

Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра комп'ютерних систем та технологій

Силабус курсу

РОЗПОДІЛЕНІ СИСТЕМИ ТА ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ

Обсяг	Загальна кількість: кредитів - 5; годин - 150
Семестр, рік навчання	2 семестр, 4 рік
Дні, час, місце	За розкладом
Викладач (-і)	Мартиневич Лариса Ярославівна
Контактний телефон	+380509360250
E-mail	larysa.yaroslavna@onu.edu.ua
Робоче місце	Кафедра комп'ютерних систем та технологій ОНУ імені І.І. Мечникова, вул. Всеволода Змієнка, 2
Консультації	Viber, Zoom, Google Class

КОМУНІКАЦІЯ

Комунікація зі студентами буде здійснюватися в аудиторії, та також з використанням додатків ZOOM, Google Клас, Google Meet, месенджери.

АНОТАЦІЯ КУРСУ

Предметом вивчення курсу є основні парадигми паралельного та розподіленого програмування та технології створення паралельних та розподілених паралельних додатків (MPI, OpenMP) для сучасних комп'ютерних архітектур.

ПРЕРЕКВІЗИТИ КУРСУ

Матеріал курсу *ґрунтується* на раніше отриманих студентами знаннях, практичних вміннях та навичках з тем та напрямів щодо програмування, алгоритмів, структур даних, операційних систем та системного програмування.

МЕТОЮ КУРСУ є вивчення основних принципів побудови паралельних та розподілених програмних додатків для різноманітних комп'ютерних систем, а також придбання практичних навичок щодо створення, тестування та експлуатації паралельного програмного продукту з використанням сучасних пакетів та стандартів паралельного програмування.

ЗМІСТ КУРСУ

Розглядаються:

- Два основних підходи до досягнення паралелізму: паралельне програмування, розподілене програмування. Найпростіша модель паралельного програмування PRAM. Найпростіша модель розподіленого програмування. Процес проектування паралельних і розподілених програм: декомпозиція, зв'язок, синхронізація. Базові рівні програмного паралелізму. Середовища для паралельного і розподіленого програмування.
- Класифікація паралельних обчислювальних систем: SMP-системи, кластери, MPP-системи. Системи із загальною й розподіленою пам'яттю. Архітектура NUMA та ccNUMA. Організація когерентності багаторівневої ієрархічної пам'яті в SMP-системах.
- Головні парадигми паралельного програмування. Ітеративний паралелізм (множення матриць). Рекурсивний паралелізм (адаптивна квадратура). «Виробники та споживачі» (конвеєри). Проблеми паралельного і розподіленого програмування: «гонка» даних, нескінченна відстрочка, взаємоблоківка, труднощі організації зв'язку.
- Паралельне програмування з використанням потоків стандарту POSIX. Створення потоку. Атрибути потоку. Очікувані й від'єднанні потоки. Скасування потоку: асинхронне скасування, синхронне скасування, потоки, які не можна скасувати. Поточкові дані. Оброблювач очищення. Очищення поточкових даних у C++. Поточкові семафори. Мютекси. Умовні змінні POSIX.
- Технологія паралельного програмування OpenMP. Модель OpenMP-додатка. Директива паралельної обробки `parallel`. Директива розподілення роботи `for`. Директиви розподілення роботи `sections` та `section`. Директиви `single` та `master`. Директиви `tasks` та `taskwait`. Директиви синхронізації `barrier`, `ordered`, `critical`, `atomic`. Спільні та приватні змінні. Функції середовища виконання. Функції блокування та синхронізації. Змінні оточення. Алгоритми планування паралельного виконання циклів (`static`, `dynamic`, `guided`, `runtime scheduling`).
- Технологія розподіленого програмування MPI. Призначення MPI. Модель MPI-додатка. Комунікатори. Функції ініціалізації й завершення роботи. Етапи передачі повідомлень між паралельними процесами MPI. Типи даних MPI. Функції передачі повідомлень між процесами типу «один-одному». Колективні комунікації. Розподілені операції в MPI. Створення нових типів даних MPI. Створення розподілених операцій. Топології процесів. Створення декартової топології процесів в MPI-додатках. Приклад використання декартової топології процесів.
- Моделювання та аналіз паралельних обчислень. Модель обчислень у вигляді графа "операції операнди". Граф інформаційних залежностей послідовного алгоритму. Опис схеми паралельного виконання алгоритму. Характеристики часу виконання паралельного алгоритму. Теореми про оцінки часу виконання паралельного алгоритму. Показники ефективності паралельного алгоритму. Каскадна схема сумування. Модифікована каскадна схема сумування. Оцінка максимально допустимого паралелізму (закон Амдаля, ефект Амдаля, закон Густавсона – Барсиса). Аналіз масштабованості паралельних обчислень.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

