

Распределенные символьные дробно-рациональные вычисления на процессорах x86 и x64

В.А. Голодов

Кафедра экономико-математических методов и статистики,
ФГБОУ ВПО ЮУрГУ (НИУ)

В рамках предыдущих исследований были созданы классы `overlong` и `rational`, реализованные в объектно-ориентированной парадигме на языке C++, а также библиотека классов `Exact Computational` [1]. Классы `overlong` и `rational` имеют MPI оболочки и, тем самым, позволяют производить безошибочные дробно-рациональные вычисления в параллельных средах. Для классов `overlong` и `rational` определены все операторы, операции и бинарные отношения, используемые для стандартных числовых типов данных.

На сегодняшний день возможность использования безошибочных вычислений представляет известная библиотека GMP (The GNU Multiple Precision Arithmetic Library) однако актуальная версия GMP 5.0.x **НЕ** предоставляет своим объектам возможность их использования в параллельных вычислениях (<http://gmplib.org/gmp5.0.html>).

Поскольку первые редакции описанных классов были выпущены в 1999 году, к настоящему времени выявился ряд недостатков относительно современного состояния вычислительной техники. Была проведена работа по актуализации кода, часть изменений, внесенных в базовый тип `overlong`, представлена ниже.

1. Основание системы счисления было заменено с 2^{16} на 2^{32} . Это существенно сокращает длину обрабатываемого массива цифр, а значит и время вычислений в целом.

2. Работа с памятью была организована более эффективно.

3. Были задействованы преимущества основания системы счисления вида 2^{32} за счет быстрых логических операций.

4. Были пересмотрены реализации операций. При делении и взятии остатка от деления удалось избавиться от порождения временных объектов.

5. Код класса был переписан в терминах обобщенного программирования. Это упростило операции со смешанными типами и сократило объем сопровождаемого кода.

6. MPI оболочка типа `overlong` изменена в соответствии с новой внутренней структурой класса.

7. Все изменения обеспечивают обратную совместимость кода (необходимое требование к вносимым изменениям).

Для демонстрации эффективности усовершенствований был проведен вычислительный эксперимент по нахождению обратной матрицы Гильберта алгоритмом Жордана-Гаусса с применением исходных классов `overlong` и `rational` и их последних модификаций. Для матриц размера порядка 300 ускорение составило более 12 раз, с ростом размерности матрицы ускорение возрастает. Объем требуемой оперативной памяти для обработки матрицы порядка $n = 1000$ на одном вычислительном узле составил менее 1 Гб.

Дальнейшие исследования направлены на оптимизацию кода, а также на параллельную реализацию базовых арифметических операций. Привлекательным является использование для этого графических ускорителей.

Литература

1. Панюков А.В., Германенко М.И., Горбик В.В. Библиотека классов «Exact Computational», // Программы для ЭВМ, базы данных, топологии интегральных микросхем: официальный бюллетень Рос. агентства по патентам и товарным знакам № 3. - 2009. - С. 251.