

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Обчислювальний кластер на основі відкритого програмного
забезпечення для надання хмарних сервісів»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

Денис ВОЛОЩУК

(підпис)

Ім'я, ПРІЗВИЩЕ здобувача

Виконав:
здобувач вищої освіти
група ІСДМ-64

Денис ВОЛОЩУК

Керівник:
науковий ступінь,
вчене звання

Віктор САГАЙДАК
доктор філософії

Рецензент:
науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

« ____ » _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ СТУДЕНТУ

Денис Волощук

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Обчислювальний кластер на основі відкритого програмного забезпечення для надання хмарних сервісів»

керівник кваліфікаційної роботи Віктор САГАЙДАК, доктор філософії
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «15» жовтня 2024 року №
320.

2. Строк подання кваліфікаційної роботи: 27 грудня 2024 року.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, основні властивості кластерів і їх види, MPI, задачі кластера для розрахунків, веб сайт для доступу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз напрямку кластеризації і їх типів
2. Аналіз операційних систем
3. Розробка веб-сайту для доступу до кластеру для проведення розрахунків

5. Перелік ілюстративного матеріалу: *презентація*

- Аналіз теми
- Аналіз ОС
- Робота з сайтом
- Перспективи подальшого розвитку проекту

6. Дата видачі завдання: 15 жовтня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	15.10 – 23.10.24	Виконав
2	Створення кластеру і веб-сайту для доступу	24.10 – 01.11.24	Виконав
3	Створення дизайну сторінки і її основних елементів	02.11 – 07.11.24	Виконав
4	Оформлення веб-сайту	08.11 – 16.11.24	Виконав
5	Додавання елементів сайту	17.11 – 21.11.24	Виконав
6	Тестування сайту	22.11 – 11.12.24	Виконав
7	Оформлення роботи: вступ, висновки, реферат	12.12 – 20.12.24	Виконав
8	Розробка демонстраційних матеріалів	21.12 – 25.12.24	Виконав

Здобувач вищої освіти

(підпис)

Денис ВОЛОЩУК
(Ім'я, ПРІЗВИЩЕ)

Керівник роботи
кваліфікаційної роботи

(підпис)

Віктор САГАЙДАК
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 81 стор., 21 рис., 1 табл., 15 джерел.

Мета роботи – проектування обчислювального кластеру і організація доступу до нього

Об’єкт дослідження – обчислювальний кластер на основі відкритого програмного забезпечення для надання хмарних сервісів

Предмет дослідження – інформаційна система для роботи кластеру.

Короткий зміст роботи: Було створено веб-сайт для ознайомлення з роботою, правилами та можливостями обчислювального кластеру. Створено код на основі HTML, та розроблено стиль оформлення. Було додано можливість реєстрації, отримання оплати для можливості розрахунків, додано перелік програмного забезпечення. Виконана робота по виставленню на безкоштовний хостинг для перегляду.

КЛЮЧОВІ СЛОВА: ХМАРНІ ОБЧИСЛЕННЯ, МЕТОД СКІНЧЕННИХ ЕЛЕМЕНТІВ, ОБЧИСЛЮВАЛЬНИЙ КЛАСТЕР, МАСШТАБУВАННЯ.

ABSTRACT

Text part of the qualification work for obtaining a master's degree: 81 pages, 21 figures, 1 tables, 15 sources.

Purpose of the work - design of a computing cluster and organisation access to it

Object of research - a computing cluster based on open source software for providing cloud services

Subject of research - an information system for the operation of the cluster.

Summary of the work: A website was created to introduce it work possibilities, rules and capabilities of the computing cluster. HTML-based code was created, and a design style has been developed. The ability to register, receive payment was added, a list of software was added. Has been complete posting on free hosting for viewing.

KEYWORDS: CLOUD COMPUTING, FINITE ELEMENT METHOD, COMPUTING CLUSTER, SCALING.

ЗМІСТ

ВСТУП.....	10
1 КОМП'ЮТЕРНИЙ КЛАСТЕР.....	13
1.1 Історія розвитку	13
1.2 Загальні відомості про обчислювальний кластер	15
1.3 Інші види розподілених обчислень	17
1.4 Типи кластерів	18
1.5 Інші випадки використання кластерів.....	19
1.4 Message Passing Interface (MPI).....	21
2 РОЗРОБКА КОМП'ЮТЕРНОГО КЛАСТЕРУ	24
2.1 Аналіз існуючих ОС	24
2.2 Система Linux для кластеру	25
2.3 Архітектура Linux-систем	25
2.4 Переваги і недоліки Linux	28
2.5 Організація зберігання даних	29
2.6 VirtualBox	39
2.6.1 Відкрита платформа віртуалізації VirtualBox.....	39
2.6.2 Огляд існуючих настільних платформ віртуалізації	41
2.6.3 Платформа VirtualBox.....	42
2.6.4 Емульоване апаратне оточення.....	44
2.6.5 Унікальні функції VirtualBox	45

2.6.6 Висновки і проміжні результати роботи.....	46
3 РОЗРОБКА ВЕБ САЙТУ	49
3.1 Мова HTML.....	49
3.1.1 Електронний гіпертекстовий документ	49
3.1.1 Інтеграція JavaScript.....	51
3.1.2 Розширені функції	52
3.1.3 Гіпертекст та Інтернет.....	53
3.1.4 Історія HTML	57
3.1.2 Основні поняття мови HTML.....	59
3.2 CSS	61
3.2.1 Структура CSS правил	62
3.2.2 Синтаксис CSS	62
3.3 Figma	64
3.4 Мова програмування PHP.....	68
3.4.1 Застосування	68
3.5 Опис роботи сайту та інструменти для розробки	73
3.5.1 Дизайн сайту	73
3.5.2 Наповнення сайту	74
3.6 Подальші плани	77
ВИСНОВОК	78
ПЕРЕЛІК ПОСИЛАНЬ	79

ВСТУП

Актуальність теми. Розвиток технологій для фізичних розрахунків міцності та потоків газів і рідин, та їх взаємодії з твердими тілами в динаміці вимагає постійного нарощування обчислювальних можливостей.

Одним з шляхів розвитку нарощування обчислювальних можливостей є кластеризація, де потужності двох або набагато більше комп'ютерів поєднуються.

На відміну від корпоративних кластерів – домашній кластер вимагає ще відповідати і балансуванню між можливостями і енергоефективністю. Що також критично важливо в теперішні часи в Україні як з огляду на надійність подачі електроенергії так і за ціну за кВт.

Актуальність теми стала важливою для автора даної роботи як його додаткова діяльність і завдання, які пов'язані з математичним і фізичним розрахунком, аналізом конструкції і взаємодією між її елементами [8], а також використання серверу для інших користувачів як на платній так і на безплатній основі.

Мета роботи. Дослідження можливості побудови і запуску обчислювального кластеру та надання доступу іншим користувачам до нього в періоди не виконання ним основних задач автора.

Для виконання поставленої мети, у магістерській роботі розроблено та виконано наступні завдання:

- дослідження кластерних систем, їх видів та типів;
- аналіз ПО для організації роботи кластера;
- аналіз видів організації HDD/SSD масивів для роботи і зберігання даних;

- аналіз програмного забезпечення для розробки сайту для організації доступу до кластера ;
- безпосередня побудова і робота кластеру.

Об'єкт дослідження. Процес побудови кластеру і сайту для доступу до нього.

Предмет дослідження. Архітектура кластерів і ПО для його роботи.

Методи дослідження. Під час виконання завдань магістерської кваліфікаційної роботи були використані методи елементів системно-аналітичного аналізу, методи теоретичного дослідження, імітаційне та моделювання та фізичні процеси.

Джерела дослідження:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/7.0_release_notes/chap-red_hat_enterprise_linux-7.0_release_notes-clustering#chap-Red_Hat_Enterprise_Linux-7.0_Release_Notes-Clustering;
- https://www.blasbenito.com/post/01_home_cluster/
- <https://www.clearlinux.org/clear-linux-documentation/tutorials/hpc.html>

Наукова новизна одержаних результатів. У магістерській роботі розглянуто всі можливі способи організації обчислювальних кластерів та зроблено практичні висновки і рекомендації щодо його роботи

Практична значущість одержаних результатів. Отримані в магістерській роботі результати будуть використані при побудові кластеру з двох машин для обчислення, з подальшим можливим розширенням кількості вузлів.

Апробація результатів магістерської роботи. Базові результати магістерської роботи опубліковано в збірнику тез:

- II Всеукраїнська Науково-Технічна Конференція Технологічні горизонти: Дослідження та застосування інформаційних технологій для технологічного прогресу України і світу.
- Науково-практична конференція «Проблеми комп'ютерної інженерії».

1 КОМП'ЮТЕРНИЙ КЛАСТЕР

1.1 Історія розвитку

Вступ і перші спроби (1960-1980-ті).

Концепція кластерних обчислень виникла в 1960-х роках, коли дослідники шукали шляхи подолання обмежень перших мейнфреймів. Перші кластерні комп'ютери були побудовані з використанням масивів взаємопов'язаних комп'ютерів, але ці перші системи були обмежені низькою швидкістю зв'язку та ненадійними з'єднаннями.

Розвиток високопродуктивних обчислень (1980-1990-ті роки).

У 1980-х і 1990-х роках відбувся розвиток високопродуктивних обчислень, включаючи використання паралельної обробки та розподілених обчислень для вирішення складних завдань. У цю епоху також були розроблені більш просунуті технології взаємозв'язку, такі як інтерфейс передачі повідомлень (MPI), який забезпечував швидший і надійніший зв'язок між комп'ютерами в кластері.

Поточні досягнення та використання (2000-ті – тепер)

Сьогодні кластерні обчислення перетворилися на потужний інструмент для вирішення складних проблем у таких галузях, як наукові обчислення, інженерія, фінанси та біомедицина. Сучасні кластерні комп'ютери здатні обробляти величезні обсяги даних і виконувати складне моделювання на високій швидкості, що дає змогу вирішувати проблеми, які раніше були нерозв'язними.

Деякі з найбільш помітних досягнень кластерних обчислень включають:

- Моделювання клімату та прогноз погоди
- Розвідка нафти і газу

- Секвенування геному та відкриття ліків
- Фінансове моделювання та управління ризиками
- Наукове моделювання, як-от моделювання великомасштабної структури Всесвіту

Кластерні обчислення є критично важливим інструментом для вирішення складних проблем і стимулювання інновацій у багатьох галузях. Його здатність обробляти величезні обсяги даних і виконувати складне моделювання з високою швидкістю робить його важливим інструментом для дослідників, науковців і компаній.

У кластерних обчисленнях використовується багато популярних пакетів з відкритим вихідним кодом, які дозволяють користувачам створювати кластери комп'ютерів і керувати ними для паралельних обчислень.

Приклади пакетів із відкритим вихідним кодом разом із їх призначенням:

Open MPI (інтерфейс передачі повідомлень)

Open MPI — це широко використовувана реалізація стандарту інтерфейсу передачі повідомлень (MPI) із відкритим кодом. Це дозволяє користувачам розробляти та запускати паралельні програми на кластері комп'ютерів, забезпечуючи інтерфейс високого рівня для зв'язку між процесами.

Rocks Cluster Distribution

Rocks Cluster Distribution — це дистрибутив на базі Linux, розроблений спеціально для кластерних обчислень. Він надає повний стек програмного забезпечення для створення та керування кластерами, включаючи керування системою, керування кластерами та інструменти керування додатками.

Slurm.

Slurm — це менеджер ресурсів із відкритим кодом і планувальник завдань для кластерних обчислень. Це дозволяє користувачам керувати ресурсами, розподіляти вузли та планувати завдання в кластері, надаючи простий і ефективний спосіб керування паралельними обчислювальними завданнями.

PBS (портативна пакетна система)

PBS — це система пакетної обробки з відкритим кодом для кластерних обчислень. Він забезпечує гнучкий і масштабований спосіб керування та виконання завдань у кластері, включаючи підтримку паралельної обробки та розподілу ресурсів.

LSF (система розподілу навантаження)

LSF — це система керування робочим навантаженням із відкритим кодом для кластерних обчислень. Він надає комплексний набір інструментів для планування завдань, керування ресурсами та моніторингу паралельних обчислювальних завдань у кластері.

Данні пакети з відкритим вихідним кодом надають ряд інструментів для керування та виконання паралельних обчислювальних завдань у кластері, що полегшує користувачам використання переваг кластерних обчислень. Використовуючи ці інструменти, користувачі можуть створювати та керувати ефективними, масштабованими та надійними кластерами для своїх обчислювальних потреб.

1.2 Загальні відомості про обчислювальний кластер

Кластерні обчислення — це тип обчислень, де кілька комп'ютерів з'єднані, щоб вони працювали разом як єдина система. Термін «кластер»

відноситься до мережі пов'язаних комп'ютерних систем, запрограмованих для виконання одного завдання.

Обчислювальні кластери зазвичай складаються із серверів, робочих станцій і персональних комп'ютерів (ПК), які спілкуються через локальну мережу (LAN) або глобальну мережу (WAN).

Кластерні обчислення — це вид розподілених обчислень, тип обчислень, який об'єднує комп'ютери в мережу для виконання обчислювального завдання, збільшення обчислювальної потужності та функціонування як єдиний комп'ютер. Кожен комп'ютер або «вузол» у комп'ютерній мережі має операційну систему (ОС) і ядро центрального процесора (ЦП), яке виконує завдання, необхідні для належної роботи програмного забезпечення.

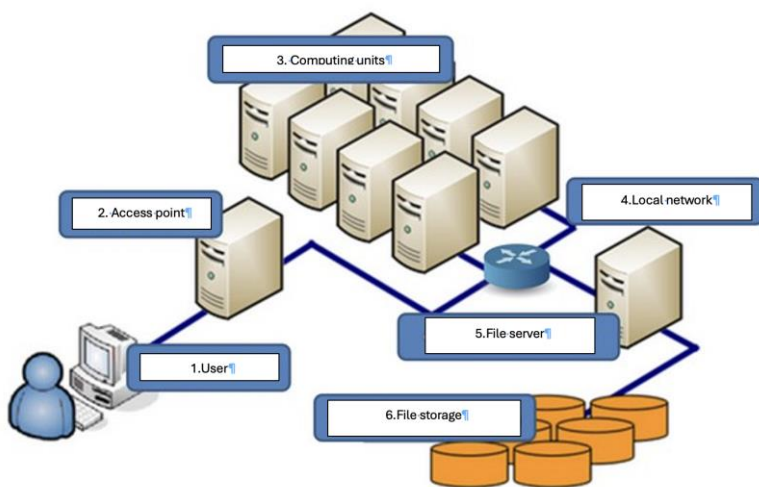


Рисунок 1.1– Схема кластеру

1.3 Інші види розподілених обчислень

Окрім кластерних обчислень, існує два інших широко використовуваних типи розподілених обчислень, які також включають об'єднані мережі комп'ютерів: Grid-обчислення та однорангові обчислення.

Grid -обчислення: в інформатиці інфраструктура Grid -обчислень створена для об'єднання обчислювальних ресурсів, які знаходяться в різних фізичних місцях. Доступні обчислювальні ресурси з різних машин об'єднуються та використовуються разом для вирішення проблеми. Як і кластеризація, Grid -обчислення використовують ресурси кількох взаємопов'язаних комп'ютерів.

