ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ НАД МНОГОРАЗРЯДНЫМИ ЧИСЛАМИ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

К.С. Исупов

Введение.

Многие прикладные задачи, решение которых возможно только с помощью суперкомпьютеров, зачастую требуют работы с *многоразрядными* числами. Вычисления над многоразрядными числами являются одной из областей, в которых хорошо разработанные в настоящее время позиционные методы являются неэффективными. В настоящее время возникает широкий спектр задач [1-4], приводящих к вычислениям, при которых значения числовых данных на порядки превышают максимум типового компьютерного диапазона современных вычислительных систем (ВС).

Многие современные программные пакеты для исследовательских расчетов (например, MATHCAD и MATLAB) справляются с решением высокоточных задач далеко не в полной мере, что может вызывать проблемы в ходе проведения исследовательских работ с их использованием. Как правило, в данных пакетах для реализации высокоточных вычислений используется рациональная арифметика на основе схемы приведения дробей. Однако такой подход не является оптимальным [5]. Кроме этого, при выполнении операций в базисе позиционной системы счисления (ПСС) образуются длинные цепочки межразрядных переносов [6], которые требуют последовательной обработки и накладывают принципиальные ограничения на эффективное распараллеливание арифметических операций. С учетом этого, а так же ввиду непрерывного роста размерности задач и требований, которые они предъявляют к точности решения, вытекает острая необходимость разработки новых методов для эффективного выполнения параллельных вычислений над многоразрядными числами.

Вычисления над многоразрядными числами в системе остаточных классов.

В свете сказанного исключительно большое значение имеют исследования, ориентированные на применение нетрадиционных способов кодирования числовой информации и соответствующих им параллельных вариантов компьютерной арифметики.

Нами предлагается один из методов выполнения высокоточных арифметических операций, заключающийся в преобразовании исходных многоразрядных целых чисел в группы независимых чисел меньшей разрядности с последующей параллельной обработкой элементов этих групп на многоядерных вычислителях (рисунок 1). Данное преобразование реализуется посредством перевода вычислений в базис системы остаточных классов (СОК). При этом становится возможной настройка базиса вычислений как под работу с числами конкретной разрядности, так и под конкретную архитектуру ВС, что дает богатые возможности оптимизации вычислительного процесса.

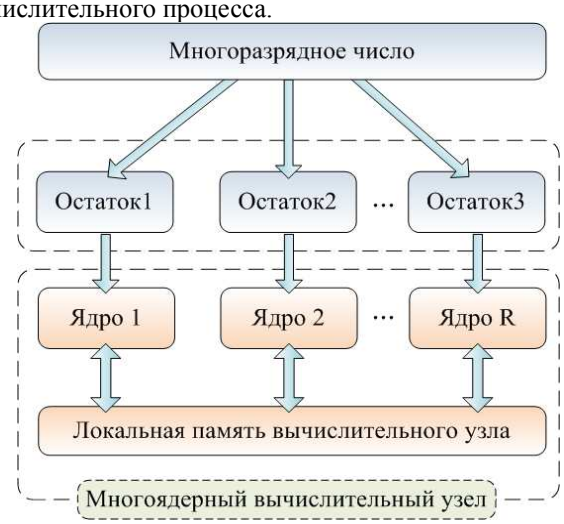

Рис. 1 – Схема распределения многоразрядного числа между ядрами вычислительного узла

Рассмотрим основные понятия СОК. *Система остаточных классов* (Residual Number System) - одна из самых известных непозиционных систем счисления. Здесь, в отличие от ПСС, числа представляются в виде независимых разрядов [7], именуемых вычетами (остатками). Все разряды чисел в СОК определяются согласно выражению $a_i = N -]N/p_i[\times p_i$, где a_i – i-ый разряд (остаток, вычет) полиномиального представления от числа N; p_i – i-ое основание системы остаточных классов, $i = 1, 2, ..., R$; R – число оснований (модульность) СОК; N – исходное число, представленное в позиционной системе счисления; $]N/p_i[$ – целая часть отношения N/p_i.

Другими словами, каждый разряд a_i в системе остаточных классов представляет собой наименьший неотрицательный остаток от деления на соответствующее основание p_i самого числа N, а не предыдущего частного, как это имеет место в позиционной системе. Система оснований СОК – совокупность всех попарно взаимно-простых оснований p_i. Число оснований, входящих в систему, именуется *модульностью* (мощностью) системы оснований СОК. В теории чисел доказано, что соблюдение условия взаимной простоты оснований гарантирует однозначное отображение позиционного числа в полиномиальное представление [7].

В таблице 1 показаны эквиваленты десятичных чисел от «0» до «29», представленные в СОК с использованием в качестве оснований p_i взаимно-простых чисел «2», «3» и «5».

