UNIVERSITÀ DEGLI STUDI DI MILANO

GPU Computing Project

Bitonic Sort using Numba on Nvidia GPUs

Author:

Manuele Lucchi

08659A

Academic Year 2022/2023

1 Abstract

This documents describe the performances of an existing algorithm, the Bitonic Sort, implemented using Python and Numba, comparing it with a serial approach.

Contents

2 Introduction

The Bitonic Mergesort [**bitonic**] is an algorithm created in ANNO by Ken Batcher. The elements to choose for the comparison are indipendent from the value (it's not data-dependant) therefore is well suited for parallel processing.

3 Algorithm

The algorithm is based on the concept of Bitonic Sequence [**bitonic**], a sequence with $x_0 \leq ... \leq x_k \geq x... \geq x_{n-1}$ for some k, $0 \leq k \leq n$, meaning that there are two subsequences sorted in opposite directions.

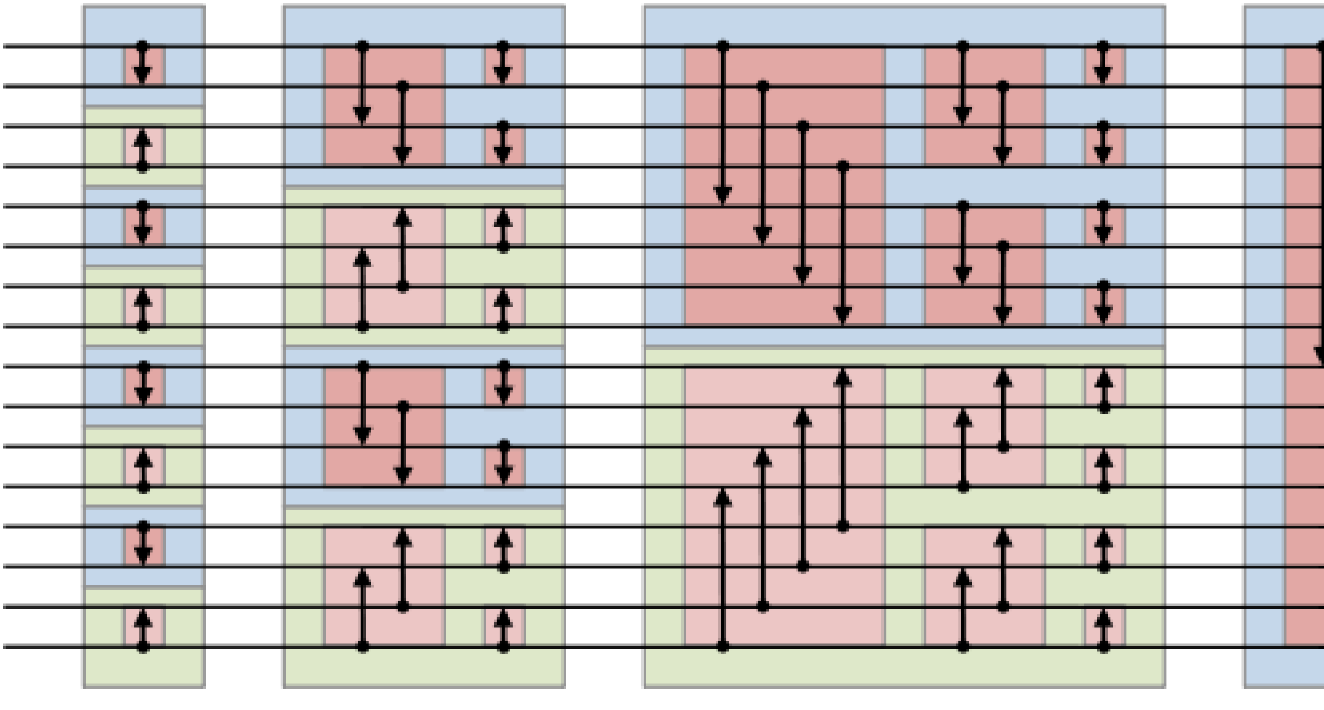
By using a sorting network, we can create a Bitonic Sequence from any sequence and then merging them to obtain the final sorted sequence, so the algorithm has two phases.
This algorithm has an asymptotic complexity of $O(nlog(n)^2)$, the same of the odd-even mergesort [**oddeven**] and shellsort [**shellsort**]

3.1 Step 1: Bitonic Sort

To create a Bitonic Sequence we need to build a sorting network. By sorting each pair of elements in the sequence in different directions pairwise (using the so called ”comparers”), we obtain a sequence full of bitonic subsequences. We can then at each phase double the size of these subsequences and half their number.

3.2 Step 2: Bitonic Merge

The last step is a variation of the first one, where we only have a single bitonic sequence and sort the two subsequences with the comparers oriented in the same direction, resulting in the sorted sequence.



3.3 Sequences of length not power of 2

4 Implementation

5 Benchmark and Profiling

Size	CPU Recursive	CPU Iterative	GPU
10	10 ms	10 ms	10 ms
10	10 ms	10 ms	10 ms

6 Conclusion