

Корнага Ярослав ІгоровичКандидат технічних наук, доцент, доцент кафедри технічної кібернетики, orcid.org/0000-0001-9768-2615

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

ВИКОРИСТАННЯ БАЗ ДАНИХ І МОВ ПРОГРАМУВАННЯ У ВИСОКОТОЧНИХ ОБЧИСЛЕННЯХ ЧИСЕЛ З ПЛАВАЮЧОЮ ТОЧКОЮ ВЕЛИКОЇ РОЗРЯДНОСТІ

***Анотація.** Для багатьох наукових обчислень, що пов'язані з емпіричними даними, використовується 32-бітна арифметика з плаваючою точкою, яка дає результати достатньої точності для розрахунків. Проблема сучасної обчислювальної техніки полягає в правильному підборі сховищ для зберігання даних та проведення обчислень, а оцінка правильності проведення обчислень повинна бути виконана з більш високою точністю. Відомі розв'язки, для яких такої точності недостатньо, тому використання методів розрахунків із 64-бітною арифметикою з плаваючою точкою є більш придатним. Для деяких дуже складних задач виникає потреба у більш високих рівнях точності при великому масиві даних. У статті розглянуто задачу використання різних сучасних мов програмування для розв'язання задач обчислень з високою точністю. Багатоядерні і багатовузлові паралельні обчислення можна виконувати з високою точністю для різних процесів в різних сферах діяльності. Особливо важливим аргументом є точність проведення розрахунків для таких галузей, як енергетична, хімічна, машинобудівна промисловість. Проведено вибір методів для проведення високоточних обчислень та аналіз пакетів для високоточної арифметики, які використовуються у мовах програмування. Розроблене середовище для проведення експериментів з можливістю підключення чотирьох комп'ютерів та бібліотек мов програмування. Також проведено експериментальне дослідження з різними базами даних з використанням великих масивів даних, за якими побудовано відповідні індекси у високоточних обчисленнях та проводиться порівняння для різних типів даних. Використовувалися реляційні бази даних Oracle та MS SQL, а також нереляційні бази даних MongoDB та Oracle NoSQL. В результаті проведених експериментів показано, що бібліотеки C++ не достатньо швидко працюють з високоточними числами, а методи, які застосовувалися для оцінювання швидкості виконання високоточних операцій класичними мовами програмування, дають результати, які відрізняються менше ніж на 5%. Експеримент з базами даних, показав, що нереляційні бази даних проводять обчислення з більшою швидкістю, ніж реляційні, причому результати відрізняються не більше, ніж на 2%. База даних Oracle проводить розрахунок більш ніж на 30% швидше за MS SQL.*

Ключові слова: високоточні обчислення; арифметика з плаваючою точкою; мови програмування; бази даних

Вступ

В наукових обчисленнях з емпіричними даними 32-бітна арифметика з плаваючою точкою є досить точною, і є кращою, оскільки вона економить оперативну пам'ять, час запуску і споживання електроенергії. А в інших обчисленнях 64-бітна арифметика з плаваючою точкою потрібна для отримання результатів з достатньою точністю. Іноді є потреба використовувати при обчисленнях результати, які отримуються об'єднанням 32-бітових і 64-бітових методів. Одна з проблем сучасної обчислювальної техніки полягає в правильному доборі сховищ для зберігання даних та проведення обчислень, які допоможуть користувачам визначити, які частини можуть бути виконані з меншою точністю, а які мають бути виконані з більш високою

точністю. При обчисленнях великих масивів даних виявлено, що з швидким зростанням об'єму інформації точність результатів була незадовільною навіть при використанні 64-бітної арифметики, а швидкість опрацювання зменшувалась [1–4].

Виклад основного матеріалу**Методи і програмне забезпечення для високоточної арифметики**

Дотепер найбільш поширена форма додаткової точність арифметики приблизно в два рази перевищує рівень стандартної 64-бітної арифметики з плаваючою точкою. Одним із варіантів є стандарт для 128-бітної арифметики з плаваючою точкою, з мантисою 113 біт, який не реалізований в апаратних засобах, але операції такого типу даних

підтримуються в програмному забезпеченні за допомогою спеціальних бібліотек. В базах даних такі обчислення доводиться проводити за допомогою написання додаткових процедур зі зміною параметрів типу даних залежно від операцій та розміру цілої частини числа.

