

## ПРОЕКТУВАННЯ ОБЧИСЛЮВАЛЬНОГО КЛАСТЕРУ ДЛЯ УЧБОВОГО ПРОЦЕСУ

*Визначено особливості побудови гетерогенного обчислювального кластеру з метою подальшого його впровадження в учбовий процес. Запропоновано рішення основних проблем проектування обчислювального кластеру. Наведено ключові питання у проектуванні архітектури паралельної обчислювальної системи.*

**Ключові слова:** обчислювальний кластер, високопродуктивні обчислення, паралельні обчислення, BEOWULF-кластер, MPI, MPICH.

У сучасному світі все частіше доводиться працювати з великими обсягами даних у таких областях знань, як квантова фізика, квантова хімія, статистичні дані, аналіз соціальних зв'язків, криптоаналіз, комп'ютерне моделювання та інші. Для оперування такими обсягами потужностей персонального комп'ютера буде недостатньо, а використання суперЕОМ може стати не вигідним з фінансових причин. Альтернативою є використання обчислювальних кластерів - декількох незалежних персональних комп'ютерів, об'єднаних на програмному та апаратному рівнях. Саме тому дана тема є актуальною і може представляти інтерес для викладачів дисциплін, пов'язаних з паралельними обчисленнями, комп'ютерними мережами, студентів ІТ-спеціальностей та всіх зацікавлених в паралельних обчислювальних системах.

**Головною метою** даної роботи є аналіз та проектування архітектури обчислювального кластеру для проведення учбових занять, формування рішення фундаментальних питань при побудові кластеру, а також аналіз існуючих рішень обчислювальних систем.

Для проектування архітектури обчислювального кластеру необхідно вирішити ключові питання щодо типу обчислювального кластеру, операційної системи, на якій буде побудовано кластер, особливостей та ключових моментів побудови мережевої архітектури та інтерфейсів передачі повідомлень, що будуть застосовуватись.

При проектуванні кластеру слід враховувати наступні потреби:

- 1) використання неспеціалізованого апаратного забезпечення;
- 2) використання безкоштовного програмного забезпечення;
- 3) гетерогенність середовища, побудова кластеру на основі комп'ютерів з різними технічними конфігураціями.

**Проектований обчислювальний кластер** використовуватиметься для проведення учбових занять, а як правило, більшість комп'ютерних класів університетів обладнані персональними комп'ютерами, які мають різну апаратну конфігурацію. Це слід враховувати при побудові обчислювального кла-

стеру, тому прийнято рішення про побудову гетерогенного кластеру типу Beowulf.

Beowulf – це кластер, який складається з множини персональних комп'ютерів, об'єднаних в одну мережу, як правило, за допомогою Ethernet, та які використовують модель MPI для паралельних обчислень. Проект виник у науково-космічному центрі NASA, почався влітку 1994 року збіркою 16-процесорного кластера (на процесорах 486DX4/100 MHz, 16 MB пам'яті і трьох мережевих адаптерах на кожному вузлі, трьох «паралельних» Ethernet-кабелів по 10 Mbit). Даний кластер, який був названий «Beowulf», створювався як обчислювальний ресурс проекту Earth and Space Sciences Project (ESS) [1].

Кластер типу Beowulf зазвичай має один головний вузол і як мінімум один або більше другорядних вузлів, з'єднаних, як правило, за допомогою Ethernet. Головний вузол виконує функції керування всіма іншими вузлами кластеру, організує загальний доступ до фалів, але при цьому також бере участь у обчисленнях. Підлеглі вузли управляються та конфігуруються з головного вузла обчислювального кластеру. Дана система будується з готових, широко розповсюджених комп'ютерних комплектуючих, на яких може працювати ОС Linux, комутаторів та ліній зв'язку.

#### **Переваги кластерних систем.**

Переваги Beowulf-систем: масштабованість системи; вартість системи нижче вартості суперкомп'ютера; широка поширеність та доступність апаратного забезпечення; можливість використання комп'ютерів з застарілою конфігурацією, що є актуальним для навчальних закладів.

