

Sequence Alignment

Marco Marasco 834482

Word count: 200

Parallelisation Approach

The approach taken for was to use a “tiling” method that exploited the data dependency in the computation. For a given cell in the DP table (i, j) , to do the required DP computation there was a dependency on cells $(i - 1, j)$, $(i - 1, j - 1)$, $(i, j - 1)$. The first approach was to sequentially compute the values for each antidiagonal (where the initial position was the top left cell, and moved right/down across the top/right border), but cells within each antidiagonal were computed in parallel.

This initial approach incurred massive thread overheads, as each cell was computed in a separate thread. To combat this, inspiration was taken from [1] to reduce these overheads with a ‘tiling’ method. Instead of parallelising along an antidiagonal of individual cells, this approach groups cells into larger (1000 cells \times 1000 cells) tiles, and the computation was parallelised over antidiagonals of tiles, where each thread would sequentially compute the DP values for a given tile. In essence, the DP table was converted to a table of much larger “tiles”, each tile contained a sub-table of the original table, and tiles in the same antidiagonal were computed in parallel.

References

- [1] Y. Liu and B. Schmidt, “Gswabe: Faster gpu-accelerated sequence alignment with optimal alignment retrieval for short dna sequences,” *Concurrency and Computation Practice and Experience*, vol. 27, Mar. 2015. DOI: 10.1002/cpe.3371.