POSTER: BenchCPU: Evaluatology-Based CPU Benchmarking

Chenxi Wang
Yuchen Su
Lei Wang
wangchenxi21s@ict.ac.cn
suyuchen24s@ict.ac.cn
wanglei_2011@ict.ac.cn
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of Sciences
Beijing, China

Guoxin Kang
Wanling Gao
Fan Zhang
Jianfeng Zhan✉
kangguoxin@ict.ac.cn
gaowanling@ict.ac.cn
zhangfan@ict.ac.cn
zhanjianfeng@ict.ac.cn
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
The International Open Benchmark Council
Hong Kong, China

Abstract

This article proposes a rigorous CPU evaluation methodology based on Evaluatology, a universal and rigorous evaluation theory. We interpret the computer system as a triad: the CPU serves as the Evaluated Object (EO); the testable system incorporating the CPU constitutes the Affected Object (AO); and other confounding components that impact the AO's quantities are designated as Confounding Objects (COs). From this perspective, traditional benchmarks function as COs, and evaluating a CPU entails inferring its true effects on the varying AOs under diverse CO configurations.

We design and implement BenchCPU, a novel CPU benchmark suite. BenchCPU mandates rigorously defining the AOs while exploring CO configuration distributions. Our analysis shows that SPEC CPU has inherent limitations because it artificially constrains COs to static empirical configurations and leaves AOs undefined. BenchCPU is open-sourced at https://benchcpu.benchcouncil.org.

CCS Concepts

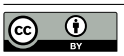
• **General and reference → Performance**; **Evaluation**; **Measurement**.

Keywords

CPU Benchmarking, Evaluatology

ACM Reference Format:

Chenxi Wang, Yuchen Su, Lei Wang, Guoxin Kang, Wanling Gao, Fan Zhang, and Jianfeng Zhan. 2026. POSTER: BenchCPU: Evaluatology-Based CPU Benchmarking. In *International Conference on Parallel Architectures and Compilation Techniques (PACT '26), October 19–22, 2026, Chicago, IL, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3838684.3846834



This work is licensed under a Creative Commons Attribution 4.0 International License.
PACT '26, Chicago, IL, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2915-7/2026/10
https://doi.org/10.1145/3838684.3846834

1 Introduction & Motivation

Evaluatology [18–20] pioneers a universal, rigorous framework for formalizing evaluation across disciplines. A core tenet of this discipline is ensuring that evaluation outcomes are strictly traceable to an explicit evaluation model. Under this paradigm, variations in evaluation outcomes are directly attributable to discrepancies between models, thereby necessitating rigorous testing to confirm the validity of the models themselves.

Rigorous CPU evaluation underpins modern processor design. From the lens of Evaluatology, a computer system could be interpreted as a triad as shown in Figure 1: a specific CPU serves as the evaluated object (EO), and for evaluation it has to be put into a computer system (the affected object, AO), which can be directly tested to obtain quantities like the wall-clock execution time. Benchmark applications and their parameters (workload, dataset, concurrency) inherently act as **Confounding Objects (COs)**, which impact the AO's quantities. Evaluating a CPU entails inferring its true effects on the varying AOs under diverse CO configurations.:

For decades, industry-standard benchmark suites, notably SPEC CPU [9–16], are the golden standard of CPU evaluation and have established a *static workload configuration* paradigm [2, 3, 8]—pairing slowly updated workloads with fixed dataset and specific concurrency configuration [4, 5]. However, guided by Evaluatology [18, 19], we recognize a fundamental flaw in this paradigm: because traditional CPU benchmarks, like SPEC CPU, statically lock these COs to static empirical configurations and leave the AO undefined, they fail to control confounding biases in CPU benchmarking. Instead of inferring the true CPU effect, they report overall system performance. This directly leads to three methodological crises:

(1) Rank Inversions: Vendors can easily over-optimize for static COs in their specific AOs. Our profiling reveals that highly common real-world configurations can yield completely contradictory CPU rankings compared to the static SPEC-recommended configuration (SPEC-R).

(2) Biased Evaluation outcomes: Because COs are artificially fixed and AO is undefined, SPEC-R frequently manifests as a statistically unrepresentative outlier rather than reflecting the CPU's true performance distribution.

