

УДК 004.02

**ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ДЛЯ ПРОЕКТИРОВАНИЯ
ПАРАЛЛЕЛЬНЫХ МАСШТАБИРУЕМЫХ ПРОГРАММ ЧИСЛЕННЫХ
РАСЧЕТОВ****К.С. Исупов, В.С. Князьков**

Многие прикладные задачи высокой размерности требуют выполнения высокоточных численных расчетов в базе матричной алгебры над полем больших чисел. Современные мультипроцессорные вычислительные системы, как правило, имеют 32-битную или 64-битную архитектуру, что затрудняет выполнение операций над большими числами и уменьшает эффективность их использования. В результате работы над проблемой предложен новый метод выполнения вычислений, основанный на использовании в качестве базиса для выполнения численных расчетов системы счисления с параллельной архитектурой – системы остаточных классов. На основании предложенного метода разрабатывается инструментальный комплекс.

Ключевые слова: программный комплекс, параллельные масштабируемые расчеты, система остаточных классов, большие числа, MPI, OpenMP, технологии параллельного программирования, параллельные программы.

Введение

Используемые в современных САД-системах математические методы и их алгоритмически-программные решения разрабатывались для вычислительных систем (ВС) с ограниченным количеством вычислительных модулей. Адаптация применяемых сегодня методов, алгоритмов и их программных реализаций для решения прикладных задач даже на 1000-ядерные ВС является весьма сложной и, в ряде случаев, невыполнимой задачей. В связи с этим эффективность использования ресурсов современных многопроцессорных многоядерных ВС крайне мала, а ряд современных прикладных задач высокой размерности и высокой вычислительной сложности вообще невозможно решить из-за отсутствия скоростных методов масштабируемых вычислений, либо задача решается неприемлемо длительное время.

Один из примеров таких задач – определение корней плохо обусловленных СЛАУ. Если коэффициенты двух или нескольких уравнений СЛАУ близки, то при малых возмущениях в матрице коэффициентов A или векторе правых частей B произойдут большие изменения в векторе неизвестных $X=A^{-1} \times B$. Это означает, что численный метод будет неустойчив и, следовательно, резко повысится вероятность получения ответа, содержащего большую погрешность. Например, при решении плохо обусловленных СЛАУ 9-го и 13-го порядков методами популярного пакета Mathcad 12 погрешности определения корней составляют 3×10^{-4} и 7×10^{-2} соответственно. Далее, с увеличением порядка СЛАУ, погрешность возрастает экспоненциально.

Постановка задачи

В результате возникает необходимость создания высокоскоростных, глубоко масштабируемых (обеспечивающих увеличение ускорения не только при росте числа процессоров, но и при росте числа ядер каждого процессора, с сохранением постоянного уровня эффективности использования процессоров) численных методов, алгоритмов и их программных реализаций для выполнения высокоточных численных расчетов по отношению к массивам данных большой размерности, с учетом особенностей и требований современных задач. Основным подклассом таких задач являются высокоточные численные расчеты в базе матричной алгебры.

Наиболее важным здесь является требование максимально быстрого выполнения операций над числами, выходящими за пределы типового компьютерного диапазона. Данные числа также принято называть большими числами. Как правило, процессоры оптимизированы для операций с 32-битными или 64-битными операндами, все подсистемы ввода/вывода также имеют такую разрядность, поэтому производительность типовых ВС при работе с большими числами катастрофически падает.

Основной результат

Одним из перспективных и многообещающих направлений в исследованиях по данной проблеме является внедрение методов вычислений на основе нетрадиционных способов кодирования и соответствующих им вариантов машинной арифметики. Особую роль в развитии указанного направления играют числовые системы с параллельной структурой и, в первую очередь, модулярные системы счисления, в которых целые числа представляются полиномами из остатков от деления на выбранные основания. Одной из самых известных таких систем является система остаточных классов (СОК). Здесь, в отличие от обобщенной позиционной системы счисления (ПСС), образование цифры каждого разряда проводится независимо друг от друга. Цифра i -го разряда определяется согласно выражению

$$a_i = N - \left\lfloor \frac{N}{P_i} \right\rfloor \cdot P_i, \quad (1)$$

где $i = 1, 2, \dots, R$; N – число, представленное в позиционной системе счисления; R – число оснований P_i (мощность системы оснований).

