# On Studying CPU Performance of CloudLab Hardware

Dmitry Duplyakin*, Alexandru Uta†, Aleksander Maricq*, Robert Ricci*
*University of Utah, †Vrije Universiteit Amsterdam
*{dmdu, amaricq, ricci}@cs.utah.edu, †a.uta@vu.nl

## I. Introduction

Empirical performance measurements of computer systems almost always exhibit variability and anomalies. Run-to-run and server-to-server variations are common for CPU, memory, disk, and network performance characteristics. In our previous work, we focused on taming performance variability for memory, disk, and network [1] and established an interactive analysis service at: `https://confirm.fyi/` to help users of the CloudLab testbed [2] better plan and conduct their experiments. In this paper, we describe our analysis of CPU variability based on over 1.3M performance measurements from nearly 1,800 servers and present our initial findings.

The focus of this work is on capturing hardware variability, which can make repeatable experiments more difficult and can impact conclusions; it it this important for systems researchers to understand. (We note that, though we do not study it in this work, in the cloud, multi-tenancy and resource sharing [3] can exacerbate the problem.) Variability also inevitably impacts performance and operation of middleware and high-level applications, contributing to the straggler problems in many domains, including HPC, Big Data, and Machine Learning, and on many types of cyberinfrastructures. We analyze the data from the CloudLab servers allocated in an exclusive fashion, with no virtualization. While our analysis focuses on the testbed that aims to promote reproducible research, we believe our approach and the findings can be of value to people who manage, analyze, and utilize shared computing resources in supercomputers, clouds, and datacenters.

## II. Performance Data and Analysis

Starting on August 15, 2018, we have been measuring CPU performance of CloudLab servers using NAS Parallel Benchmarks [4]. We ran 9 microbenchmarks (BT, CG, EP, FT, IS, LU, MG, SP, UA) on homogeneous pools of servers of 12 types, turning on/off dynamic voltage and frequency scaling (DVFS). We varied the number of running threads— tried one and the number of cores per socket— and pinned the computations to each of the sockets (for two-socket servers). Each run produced a record in the dataset with the runtime (in seconds) accompanied by 38 metadata attributes, which include OS version, kernel release, compiler version, etc. Data and code used for our analyses are available at: `https://gitlab.flux.utah.edu/emulab/cloudlab-cpu-perf`.

We begin our analysis by comparing the level of variability in the CPU results with our findings for memory, disk, and network tests from previous work [1]. We also investigate the structure of this variability and identify contributing factors.

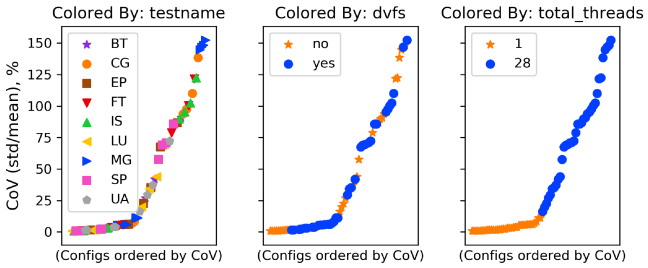


Fig. 1. High Coefficients of Variance for `c6320` CPU results. Each shown point characterizes a sample with over 1,300 measurements.

We continue our analysis with comparisons of empirical distributions observed for sockets 0 and 1 in two-socket servers. We discuss several cases where the differences are substantial.

### A. Level and Structure of Variability

We use coefficients of variance (CoVs), the ratios of the standard deviations to the means (expressed as percentages), to assess the variability. Previously, CloudLab servers have shown relatively high CoVs around 30% for network latency, followed by some memory and HDD tests with CoVs in the 10-20% range [1]. In contrast, Figure 1 shows much larger spread of CPU CoV estimates, which reach up to 150% for `c6320` hardware type. While this is the worst hardware type in terms of variability among those studied, we notice here that for the second and the third worst hardware types the highest CoVs reach nearly 60% and 30%, respectively.

Knowledge about the structure of this high variability can help inform experiment design for reliable results in research. Thus, the coloring in the plots shown in Figure 1 reveals information about the importance of the factors we controlled. The plot on the left indicates that some MG tests lead to the highest CoVs; according to the middle plot, we cannot say that DVFS increases or decreases the variability, since there is no clearly separable groups of points; and the plot on the right affirms that the 28-thread tests show much more variability than the single-thread tests in all studied cases. The last statement is not surprising, but, interestingly, it does not hold true for the `m510` and `m400` types. There, MG and EP produce results with the highest CoVs.

### B. Socket-to-Socket Comparisons

It is expected to obtain the same performance results from identical CPUs placed in both sockets of two-socket servers. While we do not know of a study with comprehensive socket-to-socket comparisons, we have no reasons to suspect

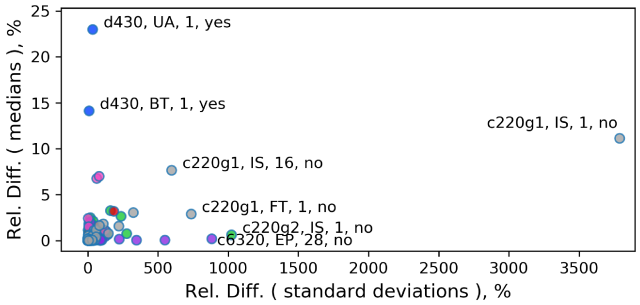


Fig. 2. Relative differences in socket 0 and socket 1 statistics.

otherwise. Our analysis of over 950K data points for 270 configurations (combinations of hardware and test parameters) provide arguments for and against this intuition.

