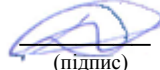


Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра комп'ютерних систем, мереж і кібербезпеки (№ 503)

ЗАТВЕРДЖУЮ

Голова НМК



(підпис)

Д. М. Крицький

(ініціали та прізвище)

«31» серпня 2021 р.

**РОБОЧА ПРОГРАМА ОBOB'ЯЗKОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Паралельні та розподілені обчислення

(назва навчальної дисципліни)

Галузь знань: 12 "Інформаційні технології"
(шифр і найменування галузі знань)

Спеціальність: 123 "Комп'ютерна інженерія"
(код та найменування спеціальності)

Освітня програма: Комп'ютерні системи та мережі

Освітня програма: Системне програмування

Освітня програма: Програмовні мобільні системи та Інтернет речей
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2021 рік

Розробник: _____Нікітіна Тетяна Сергіївна, доцент, к.т.н._____  _____
(прізвище та ініціали, посада, науковий ступінь та вчене звання) (підпис)

Робочу програму розглянуто на засіданні кафедри _____
_____ комп'ютерних систем, мереж і кібербезпеки _____
(назва кафедри)

Протокол № 1 від «30» 08 2021 р.

Завідувач кафедри _____ Д.Т.Н., професор _____  _____ В. С. Харченко _____
(науковий ступінь та вчене звання) (підпис) (ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 4	Галузь знань 12 «Інформаційні технології» Спеціальність 123 «Комп’ютерна інженерія» Освітні програми «Комп’ютерні системи та мережі», «Системне програмування» Рівень вищої освіти: перший (бакалаврський)	Обов’язкова
Кількість модулів – 2		Навчальний рік
Кількість змістовних модулів – 2		2021/2022
Індивідуальне завдання: немає		Семестр
Загальна кількість годин – 48/120		8-й
		Лекції*
Кількість тижневих годин для денної форми навчання: аудиторних – 4, самостійної роботи студента – 4		24 години
		Практичні, семінарські*
		24 години
		Лабораторні*
		0 годин
		Самостійна робота
		72 година
		Вид контролю
	модульний контроль, іспит	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: 48/72.

*Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

2. 2. Мета та завдання навчальної дисципліни

Мета вивчення: надання компетенцій, необхідних для організації паралельних та розподілених обчислень.

Завдання: вивчення базових положень організації паралельних та розподілених обчислень; оволодіння практичними навичками комплексного застосування аналізу вимог, проектування та розроблення програмного забезпечення для паралельних та розподілених систем; вивчення сучасних методик, підходів до аналізу великих даних; вивчення сучасних архітектур паралельних та розподілених систем.

Компетентності, які набуваються.

1. Здатність до абстрактного мислення, аналізу і синтезу.
2. Здатність вчитися і оволодівати сучасними знаннями.
3. Здатність застосовувати знання у практичних ситуаціях.
4. Здатність спілкуватися державною мовою як усно, так і письмово.
5. Вміння виявляти, ставити та вирішувати проблеми.
6. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
7. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
8. Здатність проектувати, впроваджувати та обслуговувати комп'ютерні системи та мережі різного виду та призначення.
9. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.
10. Готовність брати участь у роботах з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення.
11. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.
- 12.

Очікувані результати навчання:

В результаті вивчення дисципліни студенти мають досягти такі програмні результати навчання:

1. Знати новітні технології в галузі комп'ютерної інженерії.
2. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.
- 3.

Пререквізити: дисципліна є обов'язковим компонентом освітньої програми і базується на знаннях, отриманих під час вивчення дисциплін у циклі загальної і професійної підготовки, передбачених навчальним планом

спеціальності, зокрема, «Технології програмування», «Організація баз даних», «Web-технології» та є базовим підґрунтям для дисципліни «Проектування мікропроцесорних систем», «Кросплатформенні технології» та дипломної роботи бакалавра.

