

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інтелектуальних та інформаційних систем



«ЗАТВЕРДЖУЮ»
Заступник декана
з навчальної роботи
М.О. Мостопалова
_____ 2018 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
СУЧАСНІ ТЕХНОЛОГІЇ
ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЕНЬ
для здобувачів освітньо-наукового рівня «доктор філософії»

| | |
|---------------------------|------------------------------|
| галузь знань | 12 «Інформаційні технології» |
| спеціальність | 122 «Комп'ютерні науки» |
| освітній рівень | третій (освітньо-науковий) |
| освітньо-наукова програма | «Комп'ютерні науки», |
| вид дисципліни | вибіркова |

| | |
|--|------------|
| Форма навчання | денна |
| Навчальний рік | 2018/2019 |
| Рік навчання | 2 |
| Кількість кредитів ECTS | 4 |
| Мова викладання, навчання та оцінювання | українська |
| Форма заключного контролю | екзамен |

Викладачі: професор Кудін Володимир Іванович, д.т.н.

Пролонговано: на 20 ^{19/2020} н.р. (04/19) «01» 07 2019р.
на 20 /20 н.р. () « » 20 р.

КИЇВ – 2018

1. Мета дисципліни спрямована на оволодіння студентами навиків з сучасних підходів, методів та технологій високопродуктивних комп'ютерних систем. Набуття аспірантами знань з класифікацією високопродуктивних обчислювальних систем, принципами розробки високопродуктивних програм, технологічними засобами паралельного та розподіленого програмування, механізмів синхронізації та управління процесами при розробці високопродуктивних програм, патернів побудови високопродуктивних обчислень.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

- Знати:* матеріал профільних дисциплін за спеціальністю «комп'ютерні науки».
- Вміти:* Виконувати обчислення показників програм та аналізувати їх; визначати тип та характеристики наявного обладнання та обирати найбільш ефективну реалізацію залежно від вибраних характеристик; розробляти паралельні алгоритми.

3. Анотація навчальної дисципліни:

Дисципліна «Сучасні технології високопродуктивних обчислень» належить до переліку дисциплін вільного вибору аспіранта. Вона забезпечує особистісний і професійний розвиток аспіранта, спрямована на формування досконалого володіння теоретичними знаннями для вирішення практичних завдань та підготовку майбутніх фахівців для ефективного використання сучасних обчислювальних систем у процесі виконання своїх професійних обов'язків. Задача курсу ознайомлення з основними принципами розробки високопродуктивних програм, оволодіння технологічними засобами паралельного та розподіленого програмування, вивчення та оволодіння механізмів синхронізації та управління процесами при розробці високопродуктивних програм. Особлива увага приділяється патернам побудови високопродуктивних обчислень..

4. Завдання (навчальні цілі): набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно науково-освітньої кваліфікації «Доктор філософії». Зокрема, розвивати: здатність застосовувати теоретичні та практичні принципи розробки високопродуктивних програм, технологічними засобами паралельного та розподіленого програмування, механізмів синхронізації та управління процесами при розробці високопродуктивних програм.

5. Результати навчання за дисципліною:

| Результат навчання (РН) (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 | | | |
| РН 1.1 | Знати типи та різновиди архітектур у високопродуктивних комп'ютерних систем | <i>Лекція, практичне заняття</i> | <i>Контрольна робота 1 (60% правильних відповідей), екзамен, активна робота на лекції, усні відпові</i> | 20% |
| РН 1.2 | Знати основні принципи та підходи розробки високопродуктивних обчислень | | | |
| РН 1.3 | Знати патерни паралельного та розподіленого програмування, методи та засоби дослідження ефективності комп'ютерних обчислень | | | 20% |
| РН 1.4 | Знати основні технології та технологічні засоби програмування у високопродуктивному середовищі | | | |
| РН 2.1 | Вміти проводити аналіз високопродуктивних середовищ та технологій; алгоритмів та обчислень | <i>Лекція, практичне заняття, самостійна робота</i> | <i>Контрольна робота 2 (60% правильних відповідей), екзамен, захист проекту, виконання</i> | 20% |
| РН 2.2 | Вміти будувати схеми побудови обчислень у високопродуктивному середовищі | | | 20% |

