

УДК 004.413

DOI <https://doi.org/10.32838/2663-5941/2020.5/14>

Корнага Я.І.

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

Герасименко О.Ю.

Київський національний університет імені Тараса Шевченка

Базака Ю.А.

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

Базалій М.Ю.

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

Мухін О.В.

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

ВИКОРИСТАННЯ БАЗ ДАНИХ ТА МОВ ПРОГРАМУВАННЯ У ВИСОКОТОЧНИХ ОБЧИСЛЕННЯХ ЧИСЕЛ ІЗ ПЛАВАЮЧОЮ ТОЧКОЮ ВЕЛИКОЇ РОЗРЯДНОСТІ

Для багатьох наукових обчислень, що пов'язані з емпіричними даними, використовується 32-бітна арифметика із плаваючою точкою, яка дає результати достатньої точності для розрахунків. Проблема сучасної обчислювальної техніки полягає у правильному підборі сховищ для зберігання даних та проведення обчислень, а оцінка правильності проведення обчислень повинна бути виконана з більш високою точністю. Існують розв'язки, для яких такої точності не достатньо, тому використання методів розрахунків із 64-бітною арифметикою із плаваючою точкою є більш придатною. Для деяких дуже складних задач виникає потреба в більш високих рівнях точності при великому масиві даних. У даній статті розглядається задача використання різних сучасних мов програмування для вирішення задач обчислень з високою точністю. Існують багатоядерні і багатовузлові паралельні обчислення, які можна виконувати з високою точністю для різних процесів у 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 аналізу пакетів для високоточної арифметики, які використовуються у мовах програмування. Розроблене середовище для проведення експериментів із можливістю підключення чотирьох компіляторів та бібліотек мов програмування. Також було проведено експериментальне дослідження з різними базами даних із 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. У результаті проведених експериментів було показано, що бібліотеки C++ не достатньо швидко працюють із високоточними числами та методами, які застосовувалися для оцінки швидкості виконання високоточних операцій класичними мовами програмування, дають результати, які відрізняються менше ніж на 5%. Експеримент із 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, ніж MS SQL.

Ключові слова: високоточні обчислення, арифметика з плаваючою точкою, мови програмування, розподілені бази даних.

Постановка проблеми. У 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 у використанні при обчисленнях результатів, які отримуються об'єднанням 32-бітових і 64-бітових методів. Одна з проблем сучасної обчислювальної техніки полягає в правильному підборі сховищ для зберігання даних та проведення обчислень, які допоможуть користувачам визначити, які частини можуть бути виконані з меншою точністю, а які повинні бути виконані з більш високою точністю. При обчисленнях великих масивів даних було виявлено, що з швидким зростанням об'єму інформації точність результатів була не задовільною навіть при використанні 64-бітної арифметики, а швидкість обробки падала [1–4].

Мета статті – проведення порівняльного аналізу наявних систем обробки великих масивів інформації з використанням гетерогенних розподілених баз у високоточних обчисленнях з плаваючою точкою великої розрядності та дослідження швидкості обробки даних в таких системах.

Аналіз останніх досліджень і публікацій. До цих пір найбільш поширена форма додаткової точності арифметики приблизно у два рази перевищує рівень стандартної 64-бітної арифметики із плаваючою точкою. Одним з варіантів є стандарт для 128-бітної арифметики із плаваючою точкою, з мантисою 113 біт, який не реалізований в апаратних засобах, але операції даного типу даних підтримується в програмному забезпеченні за допомогою спеціальних бібліотек. У базах даних такі обчислення приходиться 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. Даний 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.

У даний час існує декілька вільно доступних високоточних пакетів програмного забезпечення, разом із супроводжуваними інтерфейсами на мові високого рівня, використовуючи переваження

операторів, які роблять перетворення відповідне коду. У більшості випадків один тільки змінює висловлювання типу тих змінних, які повинні розглядатися в якості високої точності і робить кілька інших модифікацій. Після цього, коли один із цих змінних з'являється у виразі, правильні базові підпрограми автоматично викликаються.

Ось декілька доступних у даний час пакетів для високоточної арифметики із 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.

4. MPFR: Підтримка декількох рівнів точності обчислень із плаваючою точкою із правильним округленням, ґрунтуючись на GMP.