Кореквізити – комп'ютерні системи штучного інтелекту, технології обробки великих даних.

3. Зміст навчальної дисципліни

Модуль 1.

Змістовний модуль 1.

Тема 1. Введення в теорію паралельних та розподілених обчислень.

Предмет, мета вивчення і задачі дисципліни. Структура та зміст дисципліни і методичні рекомендації щодо її вивчення. Місце дисципліни у навчальному процесі. Вимоги до знань та вмінь тих, хто навчається. Характеристика рекомендованих під час вивчення дисципліни джерел інформації. Принцип оцінювання по модульно-рейтинговій системі. Моделі паралельних та розподілених обчислень, конвеєрна обробка даних, моделі розгортання, обслуговування cloud computing. Архітектура паралельних та розподілених систем, архітектура суперкомп'ютерів, кластерних систем, GRID-систем. Особливості грід-обчислень, введення в високопродуктивні обчислення. Класифікація Флінна, Хокні, Шнайдера.

Тема 2. Паралельні обчислювальні системи.

Розглядаються типи паралельних та розподілених систем.

Паралельні обчислення – це форма обчислень, в яких кілька дій проводяться одночасно. Ґрунтуються на тому, що великі задачі можна розділити на кілька менших, кожен з яких можна розв'язати незалежно від інших.

Є кілька різних рівнів паралельних обчислень: бітовий, інструкцій, даних та паралелізм задач. Паралельні обчислення застосовуються вже протягом багатьох років, в основному в високопродуктивних обчисленнях, але зацікавлення ним зросло тільки недавно, через фізичні обмеження зростання частоти. Оскільки споживана потужність (і відповідно виділення тепла) комп'ютерами стало проблемою в останні роки, паралельне програмування стає домінуючою парадигмою в комп'ютерній архітектурі, основному в формі багатоядерних процесорів.

Паралельні комп'ютери можуть бути грубо класифіковані згідно з рівнем, на якому апаратне забезпечення підтримує паралелізм: багатоядерність, багатопроцесорність – комп'ютери, що мають багато обчислювальних елементів в межах одної машини, а також кластери, MPP, та грід – системи що використовують багато комп'ютерів для роботи над одним завданням. Спеціалізовані паралельні архітектури іноді використовуються поряд з традиційними процесорами, для прискорення особливих задач.

Програми для паралельних комп'ютерів писати значно складніше, ніж для послідовних, бо паралелізм додає кілька нових класів потенційних помилок, серед яких найпоширенішою є стан гонитви. Комунікація, та синхронізація процесів зазвичай одна з найбільших перешкод для досягнення хорошої продуктивності паралельних програм.

Максимальний можливий приріст продуктивності паралельної програми визначається законом Амдала.

Тема 3. Основні поняття теорії паралельних обчислень.

Розглядаються такі теми «Граф алгоритму», «Концепція необмеженого паралелізму», «Недоліки концепції необмеженого паралелізму», «Внутрішній паралелізм», «Граф алгоритму», «Концепція необмеженого паралелізму».

Змістовий модуль 1. Тема 4. Програмування з використанням потоків.

Розпаралелювання програмного коду з використанням багатопоточності. Особливості оптимізації багатопотоковості на прикладі Java додатків. Багатозадачність підтримується практично усіма сучасними операційними системами. Існує два типи багатозадачності: багатозадачність, заснована на процесах та багатозадачність, заснована на потоках.

Багатозадачність на основі використання процесів дає змогу одночасно виконувати на комп'ютері декілька програм. При цьому програма є найменшою одиницею, якою може керувати планувальник операційної системи. Кожний процес потребує окремий адресний простір.

Змістовий модуль 2. Тема 5. Інтерфейс передачі повідомлень mpi.

Введення в технологію Message Passing Interface (MPI). Способи виконання функцій в MPI. Способи передачі повідомлень в MPI. Структура повідомлень в MPI. MPICH (MPI реалізація) технологія (фреймворк) для паралельних обчислювальних середовищ, включаючи кластери робочих станцій та масовопаралельних систем. Приклади MPI - програм, особливості реалізації даної моделі програмування.

