

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра системного проектування

Затверджено

На засіданні кафедри системного проектування факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка (протокол № 1 від 28.08 2023 р.)

Завідувач кафедри:



Роман ШУВАР

Силабус з навчальної дисципліни
“Паралельні та розподілені обчислення”,
що викладається в межах ОПП “Комп'ютерні науки”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 122 – Комп'ютерні науки

Львів 2023 р.

Назва дисципліни	Паралельні та розподілені обчислення
Адреса викладання дисципліни	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій Кафедра системного проектування
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 122 – Комп'ютерні науки
Викладачі дисципліни	Шувар Роман Ярославович, завідувач кафедри системного проектування, кандидат фізико-математичних наук, доцент; Кулик Петро Русланович, асистент кафедри системного проектування.
Контактна інформація викладачів	roman.shuvar@lnu.edu.ua , https://electronics.lnu.edu.ua/employee/shuvar-r-ya-2 petro.kulyk@lnu.edu.ua , https://electronics.lnu.edu.ua/employee/kulyk-p-r
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 205, корпус факультету електроніки та комп'ютерних технологій, вул. Драгоманова 50, м. Львів
Сторінка курсу	https://electronics.lnu.edu.ua/course/paralelne-prohramuvannia-ipz https://moodle.elct.lnu.edu.ua/course/view.php?id=143
Коротка анотація дисципліни	Дисципліна “Паралельні та розподілені обчислення” є нормативною дисципліною з спеціальності 122 – Комп'ютерні науки для освітньої програми “Комп'ютерні науки”, яка викладається в 6-му семестрі в обсязі 4-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Інформація про дисципліну	Навчальну дисципліну розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для того, щоб оволодіти концепціями сучасного програмування в рамках парадигм паралельних та розподілених обчислень та розв'язувати складні прикладні задачі з використанням сучасних високопродуктивних обчислювальних систем. У дисципліні представлено як огляд концепцій сучасних паралельних та розподілених обчислень, так і засобів та інструментів, необхідних для розробки прикладного паралельного програмного забезпечення з використанням сучасних технологій MPI, OpenMP, CUDA, HTCCondor, Hadoop, GRID.
Мета та цілі дисципліни	<i>Мета:</i> ознайомлення студентів з сучасними підходами до організації паралельних та розподілених обчислень, архітектурами паралельних та розподілених обчислювальних систем і особливостями організації процесів паралельної та розподіленої обробки інформації для оволодіння сучасними підходами та інструментами для раціональної експлуатації паралельних та розподілених комп'ютерних систем. <i>Цілі:</i> забезпечити знайомство студентів з загальною теорією методів паралельних та розподілених обчислень і сформувати навички їх

