МОДЕЛИРОВАНИЕ ЧИСЛОВЫХ ТИПОВ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ТРОИЧНОГО КОДИРОВАНИЯ

Ю.В. Строганов, И.В. Рудаков (Москва)

В настоящее время в вычислительных задачах, как правило, используются расчёты, связанные с большим количеством операций с вещественными числами. Известно, что полного отображения множества иррациональных чисел в современных ЭВМ добиться невозможно, поэтому применяются приближенные значения, описываемые стандартом IEEE-754 для чисел с плавающей запятой, нечёткие числа, например интервальная арифметика, задокументированная стандартом IEEE-1788, особые типы данных, от известного рационального типа и длинной арифметики, до попытки создать особую алгебру на основе аппроксимации (аппроксиметы). Данные подходы пытаются устранить ошибки, возникающие в процессе вычислений, связанные как с неточностью представляемых аргументов, так и с неточностью получаемого результата. Однако, несмотря на достаточно широкий выбор различных альтернатив, повышение точности вычислений происходит и до сих пор экстенсивно, т.е. путём увеличения количества разрядов у элементарных типов данных (различные библиотеки длинной арифметики для целых и вещественных типов) [1,2].

Интенсивным можно считать переход к наилучшей системе счисления с точки зрения отображаемого объёма информации для одного разряда. При записи с помощью n знаков в p-ичной системе счисления, можно использовать $\frac{n}{p}$ разрядов, что позволит описать $f(p,n) = p^{\frac{n}{p}}$ состояний. Соответственно, $f(p,n) \to max$ при $p = 2.71828182845 \ldots = e$[3]. Использование иррационального основания теоретически возможно, однако практически не реализуемо в силу как теоретических, так и физических ограничений, связанных с оперированием иррациональными числами. Ближайшими же целочисленными основаниями являются 3 и 2. На основе троичной системы счисления была разработана и серийно выпускалась ЭВМ «Сетунь». Она обладала большим количеством преимуществ в плане надёжности, скорости работы по сравнению с современными ей двоичными аналогами, но использовала альтернативное архитектурное решение на физическом уровне, что требовало нового подхода для создания более совершенной версии. К сожалению, широкого распространения данная технология не получила.[4]

Следует отметить, что существуют две троичных системы счисления: симметричная и несимметричная. Особенностью симметричной системы счисления является использование коэффициентов, в том числе, и меньших нуля, в отличии от несимметричной, где используются лишь неотрицательные коэффициенты (1).

$$\begin{cases} 15_{10} = 120_3 \\ 15_{10} = 1(-1)(-1)0_3 \end{cases} \quad (1)$$

Причём преимуществом любой симметричной системы счисления является статистически равная нулю ошибка округления. Потому разработка системы, реализующей вычисления на основе троичной симметричной системы счисления является актуальной задачей.

Для троичной системы счисления базовым разрядом будем являться не бит, как для двоичной, а трит, находящийся в одном из трёх состояний (для симметричной системы это будут $\{-1,0,1\}$, для удобства обозначим $-1_3 = \bar{1}_3$). Соответственно, для реализации троичного целого числа нужна последовательность троичных разрядов (2), причём для отображения положительных и отрицательных чисел отсутствует необходимость в использовании знакового разряда, т.е. числа большие и меньшие нуля хранятся единообразно (3).

$$\sum_{i=0}^{n} k_i \cdot 3^i, k_i \in \{-1,0,1\} \quad (2)$$

$$\begin{aligned} 15_{10} &= 1\bar{1}\bar{1}0_3 \\ -15_{10} &= \bar{1}110_3 \end{aligned} \quad (3)$$

При использовании операции сложения двух троичных разрядов организуется разряд переноса (таблица 1).

Таблица 1 – формирование разряда переноса при операции сложения троичных разрядов с учётом разряда переноса.

(-1)	1			0	1			1	1		
1	1,0	1,1	,-1	1	1,1	,-1	,0	1	,-1	,0	,1
	1,1	,-1	, 0		,-1	,0	,1		,0	,1	,-1
	,-1	,0	, 1		,0	,1	,-1		,1	,-1	,0

Из-за хранения знака числа внутри всех разрядов, операции сложения и вычитания при таком способе кодирования чисел реализуются одним алгоритмом [5,6].