Змістовий модуль 2. Тема 6. Інтерфейс open multi-processing.

OpenMP (Open Multi-Processing) — це набір директив компілятора, бібліотечних процедур та змінних середовища, які призначені для програмування багатопоточних застосунків на багатопроцесорних системах із спільною пам'яттю на мовах C, C++ та Fortran.

Змістовий модуль 2. Тема 7. Обчислення загального призначення на графічних процесорах.

Графічний процесор (англ. Graphics Proccesing Unit, GPU) — окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний рендеринг. Сучасні графічні процесори дуже ефективно обробляють і зображують комп'ютерну графіку, завдяки спеціалізованій конвеєрній архітектурі вони набагато ефективніші в обробці графічної інформації, ніж типовий центральний процесор.

Змістовий модуль 2. Тема 8. Обчислення загального призначення на графічних процесорах.

Обчислення загального призначення на графічних процесорах (GPGPU, рідко GPGR або GP²U) це використання графічного процесора (GPU), який зазвичай обробляє обчислення тільки для комп'ютерної графіки, для того щоб виконати обчислення в додатках, традиційно виконуваних центральним процесором (CPU). Використання декількох відеокарт в одному комп'ютері, або великої кількості графічних чипів, паралелізує вже паралельну природу обробки графіки. Більше того, навіть одна GPU-CPU платформа процесора забезпечує переваги, які не пропонують декілька центральних процесорів (CPU) самі по собі за рахунок спеціалізації в кожному чипі.

Змістовий модуль 2. Тема 9. Cuda фреймворк.

Області застосування паралельних розрахунків на GPU. Особливості архітектури графічних процесорів. Модель програмування NVIDIA CUDA, програмно-апаратна платформа для організації паралельних обчислень на графічних процесорах. Приклади розв'язання задач з підтримкою NVIDIA CUDA.

Змістовий модуль 2. Тема 10. Open computing language – открытый язык вычислений.

OpenCL (англ. Open Computing Language — открытый язык вычислений) — фреймворк для написания компьютерных программ, связанных с параллельными вычислениями на различных графических и центральных процессорах, а также FPGA. В OpenCL входят язык программирования, который основан на стандарте языка программирования Си C99, и интерфейс программирования приложений. OpenCL обеспечивает параллелизм на уровне инструкций и на уровне данных и является осуществлением техники GPGPU. OpenCL является полностью открытым стандартом, его использование не облагается лицензионными отчислениями.

Цель OpenCL состоит в том, чтобы дополнить открытые отраслевые стандарты для трёхмерной компьютерной графики и звука OpenGL и OpenAL возможностями GPU для высокопроизводительных вычислений. OpenCL разрабатывается и поддерживается некоммерческим консорциумом Khronos Group, в который входят много крупных компаний, включая AMD, Apple, ARM, Intel, Nvidia, Sony Computer Entertainment и другие.

Змістовий модуль 2. Тема 11. Amazon elastic compute cloud. 88

Amazon Elastic Compute Cloud (Amazon EC2) — веб-сервіс, котрий надає обчислювальні потужності в хмарі. Сервіс входить в інфраструктуру Amazon Web Services.

Простий веб-інтерфейс сервісу дозволяє отримати доступ до обчислювальних потужностей і налаштувати ресурси з мінімальними затратами. Він надає користувачам повний контроль над обчислювальними

ресурсами, а також доступне середовище для роботи. Служба скорочує час, необхідний для отримання і завантаження нового сервера.

Модульний контроль.

