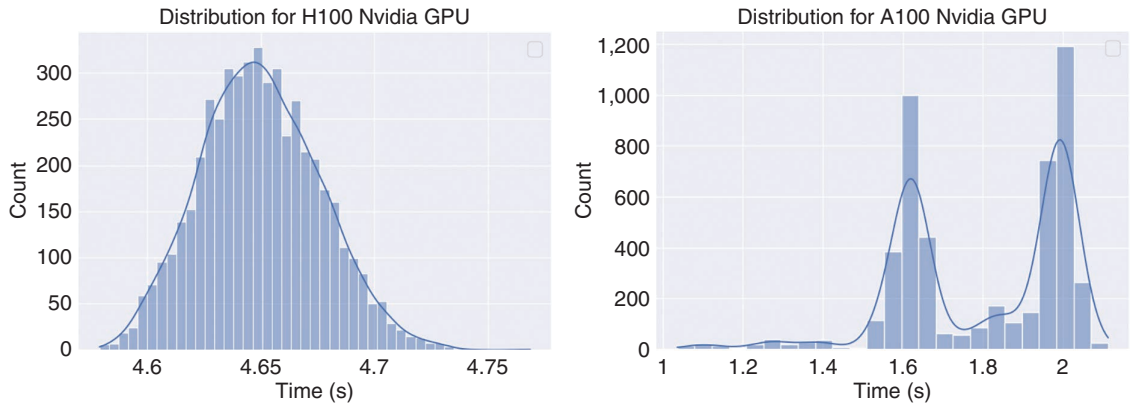

FIGURE 1. Plots for the "hotspot" benchmark from the Rodinia benchmark suite. The A100 distribution is more irregular and requires more samples to characterize accurately.[2]

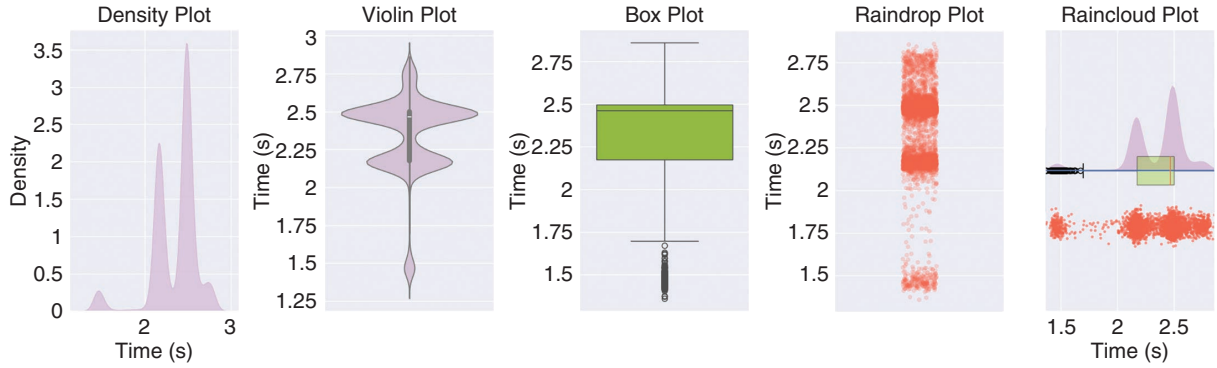

FIGURE 2. Plots for lud benchmark from Rodinia suite.

it must also be fully shared as an artifact: not only the code to generate it, but also the code to visualize and analyze it. The old formats of performance reporting (summary statistics or static graphs in PDF files) are grounded in years of analog tradition but are insufficient for full reproducibility. We should rethink performance data as fully digital, focusing on reproducibility, interactivity and exploration, archival and versioning, and future-compatible reporting standards.

3) This aspect also naturally segues into the third goal, the requirement to reproduce performance distributions. All performance evaluations should be reproducible, or their results will be suspect. Reproducibility extends across three dimensions: evaluation conditions, performance results, and interpretation. When the goal of the evaluation is a distribution, rather than a number, reproducibility raises questions, such as: how do you reproduce a distribution? and when are two distributions equal or close enough? Again, answering these questions requires a more rigorous statistical approach to performance evaluation than current practices.

4) Last, we should look not only at measuring performance as a distribution, but also at modeling and predicting performance distributions. Performance models are extremely useful to answering "what if?" questions when benchmarking is impractical, such as which future hardware to invest in or what aspects of a system can benefit most from optimizations.[12] Such questions can and should still be addressed when performance is treated as a distribution. This includes both descriptive models, which means the performance model needs to allow for performance variability (for example, outliers, modes, and tails), and predictive models, which estimate a range for performance figures with well-quantified uncertainty.

