

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«Моделі та методи створення програмних систем паралельної та розподіленої обробки даних»



Ступінь освіти	Доктор філософії
Освітньо-наукова програма	Інженерія програмного забезпечення
Тривалість викладання	Осінній та весняний семестри
Кількість кредитів	6 кредитів ЄКТС (180 год.)
Заняття:	1, 2 семестр (1, 2, 3, 4 чв.)
лекції:	1 год./тижд.
практичні заняття:	1 год./тижд.
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=7191>

Консультації: за окремим розкладом, що попередньо погоджений зі здобувачами освіти.

Онлайн-консультації: MS Teams, електронна пошта, соціальні мережі.

Інформація про викладача:



Викладач:
Швачич Геннадій Григорович
д-р техн. наук, доц., професор каф. ПЗКС
Посилання на профіль:
Сторінка кафедри ПЗКС:
https://pzks.nmu.org.ua/ua/teachers/teachers.php
Orcid ID:
https://orcid.org/0000-0002-9439-5511
Scopus ID:
https://www.scopus.com/authid/detail.uri?authorId=56509642500
ResearchGate Profile:
https://www.researchgate.net/profile/Hennadii-Shvachyuk

1. Анотація до курсу

Моделі та методи створення програмних систем паралельної та розподіленої обробки даних – це дисципліна, яка забезпечує важливий аспект професійного світогляду здобувачів PhD рівня, спрямована на формування у них ефективних дослідників, здатних використовувати передові технології високопродуктивних обчислень для розв’язування складних задач комп’ютерного моделювання у різних галузях науки і технологій. Аналізуються моделі, платформи та методи створення програмних систем паралельної та розподіленої обробки інформації, а також методи програмування таких систем. Особлива увага приділяється аналізу технік підвищення ефективності паралельних і розподілених обчислень.

2. Мета та завдання курсу

Мета дисципліни – розвинути систему знань, умінь і навичок аспірантів в області розвитку систем паралельної та розподіленої обробки даних, поглибити

знання про моделі, методи та засоби паралельного програмування, що сприятиме формуванню здатності застосовувати знання з паралельного програмування на практиці, критично переосмислювати наявні інформаційні технології, відстежувати тенденції їх розвитку, розвивати й реалізовувати нові конкурентоздатні ідеї в галузі інформаційних технологій. Надати аспірантам знання та практичні навички щодо сучасних моделей і методів створення програмних систем паралельної та розподіленої обробки даних для вирішення наукових та інженерних задач, що потребують високої продуктивності, масштабованості та надійності.

Завдання курсу:

У результаті вивчення курсу аспіранти повинні набути компетентності на рівні новітніх досягнень у програмуванні високопродуктивних обчислень на основі паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки. Зокрема, розвивати: здатність критично осмислювати наявні інформаційні технології та відстежувати тенденції їх розвитку. Здобувач має вивчити: види і рівні паралелізму операцій в комп'ютерних системах; парадигми паралельного програмування; засоби програмування систем паралельних обчислень зі спільною та розподіленою пам'яттю; критерії ефективності паралельних алгоритмів; методи розробки паралельних та розподілених програм.

3. Результати навчання:

Пропонувати нові ефективні методи і моделі розроблення, впровадження, супроводу та забезпечення якості програмного забезпечення та управління відповідними процесами на етапах життєвого циклу.

Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем для отримання нових знань та/або створення інноваційних продуктів у інженерії програмного забезпечення та дотичних міждисциплінарних напрямках.

Глибоко розуміти загальні принципи та методи інженерії програмного забезпечення, а також методологію наукових досліджень, застосовувати їх у власних дослідженнях.

Розробляти, досліджувати та вдосконалювати програмні системи на основі моделей та засобів інтелектуальної обробки даних розподілених і масштабованих систем з урахуванням вимог до надійності їх структурно-алгоритмічного забезпечення.

Дисциплінарні результати навчання сформовано на основі ПРН освітньо-наукової програми «Інженерія програмного забезпечення» третього (освітньо-наукового) рівня вищої освіти (PH03, PH07, PH08, PH14).

4. Структура курсу

Види та тематика навчальних занять	Внесок в загальну оцінку, %
ЛЕКЦІЇ	
1 семестр	36
ТЕМА 1. ПАРАЛЕЛЬНІ І РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ Поняття, класифікація архітектур, еволюція парадигм. Сучасний науковий підхід і рішення у даному питанні. 1. Вступ (поняття та концепції паралельних і розподілених	

