Linear Algebra Based Graph Analysis on RISC-V GPGPU Vortex

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Semyon Grigorev
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—In this work we evaluate Spla—sparse linear algebra based library for graph analysis—on RISC-V ISA based open source GPGPU Vortex. We show that !!!**

***Index Terms*—GraphBLAS, Sparse Linear Algebra, Graph Analysis, GPGPU, RISC-V**

I. Introduction

Sparse linear algebra has emerged as a powerful paradigm for high-performance graph analysis. A wide range of problems—from graph traversing to clustering—can be reduced to efficient algebraic operations over matrices and vectors. GraphBLAS API [1] fallows this idea and defines a standardized set of building blocks: sparse matrices and vectors, algebraic structures like monoids and semirings, and fundamental operations such as matrix-matrix multiplication. GraphBLAS is specifically designed to serve as a foundational layer for the development of scalable, linear-algebra-based graph algorithms.

While highly tuned CPU implementations of GraphBLAS—most notably SuiteSparse:GraphBLAS[1] [2]—deliver strong performance on multi-core systems, implementing the GraphBLAS API efficiently on general-purpose graphics processing units (GPGPUs) remains a significant challenge. While GPGPUs is a promising platform for linear algebra based computations, they introduce well-known obstacles for sparse workloads, including irregular memory access patterns and load imbalance. Additionally, creating generalized kernels capable of operating not only on primitive data types like floats or integers but also on user-defined custom types presents a nontrivial engineering task. Despite these challenges, several efforts have been made to create GPU-accelerated libraries for linear-algebra-based graph analysis, such as GraphBLAST[2] [3] which uses CUDA and the portable Spla[3] library which uses OpenCL.

RISC-V becoming popular. Not only CPUs, but specific devices, including GPUs. One of the actively developed RISC-V ISA based GPGPU is Vortex. Is Vortex suitable for linear algebra based graph analysis?

Identify applicable funding agency here. If none, delete this.

[1]Source code of SuiteSparse:GraphBLAS on GitHub: https://github.com/DrTimothyAldenDavis/GraphBLAS

[2]GraphBLAST project page: https://github.com/gunrock/graphblast

[3]Spla project page: https://github.com/SparseLinearAlgebra/spla

We do the following contribution in this paper.

1) Port Spla on Vortex. Technical improvements of both.
2) Investigate scaling in simulator. For BFS and Triangle Count (TC).
3) FPGA? Resources?

II. Spla Graph Analysis Library

Spla[4] [**?**] is a GPGPU-powered GraphBLAS-inspired library for graph analysis. It is based on sparse linear algebra and uses OpenCL to offload linear algebra related kernels to appropriate devices, including GPGPUs. Utilization of OpenCL make this library vendor-agnostic.

Such graph analysis algorithms as canonical single-source level BFS, !!!Sandia!!! triangle counting (TC), single-source shortest path (SSSP), PageRank are implemented as part of library.

In [**?**] it was shown that it performes and scales well across GPUs of different vendors. What about Vortex?

III. RISC-V GPGPU Vortex

Vortex[5] [4] is an open-source RISC-V ISA based GPGPU. For programming it supports OpenCL via POCL[6] [**?**] compiler. Moreover, it designed for FPGA equipped with high-bandwidth memory (HBM) that can !!! for graph processing

Configurable. Architecture (figure), Cluster, core, threads, warp. Caches.

Simulators SimX, RTL, FPGA SimX is faster than RTL simulation.

While A extension is declared, atomics supported only in SimX, not in RTL.

IV. Evaluation

Goals, research questions.

RQ1 Scaling
RQ2 FPGA resources consumption
RQ3 Performance (FPGA)

Common settings: graph, algos,

[4]Spla sources on GitHub: https://github.com/SparseLinearAlgebra/spla

[5]https://github.com/vortexgpgpu/vortex

[6]!!!

A. Scaling

Design (what, how)
Graphics
Results analysis

B. FPGA resources consumption

Design (what, how)
Graphics
Results analysis

C. Performance

Design (what, how)
Graphics
Results analysis

V. Conclusion

In this work we evaluated Spla—linear algebra based graph analysis library—on RISC-V IAS based GPGPU Vortex. We show that Spla is portable enough to be run on Vortex. Vortex ready to run. Scaling.

Future work

Evaluate Ventus [7] [5] GPU, compare with Vortex. !!!

References

[1] B. Brock, A. Buluç, T. G. Mattson, S. McMillan, and J. E. Moreira, "Introduction to graphblas 2.0," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2021, pp. 253–262.

[2] T. A. Davis, "Algorithm 1037: Suitesparse:graphblas: Parallel graph algorithms in the language of sparse linear algebra," *ACM Trans. Math. Softw.*, vol. 49, no. 3, Sep. 2023. [Online]. Available: https://doi.org/10.1145/3577195

[3] C. Yang, A. Buluç, and J. D. Owens, "Graphblast: A high-performance linear algebra-based graph framework on the gpu," *ACM Trans. Math. Softw.*, vol. 48, no. 1, Feb. 2022. [Online]. Available: https://doi.org/10.1145/3466795

[4] B. Tine, K. P. Yalamarthy, F. Elsabbagh, and K. Hyesoon, "Vortex: Extending the risc-v isa for gpgpu and 3d-graphics," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 754–766. [Online]. Available: https://doi.org/10.1145/3466752.3480128

[5] J. Li, K. Yang, C. Jin, X. Liu, Z. Yang, F. Yu, Y. Shi, M. Ma, L. Kong, J. Zhou, H. Wu, and H. He, "Ventus: A high-performance open-source gpgpu based on risc-v and its vector extension," in *2024 IEEE 42nd International Conference on Computer Design (ICCD)*, 2024, pp. 276–279.

[7] https://github.com/THU-DSP-LAB/ventus-gpgpu