



УДК [378.091.33:004.41]:159.952.1
DOI 10.32999/ksu2413-1865/2019-86-66

ПРОБЛЕМА ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПІД ЧАС ВИКЛАДАННЯ ДИСЦИПЛІНИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ

Сіциліцин Ю.О.,
старший викладач кафедри інформатики і кібернетики
Мелітопольський державний педагогічний університет

У статті порушено проблему формування у студентів бакалаврів із комп'ютерних наук навичок розроблення паралельних алгоритмів та реалізації у вигляді програм під час викладання дисципліни паралельного програмування. Проведено аналіз сучасних засобів розробки програмного забезпечення, до складу яких можуть входити бібліотеки або класи для написання паралельного коду і зовнішніх бібліотек, які у поєднанні з мовою програмування дають можливість розробки паралельних програм.

Ключові слова: паралельне програмування, паралельні обчислення, засоби розробки програмного забезпечення, OpenMP, MPI, Java, C#.

В статті затронута проблема формування у студентів бакалавров по комп'ютерним наукам навичок розробки паралельних алгоритмів та їх реалізації в формі програм при викладанні дисципліни паралельного програмування. Проведено аналіз сучасних засобів розробки програмного забезпечення, до складу яких можуть входити бібліотеки або класи для написання паралельного коду і зовнішніх бібліотек, які у поєднанні з мовою програмування дають можливість розробки паралельних програм.

Ключевые слова: параллельное программирование, параллельные вычисления, средства разработки программного обеспечения, OpenMP, MPI, Java, C#.

Sitsylitsyn Yu.O. THE PROBLEM OF CHOICE OF TOOL MEANS OF THE SOFTWARE DEVELOPMENT FOR TEACHING PARALLEL PROGRAMMING IN UNIVERSITIES

The article was written about the problem of developing parallel algorithms for students from computer sciences and their implementation in the programs in the teaching of discipline parallel programming. An analysis of modern software development tools, which may include libraries or classes for writing parallel code and external libraries, which, in combination with the programming language, provide the opportunity to develop parallel programs. The external libraries include the following: OpenMP – Open Multi-Processing and MPI – Message Passing Interface. These libraries extend the capabilities of the C++ programming language and allow the development of parallel programs. Their main drawback – to develop a parallel algorithm, you will first need to develop a sequential algorithm. This development method greatly reduces the efficiency of parallel code and prevents students from developing parallel development algorithms. Among the most significant innovations in the C++ 11 standard should include support for multithreaded applications. This will allow you to write on C++ multithreaded programs with guaranteed behavior without relying on extensions dependent. Unlike the aforementioned libraries, the programming languages Java and C# already have a set of classes and interfaces that allows starting with the use of parallel algorithms for the development of parallel programs.

Key words: parallel programming, parallel computing, software development tools, OpenMP, MPI, Java, C#.

Постановка проблеми. Нині інформаційні технології займають все більш важливу роль у розвитку економіки країни. Компаніям потрібні все більш місткі і складні програмні і програмно-апаратні рішення. Це призводить до збільшення потреби у спеціалістах з інформаційних технологій, але, незважаючи на попит у таких спеціалістах, росте і конкуренція між спеціалістами у сфері ІТ. Конкуренція зумовлена потребою роботодавців отримати спеціаліста, котрий може приступити до роботи відразу, а не витратити гроші і час на його навчання. Одним із методів підвищення конкурентоспроможності є зниження собівартості розроблення програмного забезпечення. Розробка програмного забезпечення за стандартними послідовними алгоритмами,

орієнтуючись на послідовну програму, яка залучає у своїй роботі тільки одне ядро, вже не дає необхідного приросту продуктивності у порівнянні з програмами, які використовують у своєму коді паралельні багатоядерні рішення.