Реалізація стандарту для 128-бітної арифметики, яка була реалізована, називається «подвійний подвійного» (double-double), точність якого складає 31 знак. Такий тип складається з двох 64-розрядних типів (S, T), де S являє собою значення з плаваючою точкою 64-бітної точності, а T – різниця між істинним значенням і S. Для таких рівнів точності (до кількох сотень цифр) арифметика здійснюється за допомогою пристосування відомих методів, а саме: метод Каратцуби, швидке перетворення Фур'є (ШПФ), метод Ньютона, метод Тейлора та алгоритми багаторозрядної арифметики (БРА) з модулярно-позиційним форматом.

Нині існує декілька вільно доступних високоточних пакетів програмного забезпечення, разом із супроводжуваними інтерфейсами на мові високого рівня, які використовують оператори, які роблять перетворення відповідне коду. У більшості випадків тільки один змінює висловлювання типу тих змінних, які повинні розглядатися в якості високої точності і робить кілька інших модифікацій. Після цього, коли один з цих змінних з'являється у виразі, то правильні базові підпрограми автоматично викликаються.

Ось декілька доступних на сьогодні пакетів для високоточної арифметики з плаваючою точкою:

1) ARPREC: Підтримка довільної точності з багатьма алгебраїчними і трансцендентними функціями. Включає в себе інтерфейси високого рівня для C++.

2) CLN: C++ бібліотека підтримка довільної точності з численними алгебраїчними і трансцендентними функціями.

3) Julia: середовище програмування високого рівня, який включає в себе GMP і MPFR.

3) MPFR: Підтримка декількох рівнів точності обчислень з плаваючою точкою з правильним округленням, ґрунтуючись на GMP.

4) mpmath: бібліотека Python для довільної точності арифметики з плаваючою точкою, в тому числі численного трансцендентні.

5) NTL: C++ бібліотека для довільної точності цілого числа і арифметики з плаваючою точкою.

6) Numerics: бібліотека C# для чисел з великою точністю.

7) JInterval: бібліотека Java для чисел з великою точністю.

Очевидно, що це додаткові системні витрати для виконання операцій високоточної арифметики. В деяких випадках втрати часу складають від 25 до 50

разів. Цього можна уникнути, використавши методи описані вище, тільки для невеликої частини коду, тому загальний час роботи може збільшитися в 2 рази. З появою паралельних обчислювальних систем швидкість проведення високоточних обчислень значно збільшилась.

В базах даних можливе застосування не усіх методів для розв'язання задач з великою точністю, а проведення дослідження на швидкість опрацювання даних є критерієм правильного вибору СУБД для зберігання та опрацювання інформації.

Вимоги до високоточного арифметичного програмного забезпечення

У зв'язку з виникненням багатоядерних і багатовузлових паралельних обчислень, можна виконувати з високою точністю обчислення різних радіофізичних процесів, використовуючи інтерфейси передачі повідомлень програмного забезпечення (MPI) на рівні додатків, а не розпаралелювати на окремі високоточні операції. Також на сучасних системах, які застосовують багатоядерні процесори та паралельні обчислення, більш ефективно виконуються операції з використанням загальної пам'яті (OpenMP), в межах одного вузла, навіть якщо MPI використовується для паралелізму між вузлами.

Поява графічних процесорів (GPU) на чіпі прискорювачів від Intel MIC, які зараз час використовуються для великомасштабних високопродуктивних обчислювальних додатків привела до більш ефективного застосування методів високоточних обчислень [2 – 6].

Програмне забезпечення повинно забезпечувати основні арифметичні операції на будь-якому рівні точності. Сучасні високоточні пакети мають підтримувати такі основні функції [4 – 8]:

1. Основні базові функції – експонента, логарифм, синус, косинус, тангенс, гіперболічні функції та відповідні обернені функції.

