

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ УНІВЕРСИТЕТ ІМЕНІ БОРИСА ГРІНЧЕНКА
ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І ЗАСОБІВ
НАВЧАННЯ НАПН УКРАЇНИ
ІНСТИТУТ КІБЕРНЕТИКИ ІМЕНІ В. М. ГЛУШКОВА НАН
УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ А. С. МАКАРЕНКА
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

**«ТЕОРЕТИКО-ПРАКТИЧНІ ПРОБЛЕМИ
ВИКОРИСТАННЯ МАТЕМАТИЧНИХ МЕТОДІВ І
КОМП'ЮТЕРНО-ОРІЄНТОВАНИХ ТЕХНОЛОГІЙ В
ОСВІТІ ТА НАУЦІ»**

Збірник матеріалів
II Всеукраїнської конференції

28 березня 2018 року
м. Київ

Київ – 2018

УДК 004:378(082)
ББК 32.97:74.58я73

Схвалено Вченою радою
Факультету інформаційних технологій та управління Київського
університету імені Бориса Грінченка
(Протокол № 3 від 21.03.2018 р.)

Відповідальні за випуск:

**Д. М. Бодненко,
О. М. Глушак,
О. С. Литвин,
В. В. Прошкін**

Теоретико-практичні проблеми використання математичних методів та комп'ютерно-орієнтованих технологій в освіті та науці: зб. матеріалів у II Всеукраїнської конференції, 28 березня 2018 р., м. Київ / Київ. ун-т ім. Б. Грінченка; Відповід. за вип.: Д. М. Бодненко, О. М. Глушак, О. С. Литвин, В. В. Прошкін. – К. : Київ. ун-т ім. Б. Грінченка, 2018. – 235 с.

УДК 004:378(082)
ББК 32.97:74.58я73

© Автори публікацій, 2018

© Київський університет імені Бориса Грінченка, 2018

3. Круглик В. С. Концепція сучасного педагогічного програмного засобу. // Інформаційні технології і засоби навчання. Електронне наукове фахове видання. – 2007. – Випуск 3. – С. 2-7.
4. Пантюшенко Н. Г. Використання інформаційно-комп'ютерних технологій у викладанні мови. – 2013. – №17 (387).
5. Пінаєва О. Ю. Інформатизація освіти та її застосування в навчальному процесі // Актуальні проблеми трудової і професійної підготовки молоді. – 2004. – Вип. 10. – С. 150-151.
6. Тасенко О. В. Використання комп'ютерів у викладанні хімії та біології / О. В. Тасенко // Комп'ютер у школі та сім'ї. – 2007. – № 1. – С. 16–18.
7. Шумська Н. Комп'ютерні технології у навчанні хімії / Н. Шумська // Біологія і хімія в школі. – 2006. – № 6. – С. 24.

НАВЧАЛЬНИЙ МІКРОКОМП'ЮТЕРНИЙ КЛАСТЕР ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

Гуменюк М. Д.¹, Литвин О. С.¹, Литвин П. М.²

¹ Київський університет імені Бориса Грінченка, м. Київ

² Інститут фізики напівпровідників ім. В.Є. Лашкарьова НАН України, м. Київ

Однією із найбільш швидкозростаючих тенденцій у сучасних обчислювальних системах є застосування паралельних та розподілених обчислень. Це викликано не тільки принциповим обмеженням максимально можливої швидкодії послідовних систем, але й збільшенням задач, для вирішення яких можливостей існуючих засобів обчислювальної техніки або недостатньо, або дуже затратно [1]: чисельне розв'язання задач великої розмірності із багатьма змінними; моделювання об'єктів і проведення аналізу поведінки складних систем різної природи; управління складними промисловими та технологічними процесами в режимі реального часу і в умовах невизначеності; обробка великих об'ємів інформації або даних прямих спостережень; потік запитів в пошукових або клієнт-серверних системах та ін. В останніх, крім підвищення швидкодії, при паралельній (розподіленій) обробці досягається висока відмовостійкість без значного збільшення вартості та можливість апаратно-програмного оновлення «на ходу».