обчислень; відмінності між паралельними та розподіленими обчисленнями). 2. Класифікація архітектур (архітектури паралельних обчислень; SIMD (Single Instruction, Multiple Data); MIMD (Multiple Instruction, Multiple Data); Класифікація Флінна; архітектури розподілених обчислень (кластери, ґрид-обчислення, хмарні обчислення)). 3. Еволюція парадигм (розвиток паралельних обчислень (від перших суперкомп'ютерів до сучасних багатоядерних процесорів); розвиток розподілених обчислень (від локальних мереж до глобальних розподілених систем)). 4. Сучасний науковий підхід і рішення (сучасні методи та інструменти (використання мов програмування (MPI, OpenMP для паралельних обчислень); інструменти для розподілених обчислень (Apache Hadoop, Spark)); актуальні дослідження та інновації (огляд сучасних наукових статей та досліджень). 5. Висновки (переваги використання паралельних і розподілених обчислень у різних галузях; основні виклики та проблеми, з якими стикаються дослідники).	
ТЕМА 2. МОДЕЛІ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ PRAM, BSP, Dataflow, SIMD/MIMD, MapReduce. 1. Вступ (поняття та концепції моделей паралельних обчислень, важливість моделей паралельних обчислень у сучасних системах). 2. PRAM (Parallel Random Access Machine) (принципи PRAM; модель обчислень з спільною пам'яттю; типи PRAM: EREW, CREW, CRCW; приклади використання, алгоритми для PRAM). 3. BSP (Bulk Synchronous Parallel) (принципи BSP; модель обчислень з синхронізацією; фази обчислень: обчислення, комунікація, синхронізація; приклади використання; алгоритми для BSP). 4. Dataflow (принципи Dataflow; модель обчислень, заснована на потоках даних; вузли та ребра в графах обчислень; приклади використання; реалізація Dataflow в сучасних системах). 5. SIMD/MIMD (принципи SIMD (Single Instruction, Multiple Data); виконання однієї інструкції над множиною даних; приклади використання: графічні процесори; принципи MIMD (Multiple Instruction, Multiple Data); виконання множини інструкцій над множиною даних; приклади використання: багатоядерні процесори). 6. MapReduce (принципи MapReduce; модель обчислень для обробки великих обсягів даних; фази обчислень: Map, Shuffle, Reduce; приклади використання; реалізація MapReduce в хмарних обчисленнях). 7. Висновки (переваги використання різних моделей паралельних обчислень; основні виклики та проблеми, з якими стикаються дослідники).	
ТЕМА 3. МОДЕЛІ РОЗПОДІЛЕНИХ СИСТЕМ Модель клієнт-сервер, агентно-орієнтована модель, мікросервіси. 1. Вступ (поняття та концепції моделей розподілених систем; важливість розподілених систем у сучасних інформаційних технологіях). 2. Модель клієнт-сервер (принципи моделі клієнт-сервер; взаємодія між клієнтом і сервером; типи клієнт-серверних архітектур: товстий клієнт, тонкий клієнт; приклади використання; використання CDN для	

підвищення ефективності клієнт-серверних систем). 3. Агентно-орієнтована модель (принципи агентно-орієнтованої моделі; взаємодія автономних агентів; структурні елементи та етапи побудови агентно-орієнтованих моделей; приклади використання; агентно-орієнтоване моделювання в різних системах). 4. Мікросервіси (принципи мікросервісної архітектури; декомпозиція монолітних додатків на мікросервіси; використання контейнеризації (Docker) для розгортання мікросервісів; приклади використання; організація мікросервісів на серверних групах Kubernetes). 5. Висновки (переваги використання різних моделей розподілених систем; виклики та проблеми, з якими стикаються дослідники).	
ТЕМА 4. МЕТОДИ СИНХРОНІЗАЦІЇ ТА ВЗАЄМОДІЇ В ПАРАЛЕЛЬНИХ СИСТЕМАХ Взаємовиключення, бар'єри, семафори, монітори. 1. Вступ (синхронізація та взаємодія в паралельних системах; визначення та основні концепції; важливість синхронізації для забезпечення коректності та ефективності паралельних обчислень). 2. Взаємовиключення (принципи взаємовиключення; взаємовиключення як метод запобігання одночасному доступу до спільних ресурсів; використання замків (locks) та м'ютексів (mutexes); приклади використання; реалізація взаємовиключення в різних мовах програмування). 3. Бар'єри (принципи бар'єрів; бар'єри як метод синхронізації, що забезпечує одночасне завершення певних етапів обчислень; використання бар'єрів у багатопроцесорних системах; приклади використання; реалізація бар'єрів у бібліотеках паралельного програмування). 4. Семафори (принципи семафорів; семафори як метод управління доступом до ресурсів; типи семафорів: бінарні та рахункові; приклади використання; реалізація семафорів у різних мовах програмування). 5. Монітори (принципи моніторів; монітори як метод синхронізації, що забезпечує взаємовиключення та умови синхронізації; використання моніторів у об'єктно-орієнтованих мовах програмування; приклади використання; реалізація моніторів у різних мовах програмування). 6. Висновки (переваги використання різних методів синхронізації; основні виклики та проблеми, з якими стикаються дослідники).	
ТЕМА 5. АЛГОРИТМИ ТА МОДЕЛІ УЗГОДЖЕНОСТІ В РОЗПОДІЛЕНИХ СИСТЕМАХ CAP-теорема, моделі узгодженості (strong/eventual/causal consistency). 1. Вступ (поняття узгодженості в розподілених системах; визначення та основні концепції; важливість узгодженості для забезпечення коректності та надійності розподілених систем). 2. CAP-теорема (принципи CAP-теореми; визначення та пояснення властивостей: узгодженість (Consistency), доступність (Availability), стійкість до розділення (Partition Tolerance); компроміси між цими властивостями в розподілених системах; приклади використання; реалізація CAP-теореми в сучасних розподілених системах). 3. Моделі узгодженості (Strong Consistency (сильна узгодженість), визначення та основні принципи, переваги та недоліки використання	