	практичного використання; навчити студента вибирати і обґрунтувати методи розв'язування задач за допомогою паралельного та розподіленого виконання програм, вказати область їх застосування, записати алгоритм розв'язання, реалізувати його на персональному комп'ютері.
Література для вивчення дисципліни	<p>Основна література:</p> <ol style="list-style-type: none"> 1) В. Ю. Вінник. Алгоритмічні мови та основи програмування: мова С. Навчальний посібник. – Житомир: ЖДТУ, 2007. 328 ст. 2) Рольщиков В. Б. Технології розподілених систем та паралельних обчислень. Конспект лекцій. Одеса: ОДЕКУ 2016.155 ст. 3) Коцовський В.М. Теорія паралельних обчислень: навчальний посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с. 4) Семеренко, В. П. Технології паралельних обчислень : навчальний посібник / Семеренко В. П. – Вінниця : ВНТУ, 2018. – 104 ст. 5) Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., and Melon, R. (2000). Parallel Programming in OpenMP. San-Francisco, CA: Morgan Kaufmann Publishers. 6) OpenMP Common Core: Making OpenMP Simple Again – Tim Mattson, Helen He, Alice Koniges (2019). 7) William Gropp Ewing Lusk, Anthony Skjellum. Using MPI, third edition: Portable Parallel Programming with the Message-Passing Interface (Scientific and Engineering Computation) Paperback – November 7, 2014 8) Edward Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming. Michigan, April 2012. 9) Dr. Brian Tuomanen. Hands-On GPU Programming with Python and CUDA: Explore high-performance parallel computing with CUDA 1st Edition. Packt Publishing; 1st edition. November 27, 2018 10) Tom Deakin, Timothy G. Mattson. Programming Your GPU with OpenMP. The MIT Press, November 7, 2023 11) Timothy G. Mattson, Yun (Helen) He and Alice E. Koniges. The OpenMP Common Core. Making OpenMP Simple Again. The MIT Press, November 19, 2019 12) Л.М. Олещенко Технології оброблення великих даних – Київ: КПІ ім. Ігоря Сікорського, 2021. – 227 с. <p>Додаткова література (Інтернет-ресурси):</p> <ol style="list-style-type: none"> 1) HTCondor Version 10.2.1 Manual. Available at: https://htcondor.readthedocs.io/en/latest/index.html 2) OpenMP Reference Guide. Available at: https://www.openmp.org/resources/refguides/ 3) OpenMPI Documentation. Available at: https://www.open-mpi.org/doc/ 4) CUDA Toolkit Documentation 12.3 Update 1. Available at: https://docs.nvidia.com/cuda/ 5) Hadoop 3.3.6 Documentation. Available at: https://hadoop.apache.org/docs/stable/
Обсяг курсу	Загальний обсяг: 120 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 56 год.
Очікувані результати навчання	У результаті вивчення даного курсу студент буде: знати: основні положення, що лежать в основі паралельних та розподілених обчислень, концепції сучасного програмування, сучасні підходи до

	<p>організації паралельних та розподілених обчислень, архітектури паралельних та розподілених обчислювальних систем і особливості організації процесів паралельної та розподіленої обробки інформації. Мати навички створення та відлагодження паралельних та розподілених програм. Володіти методами і засобами програмного забезпечення для паралельних і розподілених комп'ютерних систем;</p> <p>вміти: застосовувати знання для розв'язування прикладних задач, а саме керувати та реалізовувати взаємодію процесів, моделювати паралельні та розподілені обчислення, створювати та налагоджувати паралельні та розподілені програми, здійснювати побудову паралельних та розподілених алгоритмів, створювати програми із застосуванням багатьох процесів (потоків). Реалізовувати синхронні або асинхронні варіанти паралельних та розподілених процесів з використанням бібліотек MPI / OpenMP / CUDA / HTCCondor / Hadoop, за допомогою стандартних засобів мови програмування C.</p> <p>Після вивчення даного курсу «Паралельні та розподілені обчислення» здобувачі набудуть таких Загальних (ЗК), Спеціальних/Фахових (СК) компетентностей та Програмних результатів навчання (ПР):</p> <p>СК 8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p> <p>СК 9. 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.</p> <p>СК 16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.</p> <p>ПР 10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.</p> <p>ПР 17. Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.</p>
Ключові слова	Паралельні обчислення, розподілені системи, процеси, паралельне програмування, паралельне опрацювання даних, OpenMP, MPI, CUDA, HTCCondor, Hadoop, GRID-технології.
Формат курсу	Очний
Теми	Див. Схема курсу
Підсумковий контроль, форма	Іспит у кінці семестру