Иначе говоря, каждый разряд в системе остаточных классов представляет собой наименьший неотрицательный остаток от деления на соответствующее основание самого числа, а не предыдущего частного, как это имеет место в позиционной системе [1].

Если же касаться задач матричной алгебры, то здесь каждый массив, представленный в ПСС, в СОК будет представляться множеством независимых друг от друга массивов, элементы которых определяются по формуле (1). Очевидно, что мощность данного множества совпадает с числом оснований СОК.

Если в качестве оснований СОК выбрать числа, не выходящие за пределы разрядной сетки ВС, то и вычеты, взятые по этим основаниям, не будут выходить за пределы разрядной сетки. Тем самым выполнение операций над большими числами не будет отличаться от выполнения операций над числами, не выходящими за пределы разрядной сетки ВС. Это дает следующие фундаментальные преимущества при решении высокоточных задач большой размерности на многопроцессорных ВС петафлопного уровня:

- возможность более низкоуровневого распараллеливания вычислений;
- увеличение скорости выполнения операций над большими числами;
- снижение доли последовательных участков в программе.

Применительно к задачам матричной алгебры следует выделить и четко разделить два вида параллелизма, реализуемые при выполнении расчетов на мультипроцессорной ВС над числовыми данными, представленными в СОК – параллелизм на уровне элементов данных и параллелизм на уровне разрядов чисел. Реализация параллелизма на уровне элементов данных сильно зависит от выбранного алгоритма выполнения расчетов. Для эффективной реализации параллелизма на уровне разрядов чисел необходимо выбрать оптимальную мощность системы оснований СОК, учитывая при этом особенности архитектуры современных мультипроцессорных ВС. Мощность системы оснований – число оснований СОК, входящих в ее состав.

Во избежание излишних межпроцессорных взаимодействий, возникающих при сборе вычетов полиномиального числа с различных вычислительных модулей для их последующего преобразования в позиционную систему, необходимо выполнять вычисления над всеми вычетами в одном модуле. Поэтому целесообразным является использование системы оснований СОК с мощностью, равной числу ядер каждого вычислительного узла (ВУ) системы.

При использовании системы оснований с мощностью, меньшей чем число ядер каждого вычислительного узла, диапазон представления чисел сокращается, а следовательно, для достижения требуемого диапазона необходимо выбирать основания большей разрядности, что требует более высоких затрат на обработку вычетов, взятых по этим основаниям.

При использовании числа оснований, превышающего число ядер каждого ВУ, возникают накладные расходы на обработку вычетов, взятых по дополнительным основаниям. В результате этого загрузка ядер каждого процессора в ВУ будет распределена неравномерно, так как потоки, обрабатывающие дополнительные вычеты, будут выполняться на ядрах после завершения всех основных потоков, в результате чего часть ядер будет простаивать, ожидая завершения работы остальных.

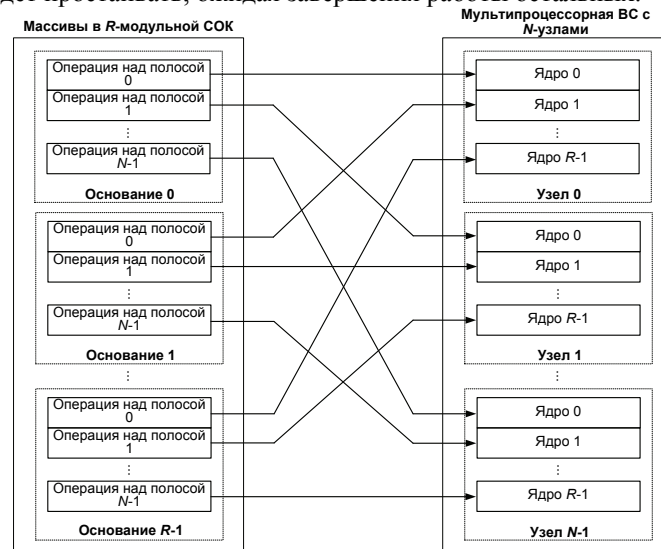


Рис. 1. Схема распределения задач по вычислительным узлам

Схема распределения задач по вычислительным узлам, необходимая для выполнения численных расчетов матричной алгебры в R -модульной системе остаточных классов на мультипроцессорной системе с кластерной архитектурой, в состав которой входит N узлов, представлена на рис. 1. Подразумевается, что каждый узел в данной системе содержит один R -ядерный процессор, а к массивам данных применена схема горизонтального ленточного разбиения.

