

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

«ЗАТВЕРДЖУЮ»

Заступник декана з навчальної роботи

_____ Світлана ЄФІМЕНКО

« ____ » _____ 2025 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Високопродуктивні обчислення»

для студентів

галузь знань	F “Інформаційні технології”
спеціальність	F7 “Комп'ютерна інженерія”
рівень вищої освіти	другий
освітньо-наукова програма	Комп'ютерні системи та мережі
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2025/2026
Семестр	2
Кількість кредитів ECTS	6
Мова викладання	українська
Форма заключного контролю	іспит

Викладач:

Євген СЛЮСАР, кандидат техн. наук, асистент кафедри комп'ютерної інженерії

Пролонговано: на 20__/20__ н. р. _____ (_____) « ____ » _____ 20__ р.

на 20__/20__ н. р. _____ (_____) « ____ » _____ 20__ р.

Розробник:

Євген СЛЮСАР, кандидат техн. наук, асистент кафедри комп'ютерної інженерії

«ЗАТВЕРДЖЕНО»

Завідувач кафедри комп'ютерної інженерії

_____ **Юрій БОЙКО**

Протокол № ____ від « ____ » _____ 2025 р.

Схвалено науково-методичною комісією факультету радіофізики, електроніки та комп'ютерних систем

Протокол № ____ від « ____ » _____ 2025 р.

Голова науково-методичної комісії

Сергій РАДЧЕНКО

« ____ » _____ 2025 року.

ВСТУП

1. Мета дисципліни — ознайомлення студентів із основними принципами, підходами та методами проведення обчислень для науково-технічних задач, які потребують великої кількості розрахунків – симуляції та моделювання процесів, аналіз великих обсягів даних, розрахунків та прогнозування характеристик, тощо.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

Навчальна дисципліна «Високопродуктивні обчислення» використовує знання та вміння базової професійної та практичної підготовки бакалавра галузі знань «Інформаційні технології», зокрема алгоритми та методи обчислень, принципи та протоколи комп'ютерних мереж, навички системного програмування та основи паралельних обчислень.

Попередні вимоги:

студент повинен знати: основні принципи побудови та характеристики апаратного та програмного забезпечення ЕОМ, концептуальні моделі комп'ютерних систем, стандартні протоколи комп'ютерних мереж та сервісів, принципи роботи системного програмного забезпечення на рівні випускника бакалаврату Київського національного університету імені Тараса Шевченка.

студент повинен вміти: здійснювати інсталяцію, адміністрування та обслуговування ОС та прикладного ПЗ, користуватись засобами командного рядка Linux, створювати та здійснювати збірку програмних засобів мовами C/C++ на рівні випускника бакалаврату Київського національного університету імені Тараса Шевченка.

3. Анотація навчальної дисципліни:

Дисципліна «Високопродуктивні обчислення» знайомить студентів із сучасними методами реалізації високопродуктивних обчислень. Сучасний підхід передбачає як застосування обчислювальних кластерів, так і спеціалізованих суперкомп'ютерів. Це вимагає застосування відповідного програмного інструментарію – засобів для розпаралелювання MPI та OpenMP, а також програмних інтерфейсів для використання спеціалізованих апаратних прискорювачів, таких як CUDA.

Вирішення обчислювальної задачі на множині зв'язаних процесорів потребує застосування паралельних алгоритмів, часто у поєднанні із розподіленими обчисленнями. Вирішення математичної задачі може бути здійснене за допомогою різних алгоритмів, ефективність застосування яких залежить від архітектури обчислювальної системи. Архітектурні особливості високопродуктивних обчислювальних систем також обумовлюють використання певних програмних інструментів та алгоритмів – спільна чи розподілена пам'ять, топологія комунікаційної мережі, гомогенність чи гетерогенність обчислювальних засобів, тощо.

В курсі розглядаються типові архітектури та на прикладі відомих обчислювальних задач розглядаються різні підходи до реалізації розрахунків із використанням середовища обчислювального кластера університету.

4. Завдання навчальної дисципліни (навчальні цілі):

1. Надати основні відомості з архітектури високопродуктивних обчислювальних систем та шляхів розв'язання складних обчислюваних задач із використанням цих систем, а також методів оцінки ефективності створених рішень.

2. Поглибити знання та навички студента в області паралельних та розподілених обчислень, зокрема – високопродуктивних обчислень, та закріпити на практиці навички здійснення розрахунків на обчислювальному кластері.

3. Навчити застосовувати основні відомості курсу у професійній діяльності, зокрема – застосовувати на практиці програмні засоби розпаралелювання обчислень MPI та OpenMP для розв'язання типових обчислювальних задач на високопродуктивних обчислювальних системах.

4. Навчити застосовувати отримані знання та уміння для запуску розрахунків в середовищі обчислювального кластера, що передбачає роботу із планувальником, збірку програмного проєкту під необхідне середовище та безпосередньо запуск обчислень в пакетному режимі.

Дисципліна спрямована на формування програмних компетентностей:

ЗК2. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК6. Здатність виявляти, ставити та вирішувати проблеми.