Пререквізити	<p>Для вивчення даного курсу студентам потрібні базові знання з курсів:</p> <ul style="list-style-type: none"> - вища математика; - дискретна математика; - основи програмування; - алгоритми і структури даних; - об'єктно-орієнтоване програмування.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	<p>Інформаційні методи (лекції, презентації, лабораторні роботи, написання рефератів, виконання індивідуальних завдань, робота у групі, командна робота, обговорення, консультації для поглибленого розуміння тем, бесіда, ілюстрація, демонстрація), дедуктивні методи на основі узагальнень, евристичні методи (проблемна лекція), інтерактивні методи (дискусія).</p>
Необхідне обладнання	<p>Для проведення лекційних занять:</p> <ul style="list-style-type: none"> • монітор TFT 23"; • системний блок (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB) ; • мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); • комутатор мережевий для доступу до мережі Internet. <p>Для проведення лабораторних занять:</p> <ul style="list-style-type: none"> • комп'ютерна лабораторія з 12-14 робочими місцями; • монітори TFT 23"; • системні блоки (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB); • мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); • комутатор мережевий для доступу до мережі Internet. <p>Обладнання кластера паралельних та розподілених обчислень та GP GPU-сервера факультету:</p> <ul style="list-style-type: none"> • системний блок GPU0 R-LINE з процесором AMD Ryzen Threadripper 1900X , 2200Mhz, 64GB оперативної пам'яті, HDD1 100 GB, HDD2 1TB, Nvidia Geforce GTX 1080ti 11 Gb; • системний блок GPU1 MPG Z590 Gaming Force, з процесором CPU 11th Gen Intel Core(TM) i7-11700K 3,60Ghz, 32GB оперативної пам'яті, HDD 240 GB, GeForce RTX 3080 Ti 12288 MB; • системний блок GPU2 Z590 PLUS (MS-7d11), з процесором CPU 11th Gen Intel Core(TM) i7-11700K 3,60Ghz, 65GB оперативної пам'яті, HDD 240 GB, GeForce RTX 3080 10018 MB; • системний блок GPU3 Z590 PLUS (MS-7d11), з процесором CPU 11th Gen Intel Core(TM) i7-11700K 3,60Ghz, 65GB оперативної пам'яті, HDD 240 GB, GeForce RTX 3080 10018 MB; • MPI cluster у складі 19 нод HP Compaq 8200, з процесором Intel i5-2400 CPU 3.10 Ghz, 8GB оперативної пам'яті, HDD 500 GB; • Gateway кластера з процесором Xeon 2.4Ghz/3*4GB/1000BG/ATX Rack/Lin , HDD1 500 GB, HDD2 500 GB; • сервер кластера з процесором Xeon 2.4GHz/6*4GB/1000BG/ATX Rack, HDD1 1TB, HDD2 1TB;

	<ul style="list-style-type: none"> • монітор Lenovo C24-20 62A8KAT1UA. <p>Необхідне програмне забезпечення:</p> <ul style="list-style-type: none"> • операційна система ОС Ubuntu 16 LTS; • середовище розробки на мові програмування C: Geany (версії 1.38)/Code::Blocks (версії 17.12); • компілятор мови програмування C (пакет GCC версії 11.2); • бібліотеки OpenMP (OpenMP версії 5.1), MPI (OpenMPI версії 4.1.1); • пакет програмного забезпечення CUDA Toolkit 11.5.1; • пакет програмного забезпечення середовища HTCondor версії 10.1.1; • пакет програмного забезпечення середовища Hadoop версії 3.3.0; • SSH клієнт Putty версії 0.77; • SFTP клієнт FileZilla версії 3.57.0.
<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> • Змістовий модуль 1: до 12.5 балів за виконання модульного завдання – написання есе за обраною тематикою з переліку. • Змістовий модуль 2: до 12.5 балів за виконання модульного завдання – написання есе за обраною тематикою з переліку. • Виконання лабораторних робіт: до 25 балів. • Написання екзаменаційної роботи: до 50 балів. <p>Підсумкова максимальна кількість балів 100. Звітність за курс – іспит в кінці семестру.</p> <p>Змістовий модуль – самостійна робота студента оформлена у вигляді есе – робота друкованим текстом, рекомендованим обсягом до 10 сторінок (шрифт Times New Roman, 14). Есе включає в себе детальний розгляд обраної індивідуальної теми, приведення прикладів та листингів коду програм, огляду технологій, літератури. Код програм повинен бути обов'язково прокоментований та пояснений, необхідно також продемонструвати його роботу у разі, якщо в якості прикладу наводяться не окремі елементи технології, а суцільна програма. Есе також повинно містити посилання на літературні джерела/інтернет ресурси, що були використані під час його написання. Фінальна версія есе здається студентом у електронному форматі .pdf викладачу для оцінки. Теми для змістових модулів див. у розділі Питання до заліку чи екзамену.</p> <p>Академічна доброчесність: Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.</p> <p>Література. Уся література, яку студенти не зможуть знайти самостійно,</p>

буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.

Політика виставлення балів. Враховуються бали, набрані при поточному контролі та бали за виконання лабораторних робіт. При цьому обов'язково враховуються присутність на заняттях та активність студента під час заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.

Оцінювання лабораторних робіт (13 лабораторних робіт, максимальна кількість балів: 25) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії (0-5 балів за одну роботу) та захисту звіту по виконаній лабораторній роботі (0-5 балів за одну роботу). У підсумку, всі набрані бали множаться на коефіцієнт (0.192) для переведення у 25-ти бальну шкалу.

Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням:

5 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутої теми, надає правильні відповіді на запитання по темі, код програми функціонує відповідно до завдання;

4 – студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання (або з несуттєвими недоліками);

3 – студент не досить добре розуміє розглянутий матеріал та написаний ним код програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, код програми функціонує неточно, або з помірними недоліками;

2 – студент погано розуміє розглянутий матеріал та написаний ним код програми, студент в більшості надає помилкові відповіді на питання по темі, код програми функціонує з суттєвими недоліками;

1 - студент погано розуміє розглянутий матеріал та написаний ним код програми, код програми не функціонує належним чином;

0 - студент зовсім не засвоїв розглянутий матеріал, написаний ним код програми не відповідає темі/не функціонує взагалі.

Оцінювання змістових модулів (2 змістових модулів, 12.5 балів за кожний) — за результатами написаних студентом есе, тестів, програм, тощо.

Бали оцінювання змістових модулів нараховуються за наступним співвідношенням:

12.5-9 - розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, містить аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади коду повністю робочі та відповідають темі. Можуть бути присутні несуттєві помилки та невідповідності;

9-7 - відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні

	<p>неточності та/або невідповідності основній темі. Наведені приклади коду частково робочі, проте в загальному відповідають темі;</p> <p>7-4 - відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);</p> <p>4-2 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;</p> <p>2 – 0 – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.</p> <p>Оцінювання екзаменаційної роботи (до 50 балів) — за результатами написаної студентом екзаменаційної роботи, що складається з 4 питань: 2 теоретичних (по 15 балів за кожне) та 2 практичних (написання прикладу коду програми, по 10 балів за кожне).</p> <p>Бали оцінювання екзаменаційної роботи нараховуються за наступним співвідношенням:</p> <p>50-40 – відповіді на запитання надано правильно, обґрунтовано, логічно, міститься аналіз і систематизація, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади коду робочі та відповідають темі питання, прокоментовані. Можуть бути присутні несуттєві помилки та невідповідності;</p> <p>40-30 - виявлено достатні знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності у відповідях на питання. Наведені приклади коду частково робочі, проте в загальному засвідчують достатнє розуміння студентом матеріалу;</p> <p>30-20 - відстежується загальне розуміння розглянутих питань. Виявлені неточності та невідповідності, пояснення наведеного коду відсутні, код наведений із значними неточностями;</p> <p>20-10 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності у відповідях на запитання. Наведені приклади коду з суттєвими недоліками, або не відповідають поставленому питанню;</p> <p>10 – 0 – студент взагалі не розуміє матеріал. Питання не розкрито, присутні значні неточності та невідповідності. Наведений код не робочий, або відсутній як такий.</p> <p>Критерії оцінювання результатів неформальної освіти: Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p>Питання до модульного</p>	<p>Орієнтовні теми для підготовки до змістових модулів/екзамену:</p> <p>1. Конструкції для створення потоків OpenMP. Директива parallel.</p>