In Figure 2, we visualize these 270 configurations in terms of the relative differences in per-socket statistics: medians and standard deviations. A majority of the points gravitate towards (0,0). We label seven points that stand out. These points trace back to 4 hardware types (not only the worst case from earlier), 5 tests, single- and multi-threaded cases, with DVFS on and off. A case with a relatively high median discrepancy and the largest observed standard deviation difference is shown in Figure 3. These scatter plots show all measurements we collected for this configuration. Not only is the median higher and the standard deviation lower for socket 1, socket 1 is missing the higher performance mode (below 4.0s runtimes) entirely. This gap between the modes in the socket 0's bimodal distribution, reaching up to 15%, needs to be considered in any analysis that uses measurements from two sockets.

Another interesting scenario is found among the rest of the labeled cases from Figure 2. However, rather than checking them one by one, we take a different approach. We create another 2-D visualization where relative differences in the medians are replaced with relative differences in the 90th percentiles (the horizontal axis stays the same). Such visualization helps to shift our attention from the common performance levels to the performance in the "tails", which are critically important in the context of running applications at scale [5]. It allows us to capture a subset of the seven configurations we labeled already, as well as several new cases with large discrepancies. For instance, Figure 4 illustrates one such case where the empirical distributions have tails that are substantially different: slow tests on socket 0 are much slower than the slow tests on socket 1. The gap here between the slow and the fast results is much larger than we saw previously, reaching up to almost $5\times$. The fact that these tail values come from a variety of servers rather than a single "bad" server (as illustrated with different colors representing different servers), speaks further for the significance of this discrepancy.

## III. Discussion and Future Work

In many cases, it is difficult to point out exact root causes of performance patterns and anomalies. Hardware capabilities

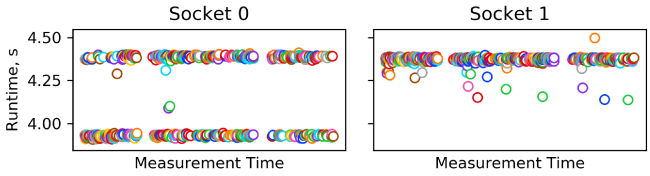


Fig. 3. CPU results for configuration: (`c220g1, IS, 1, no`).

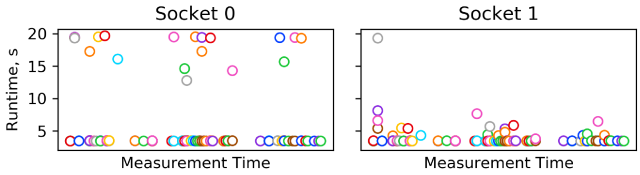


Fig. 4. CPU results for configuration: (`c6320, EP, 28, no`).

and many levels of software interact in non-deterministic and complex ways. Here, we do not set the goal of distilling causes but rather identify areas where investigations—with either additional tests or different modeling techniques—will likely yield interesting insights. Additionally, configurations with less variability and the ones that resemble each other more than the others can be candidates in the search for practical optimizations to reduce the number of tests being run frequently. We plan to pursue this as future work.

It is worth noting that the analysis enabled by CONFIRM [1] helps estimate the number of measurements needed to obtain tight confidence intervals for empirical statistics and copes well with bimodal and long-tailed distributions, such as the ones shown in Figures 3 and 4. In addition to studying such estimates, we will use the collected data to investigate the relationship between the CoV and the level of performance, looking for the evidence for and against the hypothesis suggesting that performance improvements come at the expense of increased variability.

## References

[1] A. Maricq, D. Duplyakin, I. Jimenez, C. Maltzahn, R. Stutsman, and R. Ricci, "Taming performance variability," in *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Oct. 2018. [Online]. Available: https://www.flux.utah.edu/paper/maricq-osdi18

[2] D. Duplyakin, R. Ricci, A. Maricq, G. Wong, J. Duerig, E. Eide, L. Stoller, M. Hibler, D. Johnson, K. Webb, A. Akella, K. Wang, G. Ricart, L. Landweber, C. Elliott, M. Zink, E. Cecchet, S. Kar, and P. Mishra, "The design and operation of CloudLab," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, Jul. 2019. [Online]. Available: https://www.flux.utah.edu/paper/duplyakin-atc19

[3] A. Uta and H. Obaseki, "A performance study of big data workloads in cloud datacenters with network variability," in *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering*. ACM, 2018, pp. 113–118.

[4] D. Bailey, T. Harris, W. Saphir, R. Van Der Wijngaart, A. Woo, and M. Yarrow, "The NAS parallel benchmarks 2.0," Technical Report NAS-95-020, NASA Ames Research Center, Tech. Rep., 1995.

[5] J. Dean and L. A. Barroso, "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, 2013.