A reproducible comparison between GNU MPFR and machine double-precision

Reproducible Experimental Setup

Local version

Several authors claim that GNU MPFR [1] is x times slower than double-precision floating-point numbers, for various values of x, without any way for the reader to reproduce their claim. For example in [2], Joris van der Hoeven writes “the MPFR library for arbitrary precision and IEEE-style standardized floating-point arithmetic is typically about a factor 100 slower than double precision machine arithmetic”. Such a claim typically: (i) does not say which version of MPFR was used (and which version of GMP, since MPFR being based on GMP, its efficiency also depends on GMP); (ii) does not detail the environment used (processor, compiler, operating system); (iii) does not explain which application was used for the comparison. Therefore it cannot be reproduced by the reader, which could thus have no confidence in the claimed factor of 100. In this short note we provide reproducible figures that can be checked by the reader.

Reproducible Experimental Setup

We use the programs in appendix to multiply two 1000×1000 matrices. The matrix A has coefficients $1/(i+j+1)$ for $0 \leq i, j < 1000$, and matrix b has coefficients $1/(ij+1)$. Both programs print the time for the matrix product (not counting the time to initialize the matrix), and the sum of coefficients of the product matrix (used as a simple checksum between both programs).

We used MFPR version 3.1.5, configured with GMP 6.1.2 (both are the latest releases as of the date of this document).

We used as test processor `gcc12.fsffrance.org`, which is a machine from the GCC Compile Farm, a set of machines available for developers of free software. The compiler used was GCC 4.5.1, which is installed in `/opt/cfarm/release/4.5.1` on this machine, with optimization level `-O3`. Both GMP and MPFR were also compiled with this compiler, and the GMP and MPFR libraries were linked statically with the application programs (given in appendix).

Author: Paul Zimmermann (reproduction with org-mode by Arnaud Legrand)

Created: 2018-11-23 Fri 08:50