



ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ. КУРСОВА РОБОТА

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>1 кредит (30 год)</i>
Семестровий контроль/ контрольні заходи	<i>Залік (захист курсової роботи)</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Керівники: д.т.н., професор, Стеценко Інна Вячеславівна, inna.stetsenko-fiot@ill.kpi.ua ст.викл. Дифучин Антон Юрійович, difuchin@gmail.com ст.викл. Дифучина Олександра Юріївна sashadif@gmail.com , ас. Нестеренко Костянтин Петрович k.nesterenko@kpi.ua
Розміщення курсу	Moodle: <i>Технології паралельних обчислень</i> (uc80yt) https://do.ipu.kpi.ua/course/view.php?id=2504

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни спрямовано на оволодіння практичними навичками з розробки ефективних алгоритмів паралельних обчислень з використанням технологій багатопотокових (Java Multithreading, CUDA C++, OpenMPI, MPJ Express, Java Sockets, Java RPC). Увага приділяється не тільки розробці алгоритму з використанням вказаних технологій, але й експериментальному дослідженню ефективності алгоритму. Дисципліна формує навички розробки паралельних алгоритмів, розвиває навички реалізації паралельних обчислень в одно- та багатопроцесорних обчислювальних системах, розвиває навички реалізації розподілених обчислень, а також формує базові дослідницькі навички з оцінювання швидкодії обчислень.

Предмет навчальної дисципліни – методи та технології розробки паралельних програм для багатоядерних та багатопроцесорних комп'ютерних систем.

Метою дисципліни є розуміння студентами принципів та способів розробки паралельних алгоритмів з високою швидкістю, вивчення сучасних технологій для реалізації паралельних та розподілених обчислень. За результатами вивчення дисципліни студент повинен набути **досвід** з розробки паралельних алгоритмів та їх реалізації з використанням технологій паралельних та розподілених обчислень Java Multithreading, Java MPI, Java RMI.

Програмні результати навчання студента.

Студент після засвоєння навчальної дисципліни повинен **знати**:

- базові принципи паралельних обчислень,
- архітектуру багатоядерних та багатопроцесорних обчислювальних систем;
- методи розробки паралельних алгоритмів для прикладних задач;
- технології багатопотокового програмування;
- технологію Message Passing Interface (MPI);
- методи оцінювання ефективності паралельних обчислень.

Студент повинен **вміти**:

- розробляти багатопотокові програми, використовуючи низько- та високо-рівневі засоби паралельних обчислень мови Java;
- виконувати дослідження ефективності розроблених паралельних програм експериментальними методами;
- програмувати розподілені обчислення
- розробляти алгоритми для виконання обчислень в розподілених системах з використанням технології MPI.

Згідно з вимогами освітньої програми вивчення дисципліни спрямоване на оволодіння студентом таких **компетентностей**:

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ФК 17. Здатність розробляти інформаційні системи з використанням паралельних обчислень.

Програмні результати вивчення дисципліни забезпечують такі програмні результати освітньої програми:

ПРН 12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН 31. Знати технології паралельних обчислень, віртуалізації серверних систем.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Вивчення дисципліни спирається на знання, отримані студентами при вивченні дисциплін «Основи програмування», «Алгоритми та структури даних», «Проектування алгоритмів», «Компоненти програмної інженерії. Частина 4. Якість та тестування програмного забезпечення», «Програмування веб-застосувань». Знання та навички, набуті студентом при вивченні дисципліни, використовуються в розробці дипломних проєктів здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення інформаційних систем».

3. Зміст навчальної дисципліни

Передбачено виконання курсової роботи за темою «Алгоритм _(назва вибраного алгоритму)_ та його паралельна реалізація _(назва технології чи програмного забезпечення)_». Виконання курсової роботи передбачає розробку паралельного алгоритму у відповідності до обраного варіанту та експериментальне дослідження ефективності розробленого алгоритму. Алгоритм, який розпаралелюється, та засоби його реалізації вибираються разом з викладачем у відповідності до обраного студентом рівня складності. Варіанти, в яких вказані алгоритм та програмне забезпечення, поділені за рівнем складності завдання на групи: до 100 балів, до 75 балів, до 65 балів.

