QFlex 3.0: Fast and Accurate ARM Server Simulation

Shanqing Lin
EcoCloud, EPFL
shanqing.lin@epfl.ch

Ali Ansari
EcoCloud, EPFL
ali.ansari@epfl.ch

Ayan Chakraborty
EcoCloud, EPFL
ayan.chakraborty@epfl.ch

Bugra Eryilmaz
EcoCloud, EPFL
bugra.eryilmaz@epfl.ch

Yuanlong Li
EcoCloud, EPFL
yuanlong.li@epfl.ch

Mohammad Alian
Cornell University
malian@cornell.edu

Babak Falsafi
EcoCloud, EPFL
babak.falsafi@epfl.ch

***Abstract*—This paper presents QFlex 3.0, the first microarchitectural simulation framework for full-stack ARM workloads running at 10 MIPS. Built on QEMU, a widely used ARM full-system emulator, QFlex 3.0 relies on statistical sampling to cut timing simulation time by two orders of magnitude. Future releases will include enhancements to exploit multicore hosts, interoperable checkpointing to enable versatility across timing simulators, and multi-server simulation for rack-scale computing.**

I. Introduction

ARM servers were first introduced in the earlier parts of last decade (e.g., Calxeda, HP Moonshot, AMD Seattle, AppliedMicro X-Gene, Cavium ThunderX) but did not gain much traction in the absence of a mature Linux ecosystem for ARM in the server space. Over the years, thanks first to Amazon [1] and HiSilicon [2], and now Ampere [3], the ARM server space has now started to flourish with a solid ecosystem for server software stacks. Today, there is a spectrum of new ARM server vendors including key players such as NVIDIA, Supermicro and Lenovo offering a variety of products that capitalize on the silicon efficiency of ARM cores [4] and the maturity of its server software stack.

With the slowdown in Moore's Law, computer system designers increasingly need tools, methodologies and open-source workloads to identify and properly evaluate full-stack server designs. Unfortunately, there are no timing simulators today that allow for a proper microarchitectural exploration of running full-stack server workloads. Today's fastest full-system timing simulators (e.g., gem5 [5]) run at simulation speeds of 250 kilo instructions per second (KIPS). For a stable measurement of one monolith server stack or microservices, designers require a long measurement spanning at least several seconds of the *target* machine. Such requirement for even a 64-core multicore socket at 2 GHz translates to months of simulation. Consequently, most rely on severely abbreviated measurement windows (e.g., 50 ms [6], [7]) which can lead to inconclusive results. More importantly, there is no confidence or error bounds in the reported results which could highly vary across various phases of execution [8]–[10].

Prior work on rigorous methodologies for full-system server simulation [10] has shown simulation sampling to be effective at reducing the required time for timing simulation by several orders of magnitude. In this work, we present QFlex 3.0, which enhances a family of simulators derived from SimFlex [10] by supporting the ARM ISA and statistical sampling. QFlex 3.0 runs two orders of magnitude faster than state-of-the-art full-system timing simulators, reducing the end-to-end simulation time to measure a dozen seconds of target execution time for multicore sockets to tens of hours as compared to months. We present results for CloudSuite 4.0 server workloads showing that error can be bounded with 95% confidence to 5% relative to a reference simulation that uses end-to-end timing.

II. QFlex 3.0

QFlex is a simulation framework derived from SimFlex [10] to support the ARM ISA with simulation sampling. Much like SimFlex, QFlex relies on a full-system multiprocessor emulator to generate a trace of instructions for simulation. SimFlex used Simics [11] to emulate the SPARC ISA. In contrast, QFlex uses QEMU [12], a widely used open-source full-system emulator that supports multiple ISAs including ARM. QEMU offers high emulation speeds of up to 400 million instructions per second (MIPS) per core and is compatible with the latest Linux kernel releases. In contrast to Simics, QEMU forgoes determinism, in favor of higher emulation speed with asynchronous I/O operations.

QFlex 3.0 enhances QFlex 2.0 [13] by adding support for simulation sampling. Instead of *end-to-end* timing simulation of a workload for a large period of execution to obtain stable performance metrics [9], simulation sampling allows users to sample a subset of the execution period and bound the sampling error with statistical guarantees. This approach enables practical server simulation that requires measurements from seconds to minutes of target time.