У відповідь на таке розширення кола розв'язуваних розподіленими системами завдань та значне зростання швидкодії та надійності комп'ютерних мереж виробниками обчислювальної техніки розробляються і впроваджуються нові концепції апаратної побудови таких систем, спрощується їх організація та керування. Проте залишається актуальною задача створення алгоритмів та програм для ефективного використання апаратних можливостей розподілених систем, оскільки програмні додатки

із реалізацією паралелізму складніші для проектування, розробки, налагодження та супроводу.

Це зумовлює необхідність вкладати кошти в додаткову інфраструктуру, знання та технології, щоб розробляти і досліджувати моделі ефективного паралельного обчислення та обробки інформації для конкретної задачі до їх реалізації на потужних суперкомп'ютерах або розподілених багатокомп'ютерних системах.

З іншого боку, на ринку праці спостерігається брак спеціалістів, здатних як апаратно, так і програмно реалізовувати паралельні обчислення. Щоб підготувати майбутніх фахівців до викликів, що постають при організації високо паралельних обчислень, їх підготовка повинна охоплювати вивчення розподілених моделей програмування, мережеских топологій, технологій організації обміну даними в мережах, розподілених баз даних, засобів та методів забезпечення відмовостійкості, надійності, безпеки та доступності апаратно-програмного комплексу, методик тестування, хмарних обчислення та ін.

Для багатьох вищих навчальних закладів достатнє матеріально-технічне та програмне забезпечення вивчення цих тематик є проблемою через високу вартість необхідної інфраструктури та швидке застарівання галузевих технологій.

Виходом в цій ситуації може бути створення навчальних/тестових систем з розподіленою пам'яттю і з легкодоступними готовими пристроями в якості обчислювальних вузлів. Перевагами такої архітектури є, зокрема, масштабованість – система може бути легко розширена шляхом додавання інших вузлів у міру необхідності, та надмірність – кілька вузлів можуть надавати однакові послуги, тому, якщо якийсь із них недоступний, робота системи не припиняється.

Наприклад, для зменшення витрат на обладнання та забезпечення доступності вивчення і проведення дослідження паралельних алгоритмів та програм можна побудувати модельний кластер на базі мікрокомп'ютерів Raspberry Pi [2-4]. Тому метою нашої роботи стало створення та дослідження мікрокомп'ютерної кластерної системи із використанням апаратно-програмної платформи Raspberry Pi, яка дасть можливість студентам або розробникам моделювати розв'язання задач із використанням розподілених обчислень.

Досліджувану в роботі кластерну систему побудовано на базі мікрокомп'ютера Raspberry Pi 3 модель В [5] (рис.1). Перевагами використання Raspberry Pi є: невисока вартість (35-40 \$), компактність (розміри 85.6x56x21 мм), підтримка різних операційних систем, універсальність застосування для різних проектів, можливість розгону, якщо не вистачає його звичайної продуктивності.