Однак, на відміну від кластеризації, використовуються лише невикористовувані ресурси на комп'ютерах, підключених через мережеву архітектуру. SETI, Пошук позаземного розуму, був відомим прикладом мережевих обчислень, де невикористані обчислювальні ресурси багатьох комп'ютерів використовувалися для аналізу радіосигналів із глибокого космосу на наявність ознак позаземного життя.

Однорангові обчислення: однорангові (P2P) обчислення або мережа вимагають, щоб два або більше комп'ютерів були з'єднані як «однорангові» в мережі, тобто вони мають однакову потужність і права. На відміну від кластерних обчислень, архітектура P2P не потребує централізованого підходу до керування.

У мережі P2P кожен вузол діє як клієнтська машина (комп'ютер, якому потрібен доступ до служби), і сервер (комп'ютер, який надає послугу). Кожен одноранговий вузол надає ресурси іншим користувачам мережі, включаючи сховище, пам'ять, пропускну здатність тощо.

1.4 Типи кластерів

Обчислювальні кластери сильно відрізняються як за складністю, так і за призначенням. Відносно прості двовузлові кластери, наприклад, з'єднують лише кілька комп'ютерів, тоді як суперкомп'ютер «Аврора», з іншого боку, з'єднує понад 10 000.

Кластери мають багато бізнес-випадків використання через їх високу продуктивність, масштабованість і гнучкість, але вони також використовуються університетами та медичними школами для наукових досліджень. За своїми характеристиками обчислювальні кластери поділяються на три різні типи: висока доступність, балансування навантаження та висока продуктивність.

1) Кластери високої доступності.

Кластери високої доступності автоматично передають завдання з вузла, який несподівано вийшов з ладу, на інший вузол у мережі, який все ще функціонує. Ця здатність швидко та легко адаптуватися, коли один вузол виходить з ладу, робить їх ідеальними для робочих навантажень, де критично важливо уникнути переривання обслуговування.

2) Кластери балансування навантаження

Кластери з балансуванням навантаження, або просто «балансувальники навантаження», забезпечують рівномірно-справедливий розподіл роботи між вузлами в кластері. Без балансування навантаження вузли були б перевантажені завданнями, які їм були призначені, і частіше виходили з ладу. Існують різні види балансування навантаження для різних цілей. Один із найвідоміших — віртуальний сервер Linux — є безкоштовним із відкритим кодом і використовується для розробки високопродуктивних високодоступних серверів на основі технології кластеризації.

3) Високопродуктивні обчислювальні кластери

Кластери HPC — це мережа потужних процесорів, які можуть обробляти масивні багатовимірні набори даних, також відомі як великі дані, з надзвичайно високою швидкістю. Їм потрібні потужні мережі та наднизька затримка для переміщення файлів між вузлами.

На відміну від кластерів балансування навантаження та високої доступності, кластери HPC мають більшу потужність обробки та спеціально розроблені для аналізу даних, наприклад діагностики захворювань, аналізу величезних обсягів фінансових даних і секвенування геномів. Крім того, кластери HPC використовують інтерфейс передачі повідомлень (MPI), протокол для архітектур паралельних обчислень, який забезпечує зв'язок між вузлами.

4) Кластери штучного інтелекту

Кластери AI — це обчислювальні кластери, спеціально створені для робочих навантажень AI та ML, таких як розпізнавання обличчя та мови, обробка природної мови (NLP) і автономне водіння. Кластери ШІ спеціально розроблені для алгоритмів, на яких навчаються моделі ШІ.

1.5 Інші випадки використання кластерів.

Обчислювальні кластери в першу чергу розроблені, щоб бути більш продуктивними та надійнішими, ніж інші типи обчислювальних архітектур, що робить їх незамінними для сучасного домашнього офісу або наукової діяльності. Наприклад, сучасні кластери мають вбудовану відмовостійкість, термін, який стосується їхньої здатності продовжувати функціонувати навіть у разі збою одного вузла в мережі.

Крім того, великі комп'ютерні кластери покладаються на розподілені файлові системи (DFS) і резервний масив незалежних дисків (RAID), що

дозволяє зберігати однакові дані в різних місцях на кількох жорстких дисках. Кластерні обчислення приносять багато переваг деякі з яких є критично важливими в темі даної роботи і задач автора:

1) Продуктивність

Через свою залежність від паралелізму комп'ютерні кластери є високопродуктивними та зазвичай можуть швидше обробляти дані та виконувати більші робочі навантаження, ніж один комп'ютер.

2) Надійність

Кластерні обчислення вважаються високонадійними через використання технологій DFS і RAID. У комп'ютерному кластері, навіть якщо один вузол виходить з ладу, мережа продовжує працювати, а DFS і RAID продовжуватимуть забезпечувати резервне копіювання даних у кількох місцях.

3) Доступність

Окрім високої надійності, кластерні обчислення також вважаються високодоступними завдяки своїй здатності швидко відновлюватися після збою одного вузла. Якщо кластер працює належним чином, коли один вузол виходить з ладу, його робота плавно передається іншому в кластері без перерви в обслуговуванні.

4) Масштабованість

Кластерні обчислення мають високу масштабованість, оскільки вузли кластера можна додавати в будь-який час для підвищення продуктивності. Можливість динамічно налаштовувати ресурси в кластері означає, що кластер може збільшуватися або зменшуватися залежно від попиту.

5) Вартість

Кластерні обчислення економічно ефективніші, ніж інші види обчислень. Автор покладається на кластерні обчислення для підвищення продуктивності, масштабованості та доступності своєї ІТ-інфраструктури за доступною ціною, в тому числі використання вживаних деталей і систем.

1.4 Message Passing Interface (MPI)

Щоб скористатися більш ніж одним вузлом на завдання й таким чином отримати прискорення, потрібно зробити додаткове програмування. Натомість завдання може виконуватися на кількох процесорах, які комунікують через локальну мережу.

MPI — це незалежний від мови протокол зв'язку, який використовується для програмування паралельних комп'ютерів. Існує кілька реалізацій MPI, але більшість із них складається з набору процедур, які можна викликати з Fortran, C та C++, а також будь-якої мови, яка може взаємодіяти з їхніми бібліотеками. MPI дуже портативний і, як правило, оптимізований для апаратного забезпечення, на якому він працює, тому буде достатньо швидким.

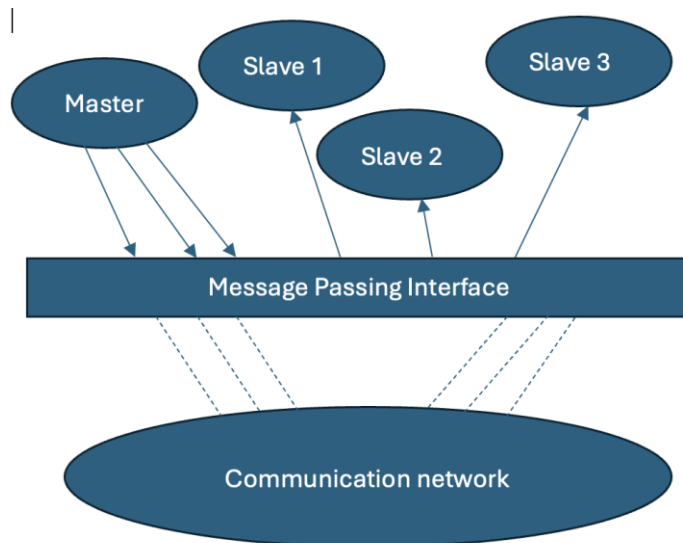


Рисунок 1.2— Схема роботи MPI

Програми, які розпаралелюються, має бути досить легко конвертувати в програми MPI, додавши до них процедури MPI.

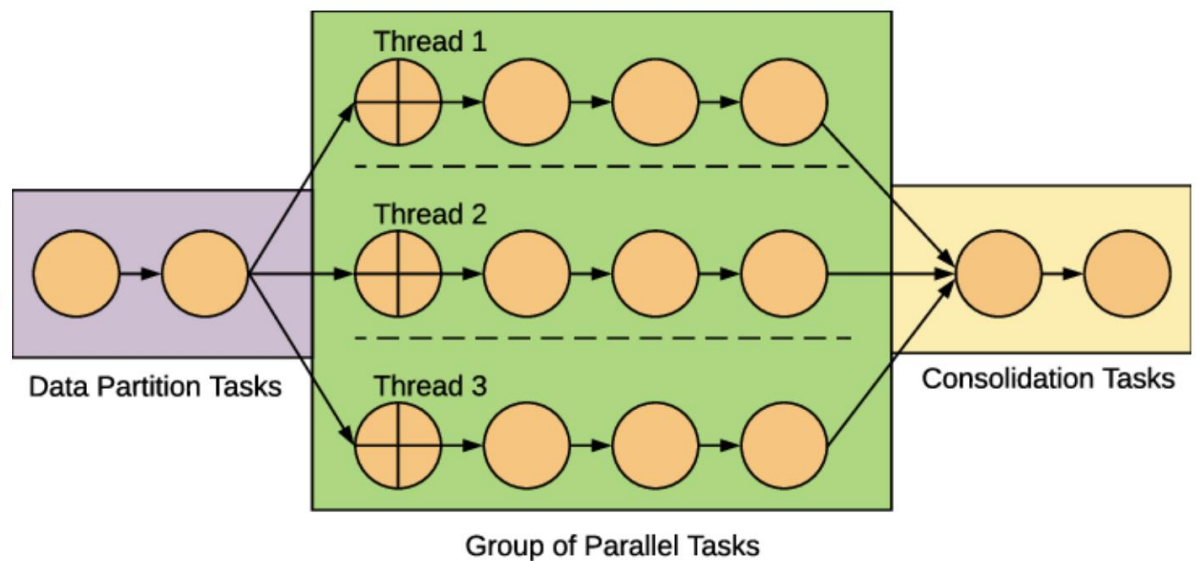


Рисунок 1.3– Розпаралелювання задач

Для того, щоб задача була розпаралелюваною, її має бути можливо розбити на менші частини, які можна розв'язувати незалежно одна від одної, а потім об'єднувати.

Загалом у паралельній програмі відбувається наступне: є «головний» процес, який контролює розділення даних тощо та передачу їх одному чи кільком «підлеглим» процесам, які виконують обчислення. Потім «підлеглі» процеси надсилають результати назад «головному». Він поєднує їх і/або, можливо, надсилає додаткові підрозділи проблеми, яку потрібно вирішити.

Приклади паралельних задач:

- Сортування
- Візуалізація комп'ютерної графіки

- Комп'ютерне моделювання, що порівнює багато незалежних сценаріїв, наприклад кліматичні моделі
- Множення матриць.

2 РОЗРОБКА КОМП'ЮТЕРНОГО КЛАСТЕРУ

2.1 Аналіз існуючих ОС

Кластерні обчислення передбачають об'єднання групи комп'ютерів (відомих як вузли), щоб вони могли працювати як єдина система. Основною метою кластерних обчислень є забезпечення високої доступності, масштабованості та паралельної обробки.

Linux: Linux має високу конфігурацію та відкритий код, що робить його популярним вибором для кластерних обчислень. Він підтримує різноманітні методи кластеризації, включаючи високоефективну кластеризацію (HPC), кластеризацію з балансуванням навантаження та кластеризацію високої доступності. Такі інструменти, як Beowulf, MOSIX і MPI (інтерфейс передачі повідомлень), використовуються в кластерних обчисленнях на базі Linux.

Windows: Windows підтримує деякі форми кластеризації, вона не така гнучка чи ефективна, як Linux для цієї мети. Версії Windows Server мають функції для відмовостійкої кластеризації та балансування мережевого навантаження, але вони не широко використовуються у високопродуктивних обчислювальних середовищах.

MacOS: MacOS зазвичай не використовується для кластерних обчислень. Хоча технічно можливо створити кластер із комп'ютерами Mac, це непрактично через вартість обладнання та обмеження операційної системи.

Android: Android — це мобільна операційна система, яка не призначена для кластерних обчислень. Їй бракує необхідних функцій і можливостей для ефективного керування та координації кластера комп'ютерів.

Підсумовуючи, хоча технічно можливо налаштувати кластерні обчислення на різних операційних системах, Linux є найбільш придатним і широко використовуваним для цієї мети завдяки своїй гнучкості, можливості конфігурації та наявності різноманітних інструментів і ресурсів.

2.2 Система Linux для кластеру

Linux — це операційна система з відкритим кодом, яка є альтернативою пропрієтарним операційним системам, таким як Microsoft Windows і macOS. Спочатку вона була розроблена Лінусом Торвальдсом як хобі в 1991 році. З тих пір вона перетворилася на потужну та універсальну операційну систему, яка використовується в усьому світі. Однією з ключових особливостей Linux є його модель розповсюдження, коли різні організації та спільноти створюють власні версії Linux, які називаються дистрибутивами або дистрибутивами. Деякі популярні дистрибутиви Linux включають Ubuntu, Fedora, Debian і CentOS..

2.3 Архітектура Linux-систем

В цьому розділі розглянемо архітектуру Linux систем

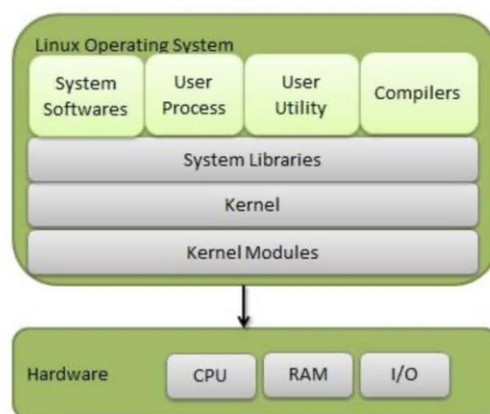


Рисунок 2.1 – Структура Linux

Linux має монолітну структуру ядра, де ядро діє як основний компонент операційної системи, керуючи апаратними ресурсами, надаючи низькорівневі послуги та полегшуючи зв'язок між програмним і апаратним забезпеченням. Він виконує такі завдання, як керування процесами, керування пам'яттю, драйвери пристроїв і доступ до файлової системи.

Окрім ядра, дистрибутиви Linux включають різні програмні компоненти, такі як бібліотеки, утиліти та графічні інтерфейси, для створення повної операційної системи. Ці компоненти працюють разом, щоб створити зручне середовище та підтримувати широкий спектр програм.

У Linux використовується багатокористувацька багатозадачна модель, що дозволяє декільком користувачам запускати кілька процесів одночасно. Він використовує ієрархічну файлову систему, подібну до інших Unix-подібних операційних систем, де файли організовані у вигляді дерева.

Є кілька вагомих причин використовувати Linux як операційну систему:

Природа з відкритим кодом: вихідний код Linux є у вільному доступі, що дозволяє користувачам переглядати, змінювати та поширювати його відповідно до їхніх вимог. Ця відкритість сприяє прозорості, безпеці та інноваціям.

Стабільність і надійність: Linux відомий своєю стабільністю і надійністю. Він широко використовується в серверах і критично важливих системах завдяки своїй здатності працювати протягом тривалого часу, не вимагаючи перезавантаження. Це робить його популярним вибором як для особистого, так і для корпоративного використання.

Гнучкість і налаштування: Linux пропонує високий ступінь гнучкості та налаштування. Користувачі можуть вибирати з широкого діапазону настільних середовищ, пакетів програмного забезпечення та системних

конфігурацій, адаптуючи свій досвід роботи з Linux відповідно до своїх потреб.

Безпека: Linux за своєю суттю безпечний, головним чином завдяки своїм принципам дизайну та спільному характеру спільноти з відкритим кодом. Швидкі виправлення помилок, регулярні оновлення безпеки та можливість перевірки вихідного коду сприяють репутації Linux як безпечної операційної системи.

