Master's Thesis

Wavelet Tree Construction on GPUs

Overview

Bit Vectors are one of the most basic data structures in computer science. Operations on bit vectors include rank and select queries.

- $rank_1(i)$ returns the number of 1-bits up to position i and
- $select_1(i)$ returns the position at which the i-th 1-bit is stored.

One of the many applications of bit vectors with rank and select support are wavelet trees. A *wavelet tree* is a binary tree data structure that can be used to answer *rank* and *select* queries on texts of size n over an alphabet of size σ in $O(\lg \sigma)$ time. Here, $rank_\alpha(i)$ queries ask for the number of occurrences of the symbol α before the position i and $select_\alpha(i)$ queries return the text position of the i-th occurrence of the symbol α.

Let T be a text of length n over an alphabet of size σ. The wavelet tree requires $n\lceil \log \sigma \rceil(1 + o(1))$ bits, see Fig. 1. In shared and distributed memory, there exist fast WT construction algorithms [1]. However, there seem to be efficient implementations of neither rank and select data structures, nor wavelet trees on GPUs. A starting point for the bit vector can be the pasta::bit_vector [2]. The Nvidia nvbio library provides an implementation but does not use state of the art algorithms[1].

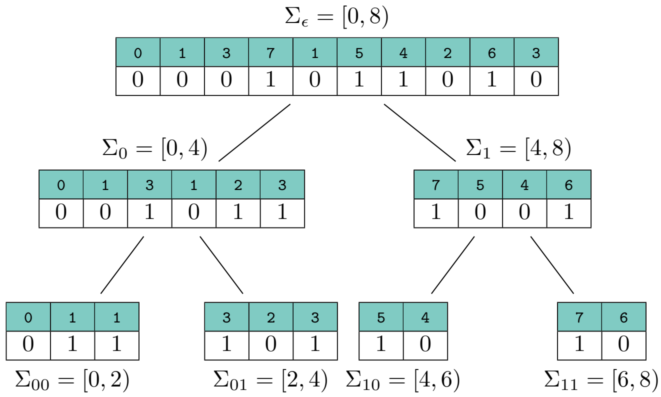

Figure 1: The wavelet tree of $T = [0, 1, 3, 7, 1, 5, 4, 2, 6, 3]$. The light teal (●) arrays contain the characters represented at the corresponding position in the bit vector and are not a part of the wavelet tree. Note that all bit vectors on the same depth can be concatenated to a single bit vector, while retaining the same functionality. Σ_α denotes the characters that are represented by the bit vector for $\alpha \in \{\epsilon, 0, 1, 00, 01, 10, 11\}$. All this auxiliary information is not stored explicitly.

Objective

The main objective of this Master's thesis is to design, develop, and benchmark a parallel construction algorithm for bit vector rank and select data structures on GPUs and use the bit vectors to design, develop, and benchmark a state of the art parallel construction algorithm for wavelet tree construction on GPUs. Contributing both algorithms back to the nvbio library is an optional goal.

Requirements

- Excellent C++ programming and CUDA skills
- Interest in string algorithms and compact data structures

[1] `https://nvlabs.github.io/nvbio/`, last accessed 2022-10-10.

Contact

- Dr. Florian Kurpicz (kurpicz@kit.edu)
- Hans-Peter Lehmann (hans-peter.lehmann@kit.edu)

References

[1] Patrick Dinklage, Jonas Ellert, Johannes Fischer, Florian Kurpicz, and Marvin Löbel. Practical wavelet tree construction. *ACM J. Exp. Algorithmics*, 26:1.8:1–1.8:67, 2021.

[2] Florian Kurpicz. Engineering compact data structures for rank and select queries on bit vectors. *CoRR*, abs/2206.01149, 2022.