Таблица 1 – Представление десятичных чисел от 0 до 29 в СОК

Число в ПСС	**Остатки по модулям «2», «3», «5»**			**Число в ПСС**	**Остатки по модулям «2», «3», «5»**			**Число в ПСС**	**Остатки по модулям «2», «3», «5»**		
	5	**3**	**2**		**5**	**3**	**2**		**5**	**3**	**2**
0	0	0	0	*10*	0	1	0	*20*	0	2	0
1	1	1	1	*11*	1	2	1	*21*	1	0	1
2	2	2	0	*12*	2	0	0	*22*	2	1	0
3	3	0	1	*13*	3	1	1	*23*	3	2	1
4	4	1	0	*14*	4	2	0	*24*	4	0	0
5	0	2	1	*15*	0	0	1	*25*	0	1	1
6	1	0	0	*16*	1	1	0	*26*	1	2	0
7	2	1	1	*17*	2	2	1	*27*	2	0	1
8	3	2	0	*18*	3	0	0	*28*	3	1	0
9	4	0	1	*19*	4	1	1	*29*	4	2	1

При переводе исходного позиционного числа N в СОК с числом оснований R образуется множество N^* независимых картежей, состоящих из неотрицательных вычетов и однозначно сопоставимых им оснований. Данное множество может быть определено как $N^*=\{(a_1, p_1); (a_2, p_2); \ldots(a_R, p_R)\}$. Это множество называется полиномиальным представлением позиционного числа N в R-модульной системе остаточных классов (либо просто - полиномиальным числом). Например, число «26» из таблицы 1 в СОК можно записать как {(1, 5); (2, 3); (0, 2)}. Одним из основных свойств множества N^* является взаимная независимость его элементов. Именно данное свойство обуславливает дополнительный уровень параллелизма, обеспечиваемый СОК, - *параллелизмом на уровне разрядов чисел* [8]. Данный параллелизм позволяет одновременно выполнять операции и группы операций над всеми полиномиальными разрядами числа параллельно и независимо друг от друга.

Еще одним немаловажным свойством N^* является малая разрядность вычетов [8, 9]. Если в качестве старшего основания p_i выбирать число, не выходящее за предел, определяемой размером разрядной сетки, то и все разряды числа в СОК не будут выходить за данный предел. При этом *максимальный диапазон представления будет определяться как произведение всех оснований* p_i. Здесь становится очевидным, что разрядность позиционного числа значительно превышает разрядность старшего основания, а соответственно, и разрядность всех вычетов при одинаковом допустимом диапазоне представления чисел. При выборе достаточно большого числа оснований p_i становится возможной корректная работа с числами, позиционное представление которых на порядки превышает предел, определяемый длиной машинного слова современных вычислителей. При этом фактически будут обрабатываться числа, не выходящие за данный предел, а длинные цепочки межразрядных переносов образовываться не будут.

Кроме этого, вычислительный процесс может быть оптимизирован под конкретную ВС путем выбора соответствующих разрядности данной ВС оснований p_i и их числа.

Обобщив, можно сказать, что использование системы остаточных классов позволяет заменить трудоемкие операции по обработке многоразрядных чисел группами значительно более простых и независимых операций по обработке чисел меньшей разрядности, не выходящих за предел, определяемый длиной машинного слова конкретной ВС.

Эксперименты.

Для подтверждения рассмотренных теоретических положений был проведен ряд экспериментов, ориентированных на сравнение быстродействия высокоточных расчетов в СОК и в позиционной двоичной системе (ПСС). В ходе экспериментов вычислялось произведение матриц, элементами которых являются целые

числа. Разрядность чисел изменялась в диапазоне от 32 до 248 бит. Запуски производились на кластерной системе Вятского Государственного Университета ENIGMA (HP Hewlett-Packard Cluster Platform 3000 BL460c, Intel EM64T Xeon 53xx, 2,3 ГГц). Были исследованы последовательный и параллельный алгоритмы умножения матриц. В качестве библиотеки для работы с многоразрядными позиционными числами был использован пакет cBigNumber [10], позволяющий работать с числами, разрядность которых ограничивается лишь доступными аппаратными ресурсами. Данный пакет является типичным представителем средств длинной арифметики, основанных на позиционной системе счисления. В качестве базиса СОК использовалось восемь взаимно-простых 32-битных чисел.

При реализации параллельного алгоритма умножения матриц в СОК была использована схема распределения массивов, представленная на рисунке 2.

