# gem5 GPU Accuracy Profiler (GAP)

Charles Jamieson, Anushka Chandrashekar, Ian McDougall, Matthew D. Sinclair
University of Wisconsin-Madison
{cjamieson2, achandrashe4, imcdougall}@wisc.edu {sinclair}@cs.wisc.edu

## I. Motivation

In recent years, we have been enhancing and updating gem5's GPU support [1]. First, we have enhanced gem5's GPU support for ML workloads such that gem5 can now run [2]. Moreover, as part of this support, we created, validated, and released a Docker image that contains the proper software and libraries needed to run GCN3 and Vega GPU models in gem5. With this container, users can run the gem5 GPU model, as well as build the ROCm applications that they want to run in the GPU model, out of the box without needing to properly install the appropriate ROCm software and libraries [2], [3]. Additionally, we have updated gem5 to make it easier to reproduce results, including releasing support for a number of GPU workloads in gem5-resources [4] and enabling continuous integration testing on future GPU commits.

However, we currently do not have a way to model validated gem5 configurations for the most recent AMD GPUs. Current support focuses on Carrizo- and Vega-class GPUs. Unfortunately, these models do not always provide high accuracy relative to real GPU runs. This leads to a mismatch between how each instruction is supposedly being executed according to the ISA and how a given GPU model executes a given instruction. These discrepancies are of interest to those developing the gem5 GPU models as they can lead to less accurate simulations. Accordingly, to help bridge this divide, we have created a new tool, **GAP** (*gem5 GPU Accuracy Profiler*), to identify discrepancies between real GPU and simulated gem5 GPU behavior. GAP identifies and verifies how accurate these configurations relative to real GPUs by comparing the simulator's performance counters to those from real GPUs.

## II. Methodology

Figure 1 shows the overall flow of GAP. To properly identify inaccuracies in the gem5's GPU simulations, we used an AMD Vega 20 (Radeon VII) as the baseline GPU. After configuring gem5 to use a similar configuration to the Vega 20, GAP's scripts run the same GPU binaries on gem5 and the physical chip. Overall, the user must specify: a) which benchmarks/binaries to run (in GAP's configuration file), b) any required command line arguments (in the gem5 configuration script), c) the path to gem5, and d) the ROCm profiler's (rocprof) configuration file [5] (to gather the appropriate hardware counter information). Given these elements, the script runs the ROCm profiler and gem5 as specified, then collects the output and parses for relevant metrics. The metrics that will be compared in the output file are specified in the ROCm profiler's input file. The GAP output file will contain the absolute values and percentage difference of collected

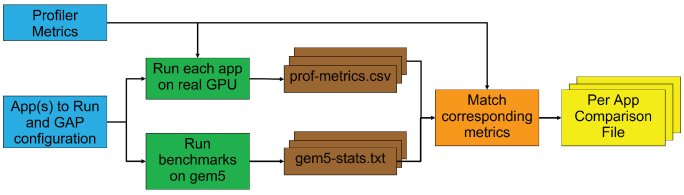


Fig. 1: GAP flowchart.

metrics. Although currently we have only tested this on a Vega 20, GAP and gem5's GPU model are both flexible enough that other GPUs could also be used for testing.

To test GAP, we used the existing benchmarks in gem5-resources [3]. For example, for square, GAP shows that the VALUUtilization is within 1% on the real GPU and gem5, but the TCC misses differ by 821%, likely indicating that further tuning of the memory sub-system is required to improve model quality. However, these benchmarks tend to be larger and thus make it difficult to isolate the behavior of specific GPU components. Thus, to help isolate and improve the behavior of specific components, we also ported a variety of GPU microbenchmarks from prior work [6]–[9] to HIP.

## III. Conclusion

Architectural simulation tools are highly important to the computer architecture community: both industry and academia rely on these tools to substantiate their findings. However this means that findings are only accurate insofar as the tools are accurate. GAP helps measure and improve gem5's GPU model. GAP gets a picture of how close gem5 is to reality by running an application on both gem5 and real hardware. The gem5 simulation is set up to closely emulate the real GPU while the hardware is profiled during the running of an application. By collecting the results of both the profiler and the simulation we can get a picture of how closely the simulation is following reality. GAP does the leg work to compare items such as cache hit and misses, utilization, instruction counts, and others. By iteratively making changes to gem5 and using GAP to ensure those changes increase the correlation of the real and simulated hardware we can improve the accuracy of gem5. Although GAP currently only improves the GPU model in this work, we believe the underlying idea can also be applied to other simulation tools. Moving forward we plan to integrate GAP into the regressions, to help parties contributing to gem5's source code to ensure their additions do not hurt the accuracy of gem5's GPU simulations.

## ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation grant ENS-1925485.

## REFERENCES

[1] A. Gutierrez, B. M. Beckmann, A. Dutu, J. Gross, M. LeBeane, J. Kalamatianos, O. Kayiran, M. Poremba, B. Potter, S. Puthoor, M. D. Sinclair, M. Wyse, J. Yin, X. Zhang, A. Jain, and T. Rogers, "Lost in Abstraction: Pitfalls of Analyzing GPUs at the Intermediate Language Level," in *2018 IEEE International Symposium on High Performance Computer Architecture*, ser. HPCA, Feb 2018, pp. 608–619.

[2] K. Roarty and M. D. Sinclair, "Modeling Modern GPU Applications in gem5," in *3rd gem5 Users' Workshop*, June 2020.

[3] B. R. Bruce, A. Akram, H. Nguyen, K. Roarty, M. Samani, M. Fariborz, T. Reddy, M. D. Sinclair, and J. Lowe-Power, "Enabling Reproducible and Agile Full-System Simulation," in *IEEE International Symposium on Performance Analysis of Systems and Software*, ser. ISPASS, 2021.

[4] gem5, "gem5 Resources," https://www.gem5.org/documentation/general_docs/gem5_resources/, 2020.

[5] AMD, "AMD ROCm Profiler," https://rocmdocs.amd.com/en/latest/ROCm_Tools/ROCm-Tools.html, 2021.

[6] T. Deakin, J. Price, M. Martineau, and S. McIntosh-Smith, "GPU-STREAM v2.0: Benchmarking the Achievable Memory Bandwidth of Many-Core Processors Across Diverse Parallel Programming Models," in *High Performance Computing*, M. Taufer, B. Mohr, and J. M. Kunkel, Eds. Cham: Springer International Publishing, 2016, pp. 489–507.

[7] M. Khairy, A. Jain, T. M. Aamodt, and T. G. Rogers, "Exploring Modern GPU Memory System Design Challenges through Accurate Modeling," *CoRR*, vol. abs/1810.07269, 2018. [Online]. Available: http://arxiv.org/abs/1810.07269

[8] M. Khairy, Z. Shen, T. M. Aamodt, and T. G. Rogers, "Accel-Sim: An Extensible Simulation Framework for Validated GPU Modeling," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ser. ISCA, 2020, pp. 473–486.

[9] H. Wong, M.-M. Papadopoulou, M. Sadooghi-Alvandi, and A. Moshovos, "Demystifying GPU Microarchitecture Through Microbenchmarking," in *IEEE International Symposium on Performance Analysis of Systems Software*, ser. ISPASS, 2010, pp. 235–246.