

## МЕТОДИ ОПТИМІЗАЦІЇ ПРОГРАМ ВИСОКОТОЧНИХ РОЗРАХУНКІВ У СИСТЕМАХ ДИСТАНЦІЙНОГО НАВЧАННЯ

Шапиро О. К.

Науковий керівник – д.т.н., проф. Четвериков Г. Г.

Харківський національний університет радіоелектроніки, каф. ПІ,  
м. Харків, Україна

e-mail: [oleksii.shapyro@nure.ua](mailto:oleksii.shapyro@nure.ua)

The topic of the research is the metrics for assessing the complexity of high-precision calculations software for distance learning system. The aim of the work is to determine the method of building a metric for assessing the complexity of high-precision calculations software for distance learning system. The result of the work is the proposal of a new method for an accurate assessment of the complexity and the identification of areas for potential optimization for high-precision calculations software for distance learning system.

Останнім часом у світі відбувається багато значних подій, які змушують переходити до дистанційного формату навчання, який реалізується за допомогою систем дистанційного навчання [1]. Під час такого формату навчання, при проведенні наукових досліджень, виникає потреба моделювати різні процеси у віртуальному середовищі (через неможливість їх фізичного/реального відтворення), що часто потребує виконувати високоточні розрахунки [2]. Для такого типу програмного забезпечення питання продуктивності є дуже чутливим, і погана оптимізація має значні наслідки. Тому цей об'єкт є гарним прикладом для ілюстрації неточності існуючих метрик оцінки складності програмного забезпечення, а також для того, щоб запропонувати новий метод побудови метрики оцінки складності програмного забезпечення для виявлення областей, що потребують комплексної оптимізації продуктивності.

Метрика програмного забезпечення – це міра, що дозволяє отримати числове значення деяких властивостей програмного забезпечення та його специфікацій. Метрики оцінки якості програмного забезпечення дають уявлення про надійність програм, їх продуктивність, складність підтримки та внесення змін, тощо. Нас цікавлять метрики оцінки складності, які поділяють на 3 основні групи: метрики розміру програми, метрики складності потоку управління програмою та метрики складності потоку даних програм.

До метрик розміру програми відносять, наприклад, SLOC, ABC, та Halstead. Дані метрики використовують для визначення загальної складності кодової бази, складності розробки та тестування, та якості коду та складності розширень. Такі метрики не можуть бути використані для оцінки складності програм з точки зору їх виконання, тому в контексті високоточних розрахунків у системах дистанційного навчання.

Прикладом метрик складності потоку управління програмою можуть бути метрики цикломатичної складності, Myers, Hansen, та Chen. Дані метрики використовують для визначення загальної складності кодової бази, загальної алгоритмічної складності програми, областей для потенційної алгоритмічної оптимізації, та тестових сценаріїв. Такі метрики не враховують складність окремих кроків/операцій програми. В контексті високоточних розрахунків у системах дистанційного навчання, складність окремих кроків/операцій програми можуть відрізнятися на порядки, тому цей аспект унеможливорює пряме використання таких метрик.

Метрики складності потоку даних програм представлені, наприклад, метриками Cherin, Span, та деякими гібридними. Дані метрики використовують для визначення рівня зв'язаності програмних модулів, складності тестування, та областей для потенційної оптимізації пов'язаної з пам'яттю. Такі метрики не можуть бути використані в контексті високоточних розрахунків у системах дистанційного навчання, оскільки не враховують ні складність кроків/операцій програми, ні складність алгоритму програми.

Розглянуті існуючі метрики оцінки складності програм можуть дати непогану оцінку якогось одного їх аспекту, проте вони слабо розраховані на оцінку складності програм, які виконують високоточні обчислення, де критично важливо врахувати декілька аспектів одночасно. Таким чином існує потреба у визначенні методу побудови метрики оцінки складності програмного забезпечення високоточних розрахунків.