СК1. Здатність до визначення технічних характеристик, конструктивних особливостей, застосування і експлуатації програмних, програмно-технічних засобів, комп'ютерних систем та мереж різного призначення.

СК2. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування.

СК4. Здатність будувати та досліджувати моделі комп'ютерних систем та мереж.

СК5. Здатність будувати архітектуру та створювати системне і прикладне програмне забезпечення комп'ютерних систем та мереж.

5. Результати навчання за дисципліною:

Результат навчання (1, знати; 2, вміти; 3, комунікація; 4, автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1	знати:	лекційні заняття	іспит	до 30
1.1	Сфери застосування високопродуктивних обчислень в IT-галузі, науці та техніці.	лекція	іспит	4
1.2	Загальні характеристики високопродуктивних обчислювальних систем	лекція	іспит	5
1.3	Основні підходи до вирішення складних обчислювальних задач на високопродуктивних комп'ютерних системах	лекція	іспит	5
1.4	Основні види технологій розпаралелювання та прискорення обчислень в високопродуктивних комп'ютерних системах	лекція	іспит	4
1.5	Загальна специфіка застосування технології OpenMP для розпаралелювання обчислень	лекція	іспит	4
1.6	Загальна специфіка застосування технології MPI для розпаралелювання обчислень	лекція	іспит	4
1.7	Загальна специфіка застосування графічних прискорювачів для проведення обчислень	лекція	іспит	4
2	вміти:	лекційні демонстрації та лабораторні роботи	захист звітів лабораторних робіт	до 60
2.1	Створювати опис пакетного завдання під задану систему керування ресурсами обчислювального кластера	лекційні демонстрації та лабораторні роботи	звіти з лабораторних робіт	12
2.2	Підключати та встановлювати необхідні програмні бібліотеки та модулі в середовище збірки та застосовувати його для компіляції власних програмних компонент	лекційні демонстрації та лабораторні роботи	звіти з лабораторних робіт	12
2.3	Реалізовувати заданий алгоритм розрахунків в програмному коді та здійснювати відлагоджування	лабораторні роботи	звіти з лабораторних робіт	12
2.4	Застосовувати анотації OpenMP в коді програми для прискорення розрахунків	лабораторні роботи	звіти з лабораторних робіт	12
2.5	Використовувати колективні операції MPI для прискорення розрахунків	лабораторні роботи	звіти з лабораторних робіт	12
3	комунікація:	лекційні заняття та лабораторні роботи	захист звітів лабораторних робіт	до 5
3.1	Здатність будувати грамотну науково-технічну комунікацію в усній та письмовій формах, підбирати правильну термінологію	лекція	звіти з лабораторних робіт	3
3.2	Здатність послідовно викладати етапи та результати розв'язання розрахункової задачі високопродуктивних обчислень	лекція	звіти з лабораторних робіт	2
4	автономність та відповідальність:	лекційні демонстрації та лабораторні роботи	захист звітів лабораторних робіт	до 5
4.1	Здатність до самостійного пошуку технічної літератури або інших джерел інформації для пошуку шляхів усунення помилок студентом під час виконання лабораторних робіт	лекційні демонстрації та лабораторні роботи	звіти з лабораторних робіт	5

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни Програмні результати навчання (назва)	Код														
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	2.1	2.2	2.3	2.4	2.5	3.1	3.2	4.1
РН8. Застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення складних задач комп'ютерної інженерії та дотичних проблем.		+	+	+				+	+			+			+
РН9. Розробляти програмне забезпечення для вбудованих і розподілених застосунків, мобільних і гібридних систем.					+	+	+		+	+	+	+			
РН11. Приймати ефективні рішення з питань розроблення, впровадження та експлуатації комп'ютерних систем і мереж, аналізувати альтернативи, оцінювати ризики та імовірні наслідки рішень.	+	+	+	+									+	+	+

7. Схема формування оцінки

7.1. Форми оцінювання

Рівень досягнення всіх запланованих результатів навчання визначається за результатами виконання та захисту звітів лабораторних робіт, а також за результатами підсумкового іспиту. Внесок результатів навчання у підсумкову оцінку, за умови їх опанування на належному рівні:

- результати навчання 1.1 – 1.7 [знання] – до 30%;
- результати навчання 2.1 – 2.5 [вміння] – до 60%;
- результати навчання 3.1 – 3.2 [комунікація] – до 5%;
- результат навчання 4.1 [автономність та відповідальність] – до 5%;

Форми оцінювання:

