

Библиотека параллельной арифметики многократной точности для высокопроизводительных систем*

К.С. Исупов, В.С. Князьков

Вятский государственный университет

При решении больших задач на высокопроизводительных системах 64-битной арифметики с плавающей точкой IEEE часто оказывается недостаточно для получения корректных результатов. Возникает необходимость использования высокоточных вычислений. В этой работе рассмотрены актуальные приложения высокоточных вычислений. Представлен обзор существующего программного обеспечения. Обсуждаются требования к перспективным программным средствам. Рассмотрена новая библиотека высокоточной арифметики MF-Library. В этой библиотеке для представления чисел с плавающей точкой произвольной длины используется система остаточных классов. Это обеспечивает эффективное выполнение основных высокоточных арифметических операций с распараллеливанием обработки отдельных цифр мантиссы. MF-Library реализует концепцию потоковой безопасности, что позволяет использовать ее на системах с общей памятью. Представлены экспериментальные оценки эффективности MF-Library.

1. Введение

Рост производительности современных компьютеров позволяет решать все более и более сложные задачи, что приводит к необходимости выполнять большое количество операций с плавающей точкой, почти каждая из которых сопровождается ошибкой округления. Эксафлопсный уровень скорости вычислений (10^{18} операций в секунду), вероятно, будет достигнут в ближайшие десять лет и получить точные результаты на таких вычислительных системах в машинной арифметике IEEE-754 будет непростой задачей [1].

Уже сейчас многие научные и промышленные вычисления требуют использования арифметики многократной точности. Перечислим наиболее распространенные из них.

1. Плохо обусловленные линейные системы. Многие проблемы нехватки точности связаны с необходимостью решения плохо обусловленных систем большого порядка. Например, в задачах строительной механики возникают системы с десятками миллионов неизвестных и с числом обусловленности $10^9 - 10^{13}$ [2]. Также задачи данного класса возникают при изучении кулоновских атомных систем [3] и при исследовании электромагнитного рассеяния [4, 5]. В задаче среднеквадратичной аппроксимации непериодических функций неортогональным степенным базисом возникает необходимость решения систем с матрицей Гильберта, которая известна плохой обусловленностью [6].
2. Рекуррентные формулы и большие суммы. Аномальные результаты часто связаны с потерей ассоциативности при суммировании, особенно при его выполнении на параллельной компьютерной системе, где порядок суммирования не может контролироваться [7, 8]. Необходимость вычисления больших сумм возникает, в частности, при использовании квадратурных формул для интегрирования полиномов высоких степеней на сетках с малым шагом. К практическим задачам, основанным на вычислении рекуррентных соотношений, чувствительных к ошибкам округления, относится реализация дискретного преобразования Фурье в форме рекурсивного фильтра – алгоритм Гёрцеля [9], а также рекурсивное дискретное косинус-преобразование [10]. Эти задачи имеют большое значение в цифровой обработке сигналов. В теории приближений

*Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 14-07-31075 мол_а.

важную роль играют полиномы Чебышева, которые также вычисляются с помощью рекуррентных соотношений.

3. Жесткие системы дифференциальных уравнений. Такие системы возникают в задачах химической кинетики с одновременным присутствием очень медленно и очень быстро протекающих химических реакций [11], при исследовании суточных колебаний озона в атмосфере [12], а также в задачах расчета динамики многозвенных систем [11].
4. Крупномасштабное моделирование. Вычисления, устойчивые на небольших задачах, выполняемые на однопроцессорных системах, могут сопровождаться значительными численными ошибками при масштабировании до уровня массивно-параллельных систем. К крупномасштабным задачам относится моделирование климата [4, 7, 13, 14], моделирование атмосферы сверхновых звезд [7], моделирование процессов, протекающих в ядерных реакторах, расчет дозвукового обтекания летательного аппарата [13].
5. Продолжительное моделирование. Практически любой процесс физического моделирования, выполняемого в течение длительного времени, в конечном итоге отходит от реальности. Это связано с накоплением ошибок округления в дополнение к ошибкам, связанным с дискретизацией по времени и пространству. Показательным примером является задача численного исследования орбитальной эволюции небесных тел [15, 16].
6. Экспериментальные математические расчеты. Многие результаты в этой сравнительно молодой области научных исследований (например, формула Бэйли – Боруэйна – Плаффа для вычисления числа π) не могли быть получены без использования вычислений с очень высокой точностью [7, 16].

Приведенные области применения высокоточных вычислений характеризуются, как правило, большой размерностью задач. Вместе с тем, даже при малом объеме вычислений в машинной арифметике может быть получен результат, не содержащий ни одной значащей цифры [17–19]. Таким образом, проблема обеспечения приемлемой точности вычислений актуальна в настоящее время и с ростом объемов производимых вычислений, ее значение, несомненно, будет возрастать.

2. Программное обеспечение высокоточных вычислений

Одним из способов повышения точности в настоящее время является 128-битный IEEE формат [20], в котором поле мантииссы расширено до 113 разрядов. Однако аппаратная поддержка этого формата требует значительных затрат [16] и, судя по всему, не предвидится в ближайшей перспективе. Более распространенным вариантом является длинная арифметика — программная обработка чисел, разрядность которых превышает стандартную длину машинного слова вычислительной машины. В настоящее время существует достаточно широкий спектр библиотек длинной арифметики. К наиболее известным относятся следующие.