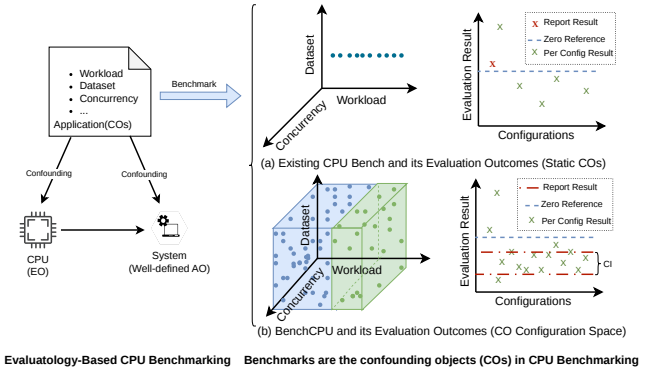


Figure 1: Evaluatology-based BenchCPU methodology.

(3) Architectural Blind Spots: By forcing CPUs down an empirical microarchitectural configuration (e.g., DRAM bandwidth-bound), static COs and undefined AO will obscure the true capabilities of modern processors with overstated or understated CPU performance scores.

2 BenchCPU Methodology

Decoupled Open CO Space: BenchCPU dismantles monolithic benchmarks by decoupling them into three orthogonal, dynamically explorable dimensions: workloads, data sets, and concurrency. *Workloads:* Selected via a rigorous pipeline, the workloads are partitioned into *fundamental Workloads* (stable baseline, updated decennially) and *emerging Workloads* (tracking cutting-edge demands, updated annually). *Datasets & Concurrency:* Datasets are scaled continuously, and concurrency is opened to a continuous interval (e.g., $[1, 128]$) to prevent the bias of blindly matching hardware thread counts. Now, BenchCPU has 54 workloads with a configuration space of 1,231.4 K, which effectively prevents over-optimization. Details are in https://benchcpu.benchcouncil.org.

Well-defined AO: According to the principle of Evaluatology, without specifying the AO, it is impossible to infer the true effect of an EO. BenchCPU rigorously defines what an AO is. When performing CPU evaluation, BenchCPU recorded execution logs in detail (e.g., OS, memory, compiler) and strictly groups evaluation outcomes according to the configurations of AO.

Cost Reduction using Sampling-Based CPU Benchmarks with the Weighted Filter: Exhaustively traversing a million-scale configuration space is computationally intractable. Instead, BenchCPU navigates it by treating performance evaluation as a *stochastic inference process*. We employ stratified random sampling[7, 17] to select a 50-configuration subset as the monthly released version. For reporting a CPU evaluation, the user or vendor needs to test the CPU under 12 released subsets (in total, 600 configurations). To maintain longitudinal consistency across annual emerging workload updates, a *filter* heavily weights (e.g., 0.8) the fundamental workloads in each version. Under the Central Limit Theorem (CLT) [1, 6], this anchors the population mean and ensures the performance distribution does not drift violently, maintaining strict backward compatibility.

Inferring true effect of CPU from the AOs' observable quantities: The execution time is the observable quantity of the AO, and we use it as the performance score, which enables a direct,

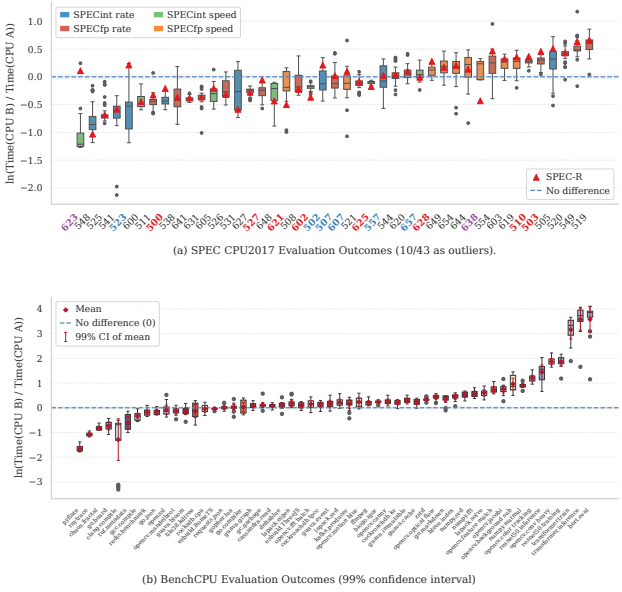


Figure 2: Cross-architecture ranking (CPU B vs. CPU A) using BenchCPU vs. SPEC CPU.

absolute comparison of performance scores without resorting to a reference machine. *Effect on a Specific AO:* For a pair of EO and AO, BenchCPU evaluates CPUs across the distribution of sampled COs. To aggregate scores properly, we calculate the mean of *log-transformed execution time differences* between AOs (B vs. A), mathematically equivalent to the log of their geometric mean ratio: $\mu = \frac{1}{n}\sum(\ln T_{B,i} - \ln T_{A,i}) = \ln(\sqrt[n]{\prod(T_{B,i}/T_{A,i})})$. By reporting the **99% Confidence Interval (CI)** of μ, we perform a rigorous hypothesis test: if the CI excludes 0, a statistically significant true CPU effect exists under this specific AO. *Effect across Varying AOs:* By further providing and comparing these performance distributions across different AO groups, BenchCPU systematically infers the true CPU effect.