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

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

7. Numerics: бібліотека C# для чисел з високою точністю.

8. JInterval: бібліотека Java для чисел з високою точністю.

Очевидно, що це додаткові системні витрати для виконання операцій високоточної арифметики. У деяких випадках втрати часу складають від 25 до 50 раз. Цього можна уникнути, використавши методи описані вище, тільки для невеликої частини коду, тому загальний час роботи може збільшитися на в 2 рази. Із появою паралельних обчислювальних систем швидкість проведення високоточних обчислень значно збільшилась.

У базах даних можливе застосування не всіх методів для вирішення задач із високою точністю, а проведення дослідження на швидкість обробки даних являється критерієм правильного вибору СУБД для зберігання та опрацювання інформації.

У зв'язку з виникненням багатоядерних і багатовузлових паралельних обчислень можна виконувати з високою точністю обчислень різних радіофізичних процесів, використовуючи інтерфейс 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 в даний 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 повинні підтримувати такі основні функції [4–8]:

1. Основні базові функції: експонента, логарифм, синус, косинус, тангенс, гіперболічні функції та відповідні зворотні функції.
2. Функції роботи з інтегралами.
3. Функції векторної алгебри.
4. Функції матричної та тензорної алгебри.
5. Гіперболічні функції.

Ці функції повинні 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 приходиться писати запити, які реалізують різні функції використання обчислень, а відсутність, наприклад, деяких базових, роботи з інтегралами та гіперболічних функцій призводить до використання методів, які були описані вище [9–10].

Виклад основного матеріалу. Для оцінки ефективності алгоритмів високоточної арифме-

тики було розроблено програмне моделююче середовища на різних мовах програмування для проведення експериментів. Вибраний персональний комп'ютер із 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 складала 72 десяткові цифри. Для проведення експерименту були вибрані наступні 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 порівняльних оцінок всіх операцій для різних мов програмування (рис. 2), який показав час виконання їх залежно від вибору мови.

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

Час у даних експериментах оцінювався за допомогою наступних функцій:

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

Третій експеримент проводився для порівняльних оцінок всіх операцій для різних баз даних (рис. 4) із використання масиву даних, рівних одному мільйону записів у таблиці, який показав час виконання їх в залежності від вибору мови. Викону-