Завдання до курсової роботи:

- 1) Виконати огляд існуючих реалізацій алгоритму, послідовних та паралельних, з відповідними посиланнями на джерела інформації (статті, книги, електронні ресурси). Зробити висновок про актуальність дослідження.
- 2) Виконати розробку послідовного алгоритму у відповідності до варіанту завдання та обраного програмного забезпечення для реалізації. Дослідити швидкість алгоритму при

зростанні складності обчислень та зробити висновки про необхідність паралельної реалізації алгоритму.

- 3) Виконати розробку паралельного алгоритму у відповідності до варіанту завдання та обраного програмного забезпечення для реалізації. Забезпечити зручне введення даних для початку обчислень.
- 4) Виконати тестування алгоритму, що доводить коректність результатів обчислень.
- 5) Виконати дослідження швидкодії алгоритму при зростанні кількості даних для обчислень.
- 6) Виконати експериментальне дослідження прискорення розробленого алгоритму при зростанні кількості даних для обчислень. Реалізація алгоритму вважається успішною, якщо прискорення більше 1.
- 7) Дослідити вплив параметрів паралельного алгоритму на отримуване прискорення.
- 8) Зробити висновки про переваги паралельної реалізації обчислень для алгоритму, що розглядається у курсовій роботі, та програмних засобів, які використовувались.

У розділах пояснювальної записки до курсової роботи мають бути описані усі етапи розробки паралельного алгоритму та його дослідження, а також відомості про обране програмне забезпечення для виконання завдання.

Курсова робота має включати такі розділи:

Анотація (не менше 650 символів)

Ключові слова (не менше 3 словосполучень)

Вступ (про актуальність паралельних обчислень та завдання, яке виконується в рамках курсової роботи).

Розділ 1 Опис алгоритму та його відомих паралельних реалізацій (якщо такі є).

Розділ 2 Розробка послідовного алгоритму та аналіз його швидкодії

Розділ 3 Вибір програмного забезпечення для розробки паралельних обчислень та його короткий опис.

Розділ 4 Розробка паралельних обчислень алгоритму з використанням обраного програмного забезпечення: проєктування, реалізація, тестування.

Розділ 5 Дослідження ефективності паралельних обчислень алгоритму (порівняльний аналіз швидкості обчислень).

Висновки (про переваги використання паралельних обчислень та обраного програмного забезпечення).

Список використаних джерел.

Додатки (схема взаємодії паралельних процесів, лістинг коду, скріншоти).

Курсова робота має бути виконана та захищена до кінця семестру. Захист усіх робіт з відбувається за умови допуску (30 з 50 балів за результатами перевірки роботи керівником) з презентацією цієї роботи перед комісією з проведення семестрового контролю.

4. Навчальні матеріали та ресурси

Базова література

1. The Java Tutorials Lesson:Concurrency [Електронний ресурс] – Режим доступу: <https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>
2. Стіренко С. Г. Засоби паралельного програмування / С. Г. Стіренко. Д. В. Грибенко. О. І. Зіненко. А. В. Михайленко – Київ. 2012. – 183 с. [Електронний ресурс] – Режим доступу: <http://hpcc.kpi.ua/hpc-book/>
3. Lea D. Concurrent programming in Java: design principles and patterns / D. Lea – Addison-Wesley Professional. 2000. – 411p.
4. Foster I. Designing and Building Parallel Programs [Електронний ресурс] – Режим доступу: <http://www.mcs.anl.gov/~itf/dbpp/text/book.html>