THE ROAD AHEAD

Figure 3 illustrates our view on the current standard and practice in performance evaluation and modeling and on the way forward: a methodology based on distribution estimates. While the current practice often assumes *iid* and normality even before experiments are conducted, we propose to start with less restrictive hypotheses and adapt as we collect data. This leaves room for experimental designs that can help uncover nonlinearities, autocorrelations, and interactions between parameters that affect performance. Using distribution estimates will also improve other steps of the performance evaluation and modeling process, contributing to better measurement, modeling and analysis, inference, and reporting.

Better measurement

When relying on the *iid* hypothesis, we can stop experimenting when the number of samples collected is sufficient to estimate the parameters of the normal distribution; that is, its mean μ and standard deviation σ. If we start without strong hypotheses, we must use other stopping criteria. For example, we can compute a bootstrapped metric of sample self-similarity, according to some divergence metric, like Kolmogorov–Smirnov, and stop measurements when the sample "no longer changes" according to our metric. We can also combine different stopping criteria based on different hypotheses, such as confidence intervals and Gaussian mixture models, and stop when one of them converges. These strategies can accommodate complex behaviors—such as multimodal and skewed distributions, autocorrelated samples, and outliers—and can guide the choice of modeling and analysis methods.

Better modeling and analysis

After having measured performance expecting a normal distribution, it is natural to rely on point estimates of its parameters. Considering that deviations from the normal are not so rare, more flexibility in modeling and analysis helps register and keep track of different properties of the measurements. Models, such as the Gaussian mixture, try to fit several Gaussian (normal) distributions to the data, typically marking each of the modes. In this case, we would be able to report multiple means $\hat{\mu}$ and multiple standard deviations $\hat{\sigma}$. Other models can provide different ways of quantifying the uncertainty of a set of measurements. Probabilistic Bayesian models,[7] such as Gaussian processes, can incorporate previous experiments directly into new ones in the form of prior assumptions that can guide experimental design and analysis.

Better inference

Inferences that we can make from measurements are also limited when all we have is the mean and standard deviation. If instead of the average or expected performance we need to minimize the worst case, we need to be able to detect and model it, and point estimates, even as percentiles (for example, p99) still fail to capture the richness of ways in which performance goes wrong. More flexible distribution-based modeling strategies also help quantify the uncertainty associated with changing parameters that impact performance.

Better reporting

If we focus on improving our data and result-reporting practices, we can ultimately choose which methods for measurement, modeling, analysis, inference, and experimental design we wish to apply to a specific performance evaluation problem. Reporting distribution estimates and complete datasets with measurements can ensure that other researchers can reproduce our processes, results, and inferences.

Table 1 summarizes some of the challenges we have listed and the opportunities for improvements we have discussed so far.

BROADENING ADOPTION

There are many opportunities for broadening the adoption of our proposed approach. Here, we focus on workflows, edge-to-cloud, new hardware, and integration with other tools.