Процес вивчення дисципліни спрямований на формування елементів наступних компетентностей:

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів

інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

Спеціальні:

СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

У результаті вивчення курсу студент повинен

знати: основні парадигми паралельного та розподіленого програмування; основні прийоми написання паралельних програм з використанням потоків стандарту POSIX; технології OpenMP, MPI написання паралельних програм.

вміти: створювати паралельні алгоритми для розв'язування різноманітних задач; аналізувати ефективність паралельного алгоритму; реалізувати паралельний алгоритм для комп'ютерних систем з різними паралельними архітектурами;

Компетентності, які отримує студент у результаті вивчення курсу: – Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

РЕЗУЛЬТАТИ НАВЧАННЯ: по завершенню курсу студент матиме навички – здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії. – виконувати експериментальні дослідження за професійною тематикою. – оцінювати отримані результати та аргументовано захищати прийняті рішення.

ОПИС КУРСУ

Форми і методи навчання

Курс буде викладений у формі лекцій (36 год.) та лабораторних занять (24 год.), організації самостійної роботи студентів (90 год.).

Основна підготовка студентів здійснюється на лекційних та лабораторних заняттях, але у значній мірі покладається на самостійне вивчення матеріалу студентами денної форми навчання протягом семестру.

Під час викладання дисципліни використовуються словесні та наочні методи навчання:

лекції, бесіда, пояснення; практичні методи навчання - виконання лабораторних робіт, розв'язання розрахункових завдань, робота з літературними джерелами.

ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ЗМІСТОВИЙ МОДУЛЬ 1. ОСНОВИ ПАРАЛЕЛЬНИХ ТА РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

Тема 1. Вступ до предмету. Паралелізм. Два основних підходи до досягнення паралелізму: паралельне програмування, розподілене програмування. Найпростіша модель паралельного програмування PRAM. Найпростіша модель розподіленого програмування. Процес проектування паралельних і розподілених програм: декомпозиція, зв'язок, синхронізація. Базові рівні програмного паралелізму. Середовища для паралельного і розподіленого програмування.

Тема 2. Класифікація паралельних обчисл. систем: SMP-системи, кластери, MPP-системи. Системи із загальною й розподіленою пам'яттю. Архітектура NUMA та ccNUMA. Організація когерентності багаторівневої ієрархічної пам'яті в SMP-системах.

Тема 3. Головні парадигми паралельного програмування. Ітеративний паралелізм (множення матриць). Рекурсивний паралелізм (адаптивна квадратура). «Виробники та споживачі» (конвейєри). Проблеми паралельного і розподіленого програмування: «гонка» даних, нескінченна відстрочка, взаємоблокування, труднощі організації зв'язку. Типові задачі синхронізації паралельних процесів: задача взаємного виключення, «виробник-споживач», «читачі-письменники».

Тема 4. Паралельне програмування з використанням потоків стандарту POSIX. Створення потоку. Атрибути потоку. Очікувані й від'єднанні потоки. Скасування потоку: асинхронне скасування, синхронне скасування, потоки, які не можна скасувати. Потоківі дані. Оброблювач очищення. Очищення поточкових даних у C++. Потоківі семафори. Мютекси. Умовні змінні POSIX. Приклад використання мютексів і поточкових семафорів.