Таким чином, кластер типу Beowulf являється оптимальним рішенням для забезпечення навчальних закладів з обмеженими ресурсами продуктивною обчислювальною системою для вивчення розподілених обчислень, систем та паралельного програмування. Витрати на його установку в університетах є мінімальними, так як більшість університетів мають комп'ютерні класи, об'єднані однією мережею.

Одною із вимог проектування кластеру є використання відкритого та безкоштовного програмного забезпечення. Саме такою є операційна система Linux, що є найбільш популярною системою з відкритим кодом. Для операційної системи Linux доступна велика кількість системного програмного забезпечення, компіляторів, засобів розробки, бібліотек, математичного та іншого спеціалізованого програмного забезпечення.

При побудові кластеру обрано ОС Ubuntu, яка є найбільш популярним дистрибутивом Linux [2]. Ця система побудована із сильним акцентом на простоту встановлення та користування для простого користувача. Вона є добре документованою системою з сильною підтримкою ком'юніті.

ОС Ubuntu має серверні версії, які працюють повністю в консольному режимі та потребують мінімум вимог до апаратного забезпечення (500 MHz (x86), 64Mb ОЗП, 500Mb HDD), що для кластерних обчислень є оптимальним, так як можливості ПК будуть затрачатися не на відображення графічної складової системи, а на виконання корисної для користувача кластеру дії – обчислень. Але якщо планується використовувати вузли кластеру і для повсяк-

денних задач, слід обрати звичайну систему з графічним інтерфейсом користувача.

Структура ліній комутації між процесорами обчислювальної системи, її топологія, визначається, як правило, з урахуванням можливостей ефективною технічною реалізацією. Важливу роль при виборі структури мережі грає і аналіз інтенсивності інформаційних потоків при паралельному вирішенні найбільш поширених обчислювальних задач.

При виборі топології мережі для обчислювального кластеру з метою проведення навчальних занять слід враховувати наявну топологію, яка вже створена при побудові комп'ютерних класів в початкових закладах. В більшості випадків це топологія типу «зірка», у якій всі комп'ютери пов'язані через центральний елемент. Дана топологія є особливо ефективною при організації централізованих схем паралельних обчислень.

### Основні відомості про обчислювальний кластер типу Beowulf

Побудова обчислювального кластеру типу Beowulf передбачає об'єднання вузлів кластеру за допомогою мережі, використовуючи Ethernet-адаптери. Схема з'єднання вузлів кластеру типу Beowulf представлена на рис. 1.

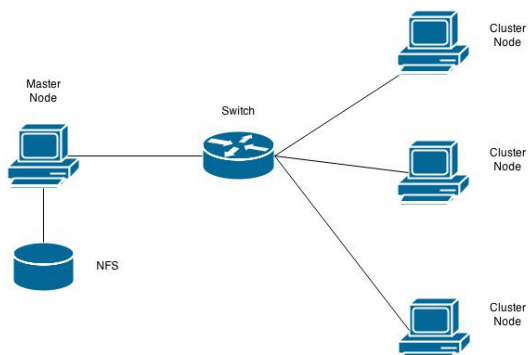


Рис. 1 – Схема мережі кластеру типу Beowulf

Швидкість доступу по мережі для обчислювальних кластерів, як правило, є його слабкою ланкою, так як швидкість пропускання каналу є набагато меншою, ніж швидкість багатьох обчислювальних елементів вузлів кластеру.

Для отримання максимальної ефективності від кластеру, особливо при роботі з великими обсягами даних, потрібно збільшувати пропускну здатність мережі шляхом використання каналів зв'язку зі швидкістю пропускання як мінімум 100 Мбіт/с та використовувати високошвидкісне мережеве обладнання.

Важливим моментом при проектуванні архітектури паралельного обчислювального кластеру є робота зі спільними даними. Прикладом таких даних може бути копія програми, що запускається, масив вхідних/вихідних та проміжних даних розрахунків. Існують різні механізми, що дозволяють виконувати подібний розподіл. В більшості випадків це різноманітні скрипти, які здійснюють синхронізацію каталогів на вузлах кластеру за допомогою команд SCP або rsync.

Для рішення даної задачі використано мережеву файлову систему NFS. NFS надає клієнтам прозорий доступ до файлів і файлової системи

сервера. На відміну від FTP, протокол NFS здійснює доступ тільки до тих частин файлу, до яких звернувся процес, і основна перевага його в тому, що він робить цей доступ прозорим [3].