сильної узгодженості; Eventual Consistency (кінцева узгодженість), визначення та основні принципи, переваги та недоліки використання кінцевої узгодженості; Causal Consistency (причинно-наслідкова узгодженість), визначення та основні принципи, переваги та недоліки використання причинно-наслідкової узгодженості). 4. Алгоритми узгодженості (алгоритми для забезпечення сильної узгодженості: Paxos, Raft; алгоритми для забезпечення кінцевої узгодженості: Dynamo, Riak; алгоритми для забезпечення причинно-наслідкової узгодженості: COPS, Eiger). 5. Висновки (переваги використання різних моделей узгодженості; основні виклики та проблеми, з якими стикаються дослідники).	
ТЕМА 6. ІНСТРУМЕНТИ ПОБУДОВИ ПАРАЛЕЛЬНОГО ПЗ MPI, OpenMP, CUDA, багатопоточність у Java/.NET. 1. Вступ (поняття паралельного програмного забезпечення; визначення та основні концепції; важливість паралельного програмування у сучасних обчислювальних системах). 2. MPI (Message Passing Interface) (принципи MPI; модель передачі повідомлень для паралельних обчислень; функції та бібліотеки MPI; приклади використання; реалізація паралельних алгоритмів з використанням MPI). 3. OpenMP (Open Multi-Processing) (принципи OpenMP; модель паралельного програмування для багатоядерних процесорів; директиви та функції OpenMP; приклади використання; реалізація паралельних обчислень з використанням OpenMP). 4. CUDA (Compute Unified Device Architecture) (принципи CUDA; модель паралельного програмування для графічних процесорів (GPU); функції та бібліотеки CUDA; приклади використання; реалізація паралельних алгоритмів з використанням CUDA). 5. Багатопоточність у Java/.NET (принципи багатопоточності; моделі багатопоточності у Java та .NET; приклади використання; реалізація багатопоточних програм у Java та .NET). 6. Висновки (переваги використання різних інструментів для побудови паралельного ПЗ; основні виклики та проблеми, з якими стикаються розробники).	
Семестрова контрольна робота за темами 1-6 (1 семестр)	36
2 семестр	
ТЕМА 7. ТЕХНОЛОГІЇ СТВОРЕННЯ РОЗПОДІЛЕНОГО ПЗ gRPC, Apache Kafka, REST, GraphQL, Docker, Kubernetes. 1. Вступ (поняття розподіленого програмного забезпечення; визначення та основні концепції; важливість розподіленого ПЗ у сучасних інформаційних технологіях). 2. gRPC (Google Remote Procedure Call) (принципи gRPC; високопродуктивний RPC фреймворк; використання Protocol Buffers для серіалізації даних; приклади використання; реалізація мікросервісів з використанням gRPC). 3. Apache Kafka (принципи Apache Kafka; платформа для обробки потоків даних у реальному часі; використання Kafka Streams для обробки даних; приклади використання; реалізація системи обробки даних у реальному часі з використанням Apache Kafka). 4. REST (Representational State Transfer) (принципи REST;	