ЗМІСТОВИЙ МОДУЛЬ 2. ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ ТА РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

Тема 5. Технологія паралельного програмування OpenMP. Призначення OpenMP. Модель OpenMP-додатка. Директива паралельної обробки parallel. Директива розподілення роботи for. Директиви розподілення роботи sections та section. Директиви single та master. Директиви tasks та taskwait. Директиви синхронізації barrier, ordered, critical, atomic. Спільні та приватні змінні. Функції середовища виконання. Функції блокування та синхронізації. Змінні оточення. Алгоритми планування паралельного виконання циклів (static, dynamic, guided, runtime scheduling).

Тема 6. Технологія розподіленого програмування MPI. Призначення MPI. Модель MPI-додатка. Комунікатори. Функції ініціалізації й завершення роботи. Етапи передачі повідомлень між паралельними процесами MPI. Типи даних MPI. Функції передачі повідомлень між процесами типу «один-одному». Колективні комунікації. Розподілені операції в MPI. Створення нових типів даних MPI. Створення розподілених операцій. Топології процесів. Створення декартової топології процесів в MPI-додатках. Приклад використання декартової топології процесів.

Тема 7. Моделювання та аналіз паралельних обчислень. Модель обчислень у вигляді графа "операції-операнди". Граф інформаційних залежностей послідовного алгоритму. Опис схеми паралельного виконання алгоритму. Характеристики часу виконання паралельного алгоритму. Теореми про оцінки часу виконання паралельного алгоритму. Показники ефективності паралельного алгоритму. Каскадна схема сумування. Модифікована каскадна схема сумування. Оцінка максимально допустимого паралелізму (закон Амдаля, ефект Амдаля, закон Густавсона – Барсиса). Аналіз масштабованості паралельних обчислень.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

Основна

1. Конспект лекцій; комплекс навчально-методичного забезпечення дисципліни; нормативні документи; презентаційні матеріали.
2. Литвинов О.А., Мартинович Л.Я., Пономарьов І.В. "Технології паралельної обробки інформації: OpenMP та MPI". / Литвинов О.А., Мартинович Л.Я., Пономарьов І.В. – Дніпро: ДНУ, 2017. – 124 с.
3. Вербіцький В. В., Максимов А. Л. Паралельне програмування з використанням технології OpenMP: метод. вказівки. Одеса: Одес. нац. ун-т ім. І.І.Мечникова, 2022. – 47 с.
4. Weinzierl T. Principles of Parallel Scientific Computing: A First Guide to Numerical Concepts and Programming Methods: Undergraduate Topics in Computer Science. / Springer Nature Switzerland AG, 2022. 314 p.
5. Pacheco P. An Introduction to Parallel Programming / Elsevier, 2011. 392 p.
6. Rauber Th., Runger G. Parallel Programming For Multicore and Cluster Systems / Springer-Verlag Berlin Heidelberg, 2010. 463 p.
7. Schmidt B., González-Domínguez J., Hundt Ch., Schlarb M. Parallel Programming. Concepts and Practice / Publisher: Katey Birtcher, 2018. 405 p.
8. Trobec R., Slivnik B., Bulić P., Robič B. Introduction to Parallel Computing. From Algorithms to Programming on State-of-the-Art Platforms / Springer Nature Switzerland AG, 2018. 259 p.
9. Robey R., Zamora Y. Parallel and High Performance Computing / Manning, Shelter Island, 2021. 705 p.

Додаткова література

1. Chapman B., Jost G., Ruud van der Pas. Using OpenMP: portable shared memory parallel programming / The MIT Press, 2007. 378 p.
2. Förster M. Algorithmic Differentiation of Pragma-Defined Parallel Regions: Differentiating Computer Programs Containing OpenMP / Springer Fachmedien Wiesbaden, 2014. 411 p.
3. OpenMP: Application Program Interface Version 5.0. (<https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5.0.pdf>)
4. Ruud van der Pas, Stotzer E., Terboven Chr. Using OpenMP-The Next Step. Affinity, Accelerators, Tasking, and SIMD / The MIT Press, 2017. 381 p.
5. Mattson T. G., He Y. H., Koniges A. E. The OpenMP common core : making OpenMP simple again / Cambridge, Massachusetts : The MIT Press, 2019. 277 p.
6. Foster I. Desinging and building parallel program / Addison-Westly, 1995.
7. Shir M., Otto St. MPI: The complete reference / MIT Press, 1996.

