Overview

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

- \( \text{rank}_\alpha(i) \) returns the number of 1-bits up to position \( i \) and
- \( \text{select}_\alpha(i) \) 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 \( \sigma \) in \( O(\log \sigma) \) time. Here, \( \text{rank}_\alpha(i) \) queries ask for the number of occurrences of the symbol \( \alpha \) before the position \( i \) and \( \text{select}_\alpha(i) \) queries return the text position of the \( i \)-th occurrence of the symbol \( \alpha \).

Let \( T \) be a text of length \( n \) over an alphabet of size \( \sigma \). 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.

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

References


Contact

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

Figure 1: The wavelet tree of \( T = [0, 1, 3, 7, 1, 5, 4, 2, 6, 3] \). The light teal (\( \bullet \)) 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. \( \Sigma_\alpha \) 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.