Также, согласно таблице 1, разряд переноса формируется меньше, чем в трети возможных исходов (4), в отличии от двоичного случая (5).

$$\frac{3+2+3}{9\cdot 3}=\frac{8}{27} \quad (4)$$

$$\frac{1+3}{4*2}=\frac{4}{8}=\frac{1}{2} \quad (5)$$

Можно представить число в виде (6)

$$\sum_{i=0}^{n} k_i\cdot 3^i=\sum_{i=0}^{\left[\frac{n}{2}\right]-1} k_i\cdot 3^i+\sum_{j=\left[\frac{n}{2}\right]}^{n} k_j\cdot 3^j=\sum_{i=0}^{\left[\frac{n}{2}\right]-1} k_i\cdot 3^i+3^{\left[\frac{n}{2}\right]}\cdot\sum_{j=0}^{\left[\frac{n}{2}\right]} k_j\cdot 3^j, \quad (6)$$

где $k_i, k_j \in \{\bar{1},0,1\}$

И осуществлять сложение чисел X и Y по частям, причём сложение частей может осуществляться независимо друг от друга, а формируемый разряд переноса статистически (4) не будет порождать больших изменений у части числа с большими степенями.

Иными словами, сложение чисел будет выполняться как (7)

$$Z=\mathrm{X}+\mathrm{Y}=\left(\sum_{i=0}^{\frac{n}{2}} x_i\cdot 3^i+\sum_{j=\frac{n}{2}+1}^{n} x_j\cdot 3^j\right)+\left(\sum_{i=0}^{\frac{n}{2}} y_i\cdot 3^i+\sum_{j=\frac{n}{2}+1}^{n} y_j\cdot 3^j\right)=\left(\sum_{i=0}^{\frac{n}{2}} x_i\cdot 3^i+\sum_{i=0}^{\frac{n}{2}} y_i\cdot 3^i\right)+\left(\sum_{j=\frac{n}{2}+1}^{n} x_j\cdot 3^j+\sum_{j=\frac{n}{2}+1}^{n} y_j\cdot 3^j\right)=\left(\sum_{i=0}^{\frac{n}{2}} x_i\cdot 3^i+\sum_{i=0}^{\frac{n}{2}} y_i\cdot 3^i\right)+\left(\sum_{\substack{j=\frac{n}{2}+1\\ k=0}}^{\substack{n\\ \frac{n}{2}-1}} x_j\cdot 3^k+\sum_{\substack{j=\frac{n}{2}+1\\ k=0}}^{\substack{n\\ \frac{n}{2}-1}} y_j\cdot 3^k\right)\cdot 3^{\frac{n}{2}+1} \quad (7)$$

Тогда результат первой скобки будет на один элемент больше, поскольку операция сложения может порождать ненулевой разряд переноса (8)

$$\left(\sum_{i=0}^{\frac{n}{2}} x_i\cdot 3^i+\sum_{i=0}^{\frac{n}{2}} y_i\cdot 3^i\right)=\sum_{i=0}^{\frac{n}{2}+1} a_i\cdot 3^i=a_{\frac{n}{2}+1}\cdot 3^{\frac{n}{2}+1}+\sum_{i=0}^{\frac{n}{2}} a_i\cdot 3^i \quad (8)$$

И тоже будет верно для второй скобки, с учётом того, что разряды имеют больший вклад в итоговое значение (9)

$$\left(\sum_{\substack{j=\frac{n}{2}+1\\ k=0}}^{\substack{n\\ \frac{n}{2}-1}} x_j\cdot 3^k+\sum_{\substack{j=\frac{n}{2}+1\\ k=0}}^{\substack{n\\ \frac{n}{2}-1}} y_j\cdot 3^k\right)=a_{\frac{n}{2}}\cdot 3^{\frac{n}{2}}+\sum_{t=0}^{\frac{n}{2}-1} a_t\cdot 3^t \quad (9)$$

Чтобы учесть разряд переноса с первой скобки, необходимо (10)

$$a_{\frac{n}{2}}\cdot 3^{\frac{n}{2}}+\sum_{t=0}^{\frac{n}{2}-1} a_t\cdot 3^t+a_{\frac{n}{2}+1}\cdot 3^0=z_{\frac{n}{2}+1}\cdot 3^{\frac{n}{2}+1}+z_{\frac{n}{2}}\cdot 3^{\frac{n}{2}}+\sum_{t=0}^{\frac{n}{2}-1} z_t\cdot 3^t \quad (10)$$

И получается, что (11)

$$Z = 3^{\frac{n}{2}+1} \cdot (z_{\frac{n}{2}+1} \cdot 3^{\frac{n}{2}+1} + z_{\frac{n}{2}} \cdot 3^{\frac{n}{2}} + \sum_{t=0}^{\frac{n}{2}-1} z_t \cdot 3^t) + \sum_{i=0}^{\frac{n}{2}-1} a_i \cdot 3^i = \sum_{j=0}^{n+2} z_j \cdot 3^j \quad (11)$$

Разработанный авторами алгоритм параллельного сложения троичных чисел оказывается эффективнее последовательного, поскольку разряд переноса, получаемый в младшей части числа, невсегда приводит к каскадной генерации разряда переноса в более старшей части, складываемой отдельно. Верно, что выполнение операции сложения в (8) и (9) могут быть выполнены рекурсивным вызовом этого же алгоритма, однако наибольшую эффективность показывает использование последовательного алгоритма для указанных частей.