При проведенні аналізу існуючих метрик оцінки складності програмного забезпечення в контексті програм високоточних розрахунків, було виявлено, що метрика цикломатичної складності дає гарні результати при оцінці загальної алгоритмічної складності програми, але вона не може бути використана в оригінальному вигляді, через її ключовий недолік, що вона не враховує складність окремих кроків/операцій програми. Тож, для вирішення поставленої задачі, пропонується модифікувати дану метрику шляхом введення оцінки складності окремих кроків/операцій програми беручи до уваги необхідну точність розрахунків (кількість знаків після коми).

Основою для програм, що здійснюють високоточні розрахунки, є отримані в ході різних досліджень математичні моделі, що складаються з систем алгебраїчних рівнянь. Для того, щоб оцінити складність і витрати часу їх розрахунку, необхідно вміти оцінювати найосновніший елемент цих моделей – рівняння і формули, написані у вигляді вихідного коду.

Для цього пропонується використовувати бінарні дерева формул, вузли якого можуть містити як типи операцій, так і змінні, які в них беруть участь (альтернативою може бути декомпозиція за допомогою алгебри предикатів [3], такий підхід був би більш універсальним, але в контексті цієї роботи він є надлишковим). Представлення у такому вигляді дає можливість оцінити складність кожної окремої операції у формулі.

Так як рівняння математичних моделей можуть включати змінні різної точності, то операції з цими змінними займають різний час обчислень і, відповідно, вони будуть мати різну складність. Незважаючи на те, що в кінцевому підсумку все доводиться до єдиної точності, вона може постійно змінюватися в процесі обчислення, саме тому важливо розглянути кожен з операцій окремо, для чого аналізується побудоване бінарне дерево.

Дерево проглядається знизу вгору, починаючи з найсвіжіших елементів, так як вони відповідають змінним формули. Вони використовуються для визначення складності вузла операції, з якого походять ці змінні, виходячи з точності цих змінних (точністю операції буде максимальна точність змінних). Далі, вузли операцій що знаходяться у вихідних вузлах з іншої операції, самі починають вважатися змінними з точністю, знайденою на попередньому кроці, і так далі. В результаті загальна точність дерева буде точністю його кореня, або максимальною точністю вузлів, що виходять від нього. Але складність дерева не залежить від загальної точності дерева, так як точність вузлів може змінюватися в міру просування від кінця дерева до його кореня, і якщо один вузол має меншу або більшу точність ніж інші, це вплине на час обчислення операції, а отже, вплине і на загальну складність дерева.

Таким чином, формули з однаковими деревами, але зі змінними різної точності матимуть різну складність і різний час обчислень, навіть якщо їх загальна точність в кінцевому підсумку виявиться однаковою.

Уміння знаходити складність формул дає можливість оцінити загальну складність програми з набагато більшою точністю, в порівнянні з базовою метрикою цикломатичної складності, а також дає змогу розрізняти складність програм з однаковими графами потоків управління. По суті, програма розбивається на мінімальні частини, кожна з яких являє собою рядок вихідного коду, що містить різні команди або математичні операції, і ці частини об'єднуються в вузли графа потоку управління. Потім оцінюється складність кожного окремого вузла цього графа на основі його складових. Отримані оцінки поєднуються з цикломатичною складністю програми.

Список використаних джерел:

1. Williams P., Nicholas D., Gunter B. E-learning: what the literature tells us about distance education: An overview // *Aslib Proceedings*. 2005. Vol. 57, No 2. P. 109-122. URL: <https://doi.org/10.1108/00012530510589083> (дата звернення: 13.03.2024).

2. Bailey D., Borwein J. High-Precision Arithmetic in Mathematical Physics // *Mathematics*. 2015. Vol. 3, No 2. P. 337-367. URL: <https://doi.org/10.3390/math3020337> (дата звернення: 13.03.2024).

3. Shubin I. Development of conjunctive decomposition tools // *CEUR Workshop Proceedings*. 2021. Vol. 2870. P. 890–900. URL: <https://ceur-ws.org/Vol-2870> (дата звернення: 13.03.2024).