архітектурний стиль для розробки веб-сервісів; використання HTTP методів для взаємодії з ресурсами; приклади використання; реалізація REST API для веб-додатків). 5. GraphQL (принципи GraphQL; запитова мова для API; використання схеми для визначення структури даних; приклади використання; реалізація API з використанням GraphQL). 6. Docker (принципи Docker; платформа для контейнеризації додатків; використання Docker для ізоляції та переносимості додатків; приклади використання; реалізація мікросервісів з використанням Docker). 7. Kubernetes (принципи Kubernetes; система оркестрації контейнерів; використання Kubernetes для автоматизації розгортання, масштабування та управління контейнерними додатками; приклади використання; реалізація кластерів контейнерів з використанням Kubernetes). 8. Висновки (переваги використання різних технологій для створення розподіленого ПЗ; основні виклики та проблеми, з якими стикаються розробники).	
ТЕМА 8. ПАРАЛЕЛЬНА ТА РОЗПОДІЛЕНА ОБРОБКА ВЕЛИКИХ ДАНИХ Hadoop, Spark, Flink: концепції, архітектура, приклади. 1. Вступ (поняття великих даних; визначення та основні концепції; важливість паралельної та розподіленої обробки великих даних у сучасних інформаційних технологіях). 2. Hadoop (принципи Hadoop; концепція та архітектура Hadoop; компоненти Hadoop: HDFS, MapReduce, YARN; приклади використання; реалізація обробки великих даних з використанням Hadoop). 3. Spark (принципи Spark; концепція та архітектура Spark; компоненти Spark: Spark Core, Spark SQL, Spark Streaming, MLlib, GraphX; приклади використання; реалізація обробки великих даних з використанням Spark). 4. Flink (принципи Flink; концепція та архітектура Flink; компоненти Flink: DataStream API, DataSet API, Flink ML; приклади використання; реалізація обробки великих даних з використанням Flink). 5. Порівняння Hadoop, Spark та Flink (переваги та недоліки; порівняння продуктивності та ефективності; вибір технології залежно від конкретних завдань). 6. Висновки (переваги використання паралельної та розподіленої обробки великих даних; основні виклики та проблеми, з якими стикаються розробники).	
ТЕМА 9. ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ РОЗПОДІЛЕНИХ СИСТЕМ Fault tolerance, реплікація, checkpointing, Paxos, Raft. 1. Вступ (поняття надійності в розподілених системах; визначення та основні концепції; важливість надійності для забезпечення коректності та безперервності роботи розподілених систем). 2. Fault Tolerance (Стійкість до відмов) (основні принципи fault tolerance; визначення та методи забезпечення стійкості до відмов; використання резервування та дублювання компонентів; приклади використання; використання глибокого навчання для забезпечення fault	

tolerance). 3. Реплікація (принципи реплікації; визначення та методи реплікації даних; типи реплікації: синхронна та асинхронна; приклади використання; реалізація розподілених ACID транзакцій з використанням реплікації). 4. Checkpointing (принципи checkpointing; визначення та методи збереження стану системи; типи checkpointing: координоване, незалежне, квазі-синхронне; приклади використання; реалізація checkpointing у розподілених системах). 5. Raхos (принципи алгоритму Raхos; визначення та основні етапи алгоритму Raхos); використання Raхos для досягнення консенсусу в розподілених системах; приклади використання; реалізація Raхos у сучасних розподілених системах). 6. Raft (принципи алгоритму Raft; визначення та основні етапи алгоритму Raft; використання Raft для досягнення консенсусу в розподілених системах; приклади використання; оптимізація використання Raft у розподілених системах). 7. Висновки (переваги використання різних методів забезпечення надійності; основні виклики та проблеми, з якими стикаються розробники).	
ТЕМА 10. ІНТЕЛЕКТУАЛЬНА ОБРОБКА ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ Моделі машинного навчання в Spark/Flink, розподілені ML-фреймворки. 1. Вступ (поняття інтелектуальної обробки даних; визначення та основні концепції; важливість інтелектуальної обробки даних у сучасних інформаційних технологіях). 2. Моделі машинного навчання в Spark (принципи Spark; концепція та архітектура Spark; компоненти Spark: Spark Core, Spark SQL, Spark Streaming, MLlib, GraphX; моделі машинного навчання в Spark; використання MLlib для реалізації моделей машинного навчання; приклади використання: класифікація, регресія, кластеризація). 3. Моделі машинного навчання в Flink (принципи Flink; концепція та архітектура Flink; компоненти Flink: DataStream API, DataSet API, Flink ML; моделі машинного навчання в Flink; використання Flink ML для реалізації моделей машинного навчання; приклади використання: обробка поточкових даних, аналіз великих даних). 4. Розподілені ML-фреймворки (принципи розподілених ML-фреймворків; TensorFlow, PyTorch, Horovod; використання розподілених ML-фреймворків для масштабування моделей машинного навчання; приклади використання; реалізація розподілених моделей машинного навчання з використанням TensorFlow та PyTorch). 5. Порівняння моделей машинного навчання в Spark та Flink(переваги та недоліки; порівняння продуктивності та ефективності; вибір технології залежно від конкретних завдань). 6. Висновки (переваги використання інтелектуальної обробки даних у розподілених системах; основні виклики та проблеми, з якими стикаються розробники).	
ТЕМА 11. МОДЕЛЮВАННЯ ТА ВЕРИФІКАЦІЯ ПРОГРАМНИХ СИСТЕМ	