QFlex 3.0 employs two simulators—a functional simulator and a timing simulator—to implement simulation sampling. A functional simulator performs *functional warming* [8], populating SRAM-based structures in the CPU including (but not limited to) caches, TLBs, branch-predictor tables, BTB, and saves both microarchitectural and architectural state together as a *checkpoint*. A timing simulator then loads this checkpoint to

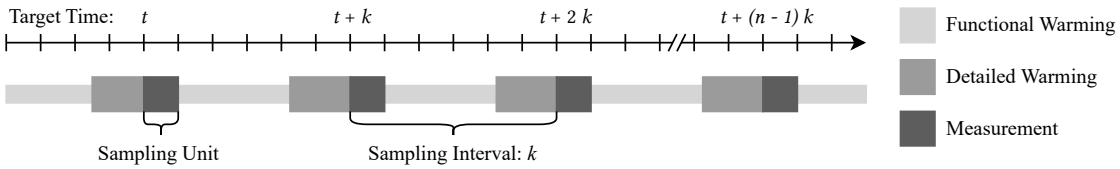

Fig. 1. Simulation sampling with QFlex 3.0.

run *detailed simulation*, first proceeding with *detailed warming* [8] to rebuild transient state (e.g., ROB, store buffer, NoC queues) and deviations arising from speculation and wrong-path execution [14], [15], then performing a *measurement* over a *sampling unit*. Sampling units are evenly distributed across the population to capture its behavior over the large timescale. Figure 1 illustrates the simulation sampling process using QFlex 3.0.

To statistically bound the sampling error ϵ under a specified confidence level, the *sample size* n—i.e, the number of sampling units—should meet the following requirement [8]:

$$n \geq \left(\frac{Z}{\epsilon} V_x\right)^2 \tag{1}$$

where, Z is the z-score of the given confidence level, and V_x is the coefficient of variation of measurements of metric x across all sampling units. A collection of sampling units with a sample size that satisfies the inequality above is called a *sample*. Note that each sampling unit requires a dedicated checkpoint to store its necessary architectural and microarchitectural state, as previously explained.

In practice, achieving a $\pm 5\%$ margin of sampling error with 95% confidence (i.e., $Z = 2$) requires only a few hundred sampling units. Detailed simulation needs to be performed on each sampling unit for just a few hundred target μs [10], resulting in only 0.1% to 0.4% of target cycles being simulated with a timing simulator. Detailed simulation of multiple sampling units are independent and can run concurrently based on the availability of host resources. Following Amdahl's law, simulation speed is primarily determined by the functional warming speed, rather than by detailed simulation speed.

Apart from supporting simulation sampling, QFlex 3.0 also features automatic checkpoint generation with incremental memory checkpointing and fast-to-run microarchitectural models for functional warming. Additionally, QFlex 3.0 extends ARM ISA support to integrate with the latest ARM software stack as featured in CloudSuite 4.0 [16]. QFlex 3.0 can also construct the microarchitectural state for various SRAM-based structure sizes using metadata recorded in a single checkpoint [17], eliminating redundant functional warming during design space exploration.

In future QFlex releases, we aim to deliver the following three key features. First, we are developing a parallel functional simulator to reduce the functional warming time, utilizing QEMU's multi-threaded emulation. Our preliminary results show that a parallel functional simulator can increase functional warming speed by up to 50× on a 64-core host. Second, we will develop a universal checkpoint format compatible with other timing simulators such as gem5 [5], extending the applicability of simulation sampling. Third, we plan to support multi-server simulation [18], [19], currently not supported by QEMU, to enable rack-scale computing exploration.

TABLE I
TARGET MACHINE PARAMETERS

Component	Parameter
Core	ARMv8 ISA, 2 GHz, 4-wide OoO, 140-entry ROB, TAGE, 4K-entry BTB
Cache Hierarchy	L1 I/D: 64 KB, 4-way LLC: 2 MB/tile, 16-way, non-inclusive
TLB Hierarchy	L1 I/D: 48-entry, fully-associative L2: 1280-entry, 5-way
Directory	MESI protocol, 16-way 4× over-provisioned

III. EVALUATION

We run QFlex[1] on an Intel Xeon Gold 5520+ server operating at 2.2 GHz with 2 TB of memory as the host machine. The hardware configuration of the target machine QFlex simulates is shown in Table I. We run CloudSuite 4.0 [16] on the target machine as representative server workloads. For online services with specific latency requirements, we tune the workloads in the functional warming simulator to maximize throughput without violating service level objectives.