<p>контролю</p>	<p>Конструкції розподілення роботи між потоками OpenMP. Директива for (parallel for). Параметри планування розподілу ітерацій. Директива sections. Приклади програм з коментарями.</p> <p>2. Конструкції для управління роботи з даними OpenMP. Ініціалізація даних. Копіювання даних. Редукція. Приклади програм з коментарями.</p> <p>3. Конструкції для синхронізації потоків OpenMP. Директива atomic. Директива critical. Директива barrier. Директива single. Директива master. Директива flush. Приклади програм з коментарями.</p> <p>4. Завдання (tasks) OpenMP. Переваги та недоліки використання завдань у порівнянні з іншими методами розпаралелення. Приклади програм з коментарями.</p> <p>5. Функції часу виконання OpenMP. Функції середовища виконання. Приклади програм з коментарями.</p> <p>6. Замки OpenMP. Навести приклади заміни інших конструкцій блокувань за допомогою замків. Приклади програм з коментарями.</p> <p>7. Змінні оточення OpenMP та відповідні їм функції. Приклади програм з коментарями.</p> <p>8. Основні поняття MPI. Загальна структура MPI програми. Отримання інформації про комунікатор та ранг процесу. Послідовна та паралельна версія обчислення числа пі.</p> <p>9. Комунікації точка-точка. Блокуючі передача та прийом даних. Написати приклад MPI програми.</p> <p>10. Комунікації точка-точка. Об'єднана передача-прийом даних. Функція MPI_Sendrecv. Функції MPI_Get_Count та MPI_Probe. Написати приклад MPI програми.</p> <p>11. Комунікаційні режими. Функції MPI_Bsend, MPI_Ssend, MPI_Rsend. Написати приклад MPI програми.</p> <p>12. Неблокуючий обмін даними. Функції MPI_Isend та Irecv. Функції MPI_Iprobe, MPI_Wait та MPI_Test. Написати приклад MPI програми.</p> <p>13. Колективні комунікаційні операції. Функція розсилки даних MPI_Bcast. Написати приклад MPI програми.</p> <p>14. Колективні комунікаційні операції. Функції MPI_Gather, MPI_Gatherv, MPI_Allgather, MPI_Allgatherv. MPI_Scatter, MPI_Scatterv, MPI_Alltoall, MPI_Alltoallv. Написати приклад MPI програми.</p> <p>15. Глобальні обчислювальні операції. Функції MPI_Reduce, MPI_Allreduce, MPI_Reduce_scatter, MPI_Scan. Написати приклад MPI програми.</p> <p>16. Групи процесів та операції з ними. Засоби доступу в групу. Конструктори та деструктори груп. Написати приклад MPI програми.</p> <p>17. Комунікатори та операції з ними. Засоби доступу до комунікатора. Конструктори та деструктори комунікаторів. Написати приклад MPI програми.</p> <p>18. Похідні типи даних. Функції MPI_Type_contiguous, MPI_Type_vector, MPI_Type_indexed, MPI_Type_struct. Написати приклад MPI програми.</p> <p>19. Пересилка – прийом запакованих типів даних. Функції MPI_Pack, MPI_Unpack. Написати приклад MPI програми.</p> <p>20. Технологія CUDA. Реалізація неграфічних обчислень на графічних процесорах. Особливості архітектури GP GPU CUDA: переваги та недоліки для розробки.</p> <p>21. Модель реалізації технології CUDA. Огляд та порівняння архітектур графічних процесорів Nvidia. Типи обчислювальних систем.</p>
------------------------	---