Используя СОК с числом оснований, равным числу ядер каждого ВУ системы, каждое ядро будет производить работу с фрагментами массивов данных только по соответствующему ему основанию. Таким образом, ядра всех процессоров отдельно взятого вычислительного узла в каждый момент времени будут выполнять одни и те же операции над различными данными, что позволит более равномерно распределить их загрузку, а также избавиться от конфликтов по доступу к разделяемым данным.

На основании предложенного подхода разрабатывается инструментальный комплекс для проектирования параллельных масштабируемых программ численных расчетов в базисах матричной алгебры и системы остаточных классов. Структурная схема инструментального комплекса представлена на рис. 2.

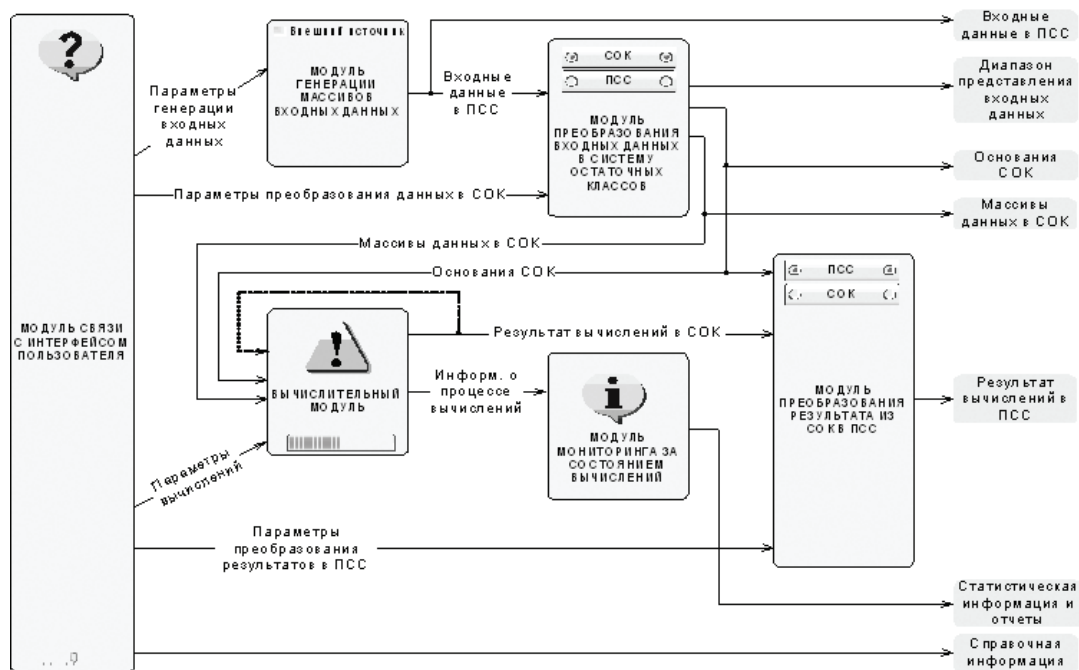


Рис. 2. Структурная схема инструментального комплекса

Ядром комплекса является вычислительный модуль, который производит масштабирование вычислений и выполнение одного из численных расчетов согласно заданному алгоритму и входным параметрам. Все вычисления производятся в СОК по выбранным основаниям. Результаты расчета передаются другим исполнительным модулям комплекса для их последующего преобразования.

При создании вычислительного модуля были использованы две технологии параллельного программирования: MPI – для реализации параллелизма на уровне элементов данных; OpenMP – для реализации параллелизма на уровне разрядов чисел, обеспечиваемого выполнением вычислений в системе остаточных классов.

Особенностью комплекса является обособленность исполнительных модулей от пользовательского интерфейса, что позволяет выполнять их в виде динамических библиотек и встраивать в современные промышленные математические системы и специализированные программные комплексы. Следует отметить, что интерфейс комплекса позволяет формировать XML-файлы заданий для программного продукта HPC Cluster Manager.