Останні десятиріччя суперкомп'ютери знаходять своє застосування у рішенні практично будь-яких завдань науки та техніки [6, с. 5]. Серед таких завдань є моделювання різних фізичних процесів, завдання обчислювальної хімії та біології, нанотехнології, автоматизація проектування та багато інших. Прогрес у галузі високопродуктивних обчислень багато в чому визначає темп розвитку науки і техніки, а тому й рівень технологічного розвитку держави в цілому. Тому можна з упевненістю ствер-



джувати, що створення і вивчення методів розробки програмно-апаратного забезпечення для високопродуктивних обчислювальних систем є одним із найважливіших завдань сучасних інформаційних технологій [17, с. 6]. Надзвичайно важливою є підготовка спеціалістів у цій галузі. Для того у програму багатьох закладів вищої освіти для студентів бакалаврів з галузі знань 12 «Інформаційні технології» введений курс «Паралельне програмування». Паралельні обчислення є узагальненням терміном, що застосовується для позначення технологій та методів розробки програмно-апаратного забезпечення для високопродуктивних комп'ютерних систем. Тому термін «паралельні обчислення» описує достатньо широку галузь, яка пов'язана з організацією розрахунків в обчислювальних системах, що містять декілька процесорних пристроїв. До таких систем входять багатоядерні процесори, багатопроцесорні системи із загальною пам'яттю, високопродуктивні обчислювальні кластери з розподіленою пам'яттю або гібридною архітектурою, системи, що реалізують загальні обчислення на основі відеоадаптерів (GPGPU), хмарні обчислення (Cloud Computing), тощо [17, с. 6]. Тому вибір засобів розробки програмного забезпечення паралельних обчислень мають велике значення для ознайомлення студентів закладів вищої освіти з основними алгоритмами та парадигмами паралельного програмування.

Аналіз останніх досліджень і публікацій. Фундаментальні дослідження у галузі підготовки майбутніх інженерів-програмістів у закладах вищої освіти здійснюють В. Биков, В. Осадчий, З. Сейдаметова, О. Спірін, Ю. Триус. Також науковці висвітлюють окремі аспекти зазначених проблем, зокрема питання якості професійної підготовки програмістів (А. Власюк, В. Круглик, К. Осадча, А. Стрюк); вимоги до професійних якостей майбутніх інженерів-програмістів (Н. Длугунович, А. J. Ko, P. L. Li, J. Zhu); організація навчання майбутніх інженерів-програмістів у вишах інших країн світу – В. Круглик. Огляд та вибір засобів паралельного програмування займалися С.А. Лупін, М.А. Посипкін, В.В. Воеводін, Е. Уїльямс та інші вчені. І.Н. Скопін досліджує навчання паралельному програмуванню, науковець А.Д. Шеметова робить огляд методичних прийомів навчання паралельному програмуванню.

Метою статті є аналіз та вибір засобів розробки програмного забезпечення які допоможуть студентам сформуванню паралельний стиль мислення, вивчаючи дисципліну паралельне програмування.

Виклад основного матеріалу. Дисципліна паралельного програмування викладається студентам бакалаврам із галузі знань 12 «Інформаційні технології» на третьому або четвертому курсах і має сприяти розвитку у студентів навиків паралельного програмування.

Головна проблема паралельного програмування заснована на відносній складності побудови схеми паралельних обчислень у голові програміста. Людина ж мислить послідовно. Саме тому послідовною є більшість дійсних сьогодні мов програмування [13, с. 6]. Нині паралельні обчислення реалізуються у вигляді паралельно-послідовних програм. При цьому в дійсному послідовному алгоритмі виділяються незалежні один від одного послідовні гілки, які можуть бути виконані паралельно, тому пишеться паралельно-послідовна програма. Нерідко береться вже дійсна послідовна програма, яка шляхом додавання деяких конструкцій розпаралелювання перетворюється в паралельно-послідовну. Чим складніше вихідна послідовна програма, чим гірше вона структурована, чим більш заплутані залежності між окремими її гілками, тим складніше виконати її розпаралелювання [2, с. 5].