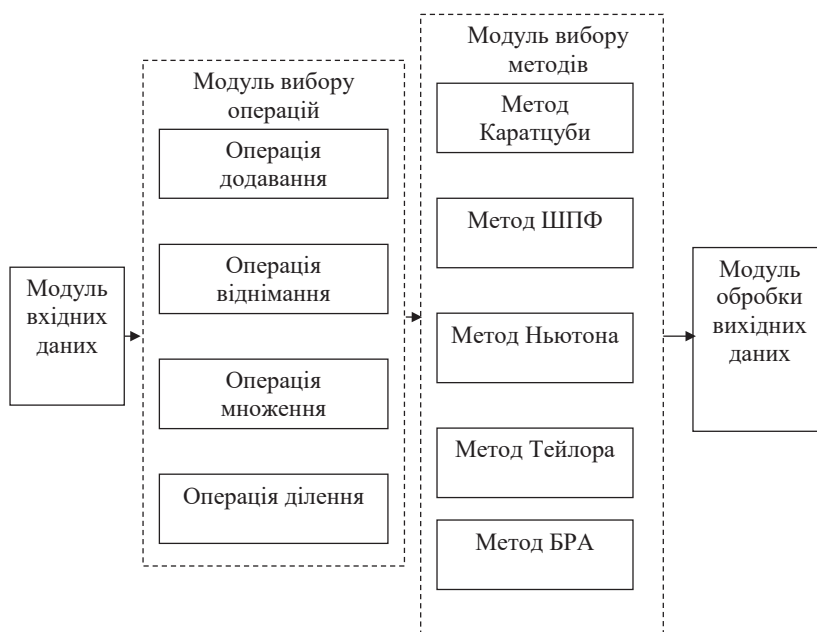


Рис. 1. Схема моделюючого середовища проведення експериментів

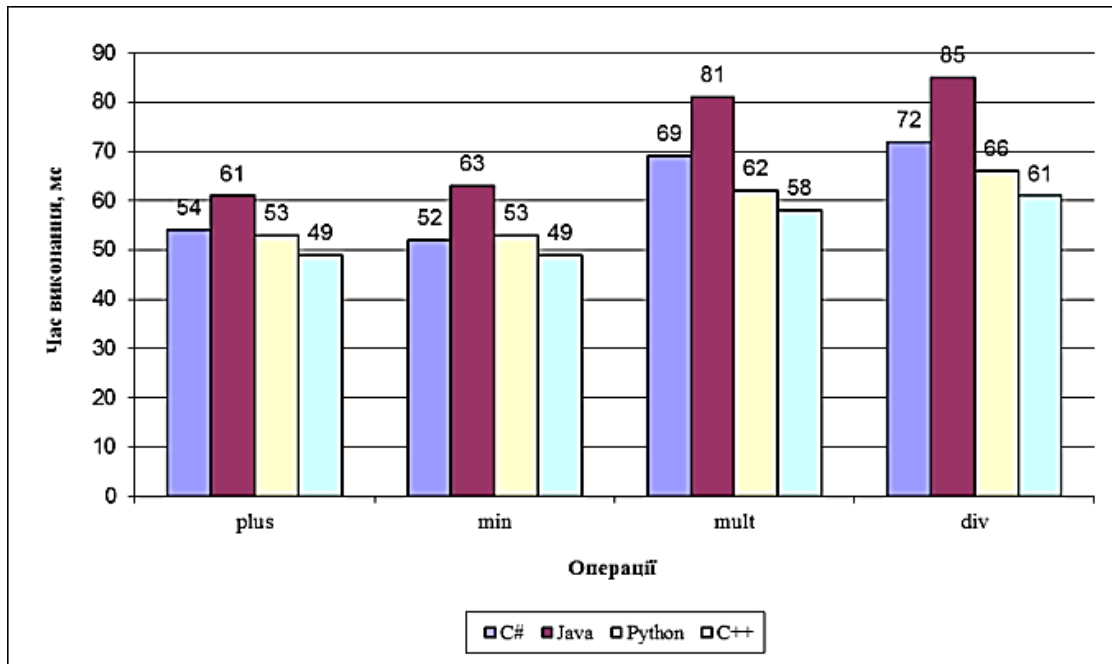


Рис. 2. Порівняльні оцінки часу виконання операцій різними мовами програмування з допомогою алгоритму БРА

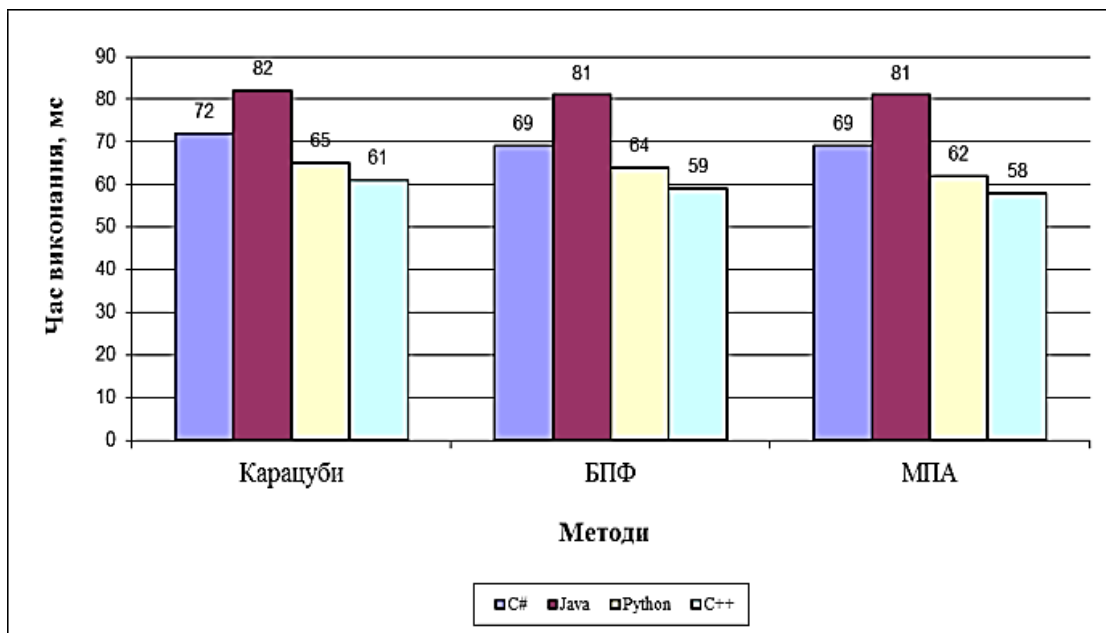


Рис. 3. Порівняльна оцінка часу виконання операції множення для різних методів

валось 20 запитів до стовпчика таблиці, за яким побудовано індекс, та бралось середнє значення.