Мережева файлова система NFS складається з двох компонентів: сервера і клієнта. Сервер здійснює мережевий доступ до каталогів базової файлової системи на основі визначених правил розмежування доступу. Клієнт використовується для підключення до спільних ресурсів кластеру.

Система NFS може бути організована як локально – на головному вузлі кластеру, так і на віддаленому файловому сховищі. Для створення обчислювального кластеру для навчальних потреб доцільніше використовувати саме локальну її організацію.

При побудові системи віддаленого виконання команд на обчислювальному кластері обрано безпарольний доступ між вузлами по протоколу SSH. Саме за допомогою цього протоколу буде здійснюватися віддалений запуск, конфігурування, адміністрування та моніторинг за віддаленими вузлами обчислювального кластеру.

Підтримка SSH реалізована у всіх UNIX-подібних операційних системах, і на більшості з них в числі стандартних утиліт присутні клієнти ssh. В операційній системі Ubuntu реалізовано ssh-клієнт (Open-SSH). SSH-сервер на кожен вузол кластеру слід встановлювати окремо.

При організації системи віддаленого виконання команд на всіх вузлах кластеру налаштовується ssh-сервер та ssh-клієнт.

Безпарольний доступ між вузлами кластеру організовується за допомогою методу створення публічного ключа на головному вузлі кластеру та додавання його вмісту в файл `authorized_keys`.

Так як сам по собі кластер являється тільки інструментом вирішення конкретних ресурсомістких задач, то серед обов'язкового програмного забезпечення обчислювального кластеру повинні бути набори компіляторів та бібліотек розробки, які необхідні для створення програм паралельних обчислень. Найбільш простими та оптимальним рішенням для установки на систему Linux є набір компіляторів GCC [6].

Для забезпечення зв'язку між окремими процесами паралельної програми, яка виконується на обчислювальному кластері, використовується стандарт передачі повідомлень MPI.

MPI – це специфікація, що розроблена в 1993–1994 роках групою MPI Forum [4], і забезпечує реалізацію моделі обміну повідомленнями між процесами. Остання версія даної специфікації – MPI-2. У моделі програмування MPI програма створює кілька процесів, що взаємодіють між собою за допомогою звертання до підпрограм прийому і передачі повідомлень.

Специфікація MPI забезпечує портативність програм на рівні вихідних кодів. Підтримується робота на гетерогенних кластерах і симетричних мультипроцесорних системах.

Важливою властивістю паралельної програми є детермінізм – програма повинна завжди давати той самий результат для того самого набору вхідних даних. Модель передачі повідомлень загалом даною властивістю не володіє, оскільки не визначений порядок одержання повідомлень від двох

процесів третім. Якщо ж один процес послідовно посилає кілька повідомлень іншому процесу, MPI гарантує, що одержувач одержить їхній саме в тому порядку, у якому вони відправлені. Відповідальність за забезпечення детермінованого виконання програми лягає на програміста [5].

Існує кілька реалізацій інтерфейсу MPI: OpenMPI, MPICH. Для побудови кластеру обрано бібліотеку MPICH за її поширеність та велику кількість документації. MPICH (MPI Chameleon) – одна з реалізацій MPI, яка підтримує роботу на великому числі платформ із різними комунікаційними інтерфейсами, у тому числі і з TCP/IP. Недоліком MPICH є неможливість запуску процесів під час роботи програми і відсутність засобів моніторингу поточного стану системи. Але вони усуваються за допомогою сторонніх програмних пакетів.

Створення експериментальної системи дозволить використати її при викладанні дисциплін: «Операційні системи», «Паралельні та розподілені системи», при виконанні наукових робіт та проектів.

#### Експериментальна частина.

Для проведення дослідження архітектурних рішень при побудові кластеру побудовано обчислювальний кластер на базі двох персональних комп'ютерів, об'єднаних в мережу за допомогою технології Fast Ethernet, під управлінням операційною системою Ubuntu Linux 14.04 LTS. з наступним технічними характеристиками:

- процесор: Intel Core i5-4440 3.1 Ghz;
- оперативна пам'ять: 2 x Kingston DDR3-1600 4096MB PC3-12800;
- жорсткий диск: WD Blue 1TB 7200rpm 64MB WD10EZEX 3.5 SATA III;
- системна плата: ASROCK B85 PRO4.