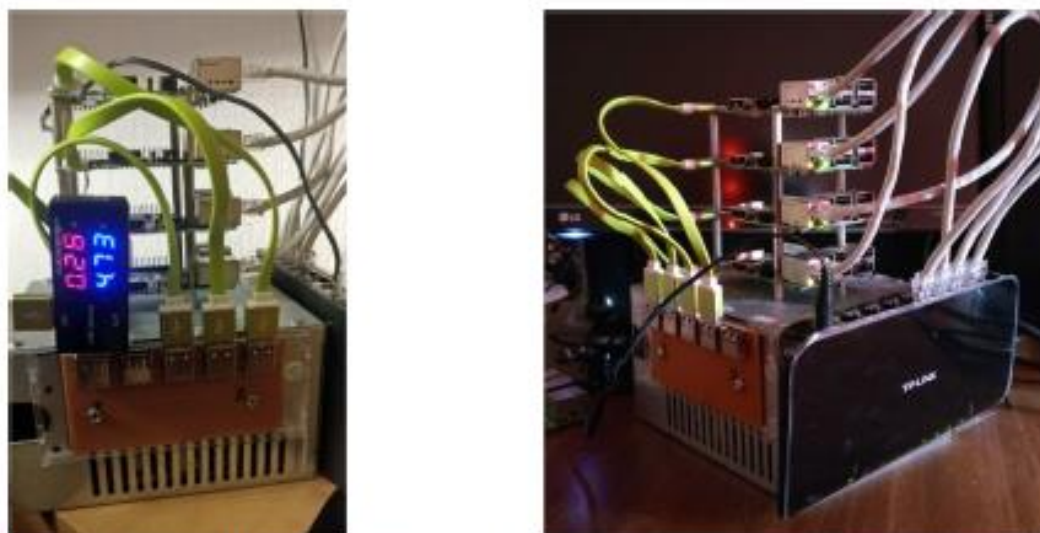


Рис. 1. Загальний вигляд кластера: блок живлення, мережевий комутатор та стійка з чотирьох плат Raspberry Pi 3.

Систему живлення реалізовано на основі стандартного блоку живлення персонального комп'ютера FSP ATX-450PNF, додатково оснащеного USB роз'ємами для під'єднання плат. Дослідження електроспоживання кластера на різних задачах показало, що обраний імпульсний блок живлення десктопного персонального комп'ютера потужністю 450 Вт забезпечує надійне живлення плат напругою 5 В при струмі до 20 А та захист від перевантажень, що, крім всього, дозволяє масштабування кластера до 10 вузлів.

Відомо, що одним із основних чинників зниження швидкодії є передача даних між вузлами кластера. Швидкісне (1 Гб в секунду) мережеве з'єднання між вузлами RPi-кластера забезпечує некерований 8-портовий гігабітний комутатор TP-Link TL-SG1008D.

Кластер побудований на операційній системі Raspbian - вільнорозповсюджуваній ОС на основі Debian, оптимізований для апаратних можливостей Raspberry Pi.

Для програмування розподілених обчислень в кластері використовується мова програмування Python, оскільки це одна найбільш популярних мов у високопродуктивних обчисленнях, поряд з C і Fortran, із безліччю бібліотек розширень для таких галузей, як біологія, обчислювальна хімія, біоінформатика, генетика, обчислювальна математика і статистика та ін., включаючи спеціальний інструментарій для паралельного програмування [6]. Крім того, у неї простий і виразний синтаксис, що підходить для навчання та швидкого створення прототипів.

У своїй реалізації Python має інтеграцію з MPI через бібліотеку mpi4py. Ця бібліотека є однією з найкращих реалізацій MPI на Python

серед конкуруючих альтернатив, має ефективний інтерфейс та покриває більшість стандарту MPI-2, у тому числі створення динамічного процесу [7].

Однією із задач для тестування побудованого кластера було обчислення числа π (використовувався ряд Грегорі-Лейбніца). При цьому дослідження включало визначення залежності прискорення розподіленої системи: 1) від різної кількості паралельно працюючих обчислювальних вузлів (нодів) від 2 до 16; 2) від різної розмірності задачі (іншими словами, кількості виконуваних операцій).

Реалізовано два способи організації обчислень: перший – розпаралелювання процесів на рівні процесорів мікроконтролерів (максимум 4 вузла), другий – розпаралелювання на ядрах плат (4x4 вузла).

Ми теоретично оцінили очікуване прискорення обчислень числа π при розпаралелюванні в залежності від кількості нодів та розмірності задачі, вважаючи, що час операцій (підсумовування дробів, обмін повідомленнями і даними та ін.) однаковий. Деякі результати обчислювальних експериментів із залученням від 1 (послідовне обчислення) до 16 вузлів (нодів) наведено в таблиці 1 та на рисунку 2. Із таблиці видно, що при збільшенні числа обчислювальних вузлів в кластері реальне прискорення значне нижче, ніж теоретично прогнозоване. Хоча, цього слід було очікувати, оскільки наші теоретичні розрахунки не враховували реальний час на виконання кожної операції та пропускну здатність мережі (в нашому випадку: 1 Гб в секунду).