Величезна екосистема програмного забезпечення: хоча Linux може не мати такої кількості програмного забезпечення, як деякі інші операційні системи, вона пропонує багату та різноманітну екосистему програм. Багато популярних програмних інструментів і фреймворків, таких як інструменти GNU, веб-сервер Apache і мова програмування Python, мають коріння в екосистемі Linux.

Linux відрізняється від інших операційних систем кількома параметрами:

Природа з відкритим кодом: на відміну від пропрієтарних операційних систем, Linux є відкритим кодом, тобто її вихідний код є у вільному доступі. Це дозволяє користувачам налаштовувати та модифікувати операційну систему відповідно до своїх потреб, сприяючи співпраці та інноваційній екосистемі.

Архітектура ядра: Linux має монолітну архітектуру ядра, де ядро виконує більшість функцій операційної системи. Навпаки, деякі інші операційні системи, такі як macOS, використовують гібридну архітектуру ядра, яка поєднує в собі монолітні та мікроядерні елементи.

Модель розповсюдження програмного забезпечення: дистрибутиви Linux, як правило, керуються спільнотою та надають широкий спектр програмних пакетів через централізовані системи керування пакетами. Це

дозволяє легко встановлювати, оновлювати програмні додатки та керувати ними.

Підтримка апаратного забезпечення: Linux має репутацію чудової підтримки апаратного забезпечення з широким набором драйверів, доступних для різних компонентів апаратного забезпечення. Він може працювати на різноманітних апаратних платформах, від вбудованих систем до суперкомп'ютерів, що робить його надзвичайно універсальним.

Філософія та модель розробки: розробка Linux відбувається за спільною та розподіленою моделлю. Його керує велика спільнота розробників у всьому світі, які роблять внесок у його постійне вдосконалення. Цей підхід, орієнтований на громаду, заохочує прозорість, інновації та спільний доступ до знань.

2.4 Переваги і недоліки Linux

Переваги:

- 1) Безкоштовність
- 2) Можливості налаштування
- 3) Простота встановлювання
- 4) Безпека
- 5) Невибагливість до ресурсів
- 6) Драйвери обладнання
- 7) Командна строка
- 8) Встановлення програм з репозиторію
- 9) Можливість налаштування графічного інтерфейсу
- 10) Продумана файлова система
- 11) Система зберігання налаштувань
- 12) Підтримка великої кількості архітектури

13) Відсутність збору даних

14) Набір безкоштовних програм

Недоліки:

1) Складність засвоєння

2) Відсутність версій популярних програм

3) Відсутність підтримки деяких програм та наявність драйверів для обладнання

2.5 Організація зберігання даних

RAID (надлишкові масиви незалежних дисків) — це техніка, яка передбачає використання комбінації кількох дисків для зберігання даних замість використання одного диска для підвищення продуктивності, резервування даних або для захисту даних у разі збою диска. Термін був визначений Девідом Паттерсоном, Гартом А. Гібсоном і Ренді Кацом з Каліфорнійського університету в Берклі в 1987 році. У цьому розділі буде детально розглянуто RAID і типи RAID, їх переваги та недоліки.

RAID (надлишковий масив незалежних дисків) — це як резервні копії важливих файлів, які зберігаються в різних місцях на кількох жорстких дисках або твердотільних накопичувачах (SSD). Якщо один диск перестає працювати, ваші дані залишаються в безпеці, оскільки на інших дисках зберігаються інші копії. Це схоже на систему безпеки, яка захищає файли від втрати в разі поломки одного з дисків.

RAID (надлишковий масив незалежних дисків) у системі керування базами даних (СУБД) — це технологія, яка об'єднує кілька фізичних дисків в один логічний блок для зберігання даних. Основною метою RAID є підвищення надійності, доступності та продуктивності даних. Існують різні рівні RAID, кожен з яких пропонує баланс цих переваг.

Для організації RAID-масиву необхідно мати спеціальний контролер.

RAID-контролер працює між операційною системою комп'ютера та реальними жорсткими дисками, організовуючи їх у групи, щоб полегшити керування ними. Це допомагає пришвидшити швидкість читання та запису даних комп'ютером, а також додає рівень захисту на випадок поломки одного з жорстких дисків.

Існує три типи контролерів RAID.

На основі апаратного забезпечення: в RAID на основі апаратного забезпечення існує фізичний контролер, який керує всім масивом. Цей контролер може обслуговувати всю групу жорстких дисків разом. Він розроблений для роботи з різними типами жорстких дисків, наприклад SATA (Serial Advanced Technology Attachment) або SCSI (Small Computer System Interface). Іноді цей контролер вбудовано прямо в головну плату комп'ютера, що полегшує налаштування та керування системою RAID.

На основі програмного забезпечення: у RAID на основі програмного забезпечення контролер не має власного спеціального обладнання. Тому для виконання своєї роботи він використовує головний процесор і пам'ять комп'ютера. Він виконує ту саму функцію, що й апаратний контролер RAID, наприклад, керує жорсткими дисками та зберігає ваші дані в безпеці. Але через те, що він ділиться ресурсами з іншими програмами на комп'ютері, він може працювати не так швидко.

На основі мікропрограми: RAID-контролери на основі мікропрограми схожі на додаткову опцію що вбудовані в головну плату комп'ютера. Вони працюють з головним процесором, як і програмний RAID. Але вони реалізуються лише під час запуску комп'ютера. Коли операційна система запущена, спеціальний драйвер бере на себе завдання RAID. Ці

контролери не такі дорогі, як апаратні, але вони посилюють роботу головного процесора комп'ютера.

Надмірність даних, хоча й займає додатковий простір але підвищує надійність диска. Це означає, що у випадку збою диска, якщо ті самі дані також зберігаються на іншому диску, ми можемо отримати дані та продовжити операцію. З іншого боку, якщо дані розподіляються на кількох дисках без технології RAID, втрата одного диска може вплинути на всі дані.

Ключові моменти оцінки системи RAID.

Надійність: скільки несправностей диска може витримати система.

Доступність: яку частину загального часу сеансу займає система в режимі безвідмовної роботи, тобто наскільки система доступна для фактичного використання.

Продуктивність: наскільки хороший час відповіді? Наскільки висока пропускна здатність (швидкість роботи обробки).

Ємність: задано набір з N дисків, кожен з B блоків, який обсяг корисної ємності доступний для користувача.

RAID дуже прозорий для основної системи. Це означає, що для хост-системи він виглядає як один великий диск, який представляє себе як лінійний масив блоків. Це дозволяє замінити старі технології на RAID, не вносячи занадто багато змін у існуючий код.

Типи RAID масивів

RAID 0 (чергування) і його оцінка.

Надійність: 0

Немає дублювання даних. Отже, втрачений блок неможливо відновити.

Місткість: $N \times B$

Весь простір використовується для зберігання даних. Оскільки немає дублювання, повністю використовується N дисків, кожен з яких має B блоків.

Переваги:

- Легко реалізувати.
- Він краще використовує ємність накопичувача.

Недоліки:

- Одна втрата накопичувача може призвести до повного виходу системи з ладу.
- Це невдалий вибір для критичної системи.

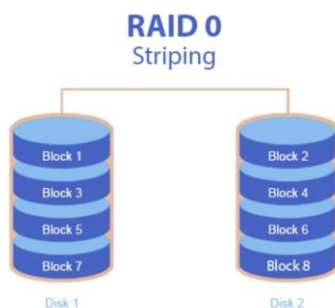


Рисунок 2.2 – Схема RAID 0

RAID 1 (дзеркалювання) і його оцінка.

Більш ніж одна копія кожного блоку зберігається на окремому диску. Таким чином, кожен блок має дві (і більше) копії, що лежать на різних дисках.

Надійність: від 1 до $N/2$

Збій 1 диска можна напевно вирішити, оскільки блоки цього диска матимуть дублікати на іншому диску. Якщо нам пощастить і диски 0 і 2 вийдуть з ладу, це знову ж таки можна вирішити, оскільки блоки цих дисків мають дублікати на дисках 1 і 3. Отже, у найкращому випадку можна

впоратися з відмовами диска $N/2$ в випадку використання більше ніж двох дисків.

Місткість: $N \times B/2$

Для зберігання даних використовується лише половина простору.

Інша половина - це лише дзеркало вже збережених даних.

Переваги:

- Він охоплює повне резервування.
- Це може підвищити безпеку даних і швидкість.

Недоліки:

- Ціна.
- Ємність зберігання менше.

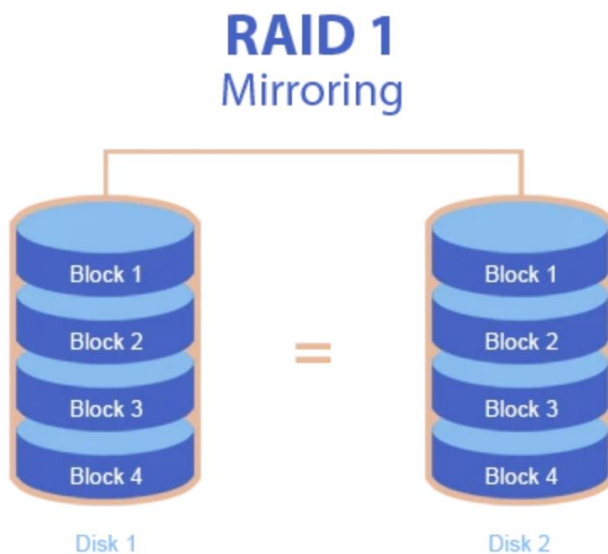


Рисунок 2.3 – Схема RAID 1

RAID 5 (чергування з розподіленою парністю)

Це невелика модифікація системи RAID-4 (яка в даній роботі не розглядалася), де єдина відмінність полягає в тому, що парність обертається між дисками.

На малюнку ми можемо помітити, як «обертається» біт парності.
Це було введено, щоб покращити продуктивність випадкового запису.

Надійність: 1

RAID-5 дозволяє відновити щонайбільше 1 збій диска (через спосіб роботи парності). Якщо більше ніж один диск виходить з ладу, неможливо відновити дані. Це ідентично RAID-4.

Ємність: $(N-1) \times B$

Загалом для зберігання паритету використовується простір, еквівалентний одному диску. Отже, $(N-1)$ дисків доступні для зберігання даних, кожен диск має B блоків.

Переваги:

- Дані можна реконструювати за допомогою бітів парності.
- Це покращує продуктивність.

Недоліки

- Його технологія складна і вимагає додаткового простору.
- Якщо обидва диски будуть пошкоджені, дані будуть втрачені назавжди.

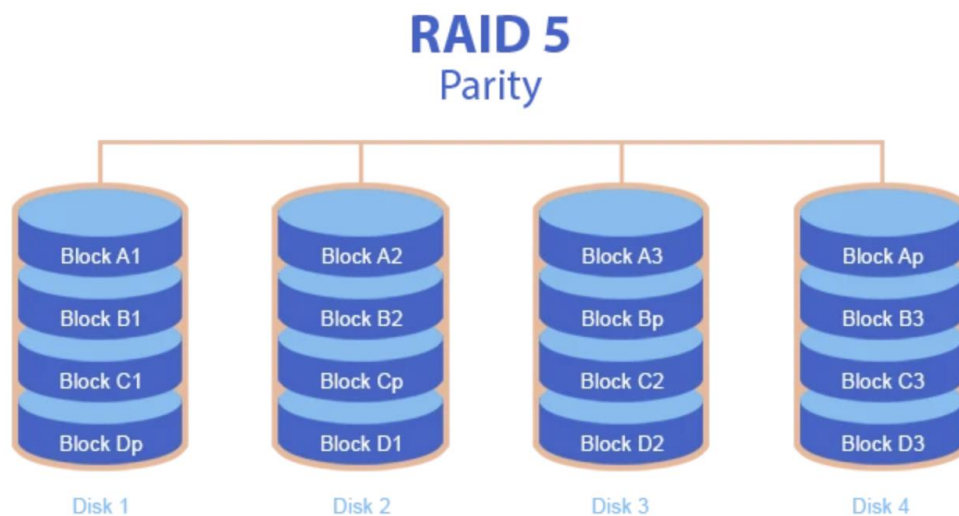


Рисунок 2.4 – Схема RAID 5

RAID 6 (чергування з подвійною парністю)

Raid-6 допомагає, коли є збій кількох дисків. Пара незалежних паритетів генерується та зберігається на кількох дисках на цьому рівні. В ідеалі для цього рівня потрібно чотири диски.

Існують також гібридні RAID, які використовують більше одного рівня RAID, вкладених один за одним, для виконання конкретних вимог.

Переваги:

- Дуже висока доступність даних.
- Швидке читання транзакцій даних.

Недоліки:

- Через подвійний паритет він має повільні транзакції запису даних.
- Потрібне додаткове місце.

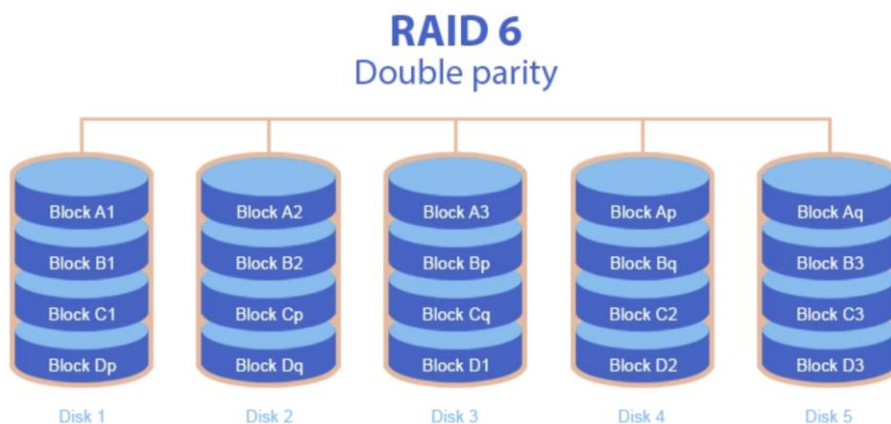


Рисунок 2.5 – Схема RAID 6

2.

JBOD-масив

JBOD (розшифровується як «Just a Bunch Of Disks») — це сукупність жорстких дисків які не налаштовані на роботу як надлишковий масив незалежних дисків (масив RAID). Кілька дисків у масиві з'єднані з одним сервером, який забезпечує більший обсяг пам'яті.

JBOD роблять диск таким, що він виглядає єдиним, об'єднуючи всі диски. Як і в системі RAID, дані зберігаються з резервуванням на кількох дисках, які відображаються як один диск в операційній системі.

Переваги JBOD:

- JBOD допомагає мінімізувати втрату даних, напр. ми можемо поєднати накопичувач на 10 ГБ і диск на 40 ГБ, щоб отримати обсяг 50 ГБ від JBOD. Це дуже невелика перевага, оскільки може виникнути проблема при розширенні існуючої системи, за умови, що нині диски дешеві.
- За допомогою JBOD ми можемо відновити відсутні файли, якщо диск потрапляє в обсяг JBOD. Це залежить від того, як операційна система керує диском. Враховуючи, що відновлення JBOD може бути складним, це незначний недолік.
- Диски та контролери SATA дешеві за своєю природою, що робить JBOD економічно ефективнішим порівняно з RAID.

Недоліки в JBOD:

- При роботі кількох дисків продуктивність диска не збільшується.