В ході тестування проведено, встановлено та налаштовано програмне забезпечення для обчислювального кластеру, протестовано їх швидкодію. Результати тестування продуктивності представлено в табл. 1 та рис. 2. Отримані результати збігаються з теоретичними відомостями про об'єднання обчислювальних ресурсів у кластер.

Таблиця 1 – Результати тестувань роботи кластера

Вузол кластера, мережеве ім'я	d-04				d-05				Заг. Продук класте- ра, MFlops	Час роз- ра- хун- ків, с
	1	2	3	4	5	6	7	8		
Кількість запущених процесів (задіяних ядер)										
Продуктивність, MFlops	1633	*	*	*	*	*	*	*	1633	1,10
	1629	1629	*	*	*	*	*	*	3258	0,55
	1592	1592	159 2	*	*	*	*	*	4778	0,38
	1532	1532	154 6	1546	*	*	*	*	6128	0,29
	1526	1526	153 7	1540	15 40	*	*	*	7631	0,24
	1537	1537	153 7	1537	16 38	163 8	*	*	9224	0,20

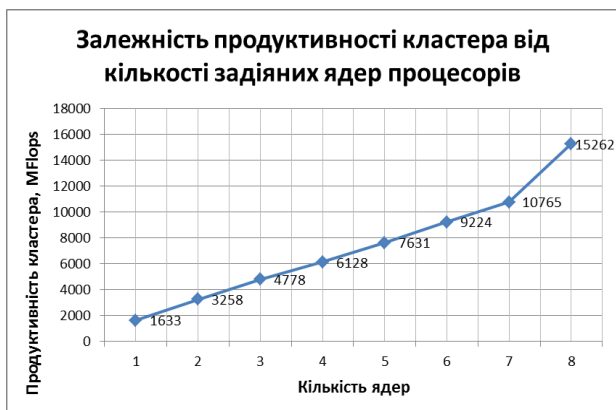


Рис. 2 – Графік залежності продуктивності кластера від кількості задіяних ядер

Результати тестувань показали, що кластер працює вірно та передбачувано, збоїв у його роботі не виявлено.

Під час тестування виявлено, що дана архітектура підходить для побудови обчислювального кластеру для навчальних потреб. Вона являє собою оптимальне рішення для побудови кластеру в навчальних закладах та в подальшому використанні його для практичних занять з дисциплін по паралельним системам, паралельним обчисленням, а також практичних задач розрахунків.

**Висновки.** В результаті виконаної роботи спроектовано архітектуру обчислювального кластеру для учбового процесу, сформовано рішення ключових питань при побудові кластеру. Вирішено основні питання типу обчислювального кластеру, операційної системи, особливостей та ключових моментів побудови мережевої архітектури, а саме: топології, мережевого забезпечення, мережевої файлової системи, системи віддаленого виконання команд, а також інтерфейсів передачі повідомлень, що будуть застосовуватись.

### Бібліографія

1. НОУ ИНТУИТ | Лекция | Принципы построения параллельных вычислительных систем [Електронний ресурс] – Режим доступу: <http://www.intuit.ru/studies/courses/1156/190/lecture/4942?page=5>
2. DistroWatch.com: Put the fun back into computing. Use Linux, BSD. [Електронний ресурс] – Режим доступу: <http://distrowatch.com/dwres.php?resource=popularity>
3. Network File System — Вікіпедія [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Network\\_File\\_System](https://uk.wikipedia.org/wiki/Network_File_System)
4. Message Passing Interface (MPI) Forum Home Page [Електронний ресурс] – Режим доступу: <http://www.mpi-forum.org/>
5. Message Passing Interface — Вікіпедія [Електронний ресурс] – Режим доступу: [http://uk.wikipedia.org/wiki/Message\\_Passing\\_Interface](http://uk.wikipedia.org/wiki/Message_Passing_Interface)
6. GCC, the GNU Compiler Collection - GNU Project - Free Software Foundation (FSF) [Електронний ресурс] – Режим доступу: <https://gcc.gnu.org/>