1. QD [21]. Пакет расширенной точности, поддерживает два формата данных: double-double (≈ 32 десятичные цифры) и quad-double (≈ 64 десятичные цифры). Имеет высокоуровневые интерфейсы языков C++ и Fortran-90, что обеспечивает конвертацию существующих программ с минимальным изменением исходного кода. Доступен для скачивания по адресу: <http://crd-legacy.lbl.gov/~dhbailey/mpdist>.
2. ARPREC [21]. Пакет произвольной точности, включает процедуры арифметических вычислений, а также многих алгебраических и трансцендентных функций. Поддерживает вычисления с вещественными, целыми и комплексными числами. Имеет интерфейсы C++ и Фортран-90. Доступен по адресу: <http://crd-legacy.lbl.gov/~dhbailey/mpdist>.

3. GMP [22]. Библиотека произвольной точности, имеет обширный набор оптимизированных процедур для поддержки вычислений с целыми, рациональными и вещественными числами. Режимы округления, совместимые со спецификациями IEEE-754, не поддерживаются. Имеет интерфейс языка С. Доступна на сайте <http://gmpmath.org>.
4. MPFR [23]. Расширение GMP, обеспечивающее вычисления многократной точности с возможностью использования одного из четырех режимов округления, соответствующих стандарту IEEE-754. Точность может быть установлена отдельно для каждой переменной. Нормализованные числа не поддерживаются. Выпускается под лицензией GNU LGPL. Доступна по адресу: <http://www.mpfr.org>. Обладает высоким быстродействием, по сравнению со многими аналогами.
5. NTL [24]. Портативная C++ библиотека для решения задач теории чисел. Включает структуры данных и алгоритмы обработки целых чисел любой длины, векторов, матриц и полиномов над целыми числами и над конечными полями, а также арифметику с плавающей точкой произвольной точности. Достоинство NTL — согласованный интерфейс с большим разнообразием классов, представляющих математические объекты. Доступна по адресу <http://www.shoup.net/ntl/doc/tour-intro.html>.
6. MPFUN2015 [25]. Пакет произвольной точности, являющийся развитием MPFUN90. Имеет интерфейс языка Фортран-90, планируется частичная поддержка интерфейса C++. Поддерживает вещественные и комплексные типы данных. К основным заявленным преимуществам пакета относится потоковая безопасность. Для сохранения приемлемой производительности при работе в режиме крайне высокой точности используются алгоритмы на базе быстрого преобразования Фурье. В состав пакета входят подпрограммы вычисления алгебраических, трансцендентных и некоторых специальных функций, таких как гамма-функция, неполная гамма-функция, дзета-функция. Доступен по адресу: <http://www.davidhbailey.com/dhsoftware>.

Представленные программные средства высокоточной арифметики являются в целом более эффективными и надежными, чем их предшественники. Вместе с тем, возникают новые требования, которые необходимо учитывать при разработке перспективного высокоточного программного обеспечения, эффективно работающего на вычислительных системах экзафлопсного класса. Рассмотрим эти требования.

1. *Высокая скорость.* Данный параметр является основополагающим. К недостаткам современного программного обеспечения высокоточной арифметики относится ярко выраженная зависимость времени вычислений от точности. Отмечается, в частности, что вычисления в формате double-double в среднем в 5 раз медленнее, чем в 64-битном формате, в формате quad-double — в 25 раз медленнее. При использовании библиотек произвольной точности время вычислений может возрасти в сотни и тысячи раз [16]. Это недопустимо при решении многих крупных задач, критичных к ошибкам округления и скорости вычислений.
2. *Потоковая безопасность.* При решении крупных задач расчеты многократной точности могут выполняться параллельно с использованием Message Passing Interface (MPI). Однако высокая производительность современных систем, построенных на базе многоядерных процессоров, обеспечивается, во многом, за счет использования разделяемой памяти. Поэтому важно иметь возможность распараллеливать высокоточные вычисления в пределах одного узла, даже если MPI используется для параллелизма между узлами. Это требует от программных средств потоковой безопасности, что означает, среди прочего, отсутствие глобальных переменных, в которые происходит запись.

Из существующих программных пакетов высокоточных вычислений потоковая безопасность задекларирована лишь в MPFUN2015 и GMP. Также библиотеки MPFR и MPFR C++ предусматривают потоково-безопасный вариант сборки.

3. *Поддержка современных параллельных архитектур.* В основе большинства суперкомпьютеров лежит гибридная архитектура, которая предполагает совместное использование центральных (CPU) и графических (GPU) процессоров. Сегодня уже очевидно, что вычислительные архитектуры эксафлопсной производительности будут гибридными. Поэтому одно из главных требований к перспективным алгоритмам и программному обеспечению арифметики многократной точности – высокая эффективность при реализации как на CPU-, так и на GPU-узлах. Существующие же средства ориентированы, главным образом, на универсальные процессоры, а GPU-реализации представлены слабо. Из высокоточных библиотек, реализующих все арифметические операции, тригонометрию и ряд других математических функций, отмечаются лишь GARPREC и GQD [26], причем в основе GARPREC лежат весьма затратные алгоритмы, а GQD поддерживает только расширенную точность (форматы “double-double” и “quad-double”). Поэтому при решении прикладных задач расчеты многократной точности выполняются на CPU, а вычислительные мощности GPU-узлов не задействуются, что, с учетом дополнительных задержек на пересылку данных, влечет существенное увеличение времени решения. Такое обстоятельство неприемлемо во многих случаях, особенно при обработке в реальном времени.