	<p>22. Інструменти розробника CUDA Toolkit. Основні бібліотеки CUDA Toolkit. CUDA C. Основи програмування за допомогою CUDA C.</p> <p>23. Модель пам'яті CUDA. Взаємодія між компонентами. Основні функції CUDA C. Модифікатори функцій. Вимірювання часу виконання програми.</p> <p>24. Типова структура програми CUDA C. Отримання параметрів GPU. Основні функції бібліотеки cuBLAS.</p> <p>25. HTCondor. Розгортання та огляд. Виконання завдань. Доступні реалізації: MPI, Java, Python, Docker, VM застосунки.</p> <p>26. HTCondor. Інтерфейс командного рядка. Адміністрування.</p> <p>27. Hadoop. Основні складові та їх взаємодія між собою. Принцип роботи. MapReduce. Інструменти аналізу та обробки даних.</p>
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

**Схема курсу “Паралельні та розподілені обчислення”
для студентів спеціальності 122 – Комп’ютерні науки**

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття) **лекція, самостійна, дискусія, групова робота)	Література. * Ресурси в інтернеті	Завдання, год	Термін виконання
1	Основні поняття паралельних обчислень та розподілених систем. Огляд основних понять паралельних обчислень та розподілених систем. Бібліотека OpenMP: історія розвитку, ключові поняття, базові функції та конструкції.	Лекція	[1], [2], [4], [5], [6], Сайт курсу – https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
1	Вступне заняття. Техніка безпеки, перевірка обладнання та налаштування програмного забезпечення. Робота з навчальним кластером факультету. Послідовна версія програм.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
2	Основні елементи OpenMP. Структура програми OpenMP. Директиви компілятора. Конструкції для створення та управління потоками. Конструкції розподілення роботи між потоками. Конструкції для управління роботи з даними. Приклади паралельних реалізацій множення матриць та векторів	Лекція	[1], [4], [5], [6], [*2], Сайт курсу	2	кінець поточного тижня
2	Лаб.1. Основні конструкції OpenMP програм.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
3	Конструкції для синхронізації потоків в OpenMP. Замки. Огляд директив синхронізації потоків. Синхронізація за допомогою технології замків. Види замків та принципи їх роботи. Приклади застосування замків для синхронізації потоків	Лекція	[1], [4], [5], [6], [9], [10] [*2], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня

3	Лаб.2. Паралельні цикли в OpenMP програмах.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
4	Функції часу виконання OpenMP. Функції середовища виконання. Змінні оточення. Замір часу виконання паралельної програми. Огляд основних функцій середовища виконання OpenMP програми. Огляд основних змінних оточення середовища OpenMP, приклади застосування. Паралельний розв'язок СЛАР методом Гауса.	Лекція	[1], [4], [4], [5], [9], [10] [*2], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
4	Лаб.3. Синхронізація в OpenMP програмах.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
5	Завдання (Task) в OpenMP. Огляд стандартів 2.5, 3.0, 3.1, 4.0, 4.5, 5.0 та 5.1. Огляд завдань (tasks) для організації паралельного виконання програми. Опції для роботи з завданнями, директиви синхронізації виконання завдань Огляд нововведень у стандартах OpenMP 2.5, 3.0, 3.1, 4.0, 4.5, 5.0 та 5.1.	Лекція	[1], [4], [5], [6], [9], [10] [*2], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
5	Лаб.4. Реалізація LU-розкладу матриці за допомогою завдань (tasks) в OpenMP програмах.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
6	Підхід до реалізації паралельних обчислень на основі MPI. Основні поняття MPI. Стандарти MPI та його реалізації. Основні поняття MPI, приклад MPI програми. Типи даних MPI, функції визначення часу виконання MPI програми. Структура програми, що використовує інтерфейс MPI. Функція MPI_Init. Функція MPI_Finalize. Функції MPI_Comm_size і MPI_Comm_rank. Функція MPI_Abort.	Лекція	[1], [7], [*3], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
6	Лаб.5. Реалізація алгоритмів сортування засобами OpenMP.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
7	Обміни повідомленнями між процесами в MPI. Комунікації точка-точка. Блокуюча передача. Блокуючий прийом. Бар'єрна синхронізація. Функція MPI_Sendrecv.	Лекція	[1], [7], [*3], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
7	Лаб. 6. Комунікації «точка-точка»: прості блоковані обміни.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/co	2	кінець поточного тижня