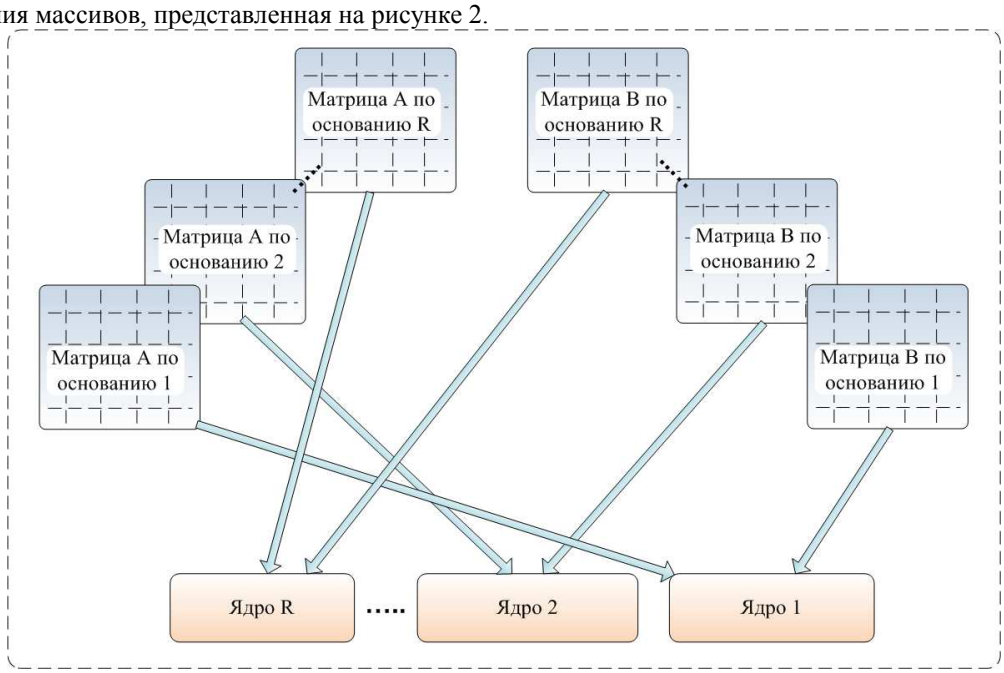

Рис. 2 Схема распределения массивов в параллельном алгоритме (СОК)

При реализации параллельного алгоритма в ПСС использована типичная схема горизонтального ленточного разбиения матриц (рисунок 3).

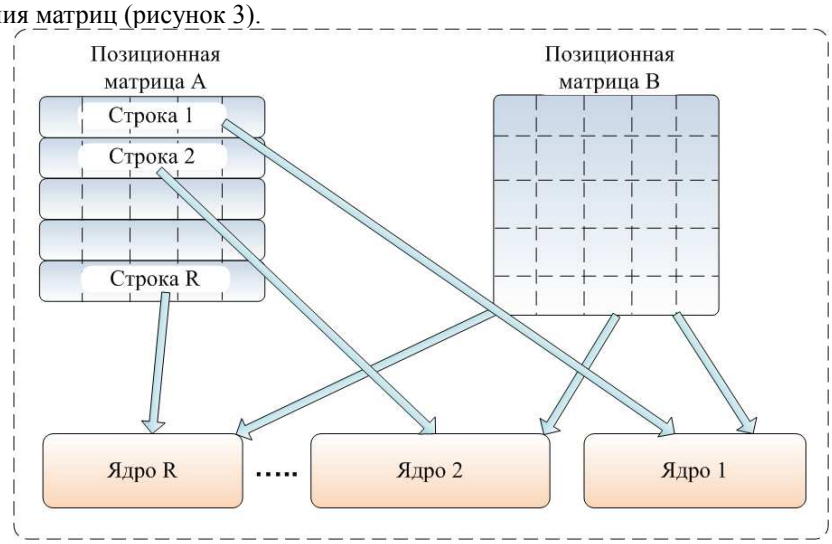

Рис. 3 Схема распределения массивов в параллельном алгоритме (ПСС)

Первым экспериментом являлось исследование изменения времени умножения матриц фиксированной размерности (700×700) с элементами, разрядность которых изменялась от 32 до 248 бит. График зависимости времени умножения от разрядности числовых данных представлен на рисунке 4. График зависимости ускорения, достигаемого при счете в СОК, представлен на рисунке 5.