4. Структура навчальної дисципліни

Назва змістовного модуля і тем	Кількість годин				
	Усього	У тому числі			
		л	п	лаб.	с. р.
1	2	3	4	5	6
Модуль 1					
Змістовний модуль 1. Основи паралельних та розподілених обчислень					
Тема 1. Введення в теорію паралельних та розподілених обчислень.	6	2			4
Тема 2. Паралельні обчислювальні системи.	10	2	4		4
Тема 3. Основні поняття теорії паралельних обчислень.	10	2	4		4
Тема 4. Програмування з використанням потоків.	10	2	4		4
Модульний контроль	1	1			
Разом за змістовним модулем 1	37	9	12		16
Модуль 2					
Змістовний модуль 2. Основи паралельних та розподілених обчислень. Технології.					
Тема 5. Інтерфейс передачі повідомлень mpi.	12	2			10
Тема 6. Інтерфейс open multi-processing.	14	2	4		8
Тема 7. Обчислення загального призначення на графічних процесорах.	12	2	2		8
Тема 8. Обчислення загального призначення на графічних процесорах.	13	2	2		9
Тема 9. Cuda фреймворк.	7	2			5
Тема 10. Open computing language – открытый язык вычислений.	12	2	2		8
Тема 11. Amazon elastic compute cloud.	12	2	2		8
Модульний контроль	1	1			
Разом за змістовним модулем 2	83	15	12	0	56
Усього годин	120	24	24	0	72

5. Теми семінарських занять

№ п/п	Назва теми	Кількість годин
1	<i>Не передбачено</i>	
	Разом	

6. Теми практичних занять

№ п/п	Назва теми	Кількість годин
1	Тема 2. Паралельні обчислювальні системи.	4
2	Тема 3. Основні поняття теорії паралельних обчислень.	4
3	Тема 4. Програмування з використанням потоків.	4
4	Тема 6. Інтерфейс open multi-processing.	4
5	Тема 7. Обчислення загального призначення на графічних процесорах.	2
6	Тема 8. Обчислення загального призначення на графічних процесорах.	2
7	Тема 10. Open computing language – открытый язык вычислений.	2
8	Тема 11. Amazon elastic compute cloud.	2
	Разом	24

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	<i>Не передбачено</i>	
	Разом	

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Вивчити наступні теми «закон Амдала», «закон Густафсона», стани «Залежності», стани гонитви, взаємне виключення, синхронізація та паралельне сповільнення, моделі узгодженості, таксономія Флінна. Вивчити наступні теми: дрібнозернистий, крупнозернистий, та приголомшливий паралелізм.	12
2	Розглянути види паралельних комп'ютерів: багатоядерні обчислення, симетрична багатопроцесорність, розподілені обчислення, кластерні обчислення, масово паралельні обчислення, обчислення Грід, спеціальні паралельні комп'ютери.	14
3	Розглянути наступні теми: перші суперкомп'ютери, суперкомп'ютери в сучасному суспільстві, суперкомп'ютери в Україні, галузі застосування суперкомп'ютерів, програмне забезпечення суперкомп'ютерів, розвиток суперкомп'ютерної техніки.	16
4	Програмування з використанням потоків. розглянути потокову модель .net framework, конкурентний доступ та блокування ресурсів,	15

	потоки з використанням делегатів.	
5	Засреструватися та вивчити основи міжнародну програму Amazon AWS Educate, що надає учням всі необхідні ресурси для придбання навичок роботи в хмарі.	15
	Разом	72

9. Індивідуальні завдання

Не передбачено.

10. Методи навчання

Проведення аудиторних лекцій, практичних занять, консультацій, а також самостійна робота студентів з використанням відповідних матеріалів (п.14, 15).

11. Методи контролю

Проведення поточного контролю, електронного тестування, підсумковий контроль у вигляді іспиту.