- Якщо помилково втрачено будь-який диск, слід пошукати резервні копії. Якщо резервної копії немає, дані зникають назавжди.

Порівняння JBOD v/s RAID.

JBOD і RAID є двома основними конфігураціями зберігання даних.

Деякі відмінності та подібності між JBOD і RAID включають:

Подібність:

- Обидва застосовуються до зберігання даних.
- JBOD і RAID покращують використання дискового простору.
- У RAID дані, що зберігаються на кількох дисках, які відображатимуться в операційних системах як один диск, так само у випадку JBOD також диски відображаються в операційній системі як один диск.

Різниця:

- RAID є порівняно дорогим, ніж JBOD, оскільки RAID використовує такі компоненти, як диски SATA та контролери, які мають високу ціну.
- JBOD дозволяє комбінувати диски різного розміру, тоді як конфігурації RAID дозволяють використовувати в масиві лише диски однакового розміру.
- Здебільшого JBOD надають перевагу над RAID, оскільки їх легко масштабувати, що робиться простим додаванням іншого диска.
- RAID має певні функції, як-от підтримка відмови жорсткого диска, підвищення продуктивності, яких немає в JBOD.



Рисунок 2.6 – Схема JBOD

Переваги RAID:

- Надмірність даних: зберігаючи численні копії даних на багатьох дисках, RAID може захистити дані від збоїв диска.
- Підвищення продуктивності: RAID може підвищити продуктивність шляхом розподілу даних між кількома дисками, дозволяючи одночасне виконання кількох операцій читання/запису.
- Масштабованість: RAID є масштабованим, тому, додавши більше дисків до масиву, можна збільшити ємність зберігання.
- Універсальність: RAID застосовний до широкого діапазону пристроїв, таких як робочі станції, сервери та персональні ПК

Недоліки RAID:

- Вартість: впровадження RAID може бути дорогим, особливо для масивів великої ємності.
- Складність: налаштування та керування RAID може бути складним завданням.

- Зниження продуктивності: обчислення парності, необхідні для деяких конфігурацій RAID, зокрема RAID 5 і RAID 6, можуть призвести до зниження швидкості.
- Єдина точка відмови: RAID не є всеосяжним рішенням для резервного копіювання, але забезпечує резервування даних. У разі несправності RAID-контролера може бути втрачено весь вміст масиву

2.6 VirtualBox

2.6.1 Відкрита платформа віртуалізації VirtualBox

Технології віртуалізації викликають великий інтерес не лише у великих компаній, що використовують їх для підвищення ефективності інформаційної інфраструктури у складі комплексних рішень. Кінцеві користувачі також застосовують настільні платформи віртуалізації як на роботі, в повсякденній діяльності, так і удома. Безліч варіантів використання віртуальних машин, що широко рекламуються вендорами платформ віртуалізації, з підвищенням потужностей призначених для користувача апаратних платформ поступово розглядаються користувачами як надійні інструменти підвищення ефективності роботи з різним програмним забезпеченням і операційними системами. Технології віртуалізації дозволяють на одному фізичному комп'ютері (хості) запускати одночасно декілька віртуальних машин з операційними системами (гостьовими ОС)

При цьому, якщо платформа віртуалізації застосовується в рамках робочої діяльності підприємства, вимоги до неї, безсумнівно, вищі за вимоги до «домашніх» віртуальних машин. Якщо в корпоративному середовищі найважливішими є надійність, функціональність, гнучкість та можливість інтеграції з існуючою інфраструктурою, то вдома — це, перш за все,

безкоштовність самої платформи, зручність використання та продуктивність.

Вдома віртуальні машини можуть використовуватися для різних цілей,

основними з яких є:

- створення «переносних» ізольованих серед користувачів з встановленим у них програмним забезпеченням, спрямованих на вирішення певного кола завдань. Наприклад, не потрібно перевантажувати операційну систему домашнього комп'ютера встановленням різних специфічних програм, які не використовуються іншими користувачами. Можна створити віртуальну машину з встановленим у ній необхідним програмним забезпеченням (наприклад, дизайнерським) і запускати її тільки при необхідності. При цьому її можна легко перенести на іншу робочу станцію і бути впевненим, що реєстр, системні файли та налаштування хостової системи залишаться недоторканими.
- Створення захищених середовищ для виходу до Інтернету. Якщо троянська програма або вірус завдадуть шкоди всередині віртуальної машини, можна видалити цю віртуальну машину і створити нову, при цьому всі дані хостової системи будуть надійно захищені.
- Експерименти з різним програмним забезпеченням та операційними системами. Можливість одночасного запуску кількох операційних систем у віртуальних машинах дозволяє використовувати різні програми в необхідних ОС і забезпечувати передачу даних між ними. Безумовно, віртуальні машини відкривають широкі простори навчання, як новим операційним системам, і налаштуванню різних мережевих служб і протоколів взаємодії між різними ОС.

2.6.2 Огляд існуючих настільних платформ віртуалізації

В індустрії платформ віртуалізації користувачі звикли до традиційних продуктів, що розвиваються протягом кількох років та надають необхідний рівень функціональності. Однак лідер у галузі виробництва ПЗ для віртуалізації, компанія VMware, орієнтується в основному на комерційне використання своїх платформ, і хоча її потужна платформа віртуалізації VMware Workstation 6 є по безлічі параметрів кращою на даний момент, вона має мінус — вона не безкоштовна. Користувачі можуть завантажити з сайту безкоштовний продукт VMware Player, призначений для «програвання» віртуальних машин, однак створювати їх у ньому не можна. Продукт VMware Server спрямований на використання у корпоративному середовищі для віртуалізації серверів малих підприємств і також не підходить для домашніх користувачів.

Компанія Microsoft пропонує користувачам безкоштовний продукт Virtual PC, який, однак, не володіє необхідним функціоналом, що відповідає вимогам всіх категорій користувачів. Слід зазначити, що Virtual PC доступний лише робочих станцій з операційними системами сімейства Windows, що також відсікає певний сегмент користувачів.

Компанія Parallels також пропонує настільні платформи віртуалізації Parallels Workstation для Windows та Linux хостів, а також Parallels Desktop для Mac OS X, які також не є безкоштовними. До того ж, у зв'язку з успішним продажем продукту Parallels Desktop, компанія Parallels, до речі, що належить російській компанії SWsoft, дещо призупинила розвиток продукту Parallels Workstation і давно не випускала його нових версій. **(Але оскільки тут є сліди російської компанії це ПЗ принципово не буде використовуватися на обладнанні!)**

Таким чином, розглядаючи ринок популярних настільних систем віртуалізації, можна сказати, що є або непогані, але платні платформи, або безкоштовні, але недостатньо функціональні продукти, що не відповідають усім вимогам користувачів, а саме:

- кросплатформність
- підтримка більшості хостових та гостьових операційних систем
- можливість використання зручних засобів керування віртуальними машинами

Варто зазначити, що прихильники рішень Open Source хотіли б використовувати вільну платформу віртуалізації з відкритим кодом, якою є, наприклад, платформа Xen.

2006 року на ринку настільних платформ віртуалізації з'явився новий гравець. Німецька компанія InnoTek представила продукт VirtualBox для віртуалізації десктопів з відкритим вихідним кодом, у розробці якого (за винятком деяких компонентів) може взяти участь будь-хто. VirtualBox є єдиним кандидатом на те, щоб використовуватися на обладнанні автора роботи.

2.6.3 Платформа VirtualBox

Платформа VirtualBox є настільною системою віртуалізації для Windows, Linux і Mac OS хостів, що підтримує операційні системи Windows, Linux, OS/2 Warp, OpenBSD і FreeBSD як гостьові. Після кількох років розробки, компанія InnoTek випустила обмежену версію продукту з відкритим вихідним кодом під ліцензією GNU General Public License (GPL) у січні 2007 року. Повна версія продукту із закритим вихідним кодом також доступна для безкоштовного використання. Якщо продукт буде використовуватися у виробничому середовищі, необхідно придбання

ліцензій, умови яких можна дізнатися у компанії InnoTek. До речі, в даний момент компанія InnoTek, окрім платформи VirtualBox, займається розробкою нової платформи віртуалізації на основі гіпервізора, тісно пов'язаної з операційною системою Windows, у співпраці з компанією Microsoft. На даний момент VirtualBox включає наступні можливості:

- нативна x86-віртуалізація, яка не потребує підтримки апаратних технік Intel VT або AMD-V (яка, однак, може бути включена в налаштуваннях)
- дружній інтерфейс користувача (побудований за допомогою Qt3)
- підтримка Windows, Linux та Mac OS хостових систем (версія для Mac OS зараз перебуває в стадії бети)
- наявність Guest VM Additions для спрощення взаємодії з хостовими ОС та оптимізації їх швидкодії
- підтримка багатопроцесорних та багатоядерних систем (тільки як гостьові, підтримка віртуального SMP для представлення багатопроцесорності в гостьових системах відсутня)
- стабільність (порівняно з іншими Open Source рішеннями)
- підтримка віртуалізації аудіопристроїв
- висока продуктивність (за відгуками багатьох експертів вище, ніж у продуктів VMware)
- підтримка різних видів мережевої взаємодії (NAT, Host Networking via Bridged, Internal)
- підтримка дерева збережених станів віртуальної машини (snapshots), до яких може бути здійснений відкат з будь-якого стану гостьової системи
- опис налаштувань віртуальної машини у форматі XML

- підтримка Shared Folders для простого обміну файлами між хостовою та гостьовою системами

2.6.4 Емульоване апаратне оточення

Продукт VirtualBox емулює наступні компоненти апаратного забезпечення у віртуальній машині:

- жорсткі диски емулюються у спеціальному форматі контейнерів VDI (Virtual Disk Images), який на даний момент не сумісний із форматами віртуальних дисків інших виробників
- відеоадаптер емулюється як стандартний VESA з 8 Мб відеопам'яті, при цьому установка Guest VM Additions (тільки для Windows та Linux хостів) дозволяє збільшити продуктивність віртуального відеоадаптера та динамічно змінювати розмір вікна віртуальної машини
- аудіоконтролер на базі Intel ICH AC'97
- мережевий адаптер емулюється як інтерфейс AMD PCNet
- у виданні із закритим вихідним кодом емулюються також контролери USB, при цьому USB-пристрої, вставлені в роз'єм хоста, автоматично підхоплюються в гостьовій системі. Також якщо віртуальна машина діє як RDP (Remote Desktop Protocol) сервер, то в клієнті також буде видно USB-пристрої

Платформа VirtualBox виконує код гостьової системи нативно (прямою передачею інструкцій процесору хоста). Цей підхід добре працює для коду, що виконується в кільці третьої гостьової системи, для коду гостьової системи, що виконується в нульовому кільці, що вимагає привілейованих інструкцій, необхідний його перехоплення платформою віртуалізації. Для цього VirtualBox використовує оригінальний підхід: код,

що виконується в нульовому кільці гостьової системи, виконується в першому кільці хостової системи, яке не використовується в архітектурі Intel.

2.6.5 Унікальні функції VirtualBox

Крім стандартних функцій, властивих більшості настільних систем віртуалізації, платформа VirtualBox має також набір унікальних можливостей, властивих тільки їй:

Яскраво виражена модульність системи

Платформа VirtualBox має модульну архітектуру з добре описаними компонентами та надає зручні інтерфейси доступу до віртуальних машин, які дозволяють контролювати гостьові системи як через GUI, так і через командний рядок та віддалено. До того ж, компанія InnoTek надає відмінний Software Development Kit, і оскільки код платформи відкритий, не потрібні додаткові зусилля, щоб написати розширення до системи. В даний момент ведеться велика робота з портування продукту на різні хостові платформи і розробникам надаються всі необхідні інструменти та інтерфейси для доопрацювання VirtualBox.

Віртуальна машина може діяти як RDP-сервер

На відміну від інших віртуалізації, VirtualBox може діяти як RDP-сервер і керуватися будь-яким клієнтом, що підтримує протокол RDP. Також підтримується функція USB over RDP. Варто відзначити, що компанія VMware в платформі VMware Workstation 6, що нещодавно вийшла, також надає функцію Act as RDP Server, тому цю функцію VirtualBox на даний момент не можна назвати такою унікальною.

iSCSI initiator

Компонент iSCSI initiator є однією із закритих частин платформи VirtualBox. Він дозволяє використовувати зовнішні пристрої за протоколом

iSCSI як віртуальні диски в гостьовій системі без додаткової підтримки з боку гостьової ОС.

2.6.6 Висновки і проміжні результати роботи

Проаналізувавши вимоги до кластерів початкового рівня для домашнього офісу та обсяг і тип розрахунків буде прийнято наступну апаратну конфігурацію і монтаж як нижче.

Таблиця 2.1– Конфігурація

	Об'єкт	Опис
1	Користувач	Будь який комп'ютер і операційна система
2	Кластер	2 units of 2 x Xeon 2690v2 3GHz with 256Gb RAM кожен (Наразі працює тільки одна машина)
3	Локальна мережа	Розглядається для з'єднання обладнання 10Gb SFP (Наразі працює через 1Gb)
4	Файл сервер	Intel Celeron J1900 16Gb RAM. Також використовується для домашніх потреб. (Енергоефективне рішення)
5	Дисковий масив	4Tb RAID0 для даних розрахунку кластера (на машині кластера) 8Tb RAID1 для резервного копіювання (на машині кластера) 40Tb JBOD для домашніх і офісних потреб (на файл сервері)



Рисунок 2.7 – Монтаж обладнання

Створення якісного кластера для персональних наукових досліджень і розрахунків можливе навіть з використанням бюджетних рішень або вживаних пристроїв.

Продуктивність 1Tflop дозволяє скоротити час обчислень і зробити візуалізацію процесів і результатів обчислень.

Монтаж виконано в стійку та на полиці шафи з дверцятами для зменшення впливу шуму особливо в нічний час. Передбачено систему примусової вентиляції.

У разі відсутності власних завдань для кластера обчислювальні потужності будуть запропоновані на веб-сайті на безоплатній основі.

Проаналізувавши наявні дистрибутиви Linux вибір був зроблений вибір стандартного дистрибутиву Linux Ubuntu [10], де організація кластеризації та функцій файл-серверу буде доповнена стандартними

відкритими доповненнями типу oneSIS, Foreman, Kubernetes та ін. що включають також застосунки для організації RAID масиву [13].

Для дискової системи вибрано організацію дискового простору як у табл. 2.1. Дослідження показали, що однією з причин виходу жорстких дисків може є проблеми з електропостачанням, тому для кожного вузлу кластеру і елементу системи передбачено відповідний до параметрів електроспоживання ДБЖ, в випадку відключень передбачено встановлення ПЗ для автоматичного завершення роботи серверів і вузлів APC PowerChute [14]. Надалі дисковий під-масив планується розширити для забезпечення організації послуг резервного копіювання інших користувачів кластеру для власних потреб.

Для віртуалізації буде використовуватися VirtualBox. Дане ПЗ необхідне для встановлення Windows систем, оскільки більшість ПЗ для розрахунків існує виключно для Windows систем [8]. Опис програмного забезпечення було зроблено вище.

3 РОЗРОБКА ВЕБ САЙТУ

3.1 Мова HTML

3.1.1 Електронний гіпертекстовий документ

Гіпертекст — це термін, який останнім часом стає все більш популярним, особливо з розвитком Інтернету та цифрових технологій. Простими словами, гіпертекст відноситься до тексту, який містить посилання на інші частини тексту, що дозволяє користувачам легко переміщатися між ними.

