

ParallelUnionFind test client and algorithm

1. Building and running the test client.

Building

The final executable 'ParallelUnionFind.exe' (Windows) or 'ParallelUnionFind' (Linux) resides in the Release folder. To obtain it one has to:

Windows: open the solution file 'ParallelUnionFind.sln', ensure that the Microsoft HPC Pack 2008 SDK is installed and the proper paths to the MPI include/lib directories are set in the project settings. Build the solution.

Linux: go to the 'Release' folder and type: make.

If some debugging information is needed, one can go to the file 'ParallelUnionFind/Makefile', uncomment the '-D _DEBUG' flag in CXXFLAGS and make the executable. In this case some additional output files will be created as well as more information will be printed to the standard output. (TODO: describe the debug output in more detail).

How to run

To run on say 4 processors (with redirecting the standard out to 'file.out'): open a terminal, go to the '/Release' folder and type:

```
mpiexec -n 4 ParallelUnionFind > file.out
```

Note: In the test client each MPI processor reads the input parameters and the picture. This will be a bottleneck when one uses a big number of processors. So it is better to run the test client on a small number of processors (1-16). Also running on big systems may create a problem, so systems with linear sizes of up to $\sim 10^3$ should be a limit (this might depend on the machine in use).

System size and the number of processors must be powers of 2.

2. Project structure.

/ParallelUnionFind/ – the source code.

/Release/ – the executable, the input file and results of tests obtained on Windows.

/test/ – input for testing. Results of the tests obtained on Windows for the input from '/test/' are in '/Release/test_results/'.

3. Code architecture. TODO: add later (patterns and corresponding UML diagrams).

4. Algorithm. TODO: add later (visual illustration of the data structures).

5. Visual Studio settings. TODO: add later (file paths for MPI, MPI debugger settings).