- **семестрове оцінювання:** Навчальний семестр має 4 лабораторні роботи. Після завершення виконання лабораторних робіт та написання звітів проводяться усні захисти звітів із демонстраціями коректного виконання завдань, аналізом продуктивності запропонованого рішення та оглядом здійснених оптимізацій. Обов'язковим для допуску до іспиту є: захист звіту з кожної лабораторної роботи з кількістю балів не менше 8 із 15.
- **підсумкове оцінювання (у формі іспиту):** форма іспиту – письмове електронне тестування. Тест складається з 20 питань із варіантами відповіді, які оцінюються по 2 бали за кожне. Всього за іспит можна отримати від 0 до 40 балів. Умовою досягнення позитивної оцінки за дисципліну є отримання не менш ніж 60 балів, оцінка за іспит не може бути меншою за **24 бали**.
- **умови допуску до підсумкового іспиту:** умовою допуску до іспиту є отримання студентом сумарно не менше, ніж *критично-розрахунковий мінімум* за семестр. Студенти, які протягом семестру сумарно набрали меншу кількість балів, ніж критично-розрахунковий мінімум **36 балів**, для одержання допуску до іспиту обов'язково повинні виконати додаткове завдання за тематикою лабораторних робіт.

У випадку відсутності студента з поважних причин відпрацювання та захист пропущених лабораторних робіт здійснюються у відповідності до «Положення про організацію освітнього процесу у Київському національному університеті»

7.2. Організація оцінювання

Оцінювання за формами контролю:

Семестрова робота	Кількість балів	
	Min. – 8	Max. – 15
Лабораторна робота №1	8	15
Лабораторна робота №2	8	15
Лабораторна робота №3	8	15
Лабораторна робота №4	8	15
Додаткові завдання	–	12

Орієнтований графік оцінювання:

Форма оцінювання	Орієнтовний період для здійснення відповідної форми оцінювання
Лабораторні роботи №1 та №2	березень
Лабораторна робота №3	квітень
Лабораторна робота №4	травень
Добір балів/додаткові завдання	травень-червень
Іспит	червень

Розрахунок балів, які отримують при успішній здачі іспиту:

Значення	Семестрова робота	Іспит	Підсумкова оцінка
Мінімум	36	24	60
Максимум	60	40	100

7.3. Шкала відповідності оцінок

Оцінка (за національною шкалою) / National grade	Рівень досягнень, % / Marks, %
Відмінно / Excellent	90-100%
Добре / Good	75-89%
Задовільно / Satisfactory	60-74%
Незадовільно / Fail	0-59%

8. Структура навчальної дисципліни

8.1. Тематичний план занять

№ з/п	Назва теми	Кількість годин		
		Лекції	Лабораторні роботи	Самостійна робота
1	Вступ до предметної області високопродуктивних обчислень	2	4	15
2	Будова та характеристики високопродуктивних обчислювальних систем	2	8	15
3	Обчислювальні кластери: архітектура, компоненти та системи керування ресурсами	2	8	15
4	Технологія OpenMP. Принципи та реалізація в мовах програмування.	2	6	15
5	Практичне застосування OpenMP до розпаралелювання програм.	2	6	15
6	Технологія MPI. Принципи та реалізація в мовах програмування.	2	6	15
7	Обмін даними в MPI. Колективні операції. Комунікатори та топології.	2	6	15
8	Особливості програмування під спеціалізовані прискорювачі.	2	—	15
Всього		16	44	120

Загальний обсяг **180** год., в тому числі:

Лекції **16** год.

Лабораторні роботи **44** год.

Самостійна робота **120** год.

8.2. Лабораторні заняття

№ з/п	Назва теми	Кількість годин
1	Запуск обчислень в середовищі обчислювального кластера	10
2	Масовий запуск пакетних завдань на обчислювальному кластері	10
3	Реалізація високопродуктивних обчислень за допомогою OpenMP	12
4	Реалізація високопродуктивних обчислень за допомогою MPI	12
Всього		44

9. Рекомендована література:

Основні джерела:

1. Thomas Sterling, Maciej Brodowicz, Matthew Anderson. High Performance Computing: Modern Systems and Practices. — Morgan Kaufmann, 2017. — 718 pp.
2. Організація паралельних обчислень: Навчальний посібник / Укладачі: Є. Ваврук, О. Лашко. – Львів: Національний університет “Львівська політехніка”, 2017. – 70 с.
3. Луцків А.М., Лупенко С.А., Пасічник В.В. Паралельні та розподілені обчислення. — Львів. Магнолія, 2017. — 566 с.
4. William Gropp, Ewing Lusk, Anthony Skjellum. Using MPI, third edition: Portable Parallel Programming with the Message-Passing Interface. — MIT Press, 2014. — 336 pp.
5. Barbara Chapman, Gabriele Jost and Ruud van der Pas. Using OpenMP – Portable Shared Memory Parallel Programming. — MIT Press, 2007. — 384 pp.

Додаткові і джерела:

1. Michael Klemm and Jim Cownie. High Performance Parallel Runtimes. — De Gruyter Oldenbourg, 2021. — 356 pp.
2. Jason Sanders, Edward Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming. — Addison-Wesley Professional, 2010. — 312 pp.
3. Timothy G. Mattson, Berna Massingill, Beverly Sanders. Parallel Programming Patterns: Working with Concurrency in OpenMP, MPI, Java, and OpenCL. — Addison Wesley, 2018. — 456 pp.