Традиційне викладання програмування виходить із передумови, що учнем слід прищепити навички алгоритмічного мислення на основі вивчення деякого простої мови програмування. У процесі освоєння матеріалу розширюється коло засобів мови, які є корисними для подання алгоритмів у вигляді програми. За своєю суттю це шлях формування шаблонів, комбінованих різними доступними способами для отримання алгоритмічно осмислених текстів. Він досить зручний у викладанні, оскільки мотивує учнів до освоєння нового матеріалу [10, с. 1].

Водночас форми подання алгоритмів у мовах програмування завжди обмежують програміста, змушують його описувати дії з набору допустимих мовою, а не викладати те, що він може собі уявити на рівні осмислення завдання. Так, крок за кроком програміст звужує свої природні можливості оперування даними та діями до рівня мовних засобів, формуються стереотипи мислення, які зумовлені мовою, що часто сприймається як образ мислення, характерний для програмування [10, с. 2].

Стереотипи та обмеження способу мислення програмістів виявляють себе наочно, коли вирішуються завдання, які передбачають паралелізм виконання. Це підтверджується підходом до розробки паралельної програми, коли спочатку будується її послідовна версія, а потім до неї застосовується розпаралелювання.



Розглянемо те, чи можна за допомогою сучасних засобів розробки програм уникнути вищезначених стереотипів та обмежень.

Перед розглядом засобів розробки програмного забезпечення для паралельних обчислень розглянемо можливі режими виконання паралельних програм та основні парадигми паралельного програмування.

Під час розгляду проблеми організації паралельних обчислень потрібно розрізняти такі можливі режими виконання незалежних частин програми [3, с. 12], як багатозадачний режим; паралельне виконання; розподілені обчислення.

Наведемо графіки, які ілюструють теоретично розраховане по закону Амдала [1, с. 66] прискорення виконання програм за умов підвищення доли паралельних обчислень у програмі (рис. 1).

Виходячи з графіку, видно, що ефективність паралельної програми залежить від доли паралельного коду в програмі в цілому. Для розробки ефективного паралельного коду розглянемо основні парадигми паралельного програмування.

Парадигма програмування – спосіб концептуалізації, який визначає те, як слід проводити обчислення, і як робота, виконувана комп'ютером, має бути структурована і організована. [1, с. 15].

Парадигми паралельного програмування поділяються на такі основні види [1, с. 15], якітеративний паралелізм; рекурсивний паралелізм; «Клієнти і сервери»; «Виробники і споживачі»; взаємодіючі рівні.

Тепер розглянемо основні бібліотеки та мови програмування, які реалізують можливість розробки паралельних програм.

Будемо розглядати засоби розробки паралельних програм у порядку їх історичної появи. Розглянемо переваги та недоліки цих технологій під час застосування її на лабораторних роботах для формування у студентів паралельного стилю мислення. Під паралельним стилем мислення будемо розуміти такий спосіб алгоритмічної мисленої діяльності, котрий за умов спеціально організованої послідовності дій дозволяє побудувати паралельний алгоритм [11, с. 4].

OpenMP (Відкрита багато процесорна обробка) – стандарт програмного інтерфейсу додатків для паралельних систем із загальною пам'яттю. Підтримує мови C, C++, Фортран [7, с. 19].

Модель паралельної програми в OpenMP можна сформулювати так: Програма складається з послідовних і паралельних секцій. У початковий момент часу створюється головна нитка, яка виконує послідовні секції програми. При вході в паралельну секцію виконується операція fork, що породжує сімейство ниток. Кожна нитка має свій унікальний числовий ідентифікатор (Головною ниткою є нитка за номером 0). При розпаралелюванні циклів всі паралельні нитки виконують один код. У загальному випадку нитки можуть виконувати різні фрагменти коду. При виході з паралельної секції виконується операція join. Завершується виконання всіх ниток, крім головної.

