Closing the Gap: Improving the Accuracy of gem5's GPU Models

Vishnu Ramadas, Daniel Kouchekinia, Ndubuisi Osuji, Matthew D. Sinclair
University of Wisconsin-Madison
{vramadas, kouchekinia, osuji}@wisc.edu sinclair@cs.wisc.edu

I. Motivation

In recent years, we have been enhancing and updating gem5's GPU support [1], including enhanced gem5's GPU support to enable running ML workloads [2]. Moreover, we created, validated, and released a Docker image with the proper software and libraries needed to run AMD's 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 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 for a variety of GPU workloads.

Current gem5 support focuses on Carrizo- and Vega-class GPUs. Unfortunately, these models do not always provide high accuracy relative to the equivalent "real" GPUs. This leads to a mismatch in expectations: when prototyping new optimizations in gem5 users may draw the wrong conclusions about the efficacy of proposed optimizations if gem5's GPU models do not provide high fidelity. Accordingly, to help bridge this divide, we design a series of micro-benchmarks designed expose the latencies, bandwidths, and sizes of a variety of GPU components on real GPUs. By iteratively applying fixes and improvements to gem's GPU model, we significantly improve its fidelity relative to real AMD GPUs.

II. Methodology

To properly identify performance inaccuracies in 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, we ran the same binaries on gem5 and the physical GPU. Next, we measure how accurate these configurations relative to real GPUs by comparing the simulator's performance counters to those from real GPUs. We used the ROCm profiler [5] on the physical chip and gem5's performance counters to collect data about the GPU for relevant 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. Finally, given these results we iteratively refined the simulation models as appropriate to more closely model the Vega 20's behavior.

To measure the accuracy of gem5's GPU models, we initially used existing benchmarks in gem5-resources [3]. For example, running square with GAP shows that the VALUUtilization is within 1% on the real GPU and gem5, but the L2 cache misses differ by 821%, likely indicating that further tuning of the memory sub-system is required to improve model quality. However, it is difficult to isolate the behavior of specific GPU components in larger benchmarks. Thus, we ported a variety of GPU micro-benchmarks [6]–[9] to HIP. These micro-benchmarks help isolate gem5's inaccuracies such as access latencies and bandwidths of L1, L2 caches, LDS, atomic operations, and global memory. To reduce these inaccuracies, we updated gem5's models by tuning gem5 configuration parameters and implementing previously not modeled features. For example, the existing AMD GPU support assumed that all atomics were system-scope [10], [11], even when cheaper scopes (e.g., device scope) was specified by the program. Thus, to improve the accuracy of gem5's GPU models for atomics we added support for performing device scope atomics at the L2 cache.

Metric	Old Accuracy	New Accuracy
L1 Latency	2.18%	0.4%
L1 Bandwidth	41.75%	9.83%
L1 Scalar Latency	41.39%	0.98%
L2 Latency	0.08%	0.07%
L2 Bandwidth	52.15%	7.81%
Atomics Latency	51.79%	0.13%
Atomics Bandwidth	47.77%	7.7%

TABLE I: gem5 GPU components accuracy before and after our updates, relative to Vega 20 GPU.

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, findings are only accurate insofar as the tools are accurate. By collecting the results of both the profiler and the simulation we can examine how closely gem5 mirrors reality. Accordingly, we improved the accuracy of various components in the GPU by debugging the performance numbers and the improvements are captured in Table I. Although we currently focus on the GPU model, the underlying idea can also be applied to other simulation tools. Moving forward we plan to provide "known good" configurations for a variety of modern AMD GPUs. Moreover, we will integrate our tests 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.

[10] D. R. Hower, B. A. Hechtman, B. M. Beckmann, B. R. Gaster, M. D. Hill, S. K. Reinhardt, and D. A. Wood, "Heterogeneous-race-free Memory Models," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS. New York, NY, USA: ACM, 2014, pp. 427–440. [Online]. Available: http://doi.acm.org/10.1145/2541940.2541981

[11] B. R. Gaster, D. Hower, and L. Howes, "HRF-Relaxed: Adapting HRF to the Complexities of Industrial Heterogeneous Memory Models," *ACM Trans. Archit. Code Optim.*, vol. 12, no. 1, pp. 7:1–7:26, Apr. 2015. [Online]. Available: http://doi.acm.org/10.1145/2701618