Поняття гіпертексту можна віднести до 1940-х років, коли Ванневар Буш вперше запропонував ідею «memex» — пристрою, який дозволяв би користувачам зберігати та отримувати доступ до інформації нелінійним способом. Однак лише з розвитком Інтернету в 1990-х роках гіпертекст по-справжньому набув прав.

Сьогодні гіпертекст є невід’ємною частиною роботи в Інтернет, що дозволяє користувачам легко переміщатися між веб-сторінками, статтями та іншими формами цифрового вмісту. Незалежно від того, чи відбувається перегляд веб-сторінки, читання електронної книги чи використовується програма для мобільних пристроїв, швидше за все, будь якимось чином є взаємодія з гіпертекстом. У наступних розділах ми дослідимо історію та еволюцію гіпертексту, а також його численні застосування в сучасному світі.

Гіпертекст — це термін, який описує тип електронного тексту, який містить посилання на інший пов’язаний вміст. Концепцію гіпертексту вперше ввів Тед Нельсон у 1960-х роках, а пізніше популяризував Дуглас Енгельбарт у 1980-х роках. Сам термін вперше ввів Нельсон у своїй книзі «Літературні машини» 1965 року. Гіпертекст — це, по суті, нелінійний спосіб подання інформації, що дозволяє користувачам переміщатися по мережі

взаємопов'язаного вмісту. Це досягається за рахунок використання гіперпосилань, які є елементами, які можна натиснути, які переводять користувачів до інших частин тексту або до зовнішніх ресурсів.

Є кілька ключових понять, які важливо розуміти, коли йдеться про гіпертекст. До них належать:

- Посилання: як уже згадувалося, посилання є основою гіпертексту. Вони дозволяють користувачам легко переміщатися між різними частинами вмісту, створюючи структуру, схожу на мережу, якою легко орієнтуватися.
- Вузли: це окремі частини вмісту в гіпертекстовому документі. Це може бути будь-що: від одного слова до цілої сторінки тексту.
- Якір: використовується для створення посилань у документі. По суті, це маркери, які дозволяють користувачам переходити до певної точки в межах вмісту.
- Перегляд:— це переміщення гіпертекстовим документом за посиланнями. Це можна зробити нелінійним способом, дозволяючи користувачам досліджувати вміст у зручний для них спосіб.

Гіпертекст став невід'ємною частиною сучасної мережі, де веб-сайти та онлайн-ресурси значною мірою покладаються на посилання для зв'язку користувачів із відповідним вмістом. За словами Merriam-Webster, гіпертекст — це «формат бази даних, у якому інформація, пов'язана з інформацією на дисплеї, доступна безпосередньо з дисплея».

Як впливає з назви, динамічний гіпертекст означає можливість змінювати вміст гіпертекстового документа на основі взаємодії користувача. Це означає, що вміст веб-сторінки може змінюватися залежно від введення або дій користувача. Динамічний гіпертекст є ключовим компонентом сучасної веб-розробки і став можливим завдяки інтеграції JavaScript.

3.1.1 Інтеграція JavaScript

JavaScript — це мова програмування, яка дозволяє розробникам створювати інтерактивні веб-сторінки. Він широко використовується у веб-розробці для додавання функцій до веб-сторінок, таких як перевірка форм, анімація та динамічний вміст. При інтеграції з гіпертекстом JavaScript дозволяє розробникам створювати динамічні та інтерактивні веб-сторінки, які можуть реагувати на введення користувача в реальному часі.

Наприклад, веб-сторінка, яка використовує динамічний гіпертекст, може відображати різний вміст залежно від місця розташування користувача або часу доби. Цього можна досягти, використовуючи JavaScript для визначення місцезнаходження користувача або поточного часу, а потім динамічно оновлюючи вміст веб-сторінки.

Окрім зміни вмісту веб-сторінки, JavaScript також можна використовувати для створення інтерактивних елементів, таких як спадні меню, повзунки та спливаючі вікна. Ці елементи дозволяють користувачам взаємодіяти з веб-сторінкою більш інтуїтивно зрозумілим і привабливим способом, роблячи загальний досвід користувача більш приємним.

Загалом динамічний гіпертекст є потужним інструментом для веб-розробників, який дозволяє їм створювати привабливі та інтерактивні веб-сторінки. Інтегруючи JavaScript у гіпертекст, розробники можуть створювати веб-сторінки, які реагують на введення користувача в режимі реального часу, забезпечуючи більш привабливий та персоналізований досвід користувача.

відео, у веб-сторінки. Ця функція допомогла зробити веб-сайти більш привабливими та інтерактивними. Завдяки мультимедійній інтеграції веб-дизайнери можуть створювати візуально приголомшливі та динамічні сторінки, які привертають увагу аудиторії.

Інтерактивні форми.

Інтерактивні форми — ще одна розширена функція гіпертексту, яка зробила веб-сторінки більш інтерактивними. Форми використовуються для збору інформації від користувачів, а інтерактивні форми дозволяють користувачам взаємодіяти з веб-сайтом у режимі реального часу. Ця функція відіграла важливу роль у створенні онлайн-магазинів, опитуваннях та інших типах веб-додатків, які вимагають введення даних від користувача.

Таблиці даних.

Таблиці даних — ще одна розширена функція гіпертексту, яка зробила веб-сторінки більш інформативними та корисними. За допомогою таблиць даних веб-дизайнери можуть представити складну інформацію в упорядкованому та зручному для читання форматі. Ця функція допомогла створити веб-сайти, які надають користувачам доступ до великих обсягів даних, таких як фінансові звіти, каталоги продуктів і наукові дослідження.

Розширені функції гіпертексту революціонізували спосіб розробки та представлення веб-сторінок. Завдяки мультимедійній інтеграції, інтерактивним формам і таблицям даних веб-дизайнери можуть створювати привабливі та інформативні сторінки, які надають користувачам багатий інтерактивний досвід.

3.1.3 Гіпертекст та Інтернет

Оскільки Інтернет набув більшого поширення, гіпертекст став невід’ємною частиною навігації та доступу до інформації. Всесвітня

павутина, або просто мережа, — це велика колекція взаємопов'язаних документів і ресурсів, які об'єднують їх разом за допомогою гіпертексту. Мережу створив Тім Бернерс-Лі в 1989 році, і вона швидко стала найпопулярнішим способом доступу до інформації в Інтернеті.

Всесвітня павутина.

Всесвітня павутина побудована на основі Інтернету, який є глобальною мережею взаємопов'язаних комп'ютерів. Інтернет дозволяє користувачам отримувати доступ до інформації та обмінюватися нею через Інтернет за допомогою веб-браузерів. Гіпертекст використовується для зв'язування веб-сторінок, що дозволяє користувачам легко переміщатися між ними.

Веб-сторінки зазвичай пишуться мовою HTML або мовою розмітки гіпертексту, яка використовується для створення веб-сторінок. HTML дозволяє веб-розробникам створювати посилання, зображення та інший мультимедійний вміст, до якого можуть отримати доступ користувачі.

Інтернет-протоколи.

Інтернет побудований на наборі протоколів, які є правилами, які керують тим, як дані передаються через мережу. Одним із найважливіших протоколів, що використовуються в Інтернеті, є протокол керування передачею/протокол Інтернету (TCP/IP). TCP/IP використовується для передачі даних між комп'ютерами в Інтернеті та забезпечує надійну та ефективну передачу даних.

Протокол передачі гіпертексту (HTTP) — ще один важливий протокол, який використовується в Інтернеті. HTTP використовується для передачі даних між веб-серверами та веб-браузерами. Коли користувач запитує веб-сторінку, веб-сервер надсилає сторінку в браузер користувача за допомогою HTTP.

Гіпертекст є важливою частиною Інтернету, і він революціонізував спосіб доступу до інформації та обміну нею. Всесвітня павутина та Інтернет-протоколи, такі як TCP/IP і HTTP, зробили можливим доступ до інформації та обмін нею з будь-якої точки світу.

Гіпертекст у базах даних.

У міру розвитку баз даних здатність зв'язувати інформацію ставала все більш важливою. Гіпертекст природно підходить для баз даних, оскільки дозволяє користувачам нелінійно переміщатися між пов'язаними даними. У цьому розділі буде виконане дослідження, як гіпертекст використовується в базах даних, зокрема з точки зору інтеграції бази даних.

Інтеграція бази даних.

Гіпертекст можна використовувати для інтеграції баз даних різними способами. Одним із поширених підходів є використання гіпертекстових посилань для з'єднання пов'язаних записів у кількох таблицях. Це дозволяє користувачам швидко переміщатися між пов'язаними даними без виконання складних запитів.

Іншим способом використання гіпертексту в інтеграції бази даних є використання гіперпосилань у звітах. Звіти можна створювати за допомогою гіперпосилань, які дозволяють користувачам переходити безпосередньо до відповідних записів у базі даних. Це може бути особливо корисним для великих, складних баз даних, де пошук пов'язаних записів може бути проблемою.

Окрім зв'язування пов'язаних даних, гіпертекст також можна використовувати для надання контекстно-залежної довідки в базі даних. Вставляючи гіпертекстові посилання в інтерфейс користувача, користувачі можуть швидко отримати доступ до довідкових файлів або іншої документації, пов'язаної з поставленим завданням.

Загалом, гіпертекст є важливим інструментом для інтеграції баз даних і робить їх більш доступними для користувачів. Забезпечуючи нелінійний спосіб навігації пов'язаними даними, гіпертекст може допомогти користувачам знайти потрібну інформацію швидко й ефективно.

Важливість гіпертексту.

Очевидно, що гіпертекст є важливим інструментом для створення та організації інформації. Гіпертекст дозволяє пов'язувати пов'язані ідеї та концепції, полегшуючи читачам орієнтуватися в своїй роботі (сайт) та отримувати доступ до додаткової інформації.

Однією з основних переваг гіпертексту є його здатність надавати нелінійний доступ до інформації. На відміну від традиційного тексту, гіпертекст дозволяє читачам вибирати власний шлях у документі, переходячи за посиланнями, які їх цікавлять, і пропускаючи розділи, які їх не цікавлять. Ця гнучкість може полегшити читачам пошук потрібної інформації швидко та ефективно.

Окрім покращення читацького досвіду, гіпертекст також може полегшити авторам створення та організацію своєї роботи. Використовуючи посилання для об'єднання споріднених ідей і концепцій, автори можуть створити більш цілісний і взаємопов'язаний документ. Це також може спростити оновлення та перегляд документа, оскільки зміни, внесені до одного розділу, можуть бути відображені в усьому документі.

Загалом, гіпертекст є важливим інструментом для кожного, хто створює або споживає інформацію. Його здатність забезпечувати нелінійний доступ до інформації та полегшувати зв'язки між спорідненими ідеями може полегшити як авторам, так і читачам навігацію та розуміння складних тем.

3.1.4 Історія HTML

HTML — це мова гіпертекстової розмітки, яка використовується для створення та організації вмісту на веб-сайті або, якщо розглядати глобально, то і у глобальній мережі. Він також використовується для відображення структури та макета у веб-браузерах, які утворюють основу для веб-сторінок.

HTML використовує систему тегів розмітки для визначення різних елементів та їхніх зв'язків на веб-сторінці. У HTML ці теги йдуть парами. Вони взяті в кутові дужки і зазвичай включають відкриваючі та закриваючі теги. У HTML ми маємо певну інформацію, таку як текст, зображення та посилання, і за допомогою всієї цієї інформації можна створити фактичний вміст веб-сайту.

На початку 1990-х фізик на ім'я Тім Бернерс-Лі, працюючи в CERN на початку 1990-х, придумав, як люди можуть ділитися інформацією та отримувати доступ до неї в усьому світі. Його ідеєю була Всесвітня павутина, яку він зробив можливою, представивши HTML у 1991 році. HTML дозволив організовувати документи за допомогою простих тегів розмітки, дозволяючи користувачам легко переміщатися між пов'язаними документами.

Еволюція розробки HTML вплинула на те, як розробник може створювати та використовувати веб-вміст. HTML — це стандартна мова розмітки, яка використовується для створення веб-сторінок, і це важливий компонент Всесвітньої павутини.

Ось короткий підсумок важливих етапів розвитку HTML:

HTML 1.0. У 1993 році Тім Бернерс-Лі, творець Всесвітньої павутини, розробив HTML 1.0, першу версію HTML. Мова розмітки була простою та створена для основних текстових документів із

гіперпосиланнями. HTML 1.0 представив теги `html>`, `head>` і `body>` на додаток до тегу прив'язки (`a`) для створення гіперпосилань.

HTML 2.0. було розроблено в 1995 році. HTML 2.0 розширив функціональність веб-сторінок, підтримуючи таблиці та вбудовування зображень. Він також представив ідею форм введення користувачами.

HTML 3.0 був розроблений у 1995 році. HTML 3.0 був проектом специфікації з більш складними функціями, такими як підтримка мультимедіа, таблиць стилів і розширених таблиць. Однак це ніколи не було широко прийнято.

HTML 3.2 був розроблений у 1997 році. HTML 3.2 був першим широко прийнятим стандартом, який приніс значні покращення. Він представив такі функції, як покращені форми, підтримка кількох наборів символів і рамки для поділу веб-сторінок.

У HTML 4.01 визначив значний прогрес, забезпечивши кращу підтримку, щоб розробники могли легко використовувати каскадні таблиці стилів, які використовуються для стилізації веб-сторінок. HTML 4.01 забезпечив із використанням таких елементів, як `div` і `span`, для створення більш точної структури документа. HTML 4.01 має різні варіанти для відповідності вимогам, зокрема HTML 4.01 перехідного періоду та HTML4.01 strict.

Розширювана мова гіпертекстової розмітки (XHTML)

XHTML прагнув надати HTML структуру XML. Він вимагав добре оформлених документів і дотримувався суворіших вказівок щодо синтаксису. Основними версіями XHTML були 1.0 і 1.1, але через його жорсткість прийняття XHTML було обмеженим.

HTML5 перелом у розвитку HTML. Це спростило розробку багатofункціональних веб-додатків, представивши широкий спектр нових

елементів і API для мультимедіа, графіки та інтерактивності. Полотно, відео, аудіо та масштабований векторний графічний елемент (svg) були важливими функціями. Завдяки таким елементам, як «заголовок», «навігація» та «нижній колонтитул», HTML5 також надає пріоритет семантичній розмітці, що полегшує пошуковим системам і допоміжним технологіям розуміння веб-вмісту.

Відео та Canvas: введення в HTML5 елемента «canvas» дозволило програмістам створювати динамічну графіку та анімацію прямо в браузері. Елемент `video` також стандартизував відтворення відео без використання зовнішніх плагінів.

Чуйний дизайн: поширення мобільних пристроїв викликало потребу в адаптивному дизайні, що й суть цього. Це стало можливим завдяки функції медіа-запитів HTML5, яка дозволяла веб-сайтам легко адаптуватися до різних розмірів екрана.

Веб-інтерфейси API: HTML5 також зробив доступними кілька API, зокрема API геолокації та API веб-сховища, дозволяючи веб-сайтам отримувати доступ до можливостей пристроїв і покращувати взаємодію з користувачами.

HTML перетворився з випуску в окремих версіях на "життєвий стандарт". Іншими словами, у міру розробки нових функцій постачальники браузерів впроваджують їх у специфікацію, яка постійно змінюється. Ця стратегія дозволяє швидко впроваджувати інновації та адаптуватися до мінливих вимог Інтернету.

