IPVS – Universität Stuttgart WS 17/18
Abteilung Simulation großer Systeme
Prof. Miriam Mehl
Malte Brunn, Amin Totounferoush, Dr.-Ing. Nehzat Emamy

Numerical Simulation
Exercise sheet 2

Parallel implementation of the flow solver
Hand in via email to amin.totounferoush@ipvs.uni-stuttgart.de until 05.12.2017, 18:00
Contact for quesions about the framework malte.brunn@ipvs.uni-stuttgart.de

2.1 Changes of the program layout

As we want to parallelize according to "Single-Program - Multiple-Data", you must not write a program for each processor but you can use one program except for some modifications of the root process. Because every subprogram performs on its assigned sub-domain, the local size of the fields must be dynamically adjusted according to total size of the fields and number of processes. In the next sections, the parts which change through parallelization of the code will be described.

2.1.1 Changes of the fields

Consider the figure in section 7.5.1 of the lecture notes (p. 48). On the process that handles sub-domain $[(il-1)\,\delta x, ir\,\delta x] \times [(jb-1)\,\delta y, jt\,\delta y]$, the fields must have the following dimensions:

- `P` `[il-1,ir+1]`$\times$`[jb-1,jt+1]`
- `U` `[il-2,ir+1]`$\times$`[jb-1,jt+1]`
- `V` `[il-1,ir+1]`$\times$`[jb-2,jt+1]`
- `F` `[il-2,ir+1]`$\times$`[jb-1,jt+1]`
- `G` `[il-1,ir+1]`$\times$`[jb-2,jt+1]`
- `RHS` `[il,ir]`$\times$`[jb,jt]`

2.1.2 Method of communication

When you want to run the flow solver in parallel, you must exchange values along the borders of processes. In order that the proneness of error in the communication step remains manageable, you should exchange the values according to a fixed scheme. The pressure will be exchanged according to the left figure in section 7.5.2 of the lecture notes. The following or similar order should be kept:

a) Send to left – received from right

b) Send to right – received from left

c) Send to top – received from bottom

d) Send to bottom – received from top

The velocities will be exchanged according to the right figure in section 7.5.2 of the lecture notes. Also here the same or similar order should be kept.

2.1.3 Approach to parallelize the flow solver

To summarize, in order to parallelize your flow solver, you should proceed like following:

a) Change the memory allocation of the fields as described in section 2.1.1.

b) The indices i and j are allowed to go up only to the size of the sub-domain.

c) Boundary values are allowed to be set just on the sub-fields, which lie on the boundary of the sub-domain.

Additionally extend the output of both visualization and VTK, to the vorticity and stream function as described in section 2.3.

2.2 Performance measurment

Execute computations with respectively $p = 2, 4, 8, \dots$ (hardware-) processors and plot the actual speedup $S(p)$ together with the theoretical speedup $S_t(p)$. Moreover, plot the parallel efficiency $E(p)$ in another diagram. To draw the diagram you can use a software of your choice, for example, gnuplot[7], octave[8], matplotlib[9], R[10] or LibreOffice[11].

Remark 1: In order to execute MPI-processes over several processors, you need a hosts/machine file, in which you specify the machines that are available for the run. The machine file is formatted so that one host exits on each line, either `hostname` oder `hostname:n`, where n stands for the number of available processes. Moreover, you should undertake the ssh-identification over ssh-keys[12] instead of passwords.