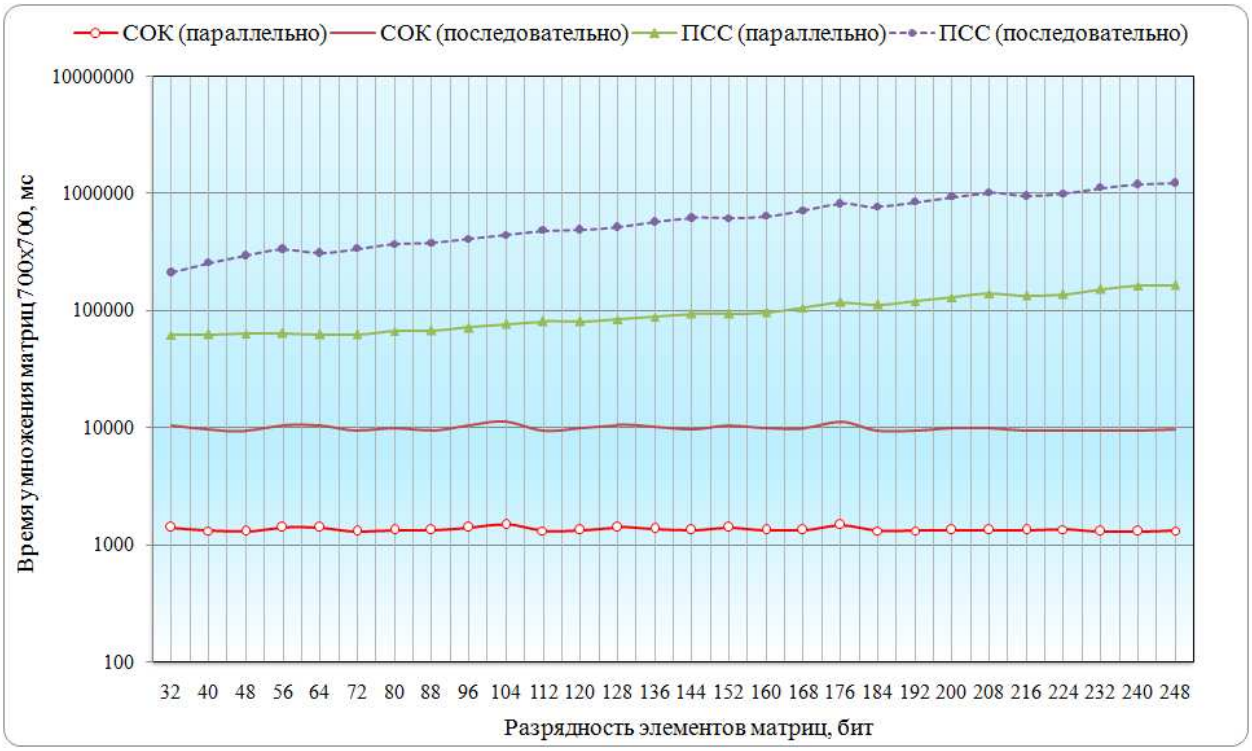

Рис. 4 Зависимость времени умножения матриц 700×700 от разрядности элементов (логарифмическая шкала)