Формальні методи, Petri net, TLA+, моделювання поведінки. 1. Вступ (поняття моделювання та верифікації програмних систем; визначення та основні концепції; важливість моделювання та верифікації для забезпечення коректності та надійності програмних систем). 2. Формальні методи (принципи формальних методів; визначення та основні техніки формальних методів; використання математичних технік для специфікації, верифікації та доказу правильності програм; приклади використання; використання формальних методів у реальних проектах, таких як системи реального часу та безпечні атомні електростанції). 3. Petri net (основні принципи мереж Петрі; визначення та основні компоненти мереж Петрі: позиції, переходи, маркування; використання мереж Петрі для моделювання динамічних комп'ютерних систем; приклади використання; реалізація моделювання паралельних процесів за допомогою мереж Петрі). 4. TLA+ (Temporal Logic of Actions) (принципи TLA+; визначення та основні концепції TLA+; використання TLA+ для специфікації та верифікації розподілених систем; приклади використання; реалізація моделей поведінки та перевірка коректності програмних систем з використанням TLA+). 5. Моделювання поведінки (принципи моделювання поведінки; використання скінчених автоматів для моделювання поведінки об'єктів; використання UML для візуального моделювання поведінки систем; приклади використання; реалізація чат-ботів з поведінкою, що керується скінченим автоматом). 6. Висновки (переваги використання різних методів моделювання та верифікації; основні виклики та проблеми, з якими стикаються розробники).	
ТЕМА 12. МЕТОДОЛОГІЯ НАУКОВИХ ДОСЛІДЖЕНЬ У ГАЛУЗІ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ Постановка задач, огляд літератури, вибір методів, оцінювання результатів. 1. Вступ (поняття методології наукових досліджень галузі розподілених обчислень; визначення та основні концепції; важливість методології у наукових дослідженнях розподілених обчислень). 2. Постановка задач (основні принципи постановки задач; визначення проблеми та формулювання дослідницьких питань та приклади постановки задач у розподілених обчисленнях; методи постановки задач; використання динамічного програмування для оптимального розподілу завдань; планування завдань у розподілених системах з обмеженою інформацією про ресурси). 3. Огляд літератури (принципи огляду літератури; важливість систематичного огляду літератури; методи пошуку та аналізу наукових статей; приклади огляду літератури; порівняльний аналіз технологій розподілених обчислень). 4. Вибір методів (принципи вибору методів; визначення критеріїв вибору методів для дослідження; порівняння потокової та пакетної обробки даних; приклади вибору методів; використання статистичних алгоритмів та кластерного аналізу).	

5. Оцінювання результатів (принципи оцінювання результатів; визначення метрик та критеріїв оцінювання; методи аналізу та інтерпретації результатів; приклади оцінювання результатів; аналіз ефективності хмарних обчислень для розподілених систем). 6. Висновки (переваги використання систематичного підходу до наукових досліджень; основні виклики та проблеми, з якими стикаються дослідники).	
Семестрова контрольна робота за темами 7-12 (2 семестр)	36
ПРАКТИЧНІ ЗАНЯТТЯ	
1 семестр	64
Практична робота №1 ПОБУДОВА БАГАТОПОТОЧНОГО ЗАСТОСУНКУ З OPENMP АБО STD::THREAD Аналіз продуктивності. <u>Мета:</u> Особливості функціонування багатопоточності та паралельного програмування. Використання OpenMP або std::thread для створення багатопоточних застосунків. Провести аналіз продуктивності створених застосунків.	22
Практична робота №2 РОЗРОБКА КЛІЄНТ-СЕРВЕРНОГО ЗАСТОСУНКУ З ВИКОРИСТАННЯМ GRPC АБО REST Реалізація взаємодії та протоколів. <u>Мета:</u> Особливості використання gRPC або REST для розробки клієнт-серверних застосунків. Реалізувати взаємодію та протоколи між клієнтом і сервером. Провести аналіз продуктивності виконаних завдань.	20
Практична робота №3 ВИКОРИСТАННЯ APACHE SPARK ДЛЯ ОБРОБКИ ВЕЛИКИХ ОБСЯГІВ ДАНИХ Завдання ETL, агрегація, обчислення. <u>Мета:</u> Ознайомитися з особливостями використання Apache Spark для обробки великих обсягів даних. Навчитися виконувати завдання ETL (Extract, Transform, Load), агрегацію та обчислення з використанням Apache Spark. Провести аналіз продуктивності виконаних завдань.	22
2 семестр	64
Практична робота №4 МОДЕЛЮВАННЯ РОЗПОДІЛЕНОГО ПРОТОКОЛУ УЗГОДЖЕННЯ (НАПРИКЛАД, RAFT) Імітаційне середовище або симулятор. <u>Мета:</u> Ознайомитися з особливостями функціонування розподілених протоколів узгодження. Навчитися моделювати протокол узгодження Raft. Використовувати імітаційне середовище або симулятор для перевірки коректності та ефективності протоколу.	20
Практична робота №5 ЗАСТОСУВАННЯ ІНСТРУМЕНТІВ РОЗПОДІЛЕНОГО	22