Допоміжна література

1. Аксак Н.Г. Паралельні та розподілені обчислення: підруч./ Н.Г.Аксак. О.Г. Руденко. А.М.Гуржій. – Х.:Компанія СМІТ. 2009. – 480с.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді на гугл-диску викладача, до якого надано доступ групі студентів та асистентам, які ведуть заняття комп'ютерного практикуму, а також, в середовищі Moodle платформи дистанційного навчання «Сікорський» (скорочена назва дисципліни uc80yt). Контент платформи доступний із будь-якого місця в мережі Інтернет зареєстрованому на курс студенту. Навчальний курс в середовищі Moodle складається з розділів, кожний з яких містить презентації лекцій, завдання та теоретичний матеріал для виконання комп'ютерного практикуму, URL-посилання на документацію з програмного забезпечення та навчальний матеріал для самостійної роботи відповідно до теми, яка вивчається.

Виконання курсової роботи поділено на частини, які можуть здаватись студентом по мірі виконання курсової роботи: розробка концептуальної моделі, розробка формалізованої моделі, реалізація імітаційної моделі, проведення експериментального дослідження на моделі. Викладач має можливість вже на перших кроках виконання завдання виявити та вказати на помилки в розробці.

В умовах дистанційного навчання усі види занять, у тому числі контрольні заходи, проводяться з використанням Zoom та Google Meet.

6. Самостійна робота студента

Матеріали для самостійного вивчення дисципліни розміщені викладачем в електронному вигляді в середовищі Moodle платформи дистанційного навчання «Сікорський». Контент платформи доступний із будь-якого місця в мережі Інтернет. До самостійної роботи студента відноситься, розробка програми у відповідності до завдання курсової роботи, робота з документацією та тьюторіалом програмного забезпечення, оформлення пояснювальної записки до курсової роботи, а також опрацювання додаткового теоретичного матеріалу та додаткової літератури. На самостійну роботу студент має витрати кількість годин, що дорівнює 1 кредиту.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Студент має виконувати завдання на курсову роботу протягом семестру від початку отримання завдання у перші три тижні семестру. Затвердження теми, керівника та дати захисту курсової роботи відбувається у перші три тижні семестру. Після затвердження теми курсової роботи та її керівника вони не можуть бути змінені.

Завдання на курсову роботу ранжовані за рівнем складності: А – до 100 балів, В – до 94 балів, D – до 74 балів. Отримана за результатом виконання роботи оцінка не може перевищувати оцінку, що відповідає його складності.

Планування днів захисту відбувається таким чином, щоб розподілити захисти по 8-10 робіт у дні, коли така комісія може бути зібрана. Роботи з низькою складністю плануються на захисти у більш ранні дати захистів, а роботи з високою складністю – на більш пізні (оскільки потребують більше зусиль на їх виконання). За складання графіку захистів в комісіях відповідає лектор.

Дата подання курсової роботи на перевірку має бути не пізніше ніж за 10 днів до дня захисту. Вчасно подані роботи мають найвищий пріоритет при перевірці керівником. Керівник зобов'язаний перевірити вчасно подану курсову роботу до дня захисту. Кількість перевірок курсової роботи обмежена виключно термінами її підготовки до визначеного дня захисту та загалом термінами виставлення оцінки у залікову відомість. За згодою студента, керівника та членів комісії день захисту може бути змінений як на більш ранній (у разі готовності роботи до

захисту), так і на більш пізній. Рекомендовано подання розроблених розділів курсової роботи по мірі їх готовності, що сприяє вчасному виявленню помилок в розробці алгоритму.

Подання курсової роботи здійснюється в дистанційному курсі «Технології паралельних обчислень», де фіксуються дата подання та результати перевірки керівником.

Оцінювання розділів курсової роботи відбувається тільки за умови відповідності їх оформлення встановленим вимогам.