2. Функції роботи з інтегралами.

3. Функції векторної алгебри.

4. Функції матричної та тензорної алгебри.

5. Гіперболічні функції.

Ці функції мають бути реалізовані за допомогою найкращих доступних алгоритмів для різних діапазонів аргументів і рівнів точності, а також повинні підтримувати як реальні, так і складні аргументи. Для проведення розрахунків мовою SQL необхідно писати запити, які реалізують різні функції використання обчислень, а відсутність (наприклад, деяких базових) роботи з інтегралами та гіперболічними функціями призводить до використання методів, які були описані вище [9 – 10].

Проведення створення моделюючого середовища для порівняння високоточних обчислень

Для оцінки ефективності алгоритмів високоточної арифметики було розроблено програмне моделююче середовище на різних мовах програмування для проведення експериментів. Вибрано персональний комп'ютер з такими характеристиками:

1. Процесор – Intel Core i7- 3.40 GHz, 10M Cache .
2. Оперативна пам'ять – 8 Gb.
3. Жорсткий диск – 1 Tb.
4. Відеокарта – 2 Gb.

У всіх експериментах точність обчислень становила 72 десяткові цифри. Для проведення експерименту були вибрані такі арифметичні операції: додавання (plus), віднімання (min), множення (mult), ділення (div). Основними мовами програмування, які були вибрані для проведення експерименту, є:

1. C#.
2. Java.
3. Python.
4. C++.

А основними базами даних, в яких проводилися експерименти, є:

1. Oracle.
2. MS SQL.
3. MongoDB.
4. Oracle NoSQL.

Розроблена схема моделюючого середовища представлена на рис. 1.

Перший експеримент проводився для порівняння оцінок всіх операцій для різних мов програмування (рис. 2), який показав час виконання їх залежно від вибору мови.

Другий експеримент проводився для порівняння оцінок операції множення для різних методів обчислень (рис. 3), який показав час виконання їх залежно від вибору методів.

Час в цих експериментах оцінювався за допомогою таких функцій:

1. C# – microtime.
2. Java – nanotime.
3. Python – timeit.
4. C++ – clock.

Перший експеримент проводився для порівняння оцінок всіх операцій для різних мов програмування (рис. 2), який показав час виконання їх залежно від вибору мови.

Другий експеримент проводився для порівняння оцінок операції множення для різних методів обчислень (рис. 3), який показав час виконання їх залежно від вибору методів.

Час в цих експериментах оцінювався за допомогою таких функцій:

