Parallel Programming 2015, Assignment 1: Sparse LU Factorization

Deadline: Monday, March 30 before 23:59 hours.

In the first programming assignment, you will put what you have learned during the Sparse Matrix Computations lecture to practice. You will do so by first implementing a Sparse LU Factorization kernel and secondly performing a benchmark of your implementation.

The deadline for the assignment is March 30. The assignment has to be completed individually. You are expected to hand in a tarball containing your source code and a report detailing how you have dealt with the various implementation issues and your benchmark results in PDF format. The assignments can be handed by e-mail to *krietvel (at) liacs (dot) nl*.

1 Implementation

The first part of the assignment consists of the implementation of a Sparse LU Factorization kernel, that applies partial pivoting. You need to implement a *sequential* program. The kernel must be written in C/C++ using solely arrays, so without the use of pointer-linked data structures.

On the website a skeleton code is provided that reads matrices in the "Matrix Market" format and stores these in Compressed Row Storage format. You can use this skeleton as a starting point for your implementation.

In your PDF report, detail how you have dealt with the various implementation issues such as pivot search, masking operations, garbage collection and permutation issues.

2 Benchmarking

In the second part of the assignment, you will benchmark the LU factorization code you have implemented. To be benchmarked are the Euclidean norm $||Ax - b||_2$ and the execution time required to compute the solution. The benchmarks need to be carried out using the Linux operating system on the Dell Precision T1650 Core i7 machines in room 302/304 at LIACS. As test data, we will use 10 matrices from the University of Florida Sparse Matrix Collection (`http://www.cise.ufl.edu/research/sparse/matrices/`):

HB/mcfe	Lucifora/cell1
Schenk_IBMNA/c-21	Gaertner/nopoly
Oberwolfach/flowmeter5	Bai/mhd4800b
Averous/epb1	FIDAP/ex10
Grund/meg4	Okunbor/aft01

Given a matrix A, you need to construct the B vector as follows: $B(i) = \text{sum}(i^{th}$ row of $A)$. This will cause the solution vector to consists of all ones.

Include the obtained results in your report, together with a brief discussion.