Основные возможности, предоставляемые пользователю:

- выбор операции и метода численного расчета;
- настройка оснований СОК на работу с числами из заданного диапазона;
- настройка комплекса на конфигурацию конкретной ВС;
- изменение диапазона представления входных данных;
- загрузка массивов входных данных из внешних источников;
- варьирование размерности массивов данных вплоть до 10^6 ;

- использование результатов i -ой операции в качестве входных данных $(i+1)$ -ой без останова вычислений.

Пользователю предоставляется один из следующих способов задания оснований:

- автоматическое задание оснований из поля чисел Мерсенна ($2^N - 1$);
- автоматическое задание оснований из поля чисел Фибоначчи;
- ручной ввод оснований с последующей проверкой их на взаимную простоту.

В комплекс включены подпрограммы, обеспечивающие выполнение 20 операций матричной алгебры. В настоящий момент разрабатываются эффективные методы решения плохо обусловленных СЛАУ. Выполнение вычислений в СОК позволяет эффективно использовать ресурсы современных мультипроцессорных ВС с кластерной архитектурой и корректно решать проблему больших чисел.

Комплекс предназначен для работы с числами как небольших, так и больших компьютерных диапазонов. В рассмотренной задаче решения плохо обусловленных СЛАУ требуемый диапазон представления чисел зависит от размерности матрицы коэффициентов: чем больше размерность матрицы, тем большая разрядность требуется для хранения ее нижних строк с заданной точностью. Поэтому предполагается, что комплекс будет производить обработку как стандартных чисел, не выходящих за пределы разрядной сетки ЭВМ, так и чисел, значительно превышающих ее.

В настоящий момент предложенные методы позволяют работать с целыми числами в формате с фиксированной запятой. Так как высокоточные вычисления, не ограничивающиеся полем целых чисел, крайне важны, в настоящее время наиболее приоритетной является задача адаптации предложенных методов и алгоритмов для работы с вещественными числами.

Традиционно недостатками СОК считается сложность реализации процедуры сравнения чисел, представленных в этой системе. В комплекс включен модифицированный алгоритм сравнения, основанный на работе [2], позволяющий выполнять на ПК (Intel Celeron(R) M CPU 1,46GHz с 1,37 Гб) за одну секунду 29 000 полиномиальных операций сравнения.

Экспериментальная апробация

В качестве матричной операции для экспериментов была выбрана операция сложения двумерных массивов. Запуски производились на четырех SMP-модулях, включающих два четырехъядерных процессора Intel Xeon под управлением ОС Windows 2008 CCS. Исходные массивы разбивались на одинаковое количество горизонтальных полос, обрабатываемых параллельно, т.е. вычислительный алгоритм подразумевает параллелизм на уровне элементов данных.

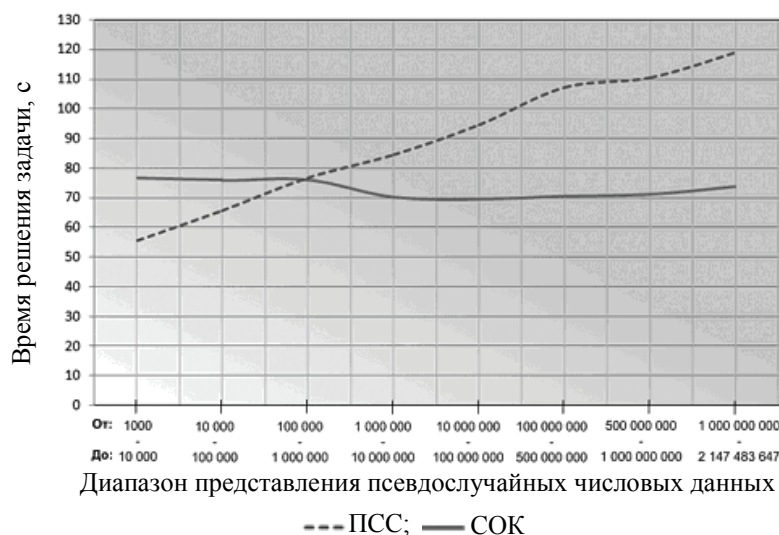


Рис. 3. Зависимость времени решения задачи от разрядности числовых данных

На рис. 3 показана зависимость времени решения задачи сложения двумерных массивов размером 16000×32000 от разрядности чисел. В качестве оснований СОК были выбраны взаимно простые числа 2047, 511, 255 и 127. При данных основаниях позиционную систему выгоднее использовать только для работы с шести- и менее разрядными числовыми данными. При работе с девятиразрядными числами ускорение решения задачи составляет 1,7 раза.