Кроме этого, необходимы дополнительные исследования, чтобы оценить возможность продуктивного использования для целей высокоточных вычислений многоядерных ускорителей, таких как Intel MIC, и программируемых логических интегральных схем, таких как FPGA.

4. *Поддержка произвольного уровня точности и широкого перечня функций.* Уже сейчас некоторые развивающиеся приложения требуют очень высокого уровня точности (10000, 50000 или более цифр). Поэтому перспективные средства высокоточной арифметики должны использовать передовые структуры данных и алгоритмы на их основе, которые остаются эффективными, в том числе и для очень высокой точности вычислений. В этом направлении уже недостаточно реализации основных арифметических операций с произвольной точностью, так как возникает удивительно широкий спектр трансцендентных и специальных функций, требующих высокоточной оценки. Выделяется, в частности следующий перечень функций, которые должны поддерживать перспективные высокоточные пакеты [16]:

- основные трансцендентные функции – \exp , \log , \sin , \cos , \tan , гиперболические функции и соответствующие им обратные функции;
- гамма, дигамма, полигамма, неполная гамма, бета и неполные бета-функции;
- дзета-функция Римана, полилогарифмы и L-функция Дирихле;
- функции Бесселя (первый, второй и третий виды, модифицированные, и т.д.);
- гипергеометрические функции;
- функции Эйри;
- эллиптические интегралы;
- эллиптические функции Якоби и Вейерштрасса;
- тэта-функции.

Ознакомиться с представленными критериями эффективности высокоточного программного обеспечения, а также найти другие критерии, можно найти в работе [16], которая и взята за основу изложенного материала.

Следует отметить, что в настоящее время значительно возрастает роль векторных (SIMD) вычислений, в том числе и на центральных процессорах, длина векторных регистров которых неизменно увеличивается. В связи с этим важным требованием является возможность распараллеливания высокоточных арифметических операций на уровне отдельных цифр многоразрядных мантисс. Но такая задача весьма трудоемка в рамках модели позиционной длинной арифметики, которая предполагает в процессе вычислений учет возможных переносов между соседними блоками цифр многоразрядной мантиссы. В результате этого алгоритмы обработки длинных чисел сильно ветвятся и не распараллеливаются, что противоречит основной концепции SIMD-вычислений. Распараллеливание на уровне отдельных цифр мантиссы наиболее актуально при реализации операций с очень высокой точностью, которые выполняются во много раз медленнее операций машинной точности.

Далее рассматривается новая библиотека высокоточных вычислений, нацеленная на удовлетворение перечисленных требований. В основе библиотеки лежит формат представления длинных чисел, ориентированный на параллельную обработку.

3. MF-Library – программная библиотека параллельных высокоточных вычислений

3.1. Тип данных

В библиотеке MF-Library для внутреннего представления длинных чисел с плавающей точкой используется модулярно-позиционный формат (MF-формат) [27,28], описание полей которого представлено в таблице 1.

Таблица 1. MF-формат для представления чисел с плавающей точкой произвольной длины

Поле	Базовый тип	Описание
sign	int	Знак числа
residue	long[n]	Мантисса числа в системе остаточных классов
exp	int	Двоичный порядок (экспонента)
ic_bot	double	Нижняя граница интервально-позиционной характеристики
ic_top	double	Верхняя граница интервально-позиционной характеристики

В MF-формате для представления длинной мантиссы M используется система счисления с параллельной структурой — система остаточных классов (СОК) [29, 30]. В соответствии с этим, мантисса представляется набором остатков $residue_1, residue_2, \dots, residue_n$ по модулям СОК p_1, p_2, \dots, p_n , где $residue_i \equiv M \pmod{p_i}$. Обработка $residue_i$ по различным модулям p_i выполняется параллельно. Мантисса может принимать значения из диапазона $[0, P)$, где P — произведение всех p_i . Следовательно, варьирование количества модулей позволяет задавать произвольную точность вычислений. Обеспечить достаточно большой динамический диапазон, что важно в научных вычислениях, где часто требуется обработка величин очень большого или очень малого масштаба, позволяет двоичный порядок (exp). Значение числа в MF-формате определяется следующим выражением:

$$x = (-1)^{\text{sign}} \times \left| \sum_{1 \leq i \leq n} residue_i \cdot B_i \right|_P \times 2^{\text{exp}},$$

где $B_i = P_i \cdot |P_i^{-1}|_{p_i}$, $P_i = P/p_i$ и $|P_i^{-1}|_{p_i}$ — мультипликативная инверсия P_i по модулю p_i .

MF-формат обладает следующими основными характеристиками точности: машинный эpsilon – $\epsilon = 2^{-\lfloor \log_2(P-1) \rfloor}$, unit in the last place – $ulp(x) = 2^{\exp - \lfloor \log_2((P-1)/M) \rfloor}$, где M – десятичное значение мантиссы. Абсолютные и относительные ошибки округления ограничены, соответственно, величинами $ulp(x)$ и 2ϵ при округлении мантиссы усечением [27]. Точность вычислений (в терминах IEEE-754 точность соответствует разрядности мантиссы) определяется половиной длины диапазона изменения мантиссы, т.е. величиной $\log_2 \sqrt{P - 1}$.