| | | | завдань, винесених на самостійну роботу | |
|--------|---|--------------------------------------|---|----|
| PH 2.3 | Вміти розробляти програмні застосунки у високопродуктивному середовищі із застосуванням паралельних та розподілених технологічних засобів | Практичне заняття, самостійна робота | Захист проекту | 5% |
| PH3.1 | Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти | | | 5% |
| PH4.1 | Демонстрація авторитетності, високий ступінь самостійності, академічна та професійна доброчесність, послідовна відданість розвитку нових ідей або процесів у передових контекстах професійної та наукової діяльності. | | | 5% |
| PH4.2 | Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість | | | 5% |

6. Схема формування оцінки.

6.1. Форми оцінювання здобувачів освітньо-наукового ступеня:

- оцінювання впродовж навчального періоду:

1. Активна робота на лекції, усні відповіді: PH1.1, PH1.2, PH1.3, PH1.4 – 5 балів/3 бали;
2. Виконання завдань, винесених на самостійну роботу: PH2.1, PH2.2 – 5 балів/3 бали;
3. Контрольна робота: PH1.1 - PH1.4, – 15 балів/9 балів;
4. Контрольна робота: PH2.1 - PH2.3 – 15 балів/9 балів;
5. Захист проекту: PH2.1 - PH2.3, PH3.1, PH4.1, PH4.2, – 20 балів/12 балів;

- підсумкове оцінювання: екзамен.

- максимальна кількість балів які можуть бути отримані: 40 балів;
- результати навчання які будуть оцінюватись: PH1.1, PH1.2, PH1.3, PH1.4;
- форма проведення і види завдань: письмова робота.

- умови допуску до підсумкового оцінювання:

Рекомендований мінімум для допуску до іспиту – 36 балів, критично розрахунковий мінімум – 20 балів.

При цьому обов'язковим є виконання всіх передбачуваних робочою програмою видів робіт не менше ніж на 60%, а також отримання позитивної оцінки з контрольної роботи.

Для здобувачів, які не набрали рекомендований мінімум (60% від максимально можливої кількості балів) обов'язковим є написання **комплексної семестрової контрольної роботи**, яка включає увесь пройдений матеріал за семестр і максимальна оцінка за яку не може перевищувати 40% підсумкової оцінки (до 40 балів за 100-бальною шкалою).

7.2. Організація оцінювання:

Обов'язковим є виконання завдань, винесених на самостійну роботу, та модульних контрольних робіт за графіком робочої програми.

Обов'язковим для отримання допуску до екзамену є виконання контрольної роботи та захист проекту до вказаної викладачем дати, перед початком екзаменаційної сесії, згідно навчального плану. Переписування чи перескладання тем не практикується. Дозволяється здача окремих завдань модульних тем у проміжках між написанням модульних контрольних робіт (наприклад, перша тема здається до здачі наступної модульної контрольної роботи у будь-який зручний для викладача та студента час).

Терміни проведення форм оцінювання:

1. Контрольна робота: на останньому лекційному занятті..
3. Захист проекту: до 15 тижня навчального періоду.

Умовою отримання позитивної результуючої оцінки з дисципліни є досягнення не менш як 60% від максимально можливої кількості балів, при цьому, оцінка за результати навчання, передбачені пунктами 2, 3, 4 не може бути меншою ніж 50% від максимального рівня. Максимальна кількість балів, яку може отримати здобувач за роботу протягом семестру становить **60 балів** за 100-бальною шкалою.

У випадку відсутності з поважних причин відпрацювання та перездачі контрольні роботи здійснюються у відповідності до „Положення про організацію освітнього процесу”.