Для допуску до захисту курсова робота має набрати **не менше** ніж 30 балів з 50 балів за результатами перевірки у керівника курсової роботи. Робота, яка не набрала достатню кількість балів, має бути доопрацьована студентом. Якщо доопрацьована робота буде подана до запланованого дня захисту і отримає достатню кількість балів для захисту, то вона отримує допуск до захисту. Інакше, за умови наявних місць на захистах в комісії, призначається пізніша дата захисту. Якщо усі дні запланованих комісій вичерпані, то захист переноситься на додаткову сесію.

У разі виявлення порушення академічної доброчесності курсова робота не допускається до захисту в основну сесію (у відомості буде зазначено «усунено»). Студент може доопрацьовувати роботу і висунути її на захист на додатковій сесії.

Під час модульного контролю виставляється накопичений на дату проведення контролю бал.

Публічний захист курсової роботи проводиться в комісії з проведення семестрового контролю і складається з доповіді студента (5 хвилин) та відповідей його на запитання членів комісії (5-10 хвилин). У разі виявлення грубої помилки в розробці алгоритму або у разі виявлення порушення академічної доброчесності комісія може відхилити захист курсової роботи. Для роботи, захист якої відхилено, приймається рішення щодо її розгляду у більш пізні дати захистів або на додатковій сесії у разі доопрацювання у відповідності до наданих зауважень та рекомендацій.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: перевірка дотримання графіку виконання курсової роботи та якості виконаних частин завдання

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.

Семестровий контроль: залік

Умови допуску до семестрового контролю: 35 балів з 50 за результатами поточного контролю.

Оцінка студента з кредитного модуля складається з балів, які він отримує за результатами:

- 1) розробки програмного забезпечення у відповідності до індивідуального завдання;
- 2) оформлення пояснювальної записки до курсової роботи;
- 3) прилюдного захисту з презентацією результатів курсової роботи.

Стартова складова оцінки містить результат перевірки поданої курсової роботи керівником. Складова захисту курсової роботи – результат оцінювання захисту роботи в комісії з проведення семестрового контролю. До відомості семестрового контролю викладач вносить оцінку, що складається з суми стартових балів та балів за захист курсової роботи. Набрана кількість балів **не може перевищувати максимальну кількість балів завдання обраної складності**.

Розподіл балів між частинами завдання:

№ з/п	Назва розділу курсової роботи	Всього	Кількість балів по видах діяльності		
			Розробка програмного забезпечення	Оформлення пояснювальної записки	Захист
Стартова складова (50 балів)					
1	Вступ	0,25	-	0,25	

№ з/п	Назва розділу курсової роботи	Всього	Кількість балів по видах діяльності		
			Розробка програмного забезпечення	Оформлення пояснювальної записки	Захист
2	Розділ 1. Опис послідовного алгоритму та його відомих паралельних реалізацій	2	-	2	-
3	Розділ 2. Розробка послідовного алгоритму та аналіз його швидкодії	5	4	1	-
4	Розділ 3. Вибір програмного забезпечення для розробки паралельних обчислень та його короткий опис	2	-	2	-
5	Розділ 4. Розробка паралельного обчислення алгоритму з використанням обраного програмного забезпечення: проєктування, реалізація, тестування.	25	23	2	-
6	Розділ 5. Дослідження ефективності паралельних обчислень алгоритму	15	13	2	-
7	Висновки	0,25	-	0,25	-
8	Список використаних джерел	0,25	-	0,25	-
9	Додатки	0,25	-	0,25	-
Складова захисту (50 балів)					
1	Доповідь	5	-	-	5
2	Рівень володіння матеріалом	20	-	-	20
3	Відповіді на запитання членів комісії	25	-	-	25
	Всього балів	100	40	10	50

Підсумкова оцінка переводиться до залікової оцінки згідно з таблицею:

Кількість балів / Інший результат виконання	Оцінка / Відмітка у відомості семестрового контролю
95...100	Відмінно
85...94	Дуже добре
75...84	Добре
65...74	Задовільно
60...64	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено
Порушення академічної доброчесності	Усунено
Курсова робота не була захищена	Не з'явився