12. Критерії оцінювання та розподіл балів, які отримують студенти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Модуль 1			
Лекції	0...2	6	0...12
Практичні роботи	0...3	6	0...18
Модульний контроль	0...20	1	0...20
Модуль 2			
Лекції	0...2	6	0...12
Практичні роботи	0...3	6	0...18
Модульний контроль	0...20	1	0...20
Усього за семестр			0...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту/заліку. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з двох теоретичних питань і одного практичного завдання. В першому і другому питанні студент повинен продемонструвати теоретичні знання. У третьому пункті – показати навички у створенні програмного забезпечення з технологіями паралельних та розподілених систем. Максимальна кількість балів за перший і другий пункт – по 25 балів за кожне питання, за третій пункт – 50 балів.

Необхідний обсяг знань для одержання позитивної оцінки:

- знати базові поняття теорії паралельних та розподілених систем, що стосується методів паралельного програмування;
- знати архітектуру сучасних паралельних та розподілених систем;

- знати сучасні технології з програмування на базі систем з паралельною та розподіленою архітектурою.

Необхідний обсяг вмінь для одержання позитивної оцінки:

- уміти використовувати сучасний інструментарій у вигляді технологій багато поточного програмування;
- уміти вирішувати задачу за допомогою технології NVIDIA CUDA;
- мати теоретичні знання «інтернет речей» (Internet of Things), «соціальні медіа» (social media, social networking services), «fog computing»;
- мати теоретичні знання про файлову систему Hadoop HDFS;
- мати теоретичні знання з теорії роботи наступних БД: Oracle NoSQL Database, Redis, Clusterpoint, Couchbase, CouchDB, DocumentDB, IBM Domino, MarkLogic, MongoDB, Cassandra, HBase;
- мати теоретичні знання з теорії роботи фреймворків для Big Data: Hive, Impala, Shark, Spark SQL, Drill. Поточна обробка даних: Spark Streaming. Імпорт даних: Kafka. Формати даних: Parquet, ORC, Thrift, Avro. Основи застосування мови Pig Latin;
- мати практичні навички роботи з NoSQL БД;
- мати теоретичні знання з побудови архітектури паралельних та розподілених систем, конвеєрної обробки даних.

Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Захистити не менше 80% від усіх завдань лабораторних занять. Уміти використовувати сучасні технології з розробки паралельних та розподілених систем. Мати теоретичні знання з технологій, архітектури побудови паралельних та розподілених систем.

Добре (75-89). Твердо знати мінімум, захистити не менше 90% завдань лабораторних занять. Уміти використовувати сучасні технології з розробки паралельних та розподілених систем. Мати теоретичні знання з технологій, архітектури побудови паралельних та розподілених систем. Мати практичні навички роботи з технологіями NVIDIA CUDA; мати практичні навички роботи з NoSQL БД; розробки додатків з використанням Hadoop.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та уміти їх застосовувати.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

13. Методичне забезпечення

1. Навчально-методичний комплекс дисципліни розміщений на кафедральному сервері у відповідному каталозі.
2. Дистанційний курс в системі дистанційного навчання Ментор, розташований за адресою: <https://mentor.khai.edu/course/view.php?id=4835>.

14. Рекомендована література

Базова.