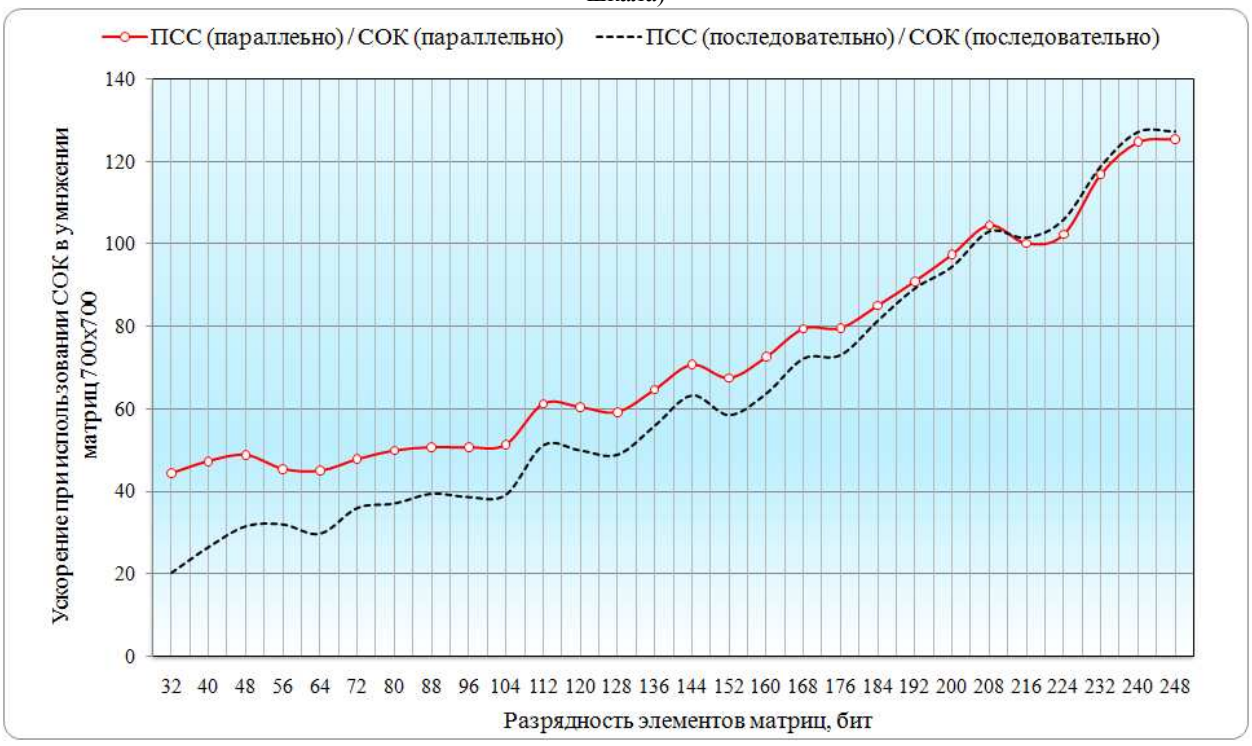

Рис. 5 Зависимость ускорения, достигаемого использованием СОК, от разрядности элементов

Из рисунков 4 и 5 видно, что уже при разрядности элементов в 32 бита время умножения матриц в СОК значительно (в 40 раз при параллельном счете) меньше времени умножения матриц в двоичной системе счисления с использованием пакета cBigNumber. При увеличении разрядности элементов ускорение от использования СОК неуклонно и достаточно резко возрастает; при оперировании с 248-битными элементами ускорение достигает 125 раз. Связано это с тем, что с увеличением разрядности чисел позиционные методы существенно замедляются, тогда как время счета в СОК никоим образом не зависит от разрядности в силу замыкания арифметических операций относительно колец вычетов по выбранным основаниям. Логарифмическая шкала искажает представление, поэтому на рисунке 6 изображен график зависимости времени решения задачи в двоичной системе от разрядности данных с линейной шкалой.