3 Evaluation

SPEC CPU2017 (Confounded Aggregated Score): Exploring 32 sampled configurations per SPEC workload (Figure 2 (a)) reveals that the SPEC recommended configuration (SPEC-R) outcomes frequently (10/43) reside at the extreme tails of performance distributions. Because it statically selects COs, the reported score is heavily confounded. For workloads straddling the zero-line (28/43), SPEC-R arbitrarily picks a side, causing rank inversions.

BenchCPU (True CPU Effect via CI): BenchCPU (Figure 2 (b)) reports the performance distribution. By exploring the COs configuration space through sampling, it captures the comprehensive performance distribution. If the **99% CI** strictly resides above or below the zero-line, it proves the performance difference is statistically significant.

Acknowledgments

This work was supported by the Innovation Funding of ICT, CAS, China under Grant No. E461070.

References

[1] Ronald Aylmer Fisher. 1970. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*. Springer, New York, NY, USA, 66–70.

[2] John L Hennessy and David A Patterson. 2019. A new golden age for computer architecture. *Commun. ACM* 62, 2 (2019), 48–60.

[3] John L. Hennessy and David A. Patterson. 2011. *Computer Architecture, Fifth Edition: A Quantitative Approach* (5th ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[4] Kenneth Hoste and Lieven Eeckhout. 2006. Comparing Benchmarks Using Key Microarchitecture-Independent Characteristics. In *Proceedings of the IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, San Jose, CA, USA, 83–92.

[5] Kenneth Hoste, Lieven Eeckhout, and Hendrik Blockeel. 2007. Analyzing commercial processor performance numbers for predicting performance of applications of interest. *ACM SIGMETRICS Performance Evaluation Review* 35, 1 (2007), 375–376.

[6] Raj Jain. 1991. *The art of computer systems performance analysis*. Vol. 182. John Wiley & Sons Chichester, Chichester, UK.

[7] David S Moore and Stephane Kirkland. 2007. *The basic practice of statistics*. Vol. 2. WH Freeman New York, New York, NY, USA.

[8] David A. Patterson and John L. Hennessy. 2009. *Computer organization and design: the hardware/software interface* (4th ed.). Morgan Kaufmann, San Francisco, CA, USA.

[9] SPEC. 1989. SPEC CPU89. Available at https://www.spec.org/cpu89.

[10] SPEC. 1992. SPEC CPU92. Available at https://www.spec.org/cpu92.

[11] SPEC. 1995. SPEC CPU95. Available at https://www.spec.org/cpu95.

[12] SPEC. 2000. SPEC CPU2000. Available at https://www.spec.org/cpu2000.

[13] SPEC. 2006. SPEC CPU2006. Available at https://www.spec.org/cpu2006.

[14] SPEC. 2017. SPEC CPU2017. Available at https://www.spec.org/cpu2017.

[15] SPEC. 2026. SPEC CPU. Available at https://www.spec.org/cpu.

[16] SPEC. 2026. SPEC CPU2026. Available at https://www.spec.org/cpu2026.

[17] Chenxi Wang, Lei Wang, Wanling Gao, Fanda Fan, Guoxin Kang, Hongxiao Li, Yuchen Su, and Jianfeng Zhan. 2026. Inference of Component Effect on System Performance. *arXiv preprint arXiv:2605.26643* (2026).

[18] Jianfeng Zhan, Lei Wang, Wanling Gao, Hongxiao Li, Chenxi Wang, Yunyou Huang, Yatao Li, Zhengxin Yang, Guoxin Kang, Chunjie Luo, Hainan Ye, Shaopeng Dai, and Zhifei Zhang. 2024. Evaluatology: The science and engineering of evaluation. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations* 4, 1 (2024), 100162.

[19] Jianfeng Zhan, Lei Wang, Wanling Gao, Chenxi Wang, Hongxiao Li, Fanda Fan, and Guoxin Kang. 2025. *Evaluatology: The Science of Uncovering the Effects* (version 1.0 (first edition, 2025) ed.). BenchCouncil Press, Hong Kong.

[20] Jianfeng Zhan, Lei Wang, Wanling Gao, Chenxi Wang, Hongxiao Li, Fanda Fan, and Guoxin Kang. 2026. *Evaluatology: The Science of Uncovering the Causes and Effects*. BenchCouncil Press, Hong Kong, China. doi:10.66834/33wi1xx8