7.3. Шкала відповідності оцінок

| | |
|----------------------------------|--------|
| Відмінно / Excellent | 90-100 |
| Добре / Good | 75-89 |
| Задовільно / Satisfactory | 60-74 |
| Незадовільно / Fail | 0-59 |

При визначені оцінки визначальною є робота в семестрі. Після завершення розгляду тем проводяться письмові контрольні роботи та теоретичне опитування.

8. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ. ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І ПРАКТИЧНИХ ЗАНЯТЬ

| № | Назва лекції | Кількість годин | | |
|---|---|-----------------|-----------|-------------------|
| | | Лекції | Практичні | Самостійна робота |
| 1 | <p>Тема 1. Вступ. Різноманітність архітектур високопродуктивних комп'ютерних систем. Паралельна обробка даних в них.</p> <p>Класифікація архітектур обчислювальних систем з паралельною обробкою даних. Симетрична багатопроцесорна архітектура SMP. Масивно-паралельна архітектура MPP. Гібридна архітектура NUMA. Організація когерентності багаторівневої ієрархічної пам'яті. Паралельна архітектура PVP з векторними процесорами. Кластерна архітектура</p> <p><i>Самостійна робота:</i> Визначити ефективну архітектуру паралельної обробкою даних для задач по темі дисертаційної роботи.</p> | 2 | | 12 |
| 2 | <p>Тема 2. Система OpenMPI. Комунікатори, типи даних, ініціалізація і завершення. Блокуючі та неблокуючі засоби комунікації. Основні функції.</p> <p>Викладаються основні відомості про пакет OpenMPI, починаючи з історії його створення та перших версій пакету, і закінчуючи командами установки і настройки сучасної четвертої версії пакету. Вивчаються структурні компоненти пакета: типи даних, класи, константи. Описуються основні функції, колективні та прості типи команд</p> <p><i>Самостійна робота:</i> На прикладі задач по темі дисертаційної роботи провести розробку алгоритму під пакет OpenMPI.</p> | 2 | | 12 |
| 3 | <p>Тема 3. Створення та виконання паралельної програми. Прості приклади.</p> <p>Вивчається створення паралельних програм, а також вивчаються програма Hello World, програми розсилки даних за допомогою одиночних команд посилки і приймання повідомлень, програми посилки і збору даних за допомогою колективних команд gather, scatter, bcast</p> <p><i>Самостійна робота:</i> Побудувати програму за допомогою колективних команд gather, scatter, bcast для задач по темі дисертаційної роботи.</p> | 2 | 2 | 12 |
| 4 | <p>Тема 4. Методи збору фрагментів масиву за допомогою процедур allGather і allToAll. Приклади програм.</p> | 2 | | 12 |

| | | | | |
|----------------------------|---|---|---|----|
| | <p>Вивчається створення паралельних програм з використанням колективних команд посилки і приймання повідомлень: allGather, allToAll, reduce, allReduce, reduceScatter, scan. Вивчається використання команди probe для визначення статусу команди прийому повідомлення</p> <p><i>Самостійна робота:</i> Реалізація паралельних програм з використанням колективних команд: allGather, allToAll, reduce, allReduce, reduceScatter, scan для задач по темі дисертаційної роботи..</p> | | | |
| <i>Контрольна робота 1</i> | | 1 | | |
| 5 | <p>Тема 5. Бібліотека процедур для пересилання об'єктів і їх масивів. На прикладах програм з пакету Math Partner вивчаються можливості розсилки і прийому масивів java-об'єктів. Пересилання будь-яких об'єктів між процесорами кластера здійснюється у вигляді байт-масивів, які є результатом процесу серіалізації об'єктів, що передбачений відповідним інтерфейсом java</p> <p><i>Самостійна робота:</i> Реалізувати пересилання об'єктів між процесорами кластера здійснюється у вигляді байт-масивів для задач дисертаційної роботи.</p> | 2 | 2 | 12 |
| 6 | <p>Тема 6. Приклади паралельних програм в пакеті Math Partner. Вивчаються алгоритми матричного множення для суперкомп'ютера. Вивчається схема множення "один-чотири", "один-вісім", програма множення матриці на вектор.</p> <p><i>Самостійна робота:</i> За допомогою пакету Math Partner реалізувати паралельні програми для задач дисертаційної роботи.</p> | 2 | | 12 |
| 7 | <p>Тема 7. Методи налагодження паралельних програм. Вивчається налагодження паралельних програм, а також програма простого відладчика і розглядається приклад використання цього відладчика в MPI програмі.</p> <p><i>Самостійна робота:</i> Провести документування результатів досліджень щодо ефективності високопродуктивних програм для задач дисертаційної роботи.</p> | 2 | | 12 |
| 8 | <p>Тема 8. Інструменти Math Partner для обчислень на кластері. Вивчається процес запуску паралельної програми з графічного додатку. Взаємодія Math Partner з кластерним пакетом PBS та запуск програми користувача кластера в Math Partner.</p> <p><i>Самостійна робота:</i></p> | 2 | | 12 |