Воспользовавшись (6), для перемножения чисел, получается (12)

$$Z = \mathrm{X} \cdot \mathrm{Y} = \left(\sum_{i=0}^{\frac{n}{2}} x_i \cdot 3^i + \sum_{j=\frac{n}{2}+1}^{n} x_j \cdot 3^j\right) \cdot \left(\sum_{i=0}^{\frac{n}{2}} y_i \cdot 3^i + \sum_{j=\frac{n}{2}+1}^{n} y_j \cdot 3^j\right) = \left(\sum_{i=0}^{\frac{n}{2}} x_i \cdot 3^i \cdot \sum_{i=0}^{\frac{n}{2}} y_i \cdot 3^i\right) + \left(\sum_{i=0}^{\frac{n}{2}} x_i \cdot 3^i \cdot \sum_{\substack{i=0 \\ j=\frac{n}{2}+1}}^{\frac{n}{2}-1 \; n} y_j \cdot 3^i + \sum_{i=0}^{\frac{n}{2}} y_i \cdot 3^i \cdot \sum_{\substack{i=0 \\ j=\frac{n}{2}+1}}^{\frac{n}{2}-1 \; n} x_j \cdot 3^i\right) \cdot 3^{\frac{n}{2}+1} + \left(\sum_{\substack{j=\frac{n}{2}+1 \\ k=0}}^{\frac{n}{2}-1 \; n} x_j \cdot 3^k \cdot \sum_{\substack{j=\frac{n}{2}+1 \\ k=0}}^{\frac{n}{2}-1 \; n} y_j \cdot 3^k\right) \cdot 3^n \quad (12)$$

Также и алгоритм параллельного умножения целых троичных чисел оказывается эффективнее при достаточном количестве разрядов у аргументов [7]. Следует отметить, что разбиение более чем на две части исходной последовательности разрядов не является эффективным с точки зрения скорости вычислений (таблица 2).

Таблица 2 – Время моделирования операций сложения и умножения в зависимости от количества задействованных потоков.

	1 поток	2 потока	3 потока	4 потока	5 потоков
сложение	4.4мс	4.03мс	4.15мс	4.4мс	5.09мс
умножение	20.3мс	17.15мс	18.8мс	17.8мс	19мс

Заключение

С помощью разработанного программного комплекса были проведены эксперименты, моделирующие поведение троичных числовых типов данных, которые подтвердили гипотезу о преимуществах троичного представления над двоичным. Разработанные алгоритмы могут быть реализованы и для двоичного представления чисел, но преимуществ от их внедрения не будет из-за отсутствия особенностей компенсирования сформированного разряда переноса в симметричных системах счисления. Реализация числовых типов в троичном симметричном коде позволяет использовать параллелизуемые алгоритмы для элементарных арифметических операций над ними. Для операции сложения получается выигрыш в 8% при использовании двух потоков, в то время как для параллельного использования операции умножения выгоднее использовать 4 потока с выигрышем в 12.3%.

Литература

1. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ. М.: Издательство «Наука ФИЗМАТЛИТ», 2007
2. Юровицкий В.М. АППРОКСИМЕТИКА: математическая теория и компьютеринг приближенных чисел. 1998. [Электронный ресурс] Режим доступа: http://www.yur.ru/science/computer/appro/monograa.htm Последнее обращение: 17.09.2019
3. Фомин С.В. Системы счисления – 5е изд. – М.:Наука, гл.ред.физ.-мат. лит., 1987 – 48с.
4. Виртуальный компьютерный музей [Электронный ресурс] Режим доступа: http://www.computer-museum.ru Последнее обращение 20.09.2019
5. Рамиль Альварес Х. Алгоритмы троичной арифметики. 20с., 2012.

6. Воеводин В.В. and Ким Г.Д. Машинные операции с точки зрения математика. Вычислительные методы и программы, (26), 1977.
7. Stroganov I.V., Volkova L., Rudakov I.V. (2019) On Parallel Addition and Multiplication via Symmetric Ternary Numeral System. In: Dolinina O., Brovko A., Pechenkin V., Lvov A., Zhmud V., Kreinovich V. (eds) Recent Research in Control Engineering and Decision Making. ICIT 2019. Studies in Systems, Decision and Control, vol 199. Springer, Cham