For server workloads, throughput serves as the primary performance metric of interest. We select U-IPC (User Instructions Per Cycle) as target performance metric because prior work has demonstrated that a workload's U-IPC is proportional to its throughput [10]. Furthermore, U-IPC exhibits lower variance than throughput measurements at significantly smaller measurement intervals, making it particularly suitable for simulation sampling.

We evaluate the accuracy of QFlex by comparing U-IPC [10] between end-to-end timing simulation and simulation sampling. QFlex is deemed accurate when simulation sampling results fall within the margin of sampling error relative to end-to-end results. Following the methodology in [10], we target a ±5% margin of sampling error with 95% confidence, requiring sample sizes up to 600 for our workloads. We empirically determine that 200 K cycles for detailed warming and 100 K cycles for measurement minimize the overall detailed simulation time while satisfying Equation 1.

Table II presents the simulation U-IPC for a single-core target machine. For all workloads, the simulation sampling

[1]In this section, we use QFlex to refer to QFlex 3.0.

TABLE II
SINGLE-CORE U-IPC REPORTED BY END-TO-END TIMING AND SIMULATION SAMPLING.

Workload	End-to-End Timing U-IPC	Simulation Sampling U-IPC	Error (%)
Data Analytics	0.64	0.66	2.83
Data Caching	0.08	0.08	-1.85
Data Serving	0.40	0.39	-2.72
Graph Analytics	1.07	1.04	-2.37
In-memory Analytics	0.92	0.94	2.49
Media Streaming	1.78	1.71	-4.11
Web Search	2.09	2.13	2.10
Web Serving	1.06	1.04	-2.21

TABLE III
64-CORE SIMULATION RESULTS SHOWING FUNCTIONAL WARMING SPEED, FRACTION OF CYCLES REQUIRING DETAILED SIMULATION, END-TO-END HOST TIME WITH FUNCTIONAL WARMING, AND ESTIMATED END-TO-END HOST TIME WITH GEM5 (WITHOUT SAMPLING).

Workload	Speed (MIPS)	Fraction (%)	Time (hours)	Estimated gem5 time (days)
Data Analytics	8	0.30	34	41
Data Caching	6	0.15	11	31
Data Serving	6	0.39	6	68
Graph Analytics	13	0.15	31	59
In-memory Analytics	13	0.15	34	53
Media Streaming	19	0.15	6	47
Web Search	16	0.15	7	59
Web Serving	11	0.30	5	50
Mean	10	0.20	17	51

results fall within a 5% margin of sampling error of the end-to-end timing simulation results. These results corroborate prior work [10], demonstrating that simulation sampling yields accurate simulation results due to statistical bounds on sampling error. Notably, Data Caching exhibits exceptionally low U-IPC [2] because it executes only 10% of all instructions in userspace with the remainder in the kernel's network stack. This workload highlights the necessity of full-system simulation for accurately evaluating server applications.

Table III summarizes the results of various metrics for simulating a 64-core server. Functional warming achieves an average speed of 10 MIPS. As mentioned in Section II, only 0.1% to 0.4% of target cycles require detailed simulation, so this 10 MIPS functional warming speed translates directly into fast simulation sampling. Consequently, generating a sample for a workload running on a 64-core server takes between 5 and 34 hours depending on the workload, making complex server workload studies feasible. In contrast, gem5 would require approximately 50 days on average (based on its 250 KIPS single-core simulation speed)—a timeframe prohibitive for practical research.

IV. CONCLUSION

In this work, we introduced QFlex 3.0, a fast and accurate multicore ARM CPU simulation framework that incorporates sampling methodology. QFlex 3.0 addresses the community's longstanding need for a simulator that enables productive research and innovation in designing next-generation server CPUs. To support broader adoption and reproducibility, QFlex 3.0 will be publicly available along with prebuilt images for CloudSuite 4.0 workloads.

[2]U-IPC is 0.076 for end-to-end timing, and 0.075 for simulation sampling.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable feedback and suggestions. This work was supported by the UrbanTwin project, the FNS Project 200021/212757, and Intel Midgard. We also acknowledge the support of the Intel Center for Heterogeneous Integrated Platforms (HIP).

REFERENCES

[1] Amazon, "ARM Processor - AWS Graviton Processor - AWS — aws.amazon.com," https://aws.amazon.com/ec2/graviton/, 22.

