

GraphBLAS API in Functional Style*

*Note: Sub-titles are not captured in Xplore and should not be used

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

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

[illegible]

Index Terms—graph analysis, sparse linear algebra, Graph-BLAS API, GPGPU, parallel programming, .NET, functional programming

I. INTRODUCTION

Graph analysis problems. Not only Graphs. Sparse Linear algebrs. GraphBLAS API

GPGPU for high-performance analysis of huge amount of data. GraphBLAST [?]

High-level programming languages for application development vs low-level for high-performance programming. Moreover, specific languages for GPGPU programming: CUDA C, OpenCL C.

Portability of OpenCL.

Type systems. Optimizations. Futhark [1], kernel fusion, etc.
.NET as a platform. F#

II. DESIGN PRINCIPLES

Functional style, types, optimizations, etc.

Code example with description and explanations.

III. IMPLEMENTATION DETAILS

Details on implementation.

Architecture.

IV. EVALUATION

Evaluation of the proposed implementation.

SuiteSparse, Math.NET Numerics, GraphBLAST, ???, and our solution on CPU and GPGPU.

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

V. CONCLUSION

Conclusion and future work.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

- [1] T. Henriksen, N. G. W. Serup, M. Elsmann, F. Henglein, and C. E. Oancea, “Futhark: Purely functional gpu-programming with nested parallelism and in-place array updates,” in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2017. New York, NY, USA: ACM, 2017, pp. 556–571. [Online]. Available: <http://doi.acm.org/10.1145/3062341.3062354>