Дополнительно в MF-формат включена атрибутивная информация – интервально-позиционная характеристика (ИПХ) мантиссы, представленная двумя направленно округленными двоичными числами с плавающей точкой – нижней (ic_bot) и верхней (ic_top) границами. ИПХ не участвует в образовании значения числа, но позволяет в значительной степени преодолеть основной недостаток СОК – высокую сложность немодульных операций, таких как сравнение, определение знака, контроль переполнения, масштабирование и пр. ИПХ локализует отношение величины мантиссы M к произведению модулей P так, что $ic_bot \leq M/P \leq ic_top$. Вопросы использования ИПХ в немодульных вычислениях обсуждаются в [27, 31].

3.2. Структура и функциональность

Структурно MF-Library включает в себя три слабосвязанных уровня, каждый из которых объединяет однотипные модули (Рис. 1). Связь уровней реализуется посредством API, который также доступен пользователю.

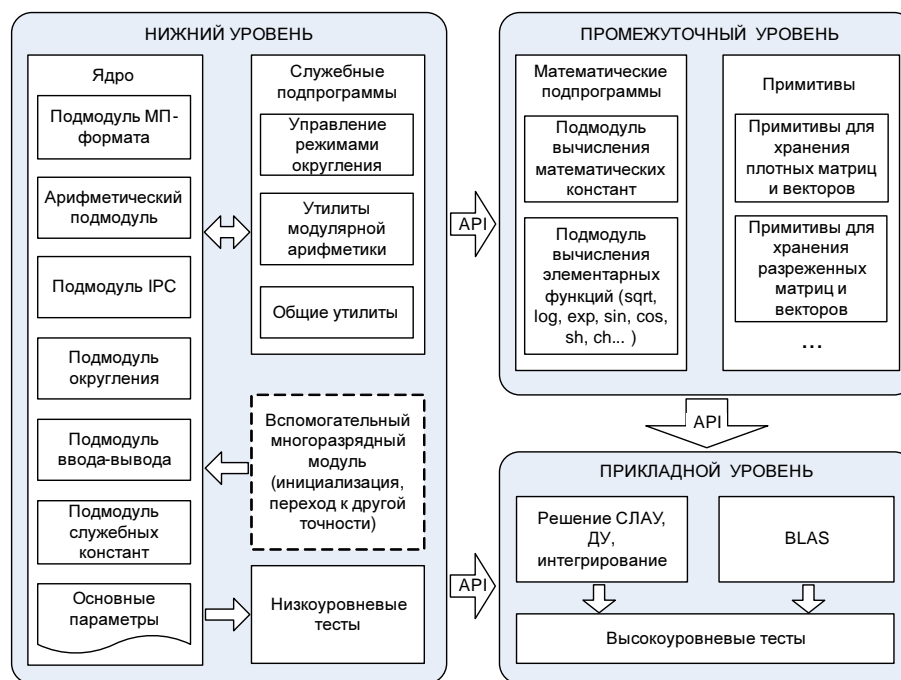


Рис. 1. Структура MF-Library

Центральной частью пакета является *модуль ядра*. В подмодуле MF-формата объявлен основной тип данных, реализованы функции по выделению и освобождению памяти, а также подпрограммы инициализации основных статических объектов, используемых другими подмодулями. Арифметический подмодуль реализует следующие алгоритмы высокоточных вычислений: сложение, вычитание, умножение, деление, выравнивание порядков, сравнение. Подробное описание этих алгоритмов с оценками эффективности можно найти в [27, 28]. Подмодуль ИПС реализует алгоритмы вычисления и анализа интервально-позиционных характеристик, необходимые для быстрого выполнения немодульных операций над мантиссами. Подмодуль округления включает в себя подпрограммы модулярного масштабирования

ния степени двойки, проверки необходимости округления и округления чисел. Подмодуль ввода / вывода обеспечивает следующие возможности: ручной ввод MF-числа, установка MF-числа из форматов double и mpfr_t (многоразрядный тип данных библиотеки MPFR), установка случайного MF-числа, преобразование MF-числа в тип double (с возможной потерей точности), форматированная печать. В подмодуле служебных констант объявлены идентификаторы статичных объектов, вычисляемых при инициализации пакета. В подмодуле основных параметров содержатся управляющие флаги и константы, определяющие все аспекты работы пакета. К их числу относятся: точность вычислений, модули СОК, флаг векторизации, флаг режима отладки, флаг работы в “тихом” режиме, в котором игнорируются некоторые некритичные исключения арифметики с плавающей точкой.

В модуле служебных подпрограмм содержатся утилиты для управления режимами округления, генераторы случайных чисел, функции преобразования типов, базовые алгоритмы модулярной арифметики: получение мультипликативной и аддитивной инверсии, алгоритм Евклида, преобразование в позиционную систему, алгоритм генерации оснований для заданной точности вычислений и пр. Для отладки ядра в состав пакета включен модуль низкоуровневых тестов. Для автоматической конфигурации MF-Library и перехода на другой базис СОК (при изменении точности вычислений) используется вспомогательный модуль длинной позиционной арифметики (MPFR+GMP).