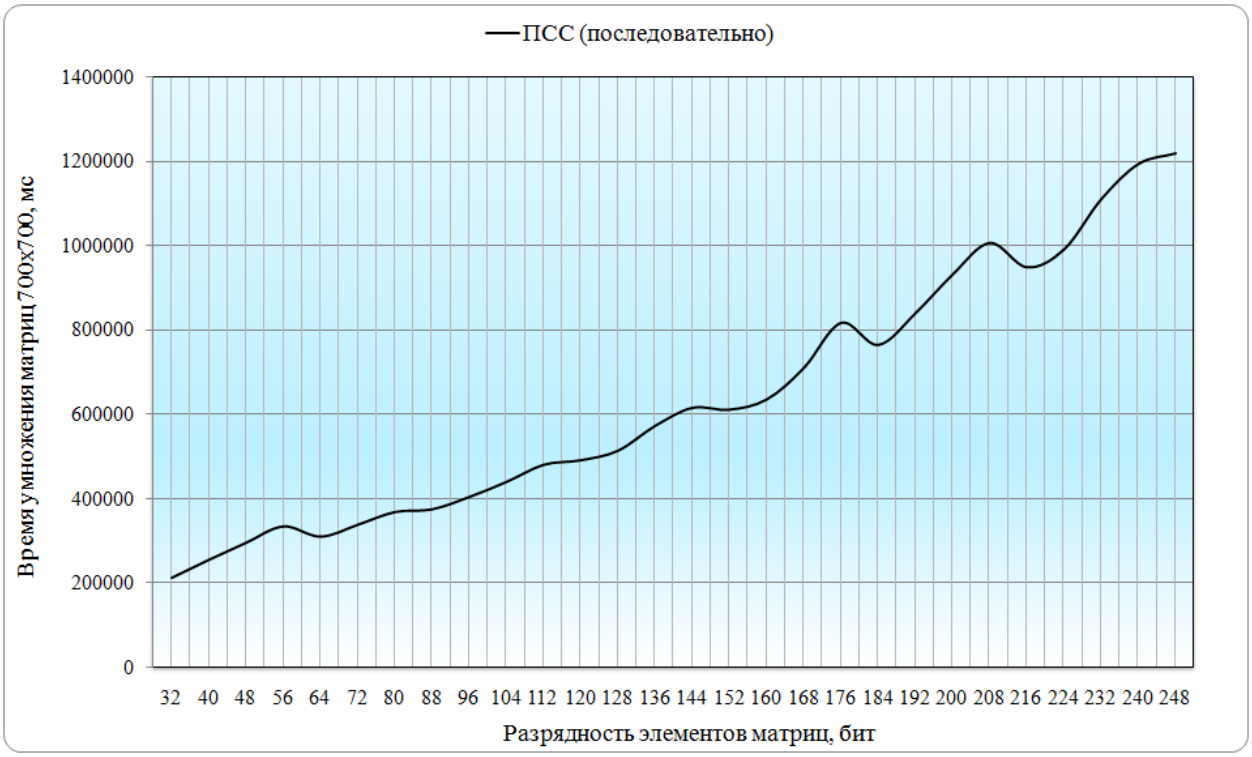

Рис. 6 Зависимость времени решения задачи в двоичной системе от разрядности данных при выполнении последовательного алгоритма умножения матриц в ПСС

Рисунок 6 показывает, что с ростом разрядности элементов матриц время решения задачи в двоичной системе возрастает практически линейно. Экстраполируя функцию изменения времени счета на сверхбольшие машинные диапазоны (1024 бит и более) можно сделать выводы о времени решения задачи, которые поставят под сомнение целесообразность применения средств длинной арифметики (по крайней мере пакета cBigNumber) для высокоточных расчетов. Кроме этого, можно заметить, что на разрядностях 176 бит и 208 бит наблюдается резкое увеличение времени решения задачи, что позволяет предположить о нерегулярности вычислительного процесса с использованием пакета cBigNumber. Для подтверждения данного предположения были построены графики времени решения задачи для других размеров матриц (рисунок 7).

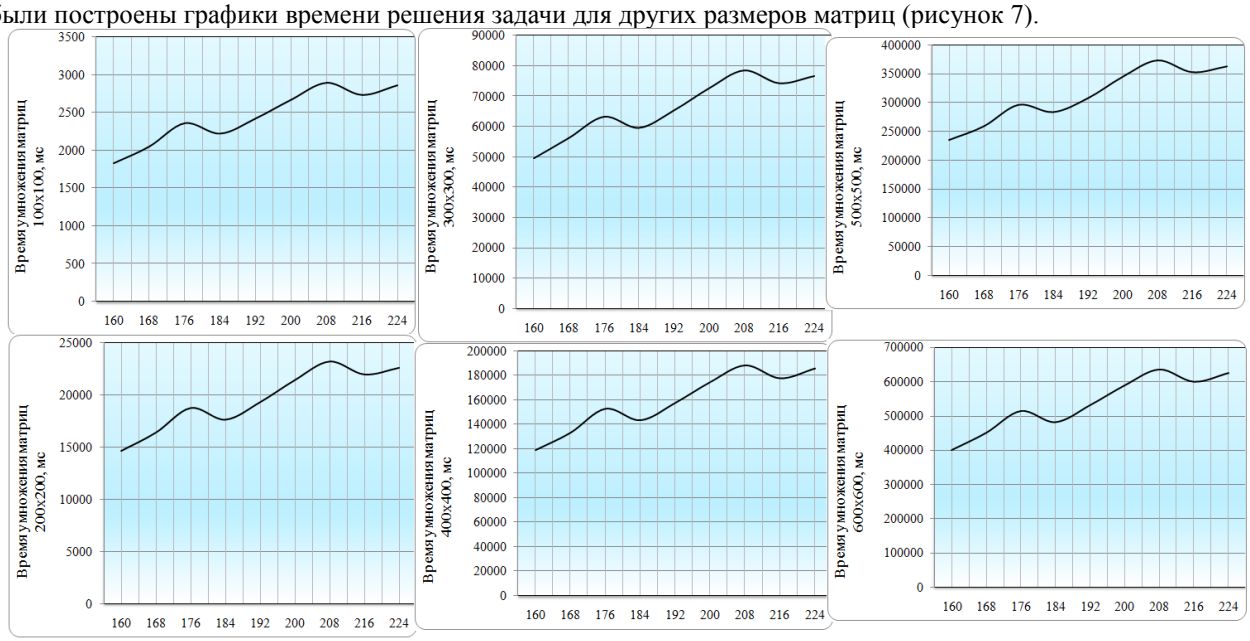

Рис. 7 «Всплески» увеличения времени решения задачи при определенных разрядностях обрабатываемых элементов

В результате были получены зависимости аналогичной формы: «всплески» увеличения времени на разрядностях 176 бит и 208 бит сохраняются, т.е. они систематичны и вызваны не случайными помехами. При счете в СОК каких-либо «всплесков» не выявлено.