[2] J. Xia, C. Cheng, X. Zhou, Y. Hu, and P. Chun, "Kunpeng 920: The First 7-nm Chiplet-Based 64-Core ARM SoC for Cloud Services." *IEEE Micro*, vol. 41, no. 5, pp. 67–75, 2021.

[3] D. Carlson, N. Simakov, R. R. Hadlich, A. Curtis, J. E. Martin, G. Verma, S. Chheda, F. Coskun, R. Gonzalez, D. G. Wood, F. Zhang, R. J. Harrison, and E. Siegmann, "The AmpereOne A192-32X in Perspective: Benchmarking a New Standard." in *HPC Asia Workshops*, 2025, pp. 23–35.

[4] A. Ansari, S. Lin, A. Chakraborty, B. Eryilmaz, M. Alian, B. Falsafi, and M. Ferdman, "Silicon efficiency in post-moore servers," in *Workshop on Hot Topics in Ethical Computer Systems (HotEthics)*, 2024.

[5] N. L. Binkert, B. M. Beckmann, G. Black, S. K. Reinhardt, A. G. Saidi, A. Basu, J. Hestness, D. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. S. B. Altaf, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator." *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[6] J. Feliu, A. Perais, D. A. Jiménez, and A. Ros, "Rebasing microarchitectural research with industry traces," in *IEEE International Symposium on Workload Characterization, IISWC 2023, Ghent, Belgium, October 1-3, 2023*. IEEE, 2023, pp. 100–114. [Online]. Available: https://doi.org/10.1109/IISWC59245.2023.00027

[7] T. A. Khan, N. Brown, A. Sriraman, N. K. Soundararajan, R. Kumar, J. Devietti, S. Subramoney, G. A. Pokam, H. Litz, and B. Kasikci, "Twig: Profile-guided BTB prefetching for data center applications," in *MICRO '21: 54th Annual IEEE/ACM International Symposium on Microarchitecture, Virtual Event, Greece, October 18-22, 2021*. ACM, 2021, pp. 816–829. [Online]. Available: https://doi.org/10.1145/3466752.3480124

[8] R. E. Wunderlich, T. F. Wenisch, B. Falsafi, and J. C. Hoe, "SMARTS: Accelerating Microarchitecture Simulation via Rigorous Statistical Sampling." in *Proceedings of the 30th International Symposium on Computer Architecture (ISCA)*, 2003, pp. 84–95.

[9] A. R. Alameldeen and D. A. Wood, "Variability in Architectural Simulations of Multi-Threaded Workloads." in *Proceedings of the 9th IEEE Symposium on High-Performance Computer Architecture (HPCA)*, 2003, pp. 7–18.

[10] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe, "SimFlex: Statistical Sampling of Computer System Simulation." *IEEE Micro*, vol. 26, no. 4, pp. 18–31, 2006.

[11] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hållberg, J. Högberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A Full System Simulation Platform." *Computer*, vol. 35, no. 2, pp. 50–58, 2002.

[12] F. Bellard, "QEMU, a Fast and Portable Dynamic Translator." in *USENIX ATC, FREENIX Track*, 2005, pp. 41–46.

[13] "QFlex 2.0," https://qflex.epfl.ch/qflex-v2-0-released/.

[14] B. R. Godala, S. P. Ramesh, K. Tibrewala, C. Pepi, G. Chacon, S. Kanev, G. A. Pokam, D. A. Jiménez, P. V. Gratz, and D. I. August, "Correct Wrong Path." *CoRR*, vol. abs/2408.05912, 2024.

[15] S. Eyerman, S. V. den Steen, W. Heirman, and I. Hur, "Simulating Wrong-Path Instructions in Decoupled Functional-First Simulation." 2023, pp. 124–133.

[16] "CloudSuite 4.0," https://www.cloudsuite.ch/.

[17] K. C. Barr, H. Pan, M. Zhang, and K. Asanovic, "Accelerating Multiprocessor Simulation with a Memory Timestamp Record." in *Proceedings of the 2005 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2005, pp. 66–77.

[18] M. Alian, U. Darbaz, G. Dózsa, S. Diestelhorst, D. Kim, and N. S. Kim, "dist-gem5: Distributed simulation of computer clusters." 2017, pp. 153–162.

[19] H. Li, J. Li, and A. Kaufmann, "SimBricks: end-to-end network system evaluation with modular simulation." in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, pp. 380–396.