Промежуточный уровень реализует итерационные методы высокоточного вычисления математических констант и некоторых классов элементарных функций.

На прикладном уровне реализованы высокоточные матричные, матрично-векторные и векторные операции, входящие в состав BLAS (GEMM, GEMV и пр.), метод сопряженных градиентов для решения больших систем уравнений и конечно-разностный метод решения краевой задачи теплопроводности.

В настоящее время завершена работа над нижним уровнем пакета. Ведутся работы по расширению функциональности промежуточного и прикладного уровней. В результате этих работ ожидается, что на промежуточном уровне будут реализованы все представленные в предыдущем разделе функции, а также будут определены оптимизированные примитивы для эффективного хранения и обработки плотных и разреженных матриц. На прикладном уровне будут реализованы методы численного интегрирования, дифференцирования и методы решения некоторых задач для дифференциальных уравнений.

3.3. Экспериментальная оценка MF-Library

Для экспериментальной оценки корректности и эффективности методов каждого уровня пакета MF-Library были разработаны и выполнены соответствующие тесты. Остановимся на рассмотрении результатов некоторых из них¹.

Расчеты с катастрофической потерей точности. Исследовалась корректность MF-Library при решении задач, которые даже при незначительном объеме вычислений сопровождаются возникновением катастрофических погрешностей. Такие задачи часто используются для верификации высокоточных программных средств. Первая задача заключалась в вычислении следующего полинома восьмой степени (S.M. Rump, 1988 [17]):

$$f(a, b) = 333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + a/(2b), \quad (1)$$

при $a = 77617.0$ и $b = 33096.0$. В качестве эталонного использовалось решение, полученное с использованием 4096-битной арифметики библиотеки MPFR. Результаты представлены в таблице 2. Более высокая точность MF-Library по сравнению с MPFR (256 бит) объясняется использованием предвычислительной схемы округления [28], которая позволяет использовать для представления мантисс весь динамический диапазон, а не его половину.

¹Во всех экспериментах точность вычислений составляла 239 бит (72 десятичные цифры). Для этого СОК была задана 32 15-битными модулями, обеспечивающими представление 479-битных мантисс.

Таблица 2. Результаты вычисления полинома (1)

Тип данных (точность)	Результат	Число верных цифр
Float (24 бит)	-6.33825×10^{29}	0
Double (53 бит)	-1.18059×10^{21}	0
MPFR (64 бит)	-5.76461×10^{17}	0
MPFR (128 бит)	0.8273960599...	39
MPFR (256 бит)	0.8273960599...	77
MF-Library (239 бит)	0.8273960599...	140

Вторая исследованная задача – вычисление рекуррентного соотношения Мюллера [18]:

$$\begin{aligned}
 f(y, z) &= 108 - (815 - 1500/z)/y, \\
 x_0 &= 4.00, \quad x_1 = 4.25, \\
 x_i &= f(x_{i-1}, x_{i-2}).
 \end{aligned}
 \tag{2}$$

Параметры этого соотношения подобраны таким образом, что при точных вычислениях $\lim_{n \rightarrow \infty} x_n = 5.0$. Однако из-за влияния ошибок округления последовательность $\{x_n\}$ отходит от верного ответа к неподвижной для данного рекуррентного соотношения точке $x^* = 100.0$. Номер итерации, с которой начинается такой переход, прямо пропорционален точности вычислений. Результаты эксперимента представлены на Рис. 2.

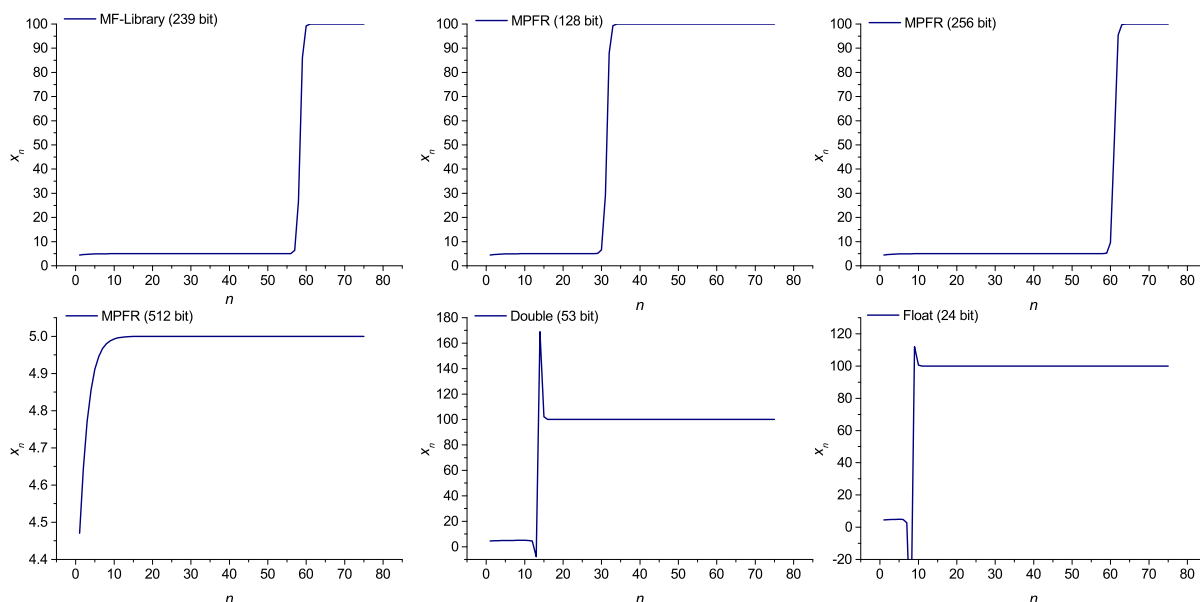


Рис. 2. Результаты вычисления рекуррентного соотношения (2). Графики показывают, что точность MF-Library сопоставима с точностью MPFR (256 бит).