3.1.2 Основні поняття мови HTML

HTML (HyperText Markup Language) — стандартна мова розмітки для розробки структури веб-сторінки. HTML — це комбінація гіпертексту та

мови розмітки. PureText визначає внутрішні зв'язки між веб-сторінками, а мова розмітки визначає макет і представлення тексту та медіа.

Основна структура HTML-документа містить декілька обов'язкових тегів, які необхідно використовувати для створення веб-сторінки. Цієї структури необхідно дотримуватися, щоб створити будь-який тип документа HTML.

Нижче наведено основні теги, які визначають базову структуру HTML:

1. `<!DOCTYPE html>`

Цей елемент визначає тип документа як HTML. Цей елемент має бути написаний перед написанням будь-якого документа HTML.

2. `<html>...</html>`

Тег `<html>` є батьківським тегом для всіх елементів HTML. Усе, що пов'язано зі створенням HTML-документа, має бути написано всередині тегу `<html>`. CSS, JavaScript і jQuery також повинні бути записані всередині цього тегу.

3. `<head>...</head>`

Тег `<head>` — це тег-контейнер для всіх елементів, які не відображаються безпосередньо на веб-сторінці, але необхідні для функціонування сторінки. Він містить мета-теги (які використовуються для цілей SEO), тег заголовка, теги сценарію тощо.

4. `<title>...</title>`

Тег `<title>` використовується для визначення назви веб-сторінки, яку ви можете побачити на вкладці браузера, у списку закладок і в результатах пошукової системи. Цей тег також дуже важливий для цілей SEO, щоб допомогти пошуковій системі зрозуміти вміст веб-сторінки.

5. `<body>...</body>`

Тег `<body>` — це тег-контейнер для всіх цих елементів, який представляє основний вміст веб-сторінки, що відображається у браузері.

6. `<h1>...</h1>`

Тег `<h1>` є одним із тегів заголовків. Це найважливіший тег заголовка, який визначає головний заголовок або заголовок веб-сторінки. Будь-який текст, написаний у `<h1>` та `</h1>`, є заголовком верхнього рівня вмісту.

7. `<p>...</p>`

Тег `<p>` визначає абзац, усе, що написано всередині `<p>` і `</p>`, відображається як абзац на веб-сторінці. Можна використовувати кілька тегів `<p>` для відображення тексту в різних абзацах.

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="DenCalc provides cloud based Calculation Software">
  <meta name="keywords" content="Calculation software, cloud based, Catia, ANSYS">

  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">

  <title>DenCalc</title>

  <link rel="icon" href="https://kuponom.com/dencalc/assets/img/favicon.ico">
  <link rel="apple-touch-icon-precomposed" sizes="152x152" href="https://kuponom.com/dencalc/assets/images/fav.png">

  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/animate.min.css">

  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/icons.css">
  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/newsboxANDnewspage.css?v=1734889208">

  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/styles.css?v=1734889208">

  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/responsive.css">

  <link rel="stylesheet" href="https://kuponom.com/dencalc/assets/css/main.css?v=1734889208">
```

Рисунок 3.3 – Приклад HTML коду веб-сайту

3.2 CSS

CSS (каскадні таблиці стилів) — це мова таблиць стилів. Він схожий на мову програмування, але замість того, щоб говорити комп'ютеру, що робити, він повідомляє веб-браузеру, як відображати елементи на веб-сайті. CSS використовується для керування стилями елементів на веб-сторінці.

3.2.1 Структура CSS правил

CSS правило складається з селектора та блока декларацій. Селектор визначає, які елементи на сторінці будуть застосовані до цих декларацій, тоді як блок декларацій містить властивості та їх значення.

Селектор може бути ідентифікатором, класом, тегом HTML, або будь-якою комбінацією їх.

Декларації вказують конкретні атрибути стилю, такі як колір тексту, розмір шрифту, відступи тощо. Кожна декларація складається з назви властивості та її значення, розділених двокрапкою та оголошені в фігурних дужках. Коментарі можуть бути включені для пояснення коду та полегшення його розуміння. Використання зрозумілої структури CSS правил допомагає підтримувати організований та легко читабельний код, що робить процес розробки ефективнішим та приємнішим.

3.2.2 Синтаксис CSS

Селектори: селектори використовуються для «вибору» елемента HTML, який потрібно стилізувати. Це може бути тип елемента (наприклад, `h1`), клас (наприклад, `.class-name`), ідентифікатор (наприклад, `#id-name`) або їх комбінація.

- Селектор типу: націлює всі елементи певного типу, наприклад `h1`, `p`, `div` тощо.
- Вибір класу: націлює елементи з певним класом. Приклад: `.my-class { }`
- Селектор ідентифікатора: націлює елемент із певним ідентифікатором. Приклад: `#my-id { }`
- Універсальний селектор: націлює всі елементи `(*)`.

Властивості: властивості — це аспекти вибраних елементів, які потрібно стилізувати (наприклад, колір, ширина, висота тощо).

- колір: визначає колір тексту.
- колір фону: визначає колір фону елемента.
- font-size: встановлює розмір шрифту.
- margin: визначає простір навколо елемента.
- padding: визначає відстань між вмістом елемента та його межею.

Значення: значення визначають специфіку властивості, яку потрібно застосувати, наприклад назву кольору, число (наприклад, 16 пікселів) або відсотки (наприклад, 50%).

```

/* PRELOADER IMAGE BACKGROUND */
.status {
    background-image: url(../images/loader-blue.gif);
    background-repeat: no-repeat;
    background-position: center;
    -webkit-background-size: cover;
    background-size: cover;
}

/* THE OVERLAY COLORS WHICH WILL SHOW IN FRONT OF BACKGROUND IMAGES*/
.home .overlay-layer {
    background: rgba(0, 5, 8, 0.75);
}

.overlay-layer-2 {
    background: rgba(0, 5, 8, 0.90);
}

/* NAVBAR AND LINK COLORS */
.navbar-inverse .navbar-toggle:hover,
.navbar-inverse .navbar-toggle:focus {
    color: red;
}