1. C# – microtime.
2. Java – nanotime.
3. Python – timeit.
4. C++ – clock.

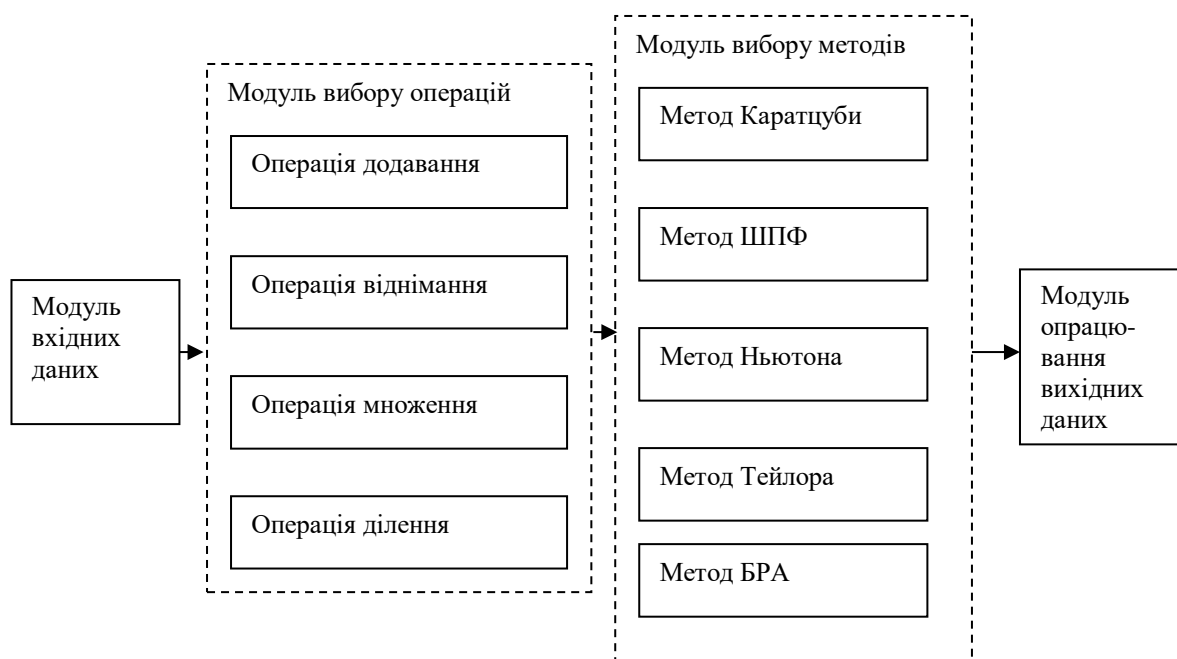


Рисунок 1 – Схема моделюючого середовища проведення експериментів

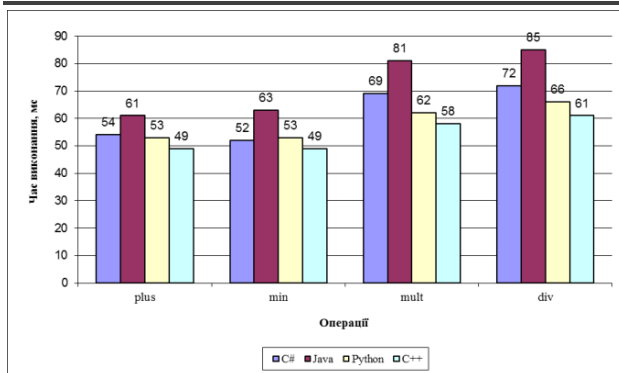


Рисунок 2 – Порівняльні оцінки часу виконання операцій різними мовами програмування за допомогою алгоритму БРА

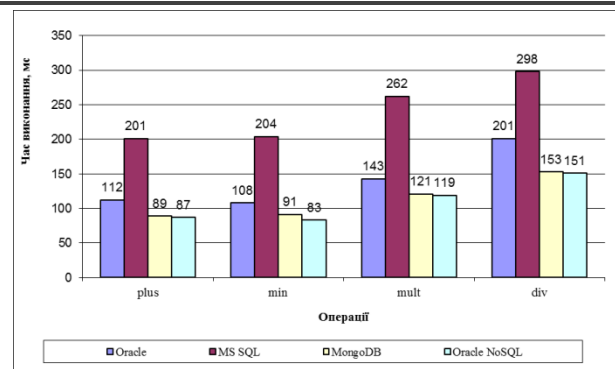


Рисунок 4 – Порівняльні оцінки часу виконання операцій різними базами даних

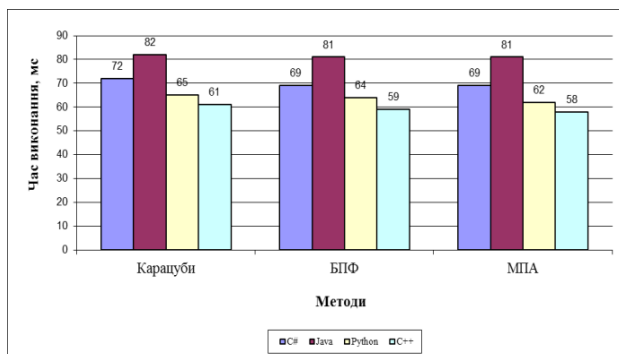


Рисунок 3 – Порівняльна оцінка часу виконання операції множення для різних методів

Третій експеримент проводився для порівняння оцінок всіх операцій для різних баз даних (рис. 4) з використання масиву даних, що дорівнюють одному мільйону записів у таблиці, який показав час виконання їх залежно від вибору мови. Виконувалось 20 запитів до стовпчика таблиці, за яким побудовано індекс та бралося середнє значення.