Из рисунка 5 так же можно заметить, что при малой разрядности пакет cBigNumber распараллеливается

значительно хуже, чем СОК (более чем в два раза). Это может говорить о том, что cBigNumber, как типичный представитель прикладных средств длинной позиционной арифметики, предназначен лишь для работы с относительно длинными числами и несет в себе дополнительные затраты при работе с числами меньшей разрядности, то есть не универсален. В этом смысле СОК – куда более универсальный аппарат, так как путем выбора соответствующих оснований можно добиться оптимальной работы программы практически на любых разрядностях числовых данных: как на малых, так и на больших.

Следующим проведенным экспериментом являлось исследование эффективности использования СОК при изменении размерности умножаемых матриц с элементами фиксированной разрядности (248 бит). График полученного ускорения представлен на рисунке 8.

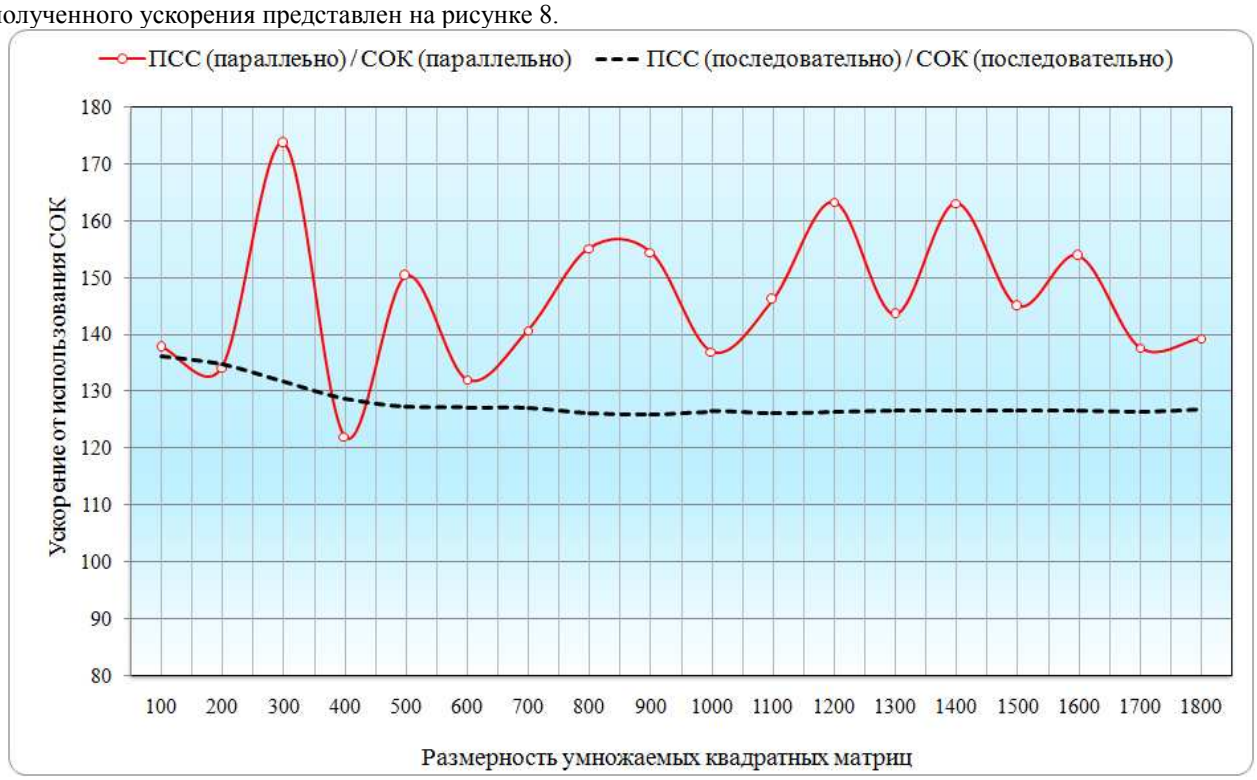

Рис. 8 Зависимость ускорения, достигаемого использованием СОК, от размеров умножаемых матриц

Из графика видно, что на всем диапазоне изменения размерности матриц, ускорение последовательного алгоритма с использованием СОК колеблется незначительно, в то время как при параллельном счете наблюдаются некие колебания ускорения, которые при увеличении числа обрабатываемых элементов затихают. Это свидетельствует о том, что с ростом сложности задачи меньше сказывается влияние случайных помех и накладных расходов, вызываемых, отчасти, затратами времени на создание и удаление параллельных потоков. Тем не менее, на всех размерностях матриц при разрядности их элементов в 248 бит, минимальным является ускорение в 120 раз, что, с нашей точки зрения, является достаточно хорошим показателем повышения эффективности высокоточных расчетов с использованием системы остаточных классов. Очевидно, что с ростом разрядности обрабатываемых данных ускорение будет возрастать согласно зависимости, представленной на рисунке 5.