ОЦІНЮВАННЯ

Методи поточного контролю: виконання завдань лабораторних робіт, контрольні роботи.

Форми і методи підсумкового контролю: іспит

Поточний та періодичний контроль							Підсумковий контроль (іспит)	Сума балів
Змістовий модуль 1				Змістовий модуль 2				
T1	T2	T3	T4	T5	T6	T7	35	100
5	5	5	5	5	5	5		
Контрольна робота за змістовим модулем 1 - 15				Контрольна робота за змістовим модулем 2 - 15				

Самостійна робота студентів.

Самостійна робота представлена у формі підготовки до лекцій та лабораторних занять та індивідуальне самостійне завдання. Підготовка до лекцій перевіряється з використанням тестових завдань.

ПОЛІТИКА КУРСУ

Викладання дисципліни відбувається відповідно до Положення про організацію освітнього процесу в Одеському національному університеті імені І.І. Мечникова.

Політика щодо дедлайнів та перескладання: роботи мають здаватися вчасно. Деякі види робіт можуть здаватися із порушенням термінів з поважних причин. Перескладання модулів відбувається за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності: дотримання академічної доброчесності здобувачами освіти передбачає дії відповідно до Положення про запобігання та виявлення академічного плагіату у освітній та науково-дослідній роботі учасників освітнього процесу та науковців Одеського національного університету імені І.І. Мечникова.

Політика щодо відвідування: відвідування лабораторних занять є обов'язковим та без запізень. За об'єктивних причин (наприклад, хвороба, міжнародне стажування, карантин тощо) навчання може відбуватись в онлайн формі (змішана форма навчання) на платформі із використанням інструментів Google WorkSpace та Classroom.

Мобільні пристрої дозволяється використовувати лише під час онлайн тестування та підготовки практичних завдань у процесі заняття.

Поведінка в аудиторії повинна відповідати загальним вимогам ділової та наукової етики: неприпустимо під час відповідей на занятті користуватися мобільними телефонами, порушувати дисципліну розмовами або в будь-який інший спосіб.

ВІДПОВІДНІСТЬ ЦІЛЯМ СТАЛОГО РОЗВИТКУ ДО 2030 РОКУ

ЦСР 4 «Якісна освіта Забезпечення всеохопної і справедливої якісної освіти та заохочення можливості навчання впродовж усього життя для всіх» (п. 4.3 та 4.4). Вивчення методів аналізу і синтезу науково-технічної, природничо-наукової та загальнонаукової інформації, зокрема тем, пов'язаних з різними професіями в ІТ (Тема 1, 2), сприяє підвищенню якості освіти та розвитку критичного мислення.

ЦСР 8 Гідна праця та економічне зростання Сприяння поступальному, всеохопному і сталому економічному зростанню, повній і продуктивній зайнятості та гідній праці для всіх (п. 8.2 та 8.3). Вдосконалення навичок роботи в команді, тайм-менеджменту та загалом SOFT SKILLS сприяє підвищенню продуктивності та якості майбутньої професійної діяльності, в тому числі інноваційної (Тема 3).

ЦСР 9 Інновації та інфраструктура Створення стійкої інфраструктури, сприяння всеохопній і сталій індустріалізації та інноваціям (п. 9.1 та 9.с). Розуміння здобувачами місця і ролі ІТ в житті людей дозволить прискорити розвиток інформаційно-комунікаційних технологій

ЦСР 17 «Партнерство заради сталого розвитку» (п. 17.8). Обмін знаннями та даними через інформаційні технології (Тема 3) сприяє глобальній співпраці та покращенню доступу до інформації для досягнення цілей сталого розвитку.