МАШИННОГО НАВЧАННЯ (НАПРИКЛАД, MLLIB, TENSORFLOW ON KUBERNETES) Побудова та тренування моделі. <u>Мета:</u> Інструменти розподіленого машинного навчання. Навчитися використовувати MLlib або TensorFlow на Kubernetes для побудови та тренування моделей машинного навчання. Провести аналіз продуктивності та ефективності тренуваних моделей.	
Практична робота №6 ФОРМАЛЬНЕ МОДЕЛЮВАННЯ ПОВЕДІНКИ ПАРАЛЕЛЬНОЇ СИСТЕМИ (НАПРИКЛАД, У TLA+ АБО PROMELA) Перевірка властивостей: безпека, завершеність, взаємовиключення. <u>Мета:</u> Ознайомитися з методами формального моделювання поведінки паралельних систем. Навчитися використовувати TLA+ або Promela для моделювання та перевірки властивостей паралельних систем. Перевірити властивості безпеки, завершеності та взаємовиключення.	22
<i>Загальна кількість</i>	100

5. Технічне обладнання та/або програмне забезпечення

Технічні засоби навчання: мультимедійні та комп'ютерні пристрої.

Засоби дистанційної освіти: Moodle, MS Teams.

6. Система оцінювання та вимоги

6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
74 – 89	добре
60 – 73	задовільно
0 – 59	незадовільно

Дисципліна (у першому та другому семестрах) оцінюється за результатами здачі 2 контрольних робіт, максимальна кількість – 100 балів за одну контрольну роботу, тобто 100% від оцінки за дисципліну у відповідному семестрі.

6.2. Здобувачі освітньо-наукового рівня PhD можуть отримати **підсумкову оцінку** з навчальної дисципліни **на підставі поточного оцінювання знань** за умови, якщо набрана кількість балів з поточного тестування та виконання і захисту практичних робіт складатиме не менше 60 балів.

У випадку якщо здобувач вищої освіти за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку проводиться підсумкове оцінювання (1-й семестр - диф. залік; 2-й семестр - іспит) під час сесії. Підсумкове оцінювання (диф. залік (після першого семестру) / іспит (після другого семестру)) проводиться у вигляді комплексної контрольної роботи, яка включає тестові та практичні запитання.

Теоретична частина оцінюється за результатами здачі тестових контрольних робіт, кожна з яких містить тестові запитання різного рівня складності (розподіл у

відсотках за окремими контрольними роботами див. в таблиці розділу 4). Загалом за контрольні тестові роботи отримується **максимум 36 балів**, тобто 36 % від загальної оцінки за дисципліну.

Практичні роботи (6 робіт – у вигляді індивідуального завдання з кожної, розподіл у відсотках див. в таблиці розділу 4) звіт з кожної роботи формується в письмовому вигляді. При несвоєчасному здаванні практичної роботи оцінка знижується вдвічі. У сумі за практичну частину курсу при поточному оцінюванні отримується **максимум 64 бали**.

Отримані бали за теоретичну частину та практичні роботи додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

Максимальне оцінювання поточного (семестрового) контролю в балах:

Теоретична частина	Практична частина	Разом
36	64	100

6.3. Критерії оцінювання підсумкової роботи. У випадку, якщо здобувач освітньо-наукового рівня PhD за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку проводиться **підсумкове оцінювання (диференційований залік / іспит)** під час сесії.

Диференційований залік / іспит проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та практичної частини курсу. Білет складається з **20 тестових завдань** з чотирма варіантами відповідей, одна правильна відповідь оцінюється в 3 бали (**разом 60 балів**) та **4 тестових завдань** з практичної частини, кожне з запитань оцінюється максимум у 10 балів (**разом 40 балів**), причому:

- 10 балів – відповідність еталону;
- 8 балів – відповідність еталону з незначними помилками;
- 5 балів – часткова відповідність еталону, питання розкрито не в повній мірі;
- 2 бали – невідповідність еталону, але відповідність темі запитання;
- 1 бал – фрагментарні результати у відповідності до теми запитання;
- 0 балів – відповідь не наведена або не відноситься до теми запитання.

Отримані бали за відкриті та закриті тести додаються і є підсумковою (семестровою) оцінкою за вивчення навчальної дисципліни. Максимально за підсумковою (семестровою) роботою здобувач вищої освіти може набрати 100 балів.

7. Політика курсу

7.1. Політика щодо академічної доброчесності. Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення

плагіату у Національному технічному університеті "Дніпровська політехніка"
<https://cutt.ly/IBesJEc>

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

7.2. Комунікаційна політика. Здобувачі вищої освіти повинні мати активовану корпоративну університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

7.3. Політика щодо перескладання. Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання підсумкового оцінювання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

7.4 Політика щодо оскарження оцінювання. Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

7.5. Відвідування занять. З 24.02.2022 реалізація освітньої діяльності відбувається в умовах правового режиму воєнного стану. Наявна низка небезпек: повітряні тривоги, ризики припинення енергозабезпечення, мобільного та Інтернет-зв'язку. Згідно з наказами по університету у 2024-2025 навчальному році освітня діяльність здобувачів всіх форм навчання здійснюється з використанням дистанційних технологій через синхронні та асинхронні комунікації.