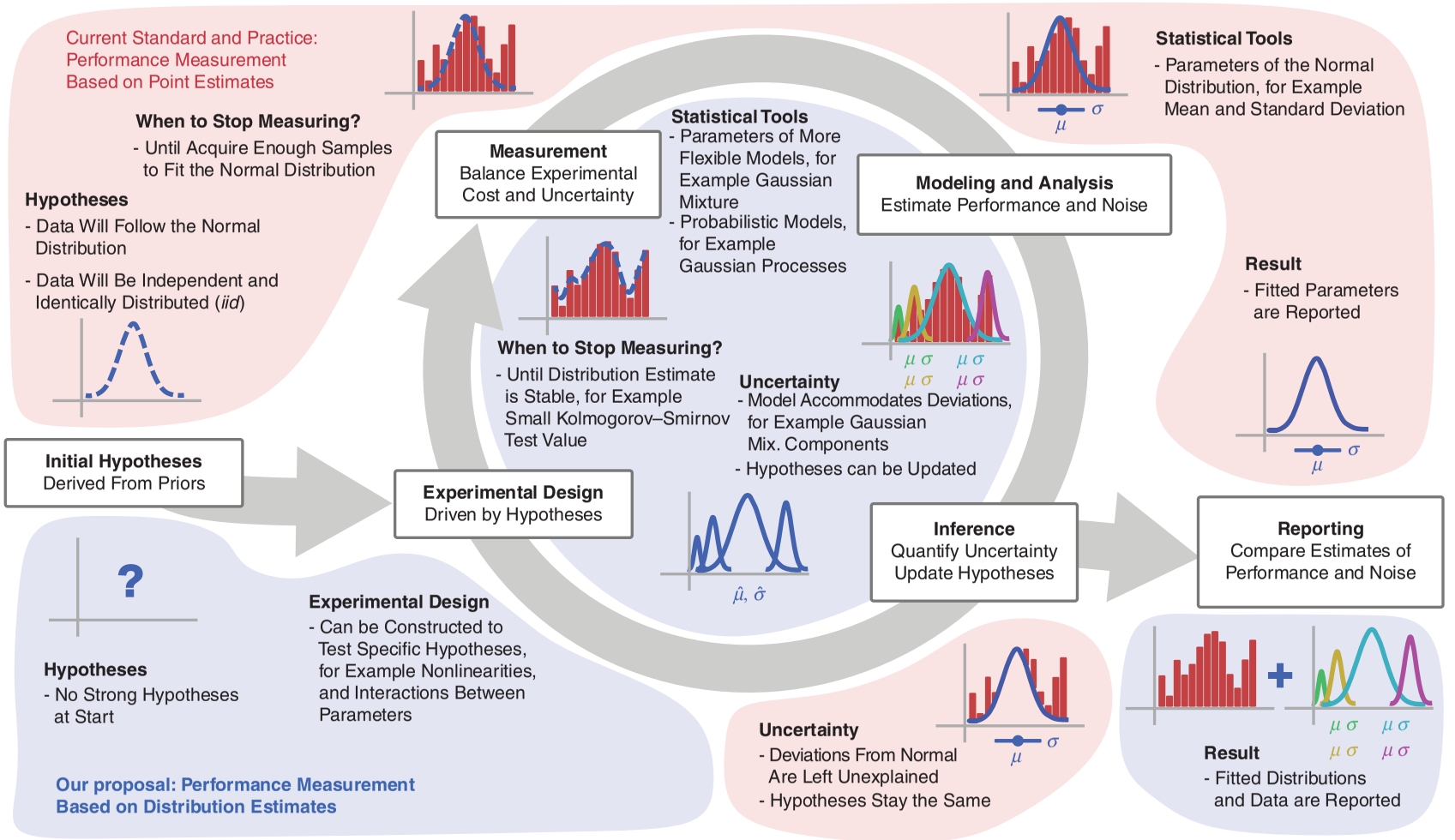

FIGURE 3. The effects of shifting from point summaries (red) to distributions (blue) on performance-evaluation life cycle.

TABLE 1. Challenges and opportunities in modern performance evaluation.

Task	Experimentation	Modeling	Simulation	Prediction
Challenges	Hardware complexity: multithreading, speculative execution, branch prediction, caching, dynamic voltage/frequency, ... System software and middleware: scheduling, background daemons, garbage collection, just-in-time compilation, ... Application variability: concurrency, configuration, workload, randomized algorithms, ...			
	Cost Completion time Hardware availability	Level of detail Deployment variability Hardware opacity	Quantifying uncertainty Validating simulation Hardware familiarity	Larger design space Lower accuracy Evolving hardware and software
Opportunities	Adaptive stopping rules Distribution analysis	Stochastic models Markov chains	Simulate sources of variability Add noise to simulation	Predict distribution parameters Predict modes, outliers, tails

As high-performance computing and artificial intelligence (AI) converge on using workflows, so should performance evaluation raise the target abstractions from workloads to workflows. With prescribed sequentiality and parallelism, workflows offer more opportunities and create more challenges. Understanding which workloads come next in the workflow helps benchmarking to better prepare for and guess distributions. However, parallelism also introduces contention on buses, interconnects, and networks that are nontrivial to account for, magnifying the need for a distribution-focused approach.

Emerging workflows encompass edge-to-cloud (or datacenter) applications, with dataflow throughput as a primary objective. Consequently, performance distributions will have to account for varying inputs, various data, and different seasonality.

The new hardware: accelerators, such as GPUs, SmartNICs, FPGAs, etc., cause additional behaviors and performance distributions. Interconnects are enhanced with switches and SmartNICs that address scalability through managing traffic contention. Understanding new hardware and how it supports emerging AI applications will be essential. This will require characterizing distributed applications at a very large scale.

Integration with other tools, such as performance prediction, schedulers, and simulators, could effectively turn performance evaluation into an integrated benchmarking environment. This in turn will improve DevOps and increase productivity. Integration with simulators and performance prediction would enable predicting the performance of new hardware.

In summary, this column advocates for a major shift in how we evaluate computer performance, moving from simple, fixed measurements to a more complex, varied approach. It highlights the need to view performance as a range of possible outcomes rather than just a single number. This change is important not only for how we understand computer performance but also for how we teach and study it. The ideas presented here also apply to other areas of computer systems, like energy use and system reliability. We urge everyone in the field—from researchers, through practitioners, to educators—to adapt their methods to this distribution-first perspective.