a {
    color: red;
}

a:hover {
    color: darkred;
}

```

Рисунок 3.4 – Приклад CSS коду веб-сайту

3.3 Figma

Figma — це потужний онлайн-інструмент, який використовується для розробки веб-сайтів, програм та інших цифрових продуктів. Він діє як цифровий робочий простір, де можна легко створювати та редагувати проекти. Однією з його видатних особливостей є те, що він дозволяє кільком людям працювати над одним проектом одночасно, що робить співпрацю простою та ефективною. Figma запускається безпосередньо у веб-браузері, тому немає необхідності встановлювати будь-яке спеціальне програмне забезпечення.

Figma — це один із найдоступніших і зручних інструментів дизайну, що робить його дуже популярним серед графічних дизайнерів, маркетологів, дизайнерів інтерфейсу користувача та UX-дизайнерів із будь-яким рівнем досвіду.

Figma зазвичай використовується для:

- Створення статичних каркасів та макетів для веб-сайтів і мобільних додатків;
- Створення інтерактивних прототипів (знову ж таки, для веб-сайтів і мобільних додатків);
- Створення системи дизайну з багаторазовими компонентами та стилями та керування ними;
- Створення презентацій, логотипів, значків, інфографіки та публікацій в соціальних мережах;
- Проводити віртуальні семінари та спільні мозкові штурми (через FigJam, цифрову дошку від Figma).

Figma не тільки неймовірно зручна для користувача, але й наповнена набором зручних функцій і функцій. Ось деякі з найкращих функцій Figma, які дійсно відрізняють цей інструмент.

Функція автоматичного макета

Розробляючи прототипи веб-сайтів і додатків, важливо думати про те, як проекти виглядатимуть і працюватимуть на різних пристроях і розмірах екрана. Для цього потрібен адаптивний або адаптивний дизайн, а функція автоматичного макета Figma спрощує процес створення адаптивних макетів.

Автоматичний макет — це, по суті, властивість, яку можна додати до фреймів і компонентів у своїх проектах, вказуючи їм автоматичну зміну розміру та розташування залежно від доступного простору.

Розумний вибір

Цю функцію було створено, щоб пришвидшити деякі з більш повторюваних завдань дизайну, скажімо, налаштувати відстань навколо всіх значків на екрані або збільшити висоту групи елементів на сторінці.

З розумним вибором не потрібно налаштовувати кожен елемент окремо; можна вибирати та маніпулювати кількома об'єктами одночасно.

Smart Selection працює автоматично з будь-яким виділенням або групою з трьох або більше елементів із рівномірним інтервалом. Вибравши вибрані вами елементи, можна відразу налаштувати їх відстань, положення та розмір. Це робить набагато легшим (і швидшим) експериментувати з різними можливостями у вашому дизайні. Мобільний додаток, що відображає на екрані програмний інтерфейс Figma.

Інструмент масштабування

Інструмент масштабування у Figma дозволяє пропорційно змінювати розмір шарів і об'єктів. Після того, як відбулася активація інструмент

масштабування можна вибрати об'єкт(и), розмір яких потрібно змінити, і настиснувши і перетягнувши, щоб збільшити або зменшити масштаб або ввести точні розміри, які потрібно застосувати.

Інструмент масштабування зберігає пропорції вибраних елементів, забезпечуючи незмінність загального дизайну.

Варіанти

Протягом усього процесу проектування можливо побачити, що потрібні компоненти, які значною мірою схожі один на одного, з деякими невеликими відмінностями чи варіаціями залежно від контексту чи випадку використання.

За допомогою функції варіантів у Figma можливо створювати різні версії компонента (скажімо, кнопки) в одному батьківському компоненті. Це корисно для керування різними станами, стилями чи розмірами елементів інтерфейсу користувача, а також допомагає впорядкувати та спростити бібліотеку компонентів.

Співпраця в реальному часі

Це не функція як така, але це одна з визначальних характеристик Figma — і одна з причин, чому вона така популярна. Figma дозволяє декільком користувачам працювати над одним файлом дизайну одночасно, сприяючи безперебійній співпраці в режимі реального часу.

Члени команди можуть бачити курсори один одного, вносити зміни та надсилати повідомлення за допомогою функції чату курсору. Це повторює відчуття спільної роботи в одній кімнаті; ідеально підходить для віддалених і розподілених команд.

Набори інтерфейсу користувача

Набір інтерфейсу користувача — це готова колекція компонентів, стилів і макетів, які можна використовувати для швидкого створення макетів

і прототипів. Figma постачається з великою бібліотекою таких комплектів інтерфейсу користувача для різних операційних систем і варіантів використання, включаючи набори для iOS і iPad, Android і навіть Microsoft Teams і Instagram.

Плагіни

Використовуючи Figma, немає обмеження по основними функціями інструменту. Він підтримує широкий спектр плагінів, які ще більше покращують функціональність.

Знайти можливо плагіни майже для будь-чого — для пришвидшення процесу створення каркасів, розробки потоків користувачів або додавання вмісту-заповнювача до нашого дизайну.

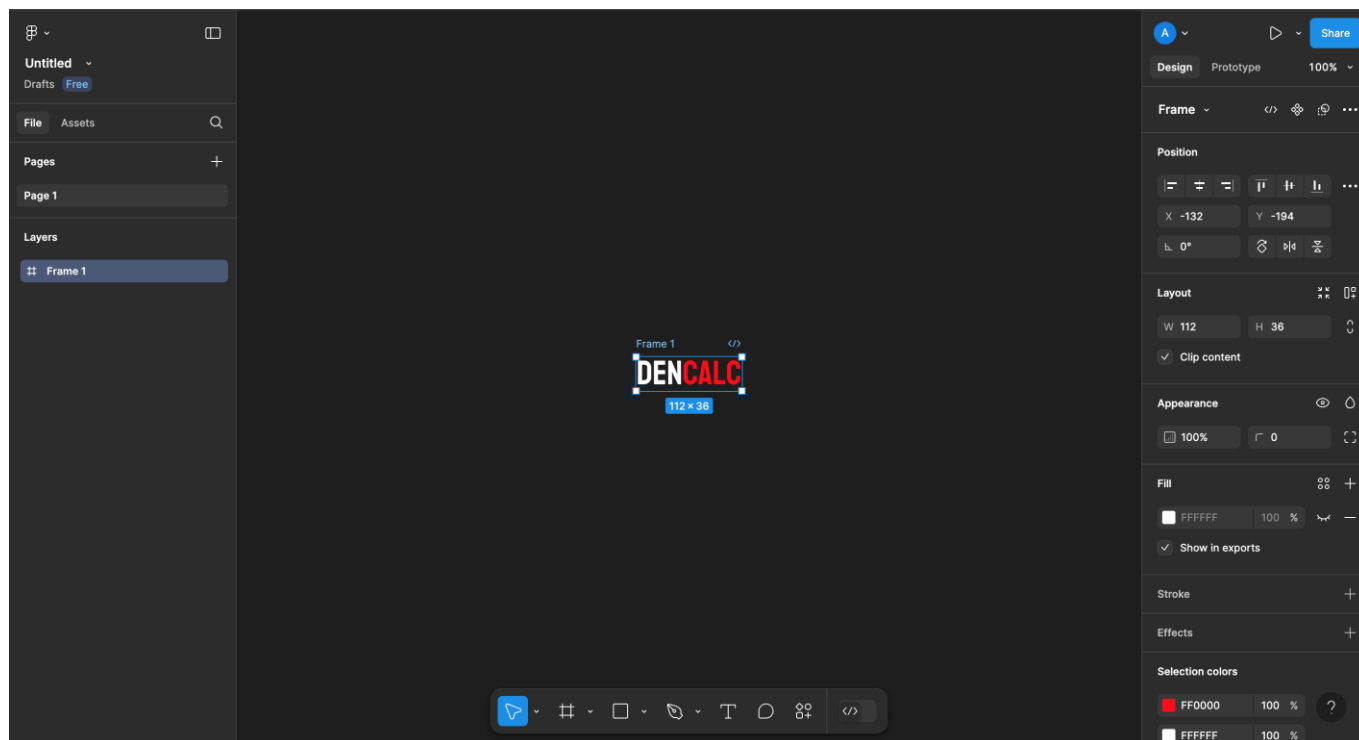


Рисунок 3.5 – Приклад розробки логотипу в Figma

3.4 Мова програмування PHP

PHP — це серверна мова програмування з відкритим вихідним кодом, яку можна використовувати для створення веб-сайтів, програм, систем управління взаємовідносинами з клієнтами тощо. Це широко використовувана мова загального призначення, яку можна вставляти в HTML. Ця функція з HTML означає, що мова PHP залишається популярною серед розробників, оскільки вона допомагає спростити код HTML.

PHP розшифровується як «PHP: Hypertext Preprocessor», а оригінальний PHP у цьому значенні означає «Personal Home Page». Акронім змінювався в міру розвитку мови з моменту її появи в 1994 році, щоб точніше відображати її природу.

З моменту випуску PHP станом на 2022 рік було випущено 8 версій, причому версія 8.1 зараз є популярним вибором серед тих, хто використовує цю мову на своїх веб-сайтах.

3.4.1 Застосування

Програмування на PHP можна використовувати для створення більшості речей, які потрібні розробнику програмного забезпечення. Однак є три основні сфери, в яких він працює. найкраще.

Серверний сценарій.

Сценарій на стороні сервера є головною перевагою PHP. Якщо ви тільки вчитеся кодувати й хочете вивчити сценарії на стороні сервера, PHP — чудова мова для вивчення. Щоб використовувати PHP за допомогою серверних сценаріїв, знадобиться PHP-парсер, веб-сервер і веб-браузер.

Сценарії командного рядка.

Сценарії командного рядка ідеально підходять для сценаріїв, створених за допомогою cron (Linux) або Task Scheduler (Windows). Він також чудово підходить для простої обробки тексту.

Написання настільних програм.

PHP, скоріш за все, не є найкраща мова для створення настільних програм, але для просунутого веб-розробника він надає набагато більше можливостей, ніж його конкуренти.

Звичайно, PHP може робити багато інших речей. Наприклад, він добре збирає дані форм, шифрує дані користувача та надсилає та отримує файли cookie. Одна з головних особливостей PHP, яка робить його таким зручним, полягає в тому, що він сумісний з усіма основними операційними системами, тому ви можете кодувати незалежно від того, яку технологію ви використовуєте.

Використовування PHP.

Для ефективного кодування на PHP потрібні три основні речі:

Синтаксичний аналізатор PHP: аналізатор бере код PHP і аналізує його, виводячи відповідне синтаксичне дерево, яке поміщає вихідний код у формат, який легше читати для розуміння машинами.

Веб-сервер: сервер — це програма, яка виконуватиме розроблені PHP-файли для створення веб-сторінок

Веб-браузер: браузер дозволяє переглядати сторінку PHP через сервер так само, як і будь-який інший вміст в Інтернеті.

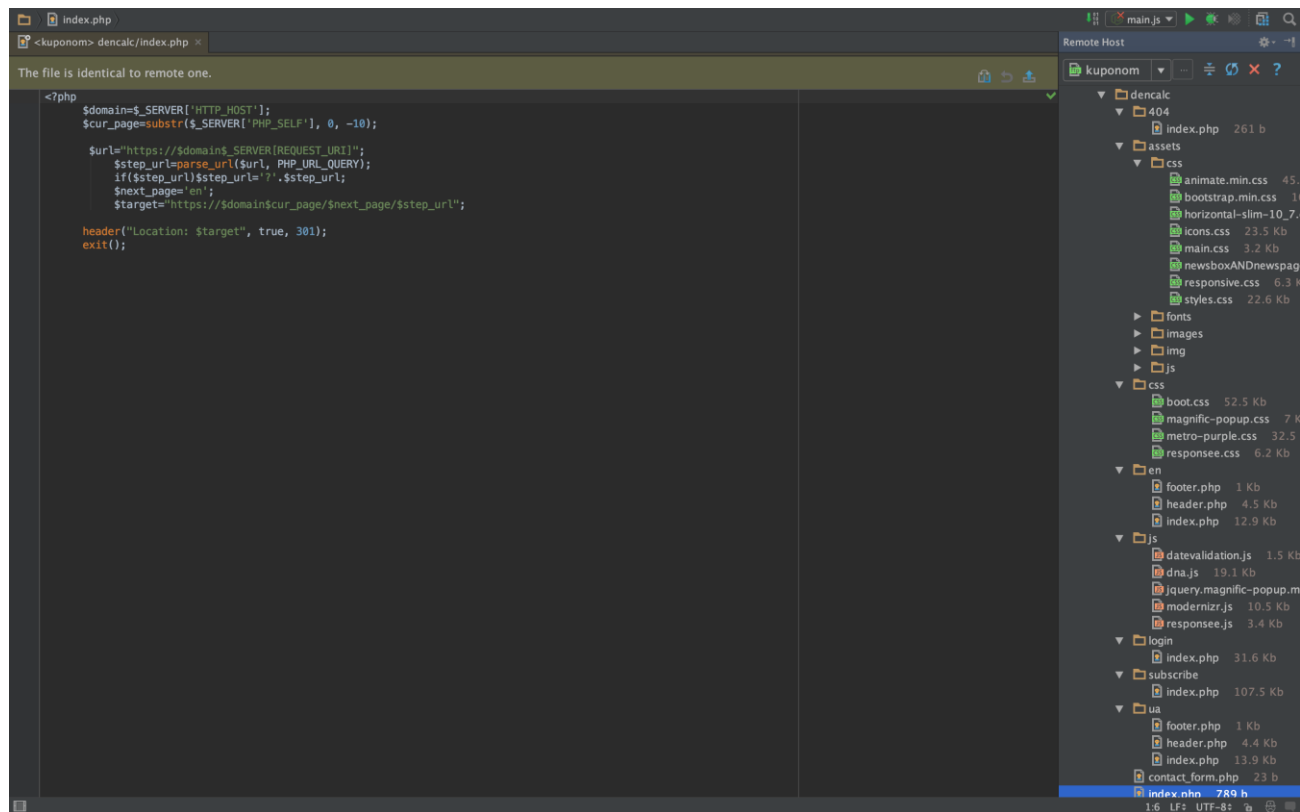


Рисунок 3.6 – Приклад коду РНР

Переваги РНР:

- Відкритий вихідний код: РНР є мовою з відкритим вихідним кодом, тобто вона вільно доступна для використання та розповсюдження. Ця відкритість сприяла створенню великої та активної спільноти розробників, які постійно роблять внесок у її зростання, вдосконалення та розвиток функцій.
- Легкість освоєння: синтаксис РНР дуже схожий на С та інші мови програмування. Це робить РНР відносно легким для вивчення, особливо для розробників, які вже мають певний досвід програмування. Початківці вважають його доступним завдяки простому синтаксису та обширним онлайн-ресурсам.

- Веб-інтеграція: PHP розроблено спеціально для веб-розробки та вбудовано в HTML. Він легко інтегрується з різними веб-технологіями, полегшуючи створення динамічних та інтерактивних веб-сторінок.
- Підтримка баз даних: PHP чудово підтримує різні бази даних, включаючи MySQL, PostgreSQL, SQLite тощо. Це полегшує підключення та взаємодію з базами даних, що є важливим аспектом багатьох веб-програм.
- Сумісність між платформами: PHP не залежить від платформи та працює на різних операційних системах, включаючи Windows, Linux, macOS та інші. Це забезпечує сумісність у різних середовищах.
- Велика спільнота та документація: у PHP є велика та активна спільнота розробників. Велика кількість онлайн-ресурсів, навчальних посібників і документації полегшує розробникам пошук рішень і звернення за допомогою за потреби.
- Фреймворки та CMS: існують такі популярні фреймворки PHP, як Laravel, Symfony та CodeIgniter, які надають готові модулі та функції, що сприяє швидкому розвитку. Крім того, PHP підтримує широко використовувані системи керування вмістом (CMS), такі як WordPress і Joomla.
- Сценарії на стороні сервера: скрипти PHP виконуються на сервері, зменшуючи навантаження на стороні клієнта. Ця можливість створення сценаріїв на стороні сервера має вирішальне значення для створення динамічного вмісту та виконання завдань, пов'язаних із сервером.

- Підтримка спільноти: спільнота PHP активно сприяє розвитку мови, забезпечуючи регулярні оновлення, виправлення безпеки та вдосконалення.

Недоліки PHP:

- Невідповідність: PHP критикували за невідповідності в назвах функцій і порядку параметрів. Це може призвести до плутанини для розробників, особливо при роботі з поєднанням старіших і новіших функцій.
- Проблеми з безпекою. Якщо з ним не поводитися належним чином, PHP-код може бути вразливим до вразливості системи безпеки, наприклад впровадження SQL і міжсайтового сценарію (XSS). Розробникам потрібно бути обережними та дотримуватися найкращих практик для захисту додатків PHP.