Відвідування онлайн лекцій та практичних занять реалізується через приєднання до «команди» Microsoft Teams. Під час повітряної тривоги заняття перериваються і продовжуються лише за умов перебування учасників освітнього процесу у захищених приміщеннях. Викладачем (за технічної та безпекової можливості) здійснюється запис заняття для підтримки асинхронного формату навчання.

У випадках відсутності енергозабезпечення, мобільного та Інтернет-зв'язку викладачем забезпечується асинхронний формат навчання та комунікація зі здобувачами за допомогою каналів зв'язку, що функціонують.

Про причини неможливості взяти участь в онлайн заняттях, ускладненні доступу до матеріалів на дистанційних платформах НТУ «ДП» тощо здобувач вищої освіти має повідомити викладача в особистих повідомленнях чатів Microsoft Teams, або листом на корпоративну е-пошту НТУ «ДП», або через старосту чи представника адміністрації факультету.

8. Рекомендовані джерела інформації

Базова:

1. Минайленко Р.М. Паралельні та розподілені обчислення: навч. посіб. — Кропивницький: Видавець Лисенко В. Ф., 2021. — 153 с.
<https://dSPACE.kntu.kr.ua/server/api/core/bitstreams/396e02d2-725b-47b5-a1c0-ae07a9bec326/content>

2. Семеренко, В. П. Технології паралельних обчислень: навчальний посібник / Семеренко В. П. – Вінниця: ВНТУ, 2018. – 104 с.

3. Коломійцев О., Рибальченко А., Голубничий Д., Третяк В., Воронін, В., Комаров, В., Пустоваров, В., Філіппенков, О., Любченко, О., & Рудаков, І. (2023). ПРОБЛЕМИ РОЗВИТКУ ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ТА ЇХ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РІШЕННЯ ЗАДАЧ

ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ. *Grail of Science*, (32), 182–197.
<https://doi.org/10.36074/grail-of-science.13.10.2023.032>

4. McCarty S. Docker really did change the world. InfoWorld. URL: <https://www.infoworld.com/article/3639596/docker-really-did-change-the-world.html>

5. Mardeni A. Why is Docker so Popular. Engineering Education (EngEd) Program | Section. URL: <https://www.section.io/engineering-education/why-is-docker-so-popular/>

6. Docker documentation. URL: <https://docs.docker.com/>

7. Kubernetes documentation. URL: <https://kubernetes.io/docs/home/>

8. Бараненко, Роман. “ТЕХНОЛОГІЇ РОЗПОДІЛЬНИХ СИСТЕМ ТА ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ.” MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, Khmelnytskyi National University.

9. Ghani, A., Ahmad, S., Khan, M. A., Khalid, S., Zohaib, S. (2021), "A Survey on Anomaly Detection Techniques for Microservices". IEEE Access. №9. P. 122766–122805. DOI: <https://ieeexplore.ieee.org/document/10479425>

10. Li, Z., Zhang, W., Li, Z., Zhao, M. (2020), "Anomaly Detection for Microservices: A Survey". ACM Computing Surveys. №53. P. 1–42. DOI: <https://dl.acm.org/doi/10.1145/3639478.3643535>

11. Luo, J., Wu, J., Zhang, Y., Sun, L., Liu, Y. (2022), "Anomaly Detection for Microservices Based on Federated Learning". In: 2022 IEEE International Conference on Web Services (ICWS). IEEE, P. 582–589. DOI: <https://ieeexplore.ieee.org/document/9461272>

12. He, P., Ranjan, R., Nogueira, J., Veiga, L. M., Zhao, W. (2021), "A Survey on Anomaly Detection for Microservices in Cloud Computing". Journal of Systems and Software. №182. 111318 p. DOI: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00296-4>

13. Patel, A. “Error Management in Data Source Connections: The Role of DSN”. Journal of Computing and Information Technology, 2023, 31(1), 75-88, DOI: 10.1109/CIT.2023.03168

14. Chen, M. “Optimizing Connection Settings for Enhanced Performance: A DSN Approach”. Database Systems Review, 2022, 33(3), 167-182, DOI: 10.1007/s11042-022-12568-0

15. Юрій Клушин. Підвищення ефективності в системах клієнт-сервер через використання мереж CDN. CSN. 2024; Випуск 6, Номер 2. : сс. 93-106. <https://doi.org/10.23939/csn2024.02.093>

