The Journal of Open Source Software

Accelerating Parallel Operation for Compacting Selected Elements on GPUs

Johannes Fett [1¶], **Urs Kober**[1], **Christian Schwarz**[1], **Dirk Habich**[1], **and Wolfgang Lehner**[1]

1 TU Dresden, Germany ¶ Corresponding author

DOI: 10.21105/joss.04589

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @robertszafa
- @wimvanderbauwhede

Submitted: 28 June 2022
Published: 22 July 2022

Summary

Compacting is a common and heavily used operation in different application areas such as statistics, database systems, simulations, and artificial intelligence. The task of this operation is to write selected elements of an input array contiguously back to a new output array, producing a smaller (i.e., compact) array. The selected elements are usually defined by means of a bit mask. With the always increasing amount of data to be processed in different application areas, better performance becomes a key factor for this operation. Thus, exploiting the parallel capabilities of GPUs to speed up the compacting operation is of great interest. We introduce smart partitioning for GPU compaction (SPACE) as a set of different optimization approaches for GPUs. A detailed guide about setting up and using the software is found in Fett et al. (n.d.). Fett et al. (2022) is a preprint of the published Euro-Par 2022 paper, in which SPACE is described in great detail.

Statement of need

SPACE is GPU-centric C++ software for compaction experiments. It consists of different data generators and a flexible experiment framework. Eight different SPACE variants can be compared against the NVIDIA-supplied CUB library for GPU compaction. Data type, percentage of selected data, and data distributions are modifiable as execution parameters for the generated C++ binary. Different Python runscripts for performing sets of experiments and reproducing the experiments from our paper (Fett et al. (2022)) are provided. The output of experiments is written in csv files. For visualizing the results, Python scripts based on Matplotlib are also provided.

SPACE was designed to allow researchers to evaluate compaction algorithms against a solid baseline across a variety of input data. It can be modified by adding additional compaction algorithms. It outperforms the current state-of-the-art algorithm (NVIDIA CUB library, n.d.). Research about compaction has been performed by Bakunas-Milanowski et al. (2017), who classify compaction on GPU into the two categories: "prefix sum based" and "atomic based". SPACE is a prefix sum based approach.

Acknowledgements

This work is funded by the German Research Foundation within the RTG 1907 (RoSI) as well as by the European Union's Horizon 2020 research and innovative program under grant agreement number 957407 (DAPHNE project).

References

Bakunas-Milanowski, D., Rego, V., Sang, J., & Chansu, Y. (2017). Efficient algorithms for stream compaction on GPUs. *International Journal of Networking and Computing*, *7*(2), 208–226. https://doi.org/10.15803/ijnc.7.2_208

Fett, J., Kober, U., Schwarz, C., Habich, D., & Lehner, W. (n.d.). *Overview*. https://github.com/yogi-tud/space_gpu/blob/main/overview.pdf.

Fett, J., Kober, U., Schwarz, C., Habich, D., & Lehner, W. (2022). *Accelerating parallel operation for compacting selected elements on GPUs*. https://github.com/yogi-tud/space_gpu/blob/main/Euro_Pars_2022_Compaction_CRC.pdf.

NVIDIA CUB library. (n.d.). https://nvlabs.github.io/cub/index.html.