Висновок

Запропоновано аналіз оцінювання високоточних операцій чисел з плаваючою точкою. Визначено методи, які допомогли провести експериментальні дослідження та показали, що при виборі мов програмування велику увагу слід приділяти бібліотекам, які застосовуються для роботи з високоточним обчисленням. В результаті проведених експериментів показано, що бібліотеки C++ не достатньо швидко працюють з високоточними числами, а методи, які застосовувалися для оцінки швидкості виконання високоточних операцій класичними мовами програмування дають результати, які відрізняються менш ніж на 5%. Експеримент з базами даних, показав, що нереляційні бази даних проводять обчислення з більшою швидкістю, ніж реляційні. Результати нереляційних відрізняються не більше ніж на 2%, а база даних Oracle проводила розрахунок даних більш як на 30% швидше за MS SQL.

Список літератури

1. Bailey D.H., Borwein J.M. High-precision arithmetic: progress and challenges. [Electronic resource]. – <http://www.davidhbailey.com/dhbpapers/hp-arith.pdf>.
2. Исупов К.С. Методика выполнения базовых немодульных операций в модулярной арифметике с применением интервальных позиционных характеристик // Известия высших учебных заведений. Поволжский регион. Технические науки. – № 3. – 2013. – С. 26 – 39.
3. Исупов К.С. Об одном алгоритме сравнения чисел в системе остаточных классов // Вестник Астраханского государственного технического университета. – № 3. – 2014. – С. 40 – 49.
4. Цюцюра М.І. Застосування генетичного алгоритму для формування функції належності нечітких множин / М.І. Цюцюра, А.В. Єрукаєв // Інформаційні технології управління. – № 36. – 2018. – С. 71 – 75.
5. Исупов К.С. Модулярное масштабирование степеню двойки с произвольным шагом / К.С. Исупов, А.Н. Мальцев // Издательство ВятГУ. – 2014. – С. 1179 – 1184.
6. Chang C.C. A division algorithm using bisection method in residue number system / C.C. Chang, J.H. Yang // International Journal of Computer, Consumer and Control. – N 1. – 2013. – С. 59 – 66.
7. Gbolagade K.A. An $O(n)$ residue number system to mixed radix conversion technique/ K.A. Gbolagade, S.D. Cotofana // IEEE International Symposium on Circuits and Systems. – 24–27 May, 2009. – P. 521 – 524.
8. Федусенко О.В. Концептуальна модель адаптивної інформаційної системи навчання / О.В. Федусенко, А.О. Федусенко, І.М. Доманецька // Інформаційні технології управління. – №32. – 2017. – С. 86 – 90.
9. Корнага Я.І. Порівняльні оцінки застосування методів підвищення швидкості пошуку та запису даних в базах даних // Адаптивні системи автоматичного управління. – № 1(22). – 2013. – С. 37 – 44.
10. Мухін В.Є. Аналіз ефективності оброблення запитів серверами гетерогенних розподілених баз даних / В.Є. Мухін, Я.І. Корнага // Технічні науки та технології. – №1. – 2016. – С. 89 – 94.

Стаття надійшла до редколегії 26.05.2020

Kornaga Yaroslav

PhD, Associate Professor, Associate Professor of the Department of technical cybernetic, orcid.org/0000-0001-9768-2615
National Technical University of Ukraine "Igor Sikosky Kyiv Polytechnic Institute", Kyiv