Эффективность векторизации. Исследованы операции сложения (add), вычитания (sub), умножения (mult), сравнения (cmp), сложения с накоплением (aac, $x \leftarrow x + y$), вычитания с накоплением (sac, $x \leftarrow x - y$), умножения с накоплением (mac, $x \leftarrow z + xy$) и

деления (div). Выполнялось 10^7 итераций. Тестовые данные – псевдослучайные числа, генерировались алгоритмом “Вихрь Мерсенна”. Тестовая конфигурация: Intel Core i5-3570K Processor / 4 Cores / 8 Gb RAM / Intel C++ Compiler 13.0. Запуск MF-Library выполнялся в двух конфигурациях: при установленном запрете на векторизацию (прописыванием директив `#pragma novector`) и при использовании средств автоматической векторизации компилятора Intel C++ Compiler (`#pragma simd`). Во втором случае векторизовались циклы вычисления модулярных мантисс и интервально-позиционных характеристик (в пределах одного вычислительного ядра). Результаты представлены на Рис. 3.

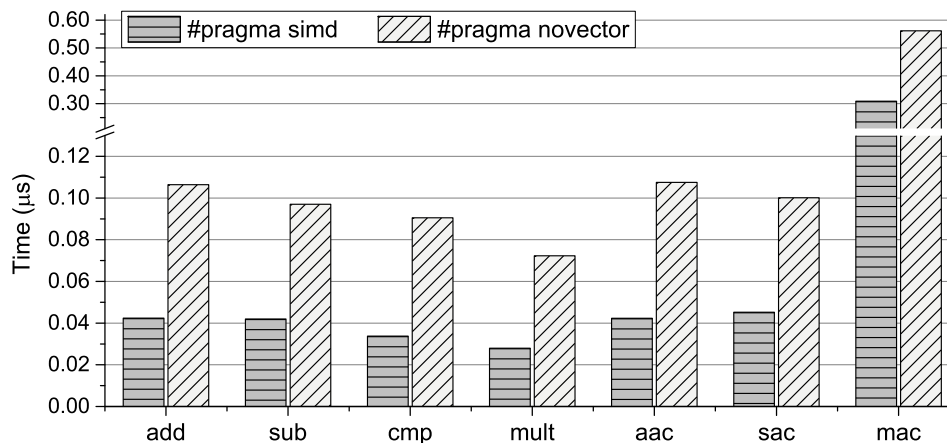


Рис. 3. Экспериментальные оценки времени выполнения операций многоразрядной арифметики в MF-Library. Время деления, не представленное на графике, составляет в среднем 6.91 мкс и 6.33 мкс соответственно без векторизации и с векторизацией.

Эффективность векторизации оценивалась по формуле

$$E_{\pi}(n) = \frac{S_{\pi}(n)}{\pi} = \frac{T_1(n)}{\pi T_{\pi}(n)},$$

где $S_{\pi}(n) = T_1(n)/T_{\pi}(n)$ – полученное ускорение, $T_1(n)$ – время вычислений без векторизации, $T_{\pi}(n)$ – время вычислений с векторизацией, π – число пар операндов, которые могут быть обработаны параллельно с использованием векторных инструкций в пределах одного вычислительного ядра (для используемого процессора $\pi = 4$).

Средняя эффективность векторизации (за исключением деления) составила 0.60. Это означает, что MF-Library обеспечивает эффективное использование более половины доступных SIMD-ресурсов вычислительного ядра. При векторизации деления получено ускорение лишь в 1.09 раза (эффективность 0.27). Это связано с необходимостью затратного преобразования модулярных мантисс в двоичную систему. В дальнейшем планируется использовать более эффективный алгоритм деления.

BLAS-операции. Быстродействие MF-Library исследовано при выполнении операций обобщенного матричного умножения (GEMM, третий уровень BLAS) и обобщенного векторного умножения (GEMV, второй уровень BLAS). В качестве аналога использовался пакет MPFR, являющийся одним из наиболее быстрых в своем классе. Исходные данные для всех операндов (матриц, векторов и скаляров) были представлены псевдо-случайными 239-битными числами с плавающей точкой, что способствовало большей представительности теста. Тестовая конфигурация: Intel Core i7-4702MQ (Haswell) / 4 Gb RAM / Intel C++ Compiler 13.0. Были рассмотрены последовательные и параллельные алгоритмы, что, ко всему прочему, позволяет оценить потоковую безопасность MF-Library. При выполнении операции GEMM ($C \leftarrow \alpha AB + \beta C$) матрицы A, B, C были плотными, их порядок n изменялся с шагом 50 в интервале от 100 до 1000. Корректность выполнения операции оценивалась по норме $\|C\|_1$. Полученные результаты представлены на Рис. 4. Среднее ускорение

MF-Library по сравнению с MPFR составило 1.9 раза и 2.3 раза соответственно при последовательных и параллельных вычислениях. При распараллеливании скорость вычислений возросла в 5.3 раза.