Таблиця 1

Прискорення паралельного обчислення
в залежності від кількості вузлів та розмірності задачі

Розмірність задачі	5000000	12500000	25000000	50000000
<i>Експеримент</i>				
2 nodes	1,258100439	1,299998154	1,316461255	1,331345418
3 nodes	2,180925072	1,906033593	1,95357111	1,985640384
4 nodes	2,450933163	2,623369896	2,623340504	2,672141176
16 nodes	8,196933017			8,381551916
<i>Теоретична оцінка</i>				
2 nodes	0,999950002	0,99998	0,99999	0,999995
3 nodes	1,999680051	1,999872008	1,999936002	1,999968001
4 nodes	2,999010327	2,999604052	2,999802013	2,999901003
16 nodes	14,89499032			14,98943245

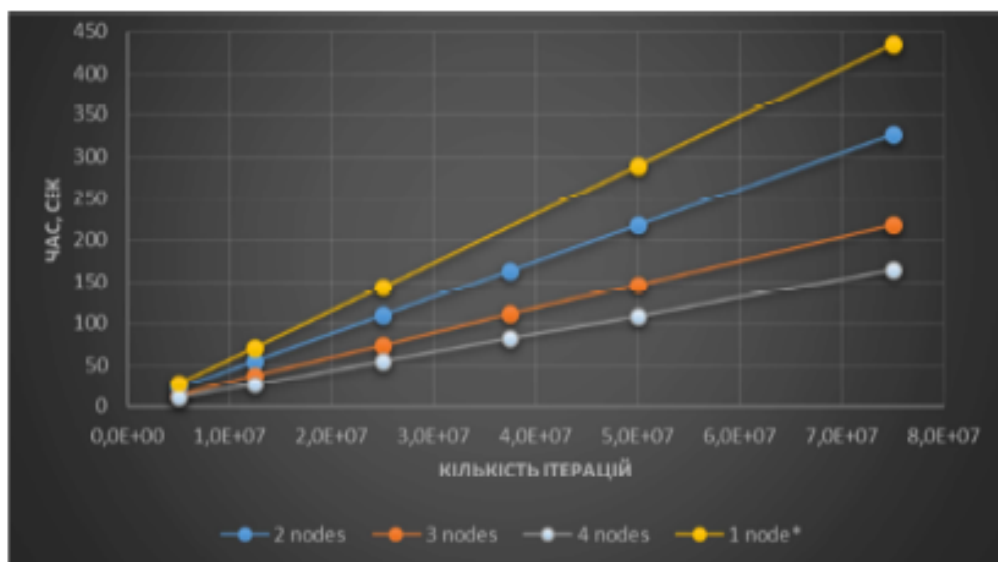


Рис. 2. Залежність часу, витраченого на обчислення числа π , від кількості ітерацій для різної кількості задіяних вузлів RPi-кластера (криві 2 nodes, 3 nodes, 4 nodes, 1 nodes, відповідно).

Слід зауважити, що динаміка зменшення часу на задане число ітерацій алгоритму при збільшенні кількості процесів різко зменшується при переході з вузлів-процесорів (4 нода) до вузлів ядер (4x4 нода) (рис.3). На нашу думку, це викликано неефективністю використання бібліотеки `mpi4py` у випадку організації розподілених обчислень на ядрах в межах одного процесора. Тим не менше, при збільшенні розмірності задачі (у нас – до 50 млн. ітерацій) спостерігається значне зростання прискорення у будь якому випадку.

