Experimental Algorithms:
Exercise sheet 3

Dr. Alexander van der Grinten

December 15, 2020

Notes

- Due by January 7, 2020, 9:00 AM.
- See the remarks on exercise sheets 1 and 2 (i.e., regarding the grading of exercises and on running your experiments).
- Write `experiments.yml` files to run experiments and use Jupyter notebooks to evaluate results.

Exercise 7: Algorithms on CSR matrices

This exercise uses `ex3-graph-csr.zip` from Moodle.

(a) Implement an algorithm that computes the transpose of a CSR matrix and stores the results in CSR. Try to optimize the performance of your algorithm. Is it easily possible to use parallelism here? What do we expect about the scalability of the algorithm? How does it scale in reality?

Use the graph `cit-patent` from `http://networkrepository.com` to evaluate the performance of your algorithm. Note that the zip already contains code to read the graph.

(b) Implement Dijkstra's algorithm (sequentially) based on the CSR matrix data structure.

To realize the priority queue, you can use the `d_ary_addressable_int_heap` class that is included in the zip file. Note that this class stores only keys (= vertex IDs, in our case) and not any priority value. To achieve the correct order, store the priorities in a vector and override the comparison function to take the priority vector into account.

(c) Evaluate the performance of your Dijkstra implementation for a single SSSP (from a random source vertex) and 1000 SSSP invocations on `gruenau1` or `gruenau2`. Use the graphs `cit-patent` (a citation network) and `roadNet-TX` (a road network) from `http://networkrepository.com` for your experiments.

Exercise 8: Bellman-Ford on CPUs and GPUs

This exercise uses `ex3-graph-csr.zip` from Moodle.

(a) Implement an OpenMP-based parallel implementation of the Bellman-Ford algorithm. Integrate your implementation into the experiments from Exercise 7 (c). Which algorithm is faster? How does your algorithm scale with the number of CPUs?

(a) Implement a CUDA-based GPU implementation of the Bellman-Ford algorithm. Integrate your implementation into the experiments from Exercise 7 (c). When does the GPU outperform the CPU and what is the speedup?

Run the GPU codes on `gruenau9` or `gruenau10`. Note: the example code from the lecture already has a working CUDA "Hello World".