REFERENCES

1. P. Bruel et al., "Predicting heterogeneity and serverless principles of converged high-performance computing, artificial intelligence, and workflows," *Computer*, vol. 57, no. 1, pp. 136–144, Jan. 2024, doi: 10.1109/MC.2023.3332973.
2. S. Che et al., "Rodinia: A benchmark suite for heterogeneous computing," in *Proc. Int. Symp. Workload Characterization (IISWC)*, Oct. 2009, pp. 44–54, doi: 10.1109/IISWC.2009.5306797.
3. A. Danalis et al., "The scalable heterogeneous computing (SHOC) benchmark suite," in *Proc. 3rd Workshop General-Purpose Comput. Graph. Process. Units (GPGPU)*, Mar. 2010, pp. 63–74, doi: 10.1145/1735688.1735702.
4. C. Delimitrou and C. Kozyrakis, "Paragon: Qos-aware scheduling for heterogeneous datacenters," *SIGPLAN Notices*, vol. 48, no. 4, pp. 77–88, Apr. 2013, doi: 10.1145/2499368.2451125.
5. C. Delimitrou and C. Kozyrakis, "Quasar: Resource-efficient and qos-aware cluster management," *SIGPLAN Notices*, vol. 49, no. 4, pp. 127–144, Apr. 2014, doi: 10.1145/2644865.2541941.
6. S. Eismann, C.-P. Bezemer, W. Shang, D. Okanović, and A. van Hoorn, "Microservices: A performance tester's dream or nightmare?" in *Proc. ACM/SPEC Int. Conf. Perform. Eng. (ICPE)*, New York, NY, USA: ACM, Apr. 2020, pp. 138–149, doi: 10.1145/3358960.3379124.
7. A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*. London, U.K.: Chapman & Hall, 2013.
8. L. Gonnord, L. Henrio, L. Morel, and G. Radanne, "A survey on parallelism and determinism," *ACM Comput. Surv.*, vol. 55, no. 10, pp. 1–28, 2023, doi: 10.1145/3564529.

9. S. Hunold, "A survey on reproducibility in parallel computing," Nov. 2015, *arXiv:1511.04217*.
10. N. Ihde et al., "A survey of big data, high performance computing, and machine learning benchmarks," in *Proc. Perform. Eval. Benchmarking, 13th TPC Technol. Conf. (TPCTC)*, Copenhagen, Denmark. Cham, Switzerland: Springer-Verlag, Jan. 2022, pp. 98–118, doi: 10.1007/978-3-030-94437-7_7.
11. V. Mittal, P. Bruel, D. Milojicic, and E. Frachtenberg, "Adaptive stopping rule for performance measurements," in *Proc. 14th IEEE Int. Workshop Perform. Model., Benchmarking Simul. High Perform. Comput. Syst.*, Denver, CO, USA, 2023, pp. 1288–1297, doi: 10.1145/3624062.3624202.
12. A. Nassereldine et al., "Predicting the performance-cost trade-off of applications across multiple systems," in *Proc. IEEE/ACM 23rd Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2023, pp. 216–228, doi: 10.1145/3624062.3624202.
13. E. Strohmaier, H. W. Meuer, J. Dongarra, and H. D. Simon, "The top500 list and progress in high-performance computing," *Computer*, vol. 48, no. 11, pp. 42–49, Nov. 2015, doi: 10.1109/MC.2015.338.
14. Y. Wang et al., "A study of database performance sensitivity to experiment settings," *Proc. VLDB Endowment*, vol. 15, no. 7, pp. 1439–1452, 2022, doi: 10.14778/3523210.3523221.

EITAN FRACHTENBERG is a researcher at Hewlett Packard Labs, Milpitas, CA 95035 USA. Contact him at eitan.frachtenberg@hpe.com.

VIYOM MITTAL is a graduate student at University of California, Riverside, Riverside, CA 92521 USA. Contact him at viyom.mittal@hpe.com.

PEDRO BRUEL is a researcher at Hewlett Packard Labs, Milpitas, CA 95035 USA. Contact him at bruel@hpe.com.

MICHALIS FALOUTSOS is a professor of computer science at University of California, Riverside, Riverside, CA 92521 USA. Contact him at michalis@cs.ucr.edu.

DEJAN MILOJICIC is a Hewlett Packard Enterprise Fellow and vice president at Hewlett Packard Labs, Milpitas, CA 95035 USA. Contact him at dejan.milojicic@hpe.com.

Digital Object Identifier 10.1109/MC.2024.3371838