Очевидно, что при выборе других оснований СОК данная зависимость будет выглядеть иначе: при выборе оснований СОК, произведение которых приближается к требуемому пределу диапазона пред-

ставления чисел, время решения задачи существенно сокращается по сравнению с выбором оснований большей разрядности, что было доказано соответствующими экспериментами.

Так, с применением тонкой настройки системы оснований СОК вокруг требуемого предела диапазона представления чисел, уже при работе с шестизрядными числами удалось ускорить решение задачи в 1,23 раза относительно позиционной системы, тогда как без настройки системы оснований при прочих равных условиях (рис. 3) ускорение решения задачи отсутствует.

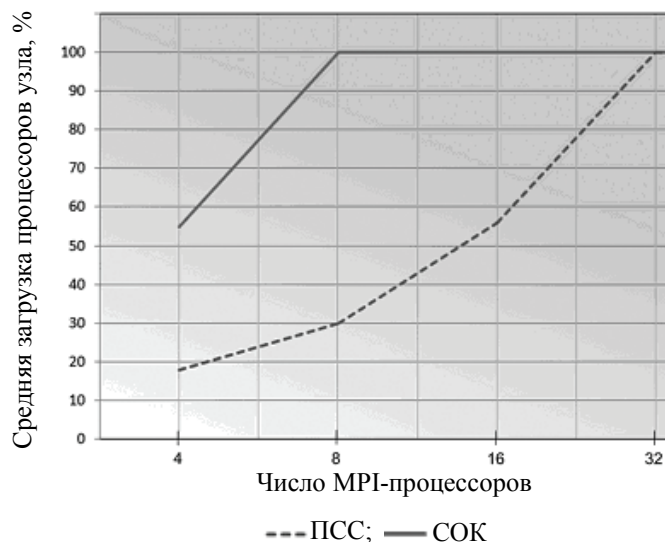


Рис. 4. Зависимость средней загрузки процессоров узла от числа MPI-процессов

На рис. 4 показана зависимость средней загрузки процессоров от количества запущенных MPI-процессов. При использовании в качестве базиса для вычислений четырехмодульной СОК полная эффективная загрузка всех восьми процессоров наблюдается уже при запуске восьми MPI-процессов, тогда как при использовании ПСС добиться полной загрузки удается лишь при запуске 32 и более MPI-процессов.

Заключение

В результате проделанной работы был предложен новый подход к решению проблемы выполнения численных расчетов в базе матричной алгебры над полем больших чисел, обеспечивающий эффективную загрузку не только всех процессоров, но и всех ядер вычислительных узлов мультипроцессорной ВС. На основании данного подхода разрабатывается инструментальный комплекс. В конечном итоге инструментальный комплекс предполагается использовать как для решения задач определения корней плохо обусловленных СЛАУ и обращения матриц Гильберта, так и для выполнения наиболее трудоемких матричных операций, таких как умножение матриц и нахождение определителя.

Наиболее приоритетным направлением дальнейших исследований является адаптация разработанных методов для работы с вещественными числами.

Литература

1. Акушский И.Я. Машинная арифметика в остаточных классах / И.Я. Акушский, Д.И. Юдицкий. – М.: Сов. радио, 1968. – 440 с.
2. Факторович М.Г., Полиский Ю.Д. Устройство для сравнения чисел, выраженных в системе остаточных классов. Авт. свид. СССР №608155 М. Кл2 G 06 F 7/04, 1978.

Исунов Константин Сергеевич
Князьков Владимир Сергеевич

- Вятский государственный университет, студент, slaer_87@hotmail.com
- Вятский государственный университет, доктор технических наук, профессор, kniazkov@list.ru