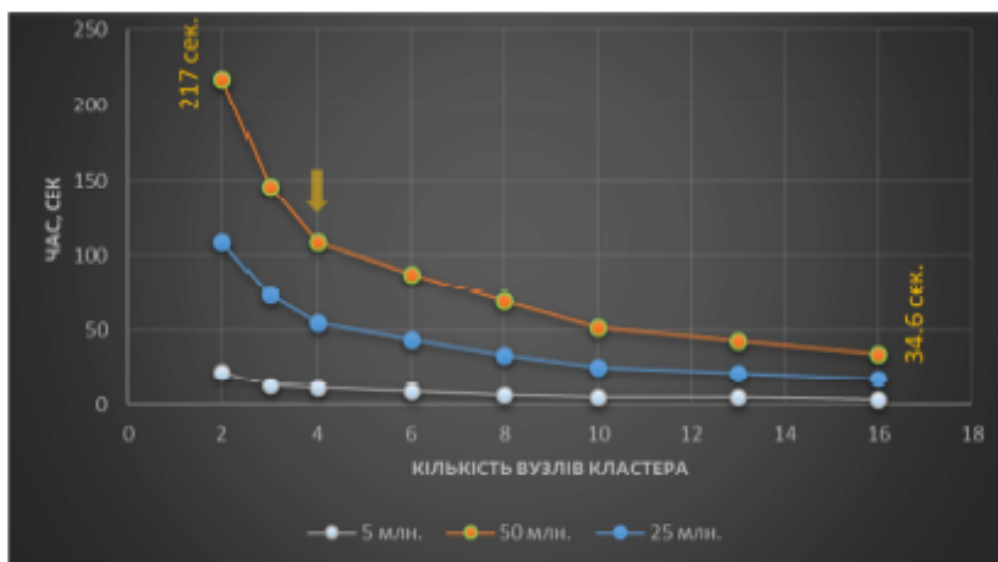


Рис. 3. Витрати часу при проведенні заданого числа ітерацій на різній кількості вузлів кластера. Криві 5, 25 та 50 млн. відповідають числу ітерацій. Стрілкою позначено перехід від збільшення кількості вузлів за рахунок кількості плат мікрокомп'ютерів до збільшення вузлів за рахунок кількості ядер процесора.

Таким чином, в роботі на базі чотирьох одноплатних мікрокомп'ютерів Raspberry Pi створено та досліджено кластер для розподілених обчислень. Проаналізовано особливості реалізації паралельних алгоритмів рішення типових завдань. Показано, що такий кластер із різною кількістю вузлів можна використовувати в якості тестового для перевірки якості розпаралелювання алгоритму для тих чи інших задач, поведінки паралельних програм та ін. При цьому у випадку розпаралелення процесів на процесори в якості вузлів динаміка росту прискорення із збільшенням вузлів вища, ніж у випадку розпаралелення на окремі ядра.

Крім навчання студентів принципам, підходам і технологіям паралельних і розподілених обчислень, розробки програмного забезпечення і тестування кластерних технологій подібні кластери можуть бути корисні як симулятори сенсорних мереж, в дослідженнях Інтернету речей та ін.

ДЖЕРЕЛА

1. Distributed Computing: 30th International Symposium DISC 2016, Paris, France, September 27-29, 2016. Proceeding.
2. Raspberry Pi supercomputer. – Режим доступу: <http://www.zdnet.com/article/raspberry-pi-supercomputer-los-alamos-to-use-10000-tiny-boards-to-test-software>
3. Ashwin Pajankar. Raspberry Pi Supercomputing and Scientific Programming. – Nashik, Maharashtra, India, 2017. – 171 p.
4. Creating a Raspberry Pi 3 Cluster - "Supercomputer", for parallel computing. – Режим доступу: <http://thundaxsoftware.blogspot.com/2016/07/creating-raspberry-pi-3-cluster.html>
5. Raspberry Pi 3 Model B. – Режим доступу: https://miniboard.com.ua/boards/272-raspberry-pi-3-model-b.html?gclid=CjwKCAiA7ovTBRAQEiwAo8dPcTCt4ftvHkBZ9EH7FivJBY1wazd0g0bNKRiRjGs7SztysWjwNf7XfxoC6wgQAvD_BwE...
6. Python. Numeric and Scientific. – Режим доступу: <https://wiki.python.org/moin/NumericAndScientific>
7. Compute Pi using Python and MPI4Py. – Режим доступу: <https://gist.github.com/jcchurch/930276>.