1. Воеводин В.В. Модели и методы в параллельных процессах. - М.: Наука, 1986.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. -С.Птб.: ВНУ-Петербург, 2002, -608 с.
3. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. – Н.Новгород, ННГУ, 2001.
4. Шпаковский Г.И., Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI: Пособие / Минск: БГУ, 2002, -324 с.
5. Антонов А.С. Параллельное программирование с использованием технологии MPI. –М.: изд. Московского университета, 2004, -71 с.
6. Эндрю Таненбаум, Мартин ван Стеен Распределенные системы. Принципы и парадигмы. — Санкт-Петербург: Питер, 2003. — 877 с. — (Классика computer science). — ISBN 5-272-00053-6
7. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.
8. Federal Plan for High-End Computing: Report of the High-End Computing Revitalization Task Force (HECRTF), 2004, http://www.nitrd.gov/pubs/2004_hecrtf/20040702_hecrtf.pdf
9. Nvidia cuda toolkit 10.0.130: Release Notes for Windows, Linux, and Mac OS.
10. Flynn MJ (1966) Very high speed computers. Proc IEEE 54:1901–1901
11. M.J. Flynn. Some computer organizations and their effectiveness. IEEE Transactions on Computers, 21(9): 948–960, 1972
12. Мартин Фаулер, Прамодкумар Дж. Садаладж. NoSQL: новая методология разработки нереляционных баз данных. NoSQL Distilled. — М.: «Вильямс», 2013. — 192 с. — ISBN 978-5-8459-1829-1.
13. Леонид Черняк. Смутное время СУБД // Открытые системы. — 2012. — № 2.
14. Dan McCreary, Ann Kelly. Making Sense of NoSQL: A guide for managers and the rest of us. — Manning Publications, 2013. — 312 p. — ISBN 978-1-61729-107-4.
15. Olivier Curé, Guillaume Blin. Chapter 2. Database Management Systems // RDF Database Systems: Triples Storage and SPARQL Query Processing. — Elsevier Science, 2014. — 256 p. — ISBN 978-0-12-800470-8.

Допоміжна.

1. Hasan Khondker S., Chatterjee Amlan, Radhakrishnan, Sridhar, and Antonio John K., «Performance Prediction Model and Analysis for Compute-Intensive Tasks on GPUs.», The 11th IFIP International Conference on Network and Parallel Computing (NPC-2014), Ilan, Taiwan, Sept. 2014, Lecture Notes in Computer Science (LNCS), pp 612-17, ISBN 978-3-662-44917-2.
2. Merrill, Duane. Allocation-oriented Algorithm Design with Application to GPU Computing. Ph.D. dissertation, Department of Computer Science, University of Virginia. Dec. 2011.
3. «A Survey of Techniques for Managing and Leveraging Caches in GPUs», S. Mittal, JCSC, 23(8), 2018.

15. Інформаційні ресурси

16. [What is the Hadoop Distributed File System \(HDFS\)?](http://ibm.com). *ibm.com*. **IBM**.
17. Malak, Michael (2014-09-19). [Data Locality: HPC vs. Hadoop vs. Spark](http://datascienceassn.org). *datascienceassn.org*. Data Science Association.
18. [Characterization and Optimization of Memory-Resident MapReduce on HPC Systems](#) (pdf). IEEE. October 2014.
19. [Resource \(Apache Hadoop Main 2.5.1 API\)](http://apache.org). *apache.org*. Apache Software Foundation. 2014-09-12.
20. Официальное руководство по программированию на CUDA, вер. 1.1 // [CUDA Programming Guide](#). Chapter 1. Introduction to CUDA → 1.2 CUDA: A New Architecture for Computing on the GPU.
21. Мельник, Антон (19 квітня 2016). [Як програматик та Big Data змінюють ринок інтернет-реклами](#). *ain.ua*.
22. Костюк, Дмитро (26 травня 2016). [Як Vodafone та інші великі компанії використовують наші дані](#). *Tech Today*.
23. [Большие Данные — новая теория и практика](#). *Открытые системы. СУБД* (ru). 20 грудня 2011.
24. Hasan Khondker S., Chatterjee Amlan, Radhakrishnan, Sridhar, and Antonio John K., «Performance Prediction Model and Analysis for Compute-Intensive Tasks on GPUs.», The 11th IFIP International Conference on Network and Parallel Computing (NPC-2014), Ilan, Taiwan, Sept. 2014, Lecture Notes in Computer Science (LNCS), pp 612-17, ISBN 978-3-662-44917-2.
25. Merrill, Duane. Allocation-oriented Algorithm Design with Application to GPU Computing. Ph.D. dissertation, Department of Computer Science, University of Virginia. Dec. 2011.
26. «A Survey of Techniques for Managing and Leveraging Caches in GPUs», S. Mittal, JCSC, 23(8), 2018.