9. Додаткова інформація з дисципліни (освітнього компонента)

- Перелік тем, які пропонуються для виконання:

Алгоритм (назва вибраного алгоритму) та його паралельна реалізація (назва мови, технології чи програмного забезпечення)

Варіанти вибору алгоритму:

Рівень А (до 100 балів)

Алгоритм Дейкстри

Δ-степінг алгоритм пошуку найкоротших шляхів
Алгоритм векторизації тексту на основі статистичного аналізу слів
Алгоритм контрольованого навчання нейромережі
Алгоритм неконтрольованого навчання нейромережі
Алгоритм розпізнавання об'єкта на графічному зображенні
Алгоритм аналізу колекції графічних зображень на співпадіння
Алгоритм сегментації зображення з використання нейромережі
Алгоритм аналізу колекції текстових фрагментів на співпадіння
Алгоритм перевірки тексту на плагіат
Алгоритм виявлення та вирішення колізій фізичних тіл
Мурашиний алгоритм
Бджолиний алгоритм
Алгоритм PageRank
Алгоритм Косараджу
Алгоритм пошуку компонент зв'язності
Багаторядний алгоритм самоорганізації моделей
Алгоритм імітації мережі масового обслуговування
Алгоритм імітації Петрі-об'єктного моделювання

Рівень В (до 84 балів)

Алгоритм Флойда-Воршелла
Алгоритм Беллмана-Форда
Алгоритм A*
Алгоритм D*
Алгоритм статистичного аналізу тексту
Алгоритм мінімакс
Еволюційний алгоритм пошуку оптимального значення
Алгоритм класифікації графічного зображення
Алгоритм виділення ознак графічного зображення
Алгоритм Крускала
Алгоритм Прима
Алгоритм Борувки
Алгоритм сортування масиву елементів об'єктного типу.
Алгоритм розв'язання задачі про пакування рюкзака методом динамічного програмування
Алгоритм пошуку в ширину
Алгоритм пошуку в глибину
Алгоритм пошуку оптимальних значень методом градієнтного спуску
Алгоритм розв'язання транспортної задачі методом потенціалів
Алгоритм коду Прюфера
Алгоритм відтворення життя популяції
Алгоритм обчислення послідовності Фібоначчі
Алгоритм пошуку розв'язку задачі лінійного програмування за симплекс-методом
Однорядний алгоритм самоорганізації моделей

Рівень С (до 74 балів)

Алгоритм бітонічного сортування
Алгоритм сортування комірками
Алгоритм парного-непарного сортування
Алгоритм пірамідального сортування
Алгоритм Фокса паралельного множення матриць
Алгоритм SUMMA (Scalable Universal Matrix Multiplication Algorithm) множення матриць
Алгоритм Кеннона паралельного множення матриць
Алгоритм розв'язання системи лінійних рівнянь методом Гауса

Алгоритм пошуку регресійного рівняння
Алгоритм пошуку оберненої матриці
Алгоритм передобробки графічного зображення
Алгоритм оптимального планування робіт

Варіанти вибору мов програмування: Java, C++, C#, Python, Go, Scala та інші.
Варіанти вибору програмного забезпечення: MPJExpress, OpenMPI, OpenMP, CUDA C++, PyCUDA та інші.

Робочу програму навчальної дисципліни (силабус):

Складено професор кафедри ІПІ, д.т.н., професор Стеценко Інна Вячеславівна, ст. викладач кафедри ІПІ, доктор філософії Дифучина О.Ю., ст. викладач кафедри ІПІ, доктор філософії Дифучин А.Ю.

Ухвалено кафедрою ІПІ (протокол № 16 від 23.06.2025р.)

Погоджено Методичною комісією факультету (протокол № 11 від 27.06.2025р.)