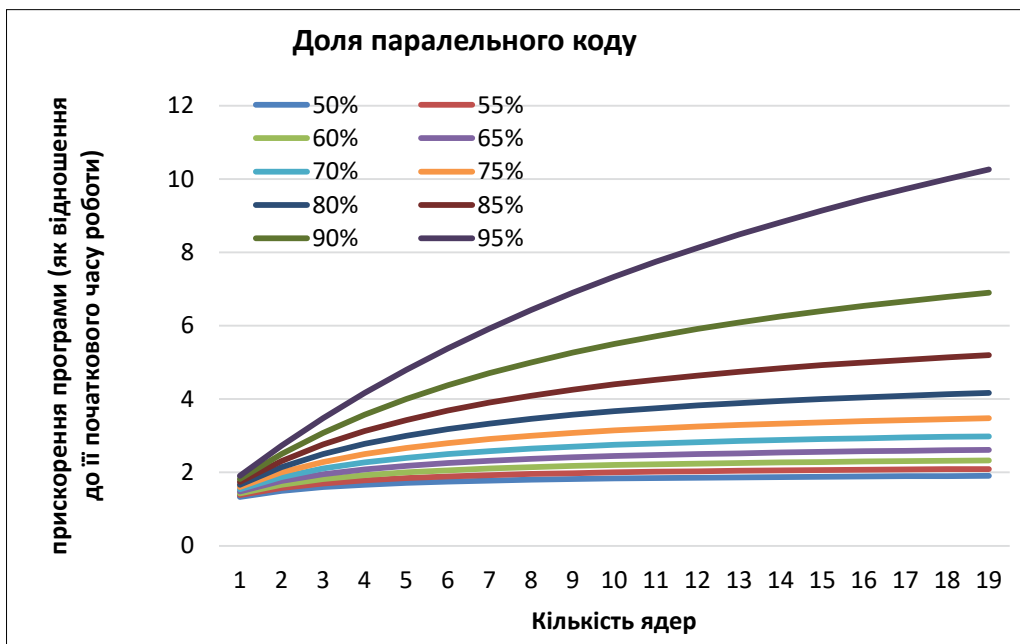


Рис. 1. Прискорення виконання програм за умов підвищення доли паралельних обчислень у програмі



Використання мови програмування C++, яку більшість студентів вивчає ще на першому році навчання та поглиблює на старших курсах на споріднених мовах програмування, дає можливість студентам зосередитись на написанні паралельного коду, не відволікаючись на синтаксис. Структура програмного коду з використанням директив OpenMP є інтуїтивно зрозумілою. Використання директив OpenMP дозволяє отримати паралельну програму з найменшою кількістю зусиль. Але структура програми з використання технології OpenMP складає використання паралельних та послідовних секцій, тобто стимулює розробку паралельної програми від послідовного алгоритму до паралельного шляхом переробки послідовного алгоритму. Набір директив OpenMP у поєднанні з мовою C або C++ дозволять розв'язати типові обчислювальні завдання, але спочатку потрібно розробити послідовний алгоритм. Також для написання програм із використання директив OpenMP програмістові потрібно мати знання з розробки послідовних програм мовою C або C++. Такий стиль програмування не дуже сприяє паралельному мисленню.

У моделі програмування MPI (Інтерфейс передачі повідомлень) програма при запуску породжує кілька процесів, що взаємодіють між собою за допомогою повідомлень. Підтримує мови C, C++, Фортран [7, с. 32].