При выполнении GEMV ($y \leftarrow \alpha Ax + \beta y$) векторы x, y и матрица A также были плотно заполнены. Размер векторов варьировался в диапазоне от 500 до 1500 с шагом 100. Корректность операции оценивалась по норме $\|y\|_1$. Результаты экспериментов представлены на Рис. 5. В данном случае по сравнению с MPFR получено ускорение 2.4 раза и 2.6 раза соответственно при последовательных и параллельных вычислениях. При распараллеливании скорость MF-Library увеличилась в среднем в 4 раза.

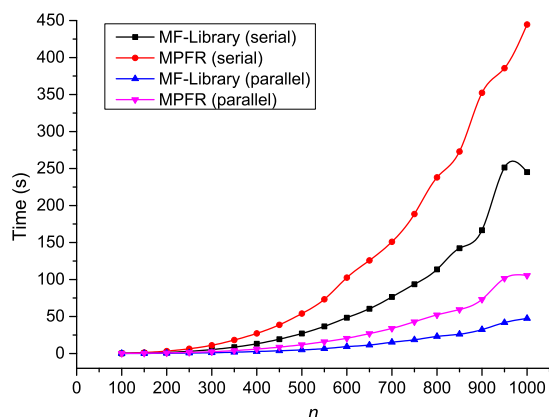


Рис. 4. Зависимость времени высокоточного выполнения операции GEMM от порядка матриц n .

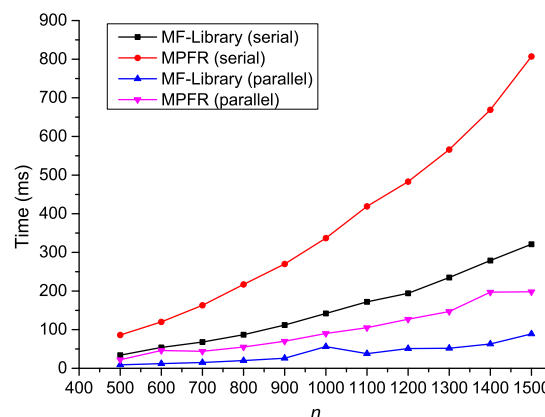


Рис. 5. Зависимость времени высокоточного выполнения операции GEMV от размера векторов n .

В работах [27, 28] представлены результаты других экспериментов по исследованию эффективности библиотеки MF-Library.

4. Выводы

С ростом производительности вычислительных систем и размерности решаемых задач возрастает значимость вычислений многократной точности. Уже сейчас имеется целый ряд актуальных приложений, которые требуют оперировать числами с разрядностью, на несколько порядков превосходящей длину машинного слова. Следует полагать, что с течением времени количество таких приложений будет увеличиваться. В связи с этим, проблема обеспечения приемлемой точности отмечается многими исследователями, как одна из наиболее актуальных проблем программного обеспечения эксафлопсных суперкомпьютеров.

Современный этап развития вычислительной техники приводит к новым требованиям, предъявляемым к высокоточному программному обеспечению. К числу основных из таких требований относятся высокая скорость, потоковая безопасность и эффективное использование современных параллельных архитектур. В этой статье рассмотрен новый программный пакет высокоточных вычислений MF-Library, нацеленный на удовлетворение этих требований. В основе пакета лежит модулярно-позиционный способ представления длинных чисел, изначально ориентированный на параллельную обработку. Алгоритмы, заложенные в MF-Library, лишены недостатков аналогов, связанных с необходимостью затратной обработки переносов между соседними цифрами длинной мантиссы, и соответствуют архитектуре современных центральных процессоров, позволяя эффективно использовать как многоядерность, так и возможности SIMD-обработки в пределах ядра. Благодаря потоковой безопасности, MF-Library может эффективно использоваться на параллельных системах с общей памятью, что подтверждается представленными результатами экспериментов.

В настоящий момент ведутся работы по расширению функциональности MF-Library. Одновременно с этим начата работа по созданию версии пакета для графических процессоров. Конечной целью исследований является создание унифицированного алгоритмического и программного обеспечения, позволяющего эффективно выполнять высокоскоростные параллельные расчеты с плавающей точкой многократной точности на современных и перспективных высокопроизводительных системах с гибридной архитектурой.