| | | | |
|---|----|---|----|
| Побудувати та реалізувати паралельний алгоритм в кластера за допомогою Math Partner для задач дисертаційної роботи. | | | |
| <i>Контрольна робота 2</i> | 1 | | |
| ВСЬОГО | 18 | 4 | 96 |

Загальний обсяг 120 годин, в тому числі:

Лекцій – **18 годин**,

Практичні – **4 години**.

Консультації - **2 години**.

Самостійна робота – **96 годин**.

9. Рекомендовані джерела

Основні:

1. Малявко, А. А. Параллельное программирование на основе технологий openmp, mpi, cuda: учебное пособие / А. А. Малявко. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2018. — 116 с.
2. Абрамян, М.Э. Практикум по параллельному программированию с использованием электронного задачника Programming Taskbook for MPI: Учебное пособие / М.Э. Абрамян. – Ростов.-Д: Издательство ЮФУ, 2010. - 172 с.
3. Соколов А.В., Барковский Е.А., Кучумов Р.И., Сазонов А.М. Методы и алгоритмы параллельных вычислений. ПетрГУ, 2016. 66с.
4. Гергель В.П. Теория и практика параллельных вычислений: учебное пособие / В.П. Гергель. –М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2013. –423 с.
5. С. Немнюгин, О. Стесик. Параллельное программирование для многопроцессорных вычислительных систем. БХВ – Петербург, 2002. – 400 с.

Додаткові:

6. Kirk D.B., Hwu W.-M. Programming Massively Parallel Processors: A Hands-on Approach / D.B. Kirk, W.-M. Hwu. –Tsinghua University Press, Beijing, 2010. –552 p.
7. Wilt N. The CUDA Handbook: A Comprehensive Guide to GPU Programming (2nd Edition) / N. Wilt. –Addison-Wesley, 2018. –494 p.
8. William D. G. Using MPI: Portable Parallel Programming with the Message-passing / D. G. William, E. Lusk, A. Skjellum. –Massachusetts institute of technology, 1999. –377 p.
9. Trobec R., Slivnik B., Bulić P., Robič B. Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms // Springer, 2018. – 268p
10. Kurgalin S., Borzunov S. A Practical Approach to High-Performance Computing// Springer, 2019. – 206 p.
11. Adamatzky A., Akl S., Sirakoulis G. From Parallel to Emergent Computing // CRC Press, 2019. – 628 p
12. Lorenzon A., Filho A. Parallel Computing Hits the Power Wall: Principles, Challenges, and a Survey of Solutions // SpringerBriefs in Computer Science, 2019 – 88 p
13. Czarnul P.Parallel Programming for Modern High Performance Computing Systems// CRC Press, 2018. – 304 p

Інформаційні ресурси:

14. TOP500 Supercomputing sites : Project ranksand details the 500 most powerful computersystems in the world [Electronic resource]. Available from : <http://www.top500.org>.

15. <http://www.cs.wisc.edu/condor/>
16. <http://setiathome.ssl.berkeley.edu/>
17. <http://www.Distributed.net/>
18. <http://mersenne.org/>
19. <http://www.globus.org/>
20. <http://www.eu-datagrid.org/>