Оскільки бібліотека MPI також використовує мову програмування C++, то все зазначене вище про OpenMP можна зазначити і про MPI. У моделі програмування MPI програма породжує кілька процесів, що взаємодіють між собою за допомогою повідомлень. Така структура коду дозволяє розробляти як паралельні, так і розподілені програми. Використання бібліотеки MPI потребує від програміста більш розвинутого алгоритмічного мислення, тому розробка програм за допомогою MPI потребує більше зусиль із боку програміста. Але структура програми, як і у випадку з OpenMP, є послідовною з включенням паралельних відрізків, тобто передбачає від програміста розробку послідовного алгоритму та переробки його у паралельно-послідовний варіант. Бібліотека MPI (у зв'язку з мовою C або C++) дозволять розв'язати усті типові обчислювальні завдання, але спочатку буде потрібно розробити послідовний алгоритм. Також для написання програм із використання бібліотеки MPI програмістові потрібно мати знання з розробки послідовних програм мовою C або C++. Такий стиль програмування також не дуже сприяє паралельному мисленню.

До найбільш істотних нововведень у стандарті C++ 11 слід віднести підтримку

багатопотікових програм. Уперше комітет по стандартах офіційно визнав існування багатопотікових додатків, написаних на C++, умістив у бібліотеку компоненти для розробки [12, с. 23]. Це дозволить писати на C++ багатопотікові програми з гарантованою поведінкою, не покладаючись на залежні від платформи розширення. В основу нової бібліотеки багатопотоковості для C++ покладено досвід, накопичений за час використання бібліотек OpenMP та MPI. Так, моделлю нової бібліотеки стала бібліотека Boost Thread Library, з якої запозичені імена і структура багатьох класів [12, с. 36]. Таким чином, для розробки програм на мові C++ для паралельних обчислень більше не треба підключення додаткових зовнішніх бібліотек. Наявність підтримки паралельного програмування мовою C++ на рівні внутрішніх стандартів дозволить ввести типові завдання паралельного програмування у курс вивчення мови C++ ще на початковій стадії та відразу з розробки паралельного алгоритму. Це буде сприяти розвитку паралельного мислення ще з першого курсу.

Розглянемо можливості розробки паралельних програм засобами мови програмування Java. Уся бібліотека класів Java спроектована так, щоб забезпечити підтримку багатопотоковості. Перевага багатопотоковості полягає у тому, що не використовується механізм циклічного опитування подій. Один потік може бути призупинений без зупинки інших [15, с. 894]. Для розробки паралельних програм у мові Java, починаючи з Java8, було додано та перероблено багато класів. У отриманий результаті набір класів та інтерфейсів дозволяє вирішити типові обчислювальні завдання паралельного програмування. Розробка паралельних програм мовою Java має починатися відразу з об'яви паралельних класів та інтерфейсів що потребує розробки відразу паралельного алгоритму. Такий підхід дуже добре сприяє розвитку паралельного мислення. Слід зазначити, що розробка паралельних програм мовою Java потребує від програміста широких знань як самої мови Java, так і основних парадигм об'єктно-орієнтованого програмування. Таким чином, розробку паралельних програм на мові Java можна розглядати як продовження розвитку паралельного мислення програмістові. Тому цю мову можна використовувати у викладанні дисципліни паралельного програмування у вищій школі.

Вищезазначене для мови програмування Java можна сказати і про мову програмування C#. C# – це об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи NET [14, с. 12]. Розроблена Андерсом Гейл-



Таблиця 1

Зведене порівняння засобів розробки паралельних програм

Засіб розробки	Мова програмування / зовнішня бібліотека	Підхід до рішення типових задач паралельного програмування
OpenMP (C++)	Зовнішня бібліотека	Послідовно-паралельний
MPI (C++)	Зовнішня бібліотека	Послідовно-паралельний
C++ 11	Мова програмування	Паралельний
Java	Мова програмування	Паралельно-об'єктно-орієнтований
C#	Мова програмування	Паралельно-об'єктно-орієнтований

сбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, переваженість операторів, указівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Мова C# також має набір класів та інтерфейсів для розробки паралельних програм. Вищезазначене про Java можна зазначити і про C#, цю мову можна використовувати як засіб паралельного програмування. Мову C# також можна використовувати у викладанні дисципліни паралельного програмування у вищій школі. Зведемо все вищезазначене до таблиці 1.

