Exercises for Advanced Topics in High Performance Scientific Computing
WS 15/16, Nr.: 0000003213, 02.10.011, Rechnerraum
Exercise Sheet 7 (until December 2, 2015)

Exercise 7*
Solve the linear system $Au = f$ with the Jacobi method.

$$u^{(0)} := u_0 \tag{1}$$

$$u^{(k+1)} := u^{(k)} + \operatorname{diag}(A)^{-1}\left(f - Au^{(k)}\right),\ k > 0 \tag{2}$$

The system matrix $A \in \mathbb{R}^{n\times n}$ is given as a sparse matrix stored in CRS format in the files `cnt.bin`, `col.bin`, and `val.bin` and the initial guess and the right hand side are stored in the files `u0.bin` and `f.bin`. The count vector $cnt \in \mathbb{N}_0^n$ and the column indices $col \in \mathbb{N}_0^m$ are stored as 32 bit integers, while the corresponding matrix entries $val \in \mathbb{R}^m$ and the initial guess and the right hand side $u_0, f \in \mathbb{R}^n$ are stored as 64 bit double precision floating point numbers. (Test data: `/share/apps/crs.zip` on `mephisto.uni-graz.at`)
In order to parallelize the sparse matrix-vector multiplication for the Jacobi iteration for P processes, split the matrix row-wise into P blocks

$$A_p \in \mathbb{R}^{I_p\times I},\ 0 \le p < P \tag{3}$$

where $I := \{i \in \mathbb{N}_0 : 0 \le i < n\}$ and the index sets I_p are defined as follows:

$$I_p := \{i \in \mathbb{N}_0 : d_p \le i < d_p + c_p\} \tag{4}$$

with the the counts c_p and displacements d_p given by

$$c_p := \begin{cases} q+1, & p < r \\ q, & p \ge r \end{cases},\quad d_p := \begin{cases} pq+p, & p < r \\ pq+r, & p \ge r \end{cases} \tag{5}$$

with the remainder and the quotient defined as

$$r := n \mod P, \quad q := (n-r)/P\ . \tag{6}$$

Use the parallel I/O functions of MPI, `MPI_File_open`, `MPI_File_read_at`, and `MPI_File_close` to read the matrix blocks $A_p \in \mathbb{R}^{I_p\times I}$ directly from the CRS files. (Hint: Every process p first reads the partial cnt vector restricted to the index set I_p and sums the counts to determine the number of non-zero elements m_p of A_p. Then the offsets into the `col.bin` and `val.bin` files can be calculated with a prefix sum over m_p using the `MPI_Scan` function.)
Note that the parallel matrix-vector multiplication requires in general the full vector u to be present on every process. Define the restriction of a vector $v \in \mathbb{R}^I$ to the index set I_p as

$$v_p := v|_{I_p} \in \mathbb{R}^{I_p} \tag{7}$$

then the matrix-vector multiplication in the Jacobi iteration on process p reads

$$u_p^{(k+1)} = D_p^{-1}(f_p - A_p u^{(k)}) \tag{8}$$

with the restriction of the diagonal matrix of A defined as $D_p := \operatorname{diag}(A)|_{I_p\times I_p}$. Use this approach to implement a first version of the parallel Jacobi iteration.
Furthermore the communication required between all the processes to collect the vector fragments u_p to assemble the full vector u can be improved by restricting u to the column index set J_p containing all column indexes appearing in the matrix A_p. Use this information to setup a sparse communication pattern to assemble $u|_{J_p}$ on every process using `MPI_Alltoallv` for an improved parallel Jacobi solver.

* Place all source files of the exercises in a folder named `Exercise7` in your home directory on the `mephisto.uni-graz.at` cluster.