			urse/view.php?id=143		
8	Комунікаційні режими в MPI. Колективні операції передачі даних. Комунікаційні режими з блокуванням та без блокування.	Лекція	[1], [7], [*3], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
8	Підсумкове заняття ЗМ 1	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
9	Синхронізація процесів. Колективні комунікаційні операції. Використання блокуючих операцій комунікації. Використання передачі по готовності. Неблокуючий обмін. Об'єднання запитів на прийом/відправлення повідомлень.	Лекція	[1], [7], [*3], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	Кінець поточного тижня
9	Лаб.7. Комунікації «точка-точка». Неблокуючі передачі в MPI.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
10	Глобальні обчислювальні операції над розподіленими даними в MPI. Користувачькі функції. Комунікатори і групи процесів. Функції виконання глобальних обчислювальних операцій. Комунікатори і групи процесів. Функції для роботи з існуючими групами. Функції для створення та знищення груп та комунікаторів. Приклади паралельних реалізацій.	Лекція	[1], [7], [*3], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
10	Лаб.8. Колективні функції.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
11	Типи даних MPI. Похідні типи даних. Функції упаковки та розпакування даних. Функції декартових топологій. Огляд основних типів даних в MPI. Приклади застосування функцій упаковки та розпаковки. Віртуальні топології, функції декартових топологій, приклади реалізацій.	Лекція	[1], [7], [*3], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
11	Лаб.9. Групи і комунікатори.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
12	Технологія CUDA. Реалізація неграфічних обчислень на графічних процесорах. GPGPU. Технологія CUDA. Інструменти CUDA. CUDA Toolkit. Огляд архітектур графічних процесорів NVidia.	Лекція	[1], [8], [9] [*4], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня

12	Лаб.10. Отримання інформації про GPU з підтримкою CUDA.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
13	CUDA C. Основи програмування. Основи програмування мовою CUDA C. Типи обчислювальних систем. Основні терміни. Модель пам'яті CUDA. Основні функції CUDA C. Практичне застосування інструментів CUDA C.	Лекція	[1], [8], [9] [*4], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
13	Лаб.11. Реалізація Single-Precision A·X Plus Y за допомогою CUDA C API.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
14	Розподілені системи. Архітектура розподілених систем. Основні принципи. Типи комунікації та взаємодії в розподілених системах. Проблеми конкуренції та синхронізації у розподілених системах. Приклади реалізації.	Лекція	[2], [3], [12] [*1], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
14	Лаб.12. Реалізація паралельного множення матриць за допомогою CUDA C API.	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
15	Розподілені обчислення. Комп'ютерні кластери. HTCondor. Знайомство та розгортання системи. Завдання. Знайомство з доступними API. Використання з MPI застосунками. Надоор: основні складові та їх взаємодія між собою. Принцип роботи. MapReduce. Інструменти аналізу та обробки даних.	Лекція	[2], [3], [12] [*1], [*5], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
15	Лаб. 13. Знайомство з HTCondor. Розгортання та робота із MPI застосунками.	Лабораторна робота	[2], [*1] Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
16	Високопродуктивні обчислювальні системи. Виконання складних паралельних обчислень на базі навчального кластеру факультету. Java, Python реалізації в HTCondor. Адміністрування в HTCondor. Приклади застосування Надоор для аналізу та обробки даних.	Лекція	[1], [2], [3], [12] [*1]. [*5], Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня
16	Підсумкове заняття ЗМ 2	Лабораторна робота	Сайт курсу - https://moodle.ct.lnu.edu.ua/course/view.php?id=143	2	кінець поточного тижня