Таким чином, як засіб розробки програмного забезпечення паралельних обчислень більш доцільно використовувати мови програмування Java та C#.

Висновки з проведеного дослідження. У статті були проаналізовані засоби розробки програмного забезпечення, які можна використовувати у викладанні дисципліни паралельного програмування у вищій школі. Під час аналізу акцент робився на таких засобах розробки, під час використання яких у студентів буде формуватися паралельний стиль мислення. Він допоможе у розробці паралельних програм із використанням саме паралельних алгоритмів та уникнути помилок застосування послідовних алгоритмів у паралельних програмах. У результаті аналізу засобів розробки програмного забезпечення можна рекомендувати такі кроки для оптимізації викладання дисципліни паралельне програмування:

– під час вивчення програмування на першому курсі розглядати паралельні бібліотеки C++;

– для лабораторних та практичних робіт за умов реалізації паралельних алгоритмів у рамках дисципліни паралельне програмування використовувати такі мови програмування, як Java та C#;

– поступово відходити від використання бібліотек OpenMP та MPI для написання паралельних програм.

ЛІТЕРАТУРА:

1. Баденко В.Л. Высокопроизводительные вычисления : учебное пособие. Санкт-Петербург : Изд-во Политехнического ун-та, 2010. 180 с.
2. Воеводин В.В. Параллельные вычисления. Санкт-Петербург : БХВ-Петербург, 2002. 608 с.
3. Гергель В.П. Теория и практика параллельных вычислений. 2-е изд. Москва : Интуит, 2016. 500 с.
4. Эккель Б. Философия Java. Санкт-Петербург : Питер, 2016. 1168 с.
5. Карпов А.Н. Введение в проблематику разработки параллельных программ. URL: <https://www.viva64.com/ru/a/0016/> (дата обращения: 12.12.2018).
6. Лупин С.А., Посыпкин М.А. Технологии параллельного программирования. Москва : ИД «ФОРУМ» ; ИНФРА-М, 2011. 208 с.
7. Немногин С.А. Средства программирования для многопроцессорных вычислительных систем. Санкт-Петербург, 2007. 88 с.
8. Осадчий В.В. Анализ проблемы профессиональной подготовки программиста и пути ее решения. *Образовательные технологии и общество*. 2014. № 3. Т. 17. С. 362–377.
9. Парадигмы параллельного программирования. URL: http://staff.mmcs.sfedu.ru/~dubrov/files/sl_parallel_05_paradigm.pdf (дата обращения: 26.11.2018).
10. Скопин И.Н. Раннее обучение параллельному программированию. *Вестник Московского городского педагогического университета*. 2011. № 2. С. 46–22.
11. Сокольская М.А., Степанова Т.А. Уточнение понятия «параллельный стиль мышления» на основе информационно-деятельностного подхода. URL: <https://cyberleninka.ru/article/n/utochnenie-ponyatiya-parallelnyy-stil-myshleniya-na-osnove-informatsionno-deyatelnostnogo-podhoda> (дата обращения: 10.11.2018).
12. Уильямс Э. Параллельное программирование на C++ в действии / пер. А.А. Слинкин. Москва : ДМК Пресс, 2016. 671 с.
13. Федотов И.Е. Модели параллельного программирования. Москва : Солон-Пресс, 2012. 384 с.
14. Фленов М.Е. Библия C#. Санкт-Петербург : БХВ-Петербург, 2009. 560 с.
15. Хьюз К., Хьюз Т. Параллельное и распределенное программирование с использованием C++ / пер. с англ. Москва : ИД «Вильямс», 2004. 672 с.
16. Шеметова А.Д. Методические приёмы обучения параллельному программированию. *Прикладная информатика*. 2016. № 6. С. 43–48.
17. Яровий А.А. Методи та засоби організації високопродуктивних паралельно-ієрархічних обчислювальних систем із рекурсивною архітектурою : дис. ... докт. техн. наук : 05.13.05. Вінниця, 2013. 439 с.