[7] http://www.gnuplot.info/

[8] https://www.gnu.org/software/octave/

[9] http://matplotlib.org/

[10] http://www.r-project.org/

[11] https://www.libreoffice.org/

[12] https://wiki.archlinux.org/index.php/SSH_Keys

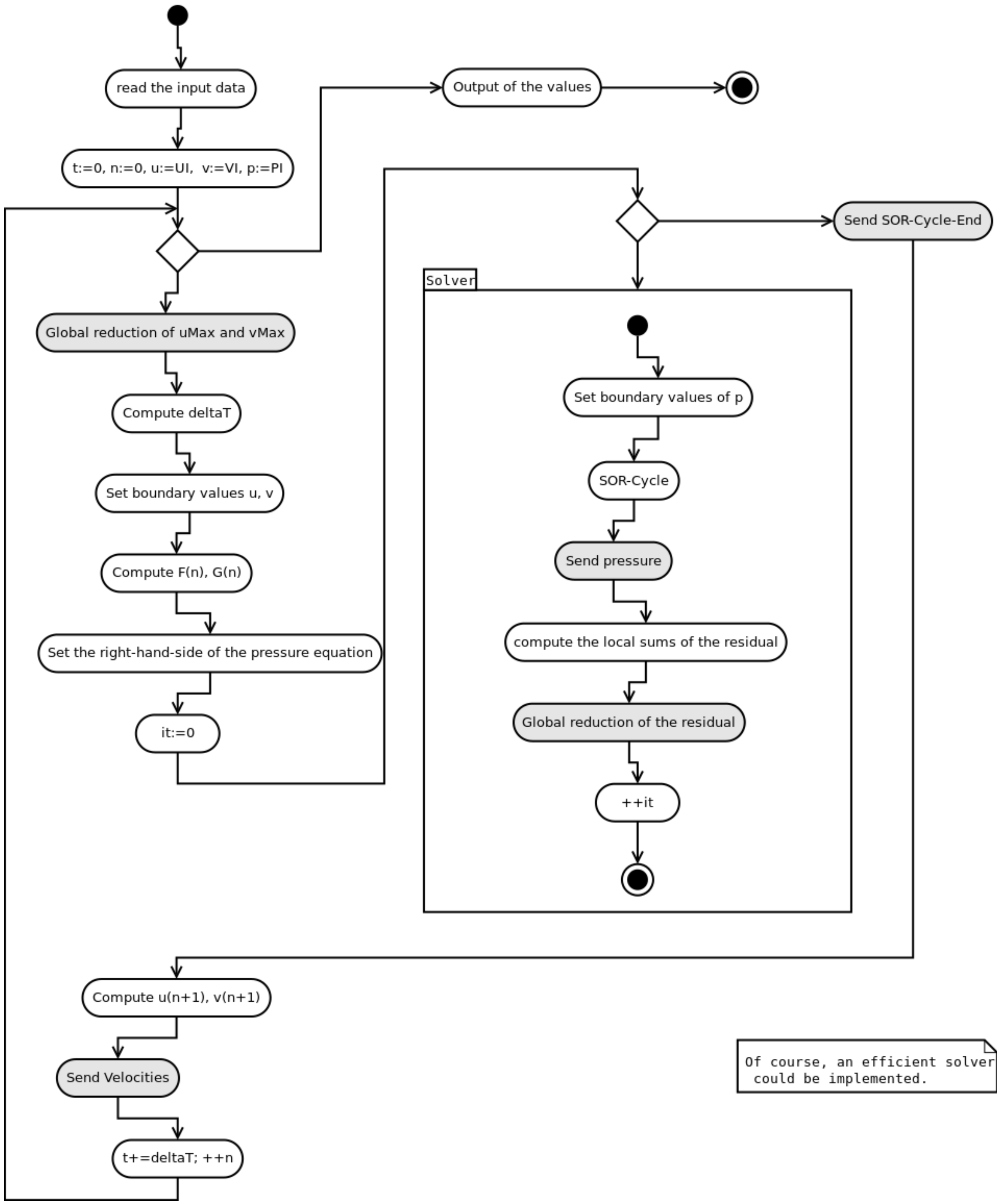

Figure 8: Activity diagram with communication methods.

2.3 Extended visualization

There exists the possibility of formulating the Navier-Stokes equations not only with the "primitive" variables u, v and p, but also Ψ and ζ, the stream function and vorticity, respectively. Both values are important for a better analysis of the simulated flow.

2.3.1 Implementation

Extend the components of *IO* from the material from ILIAS such that the stream function and vorticity, the discrete values $\varphi_{i,j}$ and $\zeta_{i,j}$, would be computed, saved in the VTK file and visualized with Paraview. Besides, visualize the contours of the stream function.

2.4 Questions

- For one iteration of the pressure equation, provide the magnitude of the computation and communication times in dependence of number of processes.
- Provide the computational times for the global reduction of the residual and finding out the time step in dependence of the number of processes.
- Study the strong-scaling of your implementation, in which the run time and parallel efficiency are plotted over the number of processes.