USING OF DATABASES AND PROGRAMMING LANGUAGES IN HIGH-PRECISION CALCULATIONS OF FLOATING-POINT NUMBERS

Abstract. Many scientific calculations involving empirical dates use 32-bit floating-point arithmetic, which gives results of sufficient accuracy for calculations. The problem of modern computers is the correct selection of repositories for data storage and calculations, and the assessment of the correctness of the calculations has been performed with greater accuracy. There are solutions for which such accuracy is not sufficient, so using of calculation methods with 64-bit floating-point arithmetic is more appropriate. There is a requirement for higher levels of accuracy for some very complex tasks with a large data set. This article considers the problem of using various modern programming languages to solve computational problems with high accuracy. There are multi-core and multi-node parallel calculations that can be performed with high accuracy for different processes in different areas of activity. A particularly important argument is the accuracy of calculations for industries such as energy, chemical, engineering. The choice of methods is made for high-precision calculations and analysis of packages for high-precision arithmetic, which are used in programming languages. An environment has been developed for conducting experiments with the possibility of connecting four compilers and libraries of programming languages. An experimental study was also conducted with different databases using large data sets, on which the corresponding indices in high-precision calculations were constructed and comparisons were made for different types of data. Relational Oracle and MS SQL databases were used, as well as non-relational MongoDB and Oracle NoSQL databases. As a result of experiments, it was shown that C++ libraries do not work fast enough with high-precision numbers and the methods used to estimate the speed of high-precision operations in classical programming languages give results that differ by less than 5%. The experiment with databases showed that non-relational databases perform calculations with a faster rate than relational ones, and the results differ by no more than 2%. The Oracle database calculated more than 30% faster than MS SQL.

Keywords: high-precision calculations; floating point arithmetic; programming languages; databases

References

1. Bailey, D.H. & Borwein, J.M., (2018). High-precision arithmetic: progress and challenges. [Electronic resource]. – <http://www.davidhbailey.com/dhbpapers/hp-arith.pdf>.
2. Isupov, K.S., (2013). Methods of performing basic non-modular operations in modular arithmetic using interval positional characteristics. *Izvestiya vysshikh uchebnykh zavedenii. Volga region. Technical sciences*, 3, 26 – 39.
3. Isupov, K.S., (2014). On an algorithm for comparing numbers in the system of residual classes. *Bulletin of the Astrakhan State Technical University*, 3, 40 – 49.
4. Tsyutsyura, M.I. & Yerukaev, A.V., (2018). Application of genetic algorithm for formation of fuzzy set membership function. *Information technologies of management*, 36, 71 – 75.
5. Isupov, K.S. & Maltsev, A.N., (2014). Modular scaling by the degree of two with an arbitrary step. *Vyatchevo Publishing House*, 1179 – 1184.
6. Chang, C.C. & Yang, J.H., (2013). A division algorithm using bisection method in residue number system. *International Journal of Computer, Consumer and Control*, 1, 59 – 66.
7. Gbolagade, K.A. & Cotofana, S.D., (2009). An $O(n)$ residue number system to mixed radix conversion technique. *IEEE International Symposium on Circuits and Systems*, 521 – 524.
8. Fedusenko, O.V., Fedusenko, A.O. & Domanetskaya, I.M., (2017). Conceptual model of adaptive information system of education. *Information technologies of management*, 32, 86 – 90.
9. Kornaga, Ya.I., (2013). Comparative evaluations of application of methods of increase of speed of search and record of data in databases. *Adaptive systems of automatic control*, 1 (22), 37 – 44.
10. Mukhin, V.E. & Kornaga, Ya.I., (2016). Analysis of the efficiency of query processing by servers of heterogeneous distributed databases. *Technical sciences and technologies*, 1, 89 – 94.

Посилання на публікацію

APA Kornaga, Ya. (2020). *Using of Database and Programming Languages in High Precision Calculations of Floating-Point Numbers. Management of Development of Complex Systems*, 42, 51 – 55; dx.doi.org/10.32347/2412-9933.2020.42.51-55.

ДСТУ Корнага Я.І. Використання баз даних та мов програмування у високоточних обчисленнях чисел з плаваючою точкою великої розрядності [Текст] / Я.І. Корнага // *Управління розвитком складних систем*. – 2020. – № 42. – С. 51 – 55; dx.doi.org/10.32347/2412-9933.2020.42.51-55.