Усі експерименти проводилися як із засобів керування базами даних, так і з використанням системи сервісів доступу до даних. Для системи сервісів використовувалося імітаційне тестування, яке дозволило тестувати кожен модуль окремо, що необхідно для зменшення навантаження на саму розподілену мережу.

Висновки. У цій статті запропонований аналіз оцінки високоточних операцій чисел із плаваючою точкою. Було визначено методи, які дозволили провести експериментальні дослідження та показали, що під час вибору мов програмування велику увагу слід приділяти бібліотекам, які застосовуються для роботи з високоточним обчисленням. У результаті проведених експериментів було показано, що бібліотеки C++ не достатньо швидко

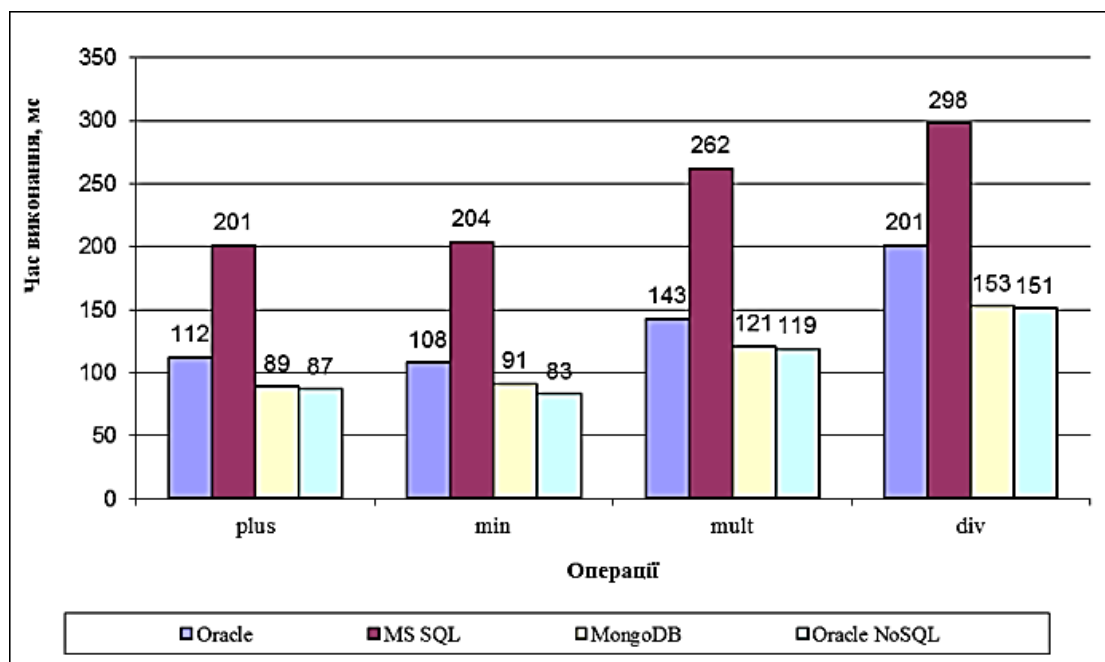


Рис. 4. Порівняльні оцінки часу виконання операцій різними базами даних

працюють з високоточними числами, та методи, які застосовувалися для оцінки швидкості виконання високоточних операцій класичними мовами програмування, дають результати, які відрізняються менше ніж на 5%. Експеримент із 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 розрахунків даних більше як на 30% швидше ніж MS SQL.

Список літератури:

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

Kornaha Ya.I., Herasymenko O.Yu., Bazaka Yu.A., Bazalii M.Yu., Mukhin O.V.

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

Many scientific calculations involving empirical dates uses 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.

Key words: *high-precision calculations, floating point arithmetic, programming languages, databases.*