Заключение.

В статье рассматривается недвоичная система счисления – система остаточных классов, как высокоэффективный параллельный базис для выполнения вычислений над многоразрядными числами. Использование СОК позволяет преобразовать исходные многоразрядные числа в группы независимых чисел меньшей разрядности с последующей параллельной обработкой элементов этих групп без образования длинных цепочек переносов между полиномиальными разрядами. Благодаря этому предлагаемый метод выполнения арифметических операций над многоразрядными числами, основанный на использовании СОК, имеет следующие преимущества перед его позиционными аналогами:

1. Возможность эффективного более быстрого выполнения элементарных операций над многоразрядными числами, без необходимости применения специальных алгоритмов.
2. Наличие в самой системе остаточных классов еще одного вида параллелизма, не зависящего от алгоритма выполнения численных расчетов.
3. Возможность эффективного полномерного использования ресурсов современных многоядерных CPU и других вычислителей, обеспечивающих возможность многопоточного выполнения вычислений.
4. Возможность более гибкого размещения структур данных в памяти вычислительной системы.

Для обоснования этих преимуществ был проведен ряд экспериментов, которые подтвердили выдвигаемые теоретические положения, продемонстрировав существенное ускорение счета в СОК относительно традиционных позиционных методов длинной арифметики. Кроме этого, в ходе экспериментов выявлено такое преимущество СОК, как более регулярный вычислительный процесс, т.е. отсутствие необоснованных систематических «всплесков» увеличения времени решения задачи на протяжении всего процесса вычислений. В ПСС напротив, такие «всплески» наблюдаются на разрядностях 176 и 208 бит из диапазона [32; 248].

ЛИТЕРАТУРА:

1. Суперкомпьютерное конструирование биоорганических нанопроводов [Текст] / А.Л. Шайтан, П.Г. Халатур, А.Р. Хохлов // Суперкомпьютерные технологии в науке, образовании и промышленности / под ред. В.А. Садовничего, Г.И. Савина, Вл.В. Воеводина. – М., 2009. – С. 66–71.
2. Метод динамического программирования для подсчета числа циклов на прямоугольной решетке [Электронный ресурс]. – Режим доступа: http://habrahabr.ru/blogs/algorithm/105705. – Загл. с экрана. – 8.12.2010.
3. Сейсморазведка и суперкомпьютеры [Электронный ресурс] / Е.А. Курин // Международная суперкомпьютерная конференция «Научный сервис в сети Интернет: суперкомпьютерные центры и задачи» : г. Новороссийск, 20-25 сентября 2010 г. : сб. трудов / РАН, Суперкомпьютер. консорциум ун-тов России. – Электрон. изд. – [М.] : Изд-во МГУ, 2010. – 1 эл. опт. диск (CD-ROM). - Заглавие с экрана.
4. Суперкомпьютерные технологии в медицине [Текст] / В.А. Садовничий, В.Б. Сулимов // Суперкомпьютерные технологии в науке, образовании и промышленности / под ред. В.А. Садовничего, Г.И. Савина, Вл.В. Воеводина. – М., 2009. – С. 16–23.
5. М.В. Лобес Разработка методов и алгоритмов модулярных вычислений для задач большой алгоритмической сложности: дис. … канд. физ.мат. наук. – Ставрополь, 2009. – 192 с.
6. Д.Б. Малашевич Недвоичные системы в вычислительной технике [Электронный ресурс] / Д.Б. Малашевич // Материалы международной научной конференции «Модулярная арифметика». – Режим доступа: http://www.computer-museum.ru/books/archiv/sokcon27.pdf. – Загл. с экрана. – 8.12.2010.
7. И.Я. Акушский Машинная арифметика в остаточных классах [Текст] / И.Я. Акушский, Д.И. Юдицкий. – М.: Сов. Радио, 1968. – 440 с.
8. К.С. Исупов ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ДЛЯ ПРОЕКТИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ МАСШТАБИРУЕМЫХ ПРОГРАММ ЧИСЛЕННЫХ РАСЧЕТОВ [Текст] / К.С. Исупов, В.С. Князьков // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики / главный ред. д.т.н., проф. В.О. Никифоров. – СПб., 2010. – Выпуск 6(70). – С. 68-72.
9. A. Omondi Residue Number Systems: Theory and Implementation (Advances in Computer Science and Engineering Texts) [Text] / Amos Omondi, Benjamin Premkumar. - London : Imperial College Press, 2007. - 312 pages.
10. C++ class for integers of unlimited range [Electronic resource]. – Режим доступа: http://www.imach.uran.ru/cbignum/. – Загл. с экрана. – 11.04.2011.