Литература

1. Collange S., Defour D., Graillat S., Iakymchuk R. Reproducible and Accurate Matrix Multiplication for High-Performance Computing // 16th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2014), September 21–26, 2014, Würzburg, Germany, Book of Abstracts. 2014. P. 42–43.
2. Якушев В.Л. Решение плохообусловленных симметричных СЛАУ для задач строительной механики параллельными итерационными методами / В.Л. Якушев, [и др.] // Вестник ННГУ. 2012. № 4(1). С. 238–246.
3. Frolov A.M., Bailey D.H. Highly Accurate Evaluation of the Few-Body Auxiliary Functions and Four-Body Integrals // Journal of Physics B: Atomic, Molecular and Optical Physics. 2003. Vol. 36, No. 9. P. 1857–1867.
4. Bailey D.H. High-Precision Floating-Point Arithmetic in Scientific Computation // Computing in Science and Engineering. 2005. Vol. 7, Issue 3. P. 54–61.
5. Barrowes B.E. On the Asymptotic Expansion of the Spheroidal Wave Function and its Eigenvalues for Complex Size Parameter / B.E. Barrowes [et al.] // Studies in Applied Mathematics. 2004. Vol. 113, Issue 3. P. 271–301.
6. Калиткин Н.Н., Кузьмина Л.В. Об аппроксимации неортогональными системами // Математ. моделирование. 2004. Т. 16, № 3. С. 95–108.
7. Bailey D.H., Barrio R., Borwein J.M. High-Precision Computation: Mathematical Physics and Dynamics // Applied Mathematics and Computation. 2012. Vol. 218, Issue 20. P. 10106–10121.
8. Robey R.W., Robey J.M., Aulwes R. In Search of Numerical Consistency in Parallel Programming // Parallel Computing. 2011. Vol. 37, Issue 4–5. P. 217–229.
9. Опенгейм А.В., Шафер Р. В. Цифровая обработка сигналов : пер. с англ.; под ред. С.Я. Шаца. М. : Связь, 1979. 416 с.
10. Yang J.-F., Fan C.-P. Compact Recursive Structures for Discrete Cosine Transform // IEEE Transactions on CAS-II: Analog and Digital Signal Processing. 2000. Vol. 47, Issue 4. P. 314–321.
11. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи : пер. с англ. М. : Мир, 1999. 685 с.
12. Каханер Д., Моулер К., Нэш С. Численные методы и программное обеспечение : пер. с англ. М. : Мир, 1998. 575 с.
13. Воеводин В.В. Вычислительная математика и структура алгоритмов. М. : Изд-во Московского ун-та, 2010. 168 с.

14. He Y., Ding C. Using Accurate Arithmetics to Improve Numerical Reproducibility and Stability in Parallel Applications // Journal of Supercomputing. 2001. Vol. 18, No. 3. P. 259–277.
15. Lake G., Quinn T., Richardson D.C. From Sir Isaac to the Sloan survey: Calculating the Structure and Chaos due to Gravity in the Universe // 8th ACM – SIAM Symposium on Discrete Algorithms, 1997, Philadelphia, USA, Proceedings. SIAM. 1997. 10 p.
16. Bailey D.H., Borwein J.M. High-Precision Arithmetic: Progress and Challenges, <http://www.davidhbailey.com/dhbpapers/hp-arith.pdf>.
17. Rump S.M. Algorithms for Verified Inclusions – Theory and Practice. In: Moore, R.E. (ed.) Reliability in Computing, pp. 109–126. Academic Press, New York (1988)
18. Kahan W. How Futile are Mindless Assessments of Roundoff in Floating-Point Computation, <https://www.cs.berkeley.edu/~wkahan/Mindless.pdf>.
19. Ghazi K.R., Lefèvre V., Théveny P., Zimmermann P. Why and How to Use Arbitrary Precision // Computing in Science and Engineering. 2010. Vol. 12, No. 3. P. 62–65.
20. IEEE Standard for Floating-Point Arithmetic. Introduced 2008-08-29. New York : Institute of Electrical and Electronics Engineers, 2008. 70 p.
21. High-Precision Software Directory, <http://crd-legacy.lbl.gov/~dhbailey/mpdist>.
22. The GNU Multiple Precision Arithmetic Library, <https://gmplib.org>.
23. Fousse L., Hanrot G., Lefèvre V., Pélissier P., Zimmermann, P.: MPFR: A Multiple-Precision Binary Floating-Point Library With Correct Rounding // ACM Transactions on Mathematical Software. 2007. Vol. 33, No. 2.
24. NTL: A Library for doing Number Theory, <http://www.shoup.net/ntl>.
25. MPFUN2015: A Thread-Safe Arbitrary Precision Computation Package (Full Documentation), <http://www.davidhbailey.com/dhbpapers/mpfun2015.pdf>.
26. Lu M., He B., Luo Q. Supporting extended precision on graphics processors // 6th International Workshop on Data Management on New Hardware (DaMoN 2010), June 7, 2010, Indianapolis, Indiana, USA, Proceedings. 2010. P. 19–26.
27. Исупов К.С. Методы и алгоритмы организации высокоточных вычислений в арифметике остаточных классов для универсальных процессорных платформ : диссертация ... кандидата технических наук : 05.13.15 / Исупов Константин Сергеевич; 2014.
28. Исупов К.С., Мальцев А.Н. Способ представления чисел с плавающей точкой большой разрядности, ориентированный на параллельную обработку // Вычислительные методы и программирование. 2014. Т. 15, № 4. С. 631–643.
29. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М. : Сов. Радио, 1968. 440 с.
30. Garner H.L. The Residue Number System // IRE Transactions on Electronic Computers. 1959. Vol. EC-8, No. 2. P. 140–147.
31. Исупов К.С. Немодульные вычисления в системах остаточных классов с интервально-позиционными характеристиками / К.С. Исупов, В.С. Князьков ; ВятГУ. Киров, 2015. 92 с. Деп. в ВИНТИ РАН 26.03.2015.