16. <https://www.webscale.com/engineering-education/>

17. Дивак М., Кіндзерський О. "Дослідження ефективності паралельної обчислювальної схеми ідентифікації інтервальних дискретних моделей на основі ройового інтелекту". Herald of Khmelnytskyi National University. Technical Sciences, 2024, 331(1), pp. 29-37. <https://doi.org/10.31891/2307-5732-2024-331-3>

18. Duchenchuk, V. i Boublik, V. 2020. Моделі узгодженості розподілених систем віртуальної реальності. Наукові записки НаУКМА. Комп'ютерні науки. 3, (Груд 2020), 7–11. DOI: <https://doi.org/10.18523/2617-3808.2020.3.7-11>

19. IEEE Article: The Ongoing Evolution of OpenMP. URL: <https://www.openmp.org/updates/ieee-article-the-ongoing-evolution-of-openmp/>

20. OpenMP in a Modern World: From Multi-device Support to Meta Programming. 18th International Workshop on OpenMP, IWOMP 2022, Chattanooga, TN, USA, September 27–30, 2022, Proceedings. URL: <https://link.springer.com/book/10.1007/978-3-031-15922-0>

21. Новини і статті по темі CUDA. URL: <https://itc.ua/ua/tag/cuda/>

22. Kafka. BOOKS AND PAPERS URL: <https://kafka.apache.org/books-and-papers>
23. B. Yuen, A. Matyushentsev, T. Ekenstam, and J. Suen, "GitOps and Kubernetes: Continuous Deployment with
24. Argo CD, Jenkins X, and Flux" O'Reilly Media, 2021, ISBN: 978-1492094317
25. Bohdan Fedoryshyn, Olena Krasko. MIGRATION OF SERVICES IN A KUBERNETES CLUSTER BASED ON WORKLOAD FORECASTING. ICTEE. 2024; Volume 4, Number 2 : 82-92 <https://doi.org/10.23939/ict2024.02.082>
26. Mazurova, O., Naboka, A. i Shirokopetleva, M. (2021) «ДОСЛІДЖЕННЯ МЕТОДІВ РЕАЛІЗАЦІЇ РОЗПОДІЛЕНИХ АСІД ТРАНЗАКЦІЙ ЗА ТЕХНОЛОГІЄЮ РЕПЛІКАЦІЇ», *СУЧАСНИЙ СТАН НАУКОВИХ ДОСЛІДЖЕНЬ ТА ТЕХНОЛОГІЙ В ПРОМИСЛОВОСТІ*, (2 (16), с. 19–31. doi: 10.30837/ITSSI.2021.16.019.
27. Белоус Р.В., Крилов Є.В. ОПТИМІЗАЦІЯ ВИКОРИСТАННЯ RAFT В РОЗПОДІЛЕНИХ СИСТЕМАХ З БАЗАМИ ДАНИХ. Інформатика, обчислювальна техніка та автоматизація. URL: <https://doi.org/10.32782/2663-5941/2024.3.1/07>
28. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. URL: <https://www.tensorflow.org/about/bib>
29. Introduction to TensorFlow. URL: <https://www.tensorflow.org/learn>
30. Scikit-learn URL: <https://scikit-learn.org/stable/index.html>
31. Shantyr, A., & Zinchenko, O. (2024). Forecasting software system quality metrics using modifications of the DeepLIFT interpretation method. *INNOVATIVE TECHNOLOGIES AND SCIENTIFIC SOLUTIONS FOR INDUSTRIES*, (4(30), 110–124. <https://doi.org/10.30837/2522-9818.2024.4.110>
32. Козирєв, А. і Шубін, І. (2023) «Метод планування завдань оброблення даних у розподілених системах з обмеженою інформацією про доступні ресурси», *СУЧАСНИЙ СТАН НАУКОВИХ ДОСЛІДЖЕНЬ ТА ТЕХНОЛОГІЙ В ПРОМИСЛОВОСТІ*, (3 (25), с. 27–39. doi: 10.30837/ITSSI.2023.25.027.
33. Stanislav Zhuravel, S. Dumych, Olha Shpur. Дослідження методів збору та обробки даних в розподілених інформаційних системах. ICTEE. 2021; Volume 1, Number 1 : 20-38 <https://doi.org/10.23939/ict2021.01.020/>
34. С. Журавель, С. Думич, О. Шпур. ДОСЛІДЖЕННЯ МЕТОДІВ ЗБОРУ ТА ОБРОБКИ ДАНИХ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ. Information and communication technologies, electronic engineering. Vol. 1, No. 1. pp. 20–38 (2021)
35. Nagel Chrisian, Professional C# and .NET. Wrox; 2021. - 1008 p.
36. Oaks S. Java Performance: In-depth Advice for Tuning and Programming Java 8, 11, and Beyond. O'Reilly Media, Inc.; 2end Edition, 2020. – 452 p.