- Продуктивність. Незважаючи на те, що PHP добре працює з багатьма веб-додатками, він може бути не таким швидким, як деякі скомпільовані мови, такі як C або Java. Однак удосконалення та оптимізація в останніх версіях підвищили продуктивність.
- Відсутність сучасних функцій: порівняно з новими мовами, у PHP можуть бути відсутні деякі сучасні функції мови. Однак останні версії PHP представили вдосконалення та функції для вирішення цієї проблеми.
- Виклики масштабованості: PHP може зіткнутися з проблемами, коли справа доходить до масштабування великих і складних програм. Розробникам може знадобитися застосувати додаткові інструменти або фреймворки для вирішення проблем масштабованості.

- Не підходить для великомасштабних додатків: хоча РНР підходить для проектів малого та середнього розміру, він може бути не найкращим вибором для надзвичайно великих і складних додатків, де можна віддати перевагу більш структурованим мовам.
- Обмежена підтримка об'єктно-орієнтованого програмування (ООП): хоча РНР підтримує ООП, його реалізацію критикували за те, що вона не така надійна, як у деяких інших мовах. Однак останні версії внесли вдосконалення для покращення можливостей ООП.

3.5 Опис роботи сайту та інструменти для розробки

Функції веб сторінки мають забезпечити доступ до кластеру і хмарного сховища, дати можливість зареєструватися користувачеві, отримати контактні дані для звернення.

3.5.1 Дизайн сайту

Дизайн сайту було розроблено в середовищі Figma.

Мова розмітки сайту – HTML

Оздоблення сайту – CSS.

Сайт має містити основні розділи такі як:

- Сервіс розрахунків
- Форма реєстрація
- Новини проекту
- Команда
- Ціни
- Контактні дані

Згодом після апаратного розширення дискового масиву буде додано розділ що буде пропонувати послуги з хмарного збереження інформації.

Посилання на сайт:

<https://kuponom.com/dencalc/>

3.5.2 Наповнення сайту

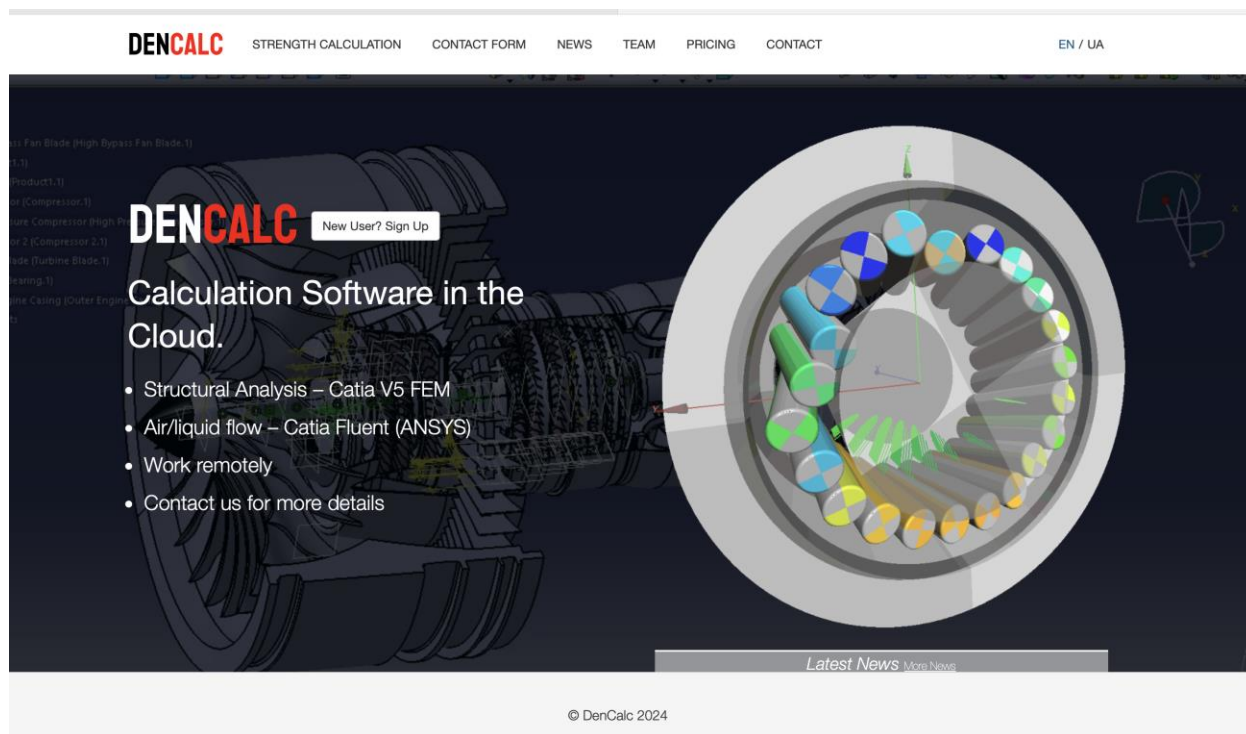


Рисунок 3.7 – Головна сторінка

На головній сторінці є загальний опис, опис можливостей сайту і програмного забезпечення з яким можлива робота на момент релізу сайту.

STRENGTH CALCULATION

We do it very easy!

- Upload model in any format
 - Provide with calculation data and rules
 - CATIA interface
 - Don't afraid - mistakes are allowed! Make endless changes in model and recalculations
-
- Static and dynamic analysis in the CATIA FEM shell addon
 - Creating video animation of processes plus different options of vibration and spectrum analysis representation
 - Generation of CATIA FEM add-on reports



See DenCalc's analysis capabilities in action

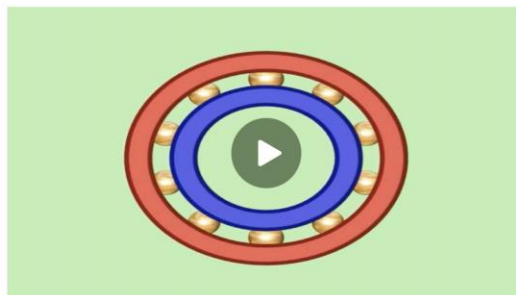


Рисунок 3.8 – Розділ розрахунків

Тут вказано формат даних які мають надсилатися, програмне забезпечення яке буде використовуватися, і розрахунки які можуть бути виконані зі звітом. Показано анімацію зі зміною конструкції підшипника масляного насосу і анімація підшипника що відображає сутність радіального зазору.

CONTACT FORM

Let's connect!

Personal Information

First Name:

Last Name:

Email:

Рисунок 3.9 – Форма для звязку

DenCalc News

Latest

- [Web site start running December 2024.](#)

The DenCalc Team

Built by fair engineers, for fair engineers.



Denis Voloshchuk
Founder

CATIA V5 ANSYS SolidWorks
SolidEdge AutoCAD

Рисунок 3.10 – Новини та команда

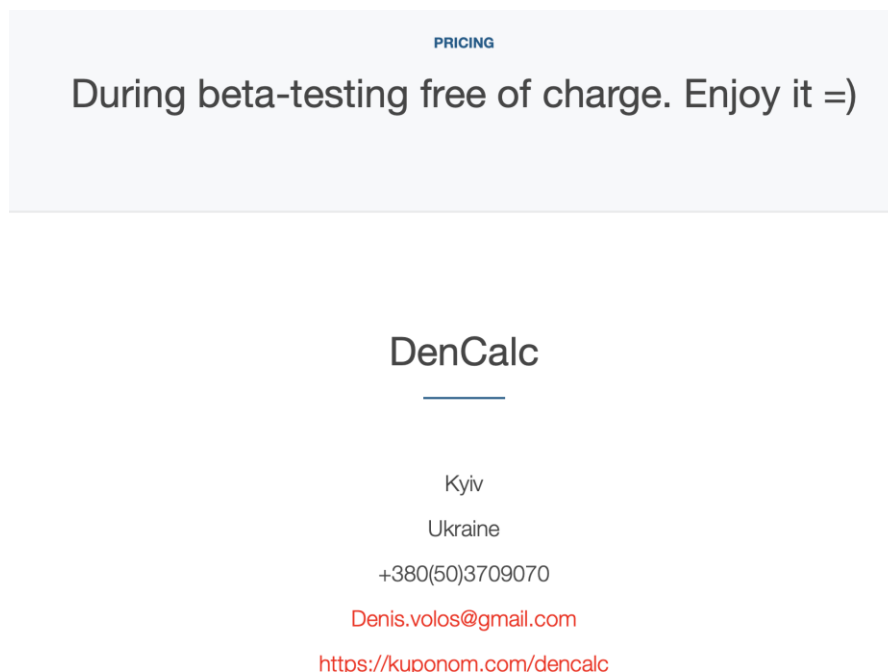


Рисунок 3.11 – Ціни та контактна інформація

3.6 Подальші плани

Сайт функціонує і згодом його можливості буде розширено разом з більшою пропозицією щодо програмного забезпечення для розрахунків та прямого доступу після реєстрації.

Разом з цим планується додати додаткові 2 модулі кластеру до запланованих 2 оскільки є сумніви щодо достатності обчислюваних ресурсів для підтримання віртуальних машин з багатьма користувачами.

ВИСНОВОК

В роботі проведено глибокий аналіз видів і вимог до обчислювальних кластерів. Це дозволило зробити висновок що організація HPC Cluster в повній мірі відповідає поставленим задачам і дозволить надалі масштабувати систему.

Було досліджено ОС для роботи головного і додаткових вузлів кластеру. Визначено вид організації HDD/SSD накопичувачів для роботи як самих вузлів кластерів, так і файл сервера. Розглянуто безпосередній монтаж обладнання, налагодження високошвидкісного інтерфейсу комунікації між вузлами кластеру, описані мови програмування та інструментарій для запуску веб-сайту для доступу до кластеру.

Представлено та описано види кластерів по задачам на які вони розраховані.

У цій роботі описано кілька варіантів реалізації мережі для приватних мереж на основі специфікацій 3GPP. Вони варіюються від повністю самостійних автономних, які не мають підключення до загальнодоступної мережі, до приватних мереж, які повністю розміщені операторами загальнодоступних мереж. Зроблено аналіз доцільності вибору кожного з варіантів в залежності від специфіки і розміру підприємства.

В тезах було описана можливість роботи кластеру на відновлюваних джерелах енергетики.

ПЕРЕЛІК ПОСИЛАНЬ

1. Комп'ютерний кластер – Wiki ТНТУ. *Wiki ТНТУ*.
URL: https://wiki.tntu.edu.ua/Комп'ютерний_кластер
2. www.wikidata.uk-ua.nina.az. Кластер (комп'ютерна техніка).
www.wikidata.uk-ua.nina.az. URL: [https://www.wikidata.uk-ua.nina.az/Кластер_\(комп'ютерна_техніка\).html](https://www.wikidata.uk-ua.nina.az/Кластер_(комп'ютерна_техніка).html)
3. Кластерні обчислення: архітектура та її типи-Electron-FMUSER
FM/TV Broadcast One-Stop постачальник. FMUSER FM / TV
трансляція Одноразовий постачальник. URL:
<https://uk.fmuser.net/content/?11645.html>
4. Chapter 9. Clustering | Red Hat Product Documentation. Red Hat
Product Documentation. URL:
[https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/
7/html/7.0_release_notes/chap-red_hat_enterprise_linux-
7.0_release_notes-clustering#chap-Red_Hat_Enterprise_Linux-
7.0_Release_Notes-Clustering](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/7.0_release_notes/chap-red_hat_enterprise_linux-7.0_release_notes-clustering#chap-Red_Hat_Enterprise_Linux-7.0_Release_Notes-Clustering)
5. Azure Machine Learning. Microsoft Learn: Build skills that open
doors in your career. URL: [https://learn.microsoft.com/ru-
ru/Azure/machine-learning/how-to-create-attach-compute-
cluster?view=azureml-api-1&tabs=python](https://learn.microsoft.com/ru-ru/Azure/machine-learning/how-to-create-attach-compute-cluster?view=azureml-api-1&tabs=python).
6. Benito B. M. Setting up a home cluster | Blas M. Benito, PhD. *Blas
M. Benito, PhD*.
URL: https://www.blasbenito.com/post/01_home_cluster/
7. HPC Cluster – Documentation for Clear Linux* project. Home Clear
Linux* Project | Clear Linux* Project. URL:

<https://www.clearlinux.org/clear-linux-documentation/tutorials/hpc.html>

8. Zamani N. CATIA V5 FEA Tutorials, Release 15. SDC Publications, 2006. 508 p.
9. Пример построения отказоустойчивого кластера | Статьи | Компьютерное Обозрение. Компьютерное Обозрение. URL: https://ko.com.ua/primer_postroeniya_otkazoustojchivogo_klastera_126576
10. Найкращі дистрибутиви GNU / Linux 2020 року для вашого ПК. Linux Adictos. URL: https://uk.linuxadictos.com/найкращі-дистрибутиви-gnu-linux-2020.html#google_vignette
11. Загальні відомості про обчислювальні кластери - Про кластери - Суперкомп'ютер ІК НАН України. Головна - Суперкомп'ютер ІК НАН України. URL: http://icybcluster.org.ua/index.php?lang_id=2&menu_id=6
12. Что такое Linux? История создания Linux / Ravesli. Ravesli. URL: <https://ravesli.com/chto-takoe-linux-ego-struktura-i-preimushhestva/>
13. Єрмолова М. RAID-масив: що це таке і який тип рейду обрати саме тобі?. *Magicuneraser – Advanced Data Recovery Software Provider*. URL: https://www.magicuneraser.com/ua/press/types_of RAID_arrays.php
14. PowerChute available for Ubuntu 20 ?. Schneider Electric Community. URL: <https://community.se.com/t5/APC-UPS-Data-Center-Enterprise/PowerChute-available-for-Ubuntu-20/td-p/337993>

15.Figma: The Collaborative Interface Design Tool. Figma. URL:
<https://www.figma.com>

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Обчислювальний кластер на основі відкритого програмного забезпечення для надання хмарних сервісів»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав: здобувач вищої освіти гр. ІСДМ-64

Денис ВОЛОЩУК

Керівник: доктор філософії

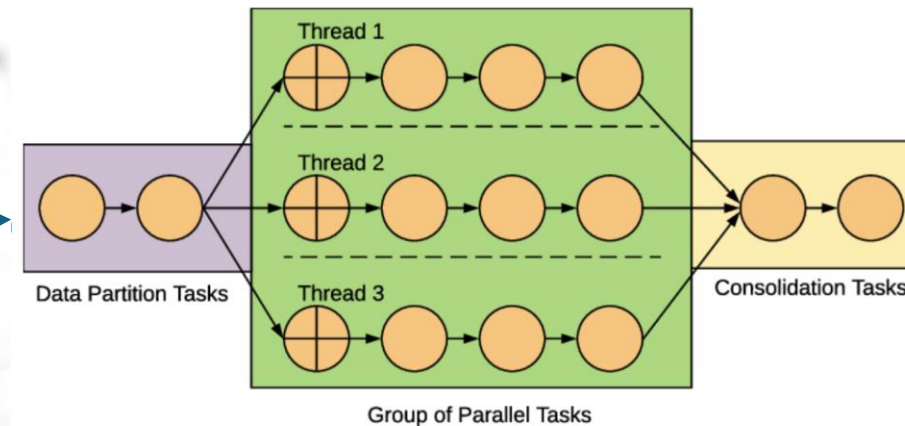
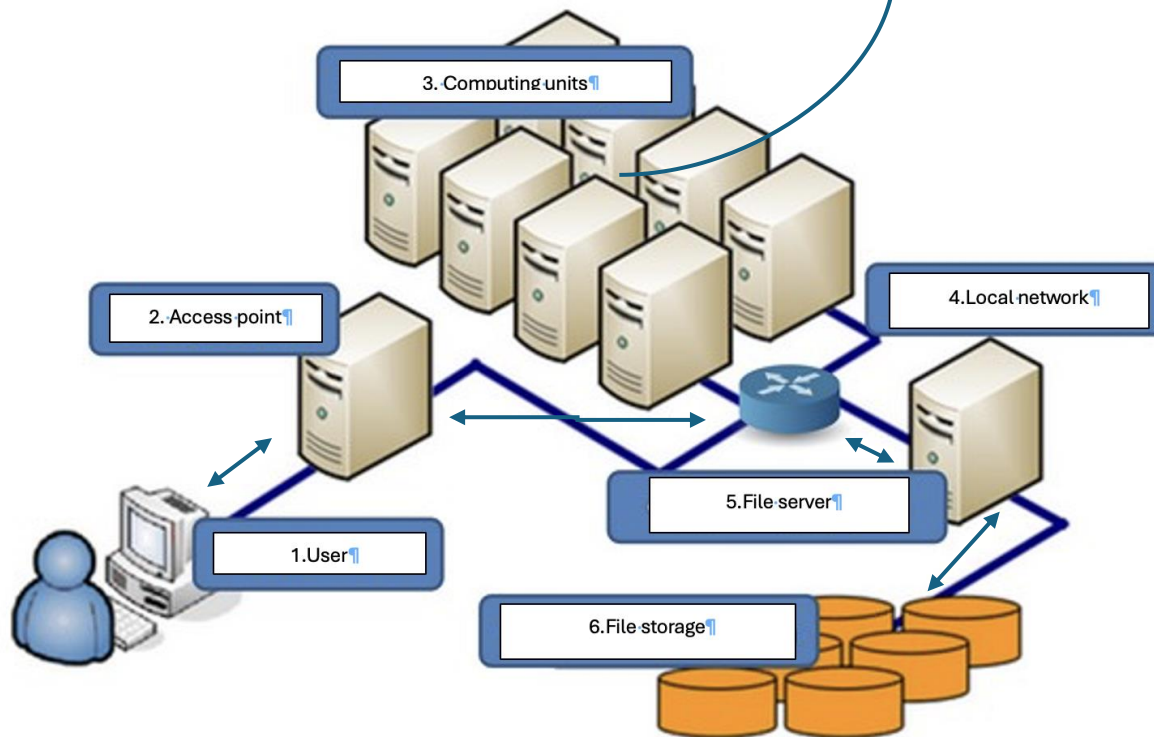
Віктор САГАЙДАК

Актуальність теми:

Розвиток технологій для фізичних розрахунків міцності та потоків газів і рідин, та їх взаємодії з твердими тілами в динаміці вимагає постійного нарощування обчислювальних можливостей. Одним з шляхів розвитку нарощування обчислювальних можливостей є кластеризація, де потужності двох або набагато більше комп'ютерів поєднуються. На відміну від корпоративних кластерів – домашній кластер вимагає ще відповідати і балансуванню між можливостями і енергоефективністю. Що також критично важливо в теперішні часи в Україні як з огляду на надійність подачі електроенергії так і за ціну за кВт. Актуальність теми стала важливою для автора даної роботи як його додаткова діяльність і завдання, які пов'язані з математичним і фізичним розрахунком, аналізом конструкції і взаємодією між її елементами, а також використання серверу для інших користувачів як на платній так і на безплатній основі.

Об'єкт дослідження:	Процес побудови кластеру і сайту для доступу до нього.
Предмет дослідження:	Архітектура кластерів і ПО для його роботи.
Мета дослідження:	Дослідження можливості побудови і запуску обчислювального кластеру та надання доступу іншим користувачам до нього в періоди не виконання ним основних
Наукова новизна та практична значущість:	У магістерській роботі розглянуто всі можливі способи організації обчислювальних кластерів та зроблено практичні висновки і рекомендації щодо його роботи. Отримані в магістерській роботі результати будуть використані при побудові кластеру з двох машин для обчислення, з подальшим можливим розширенням кількості вузлів.

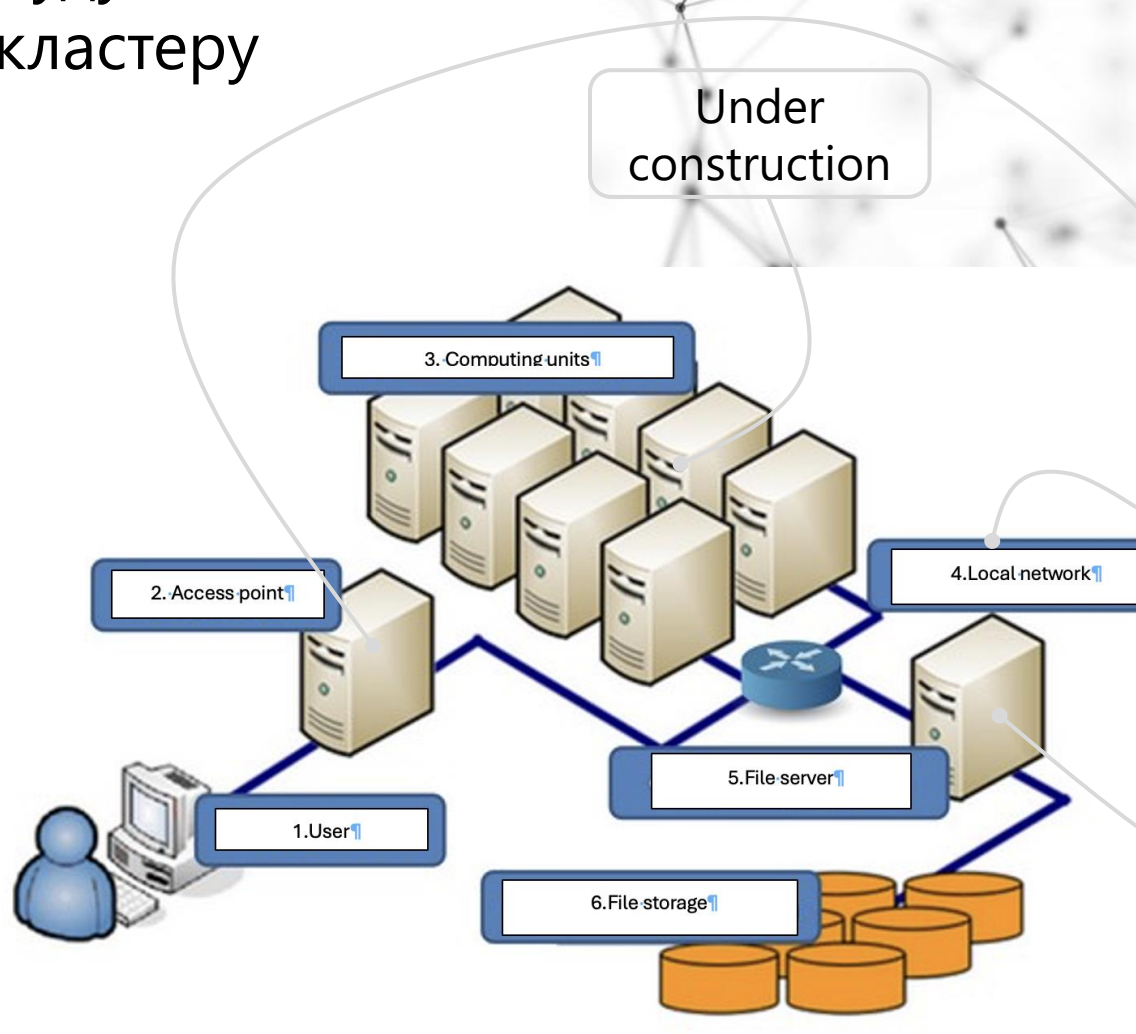
Теорія роботи кластеру



Кластер побудовано на моделі НРС
Користувач через будь яку ОС з визначеними правами отримує доступ до кластеру або його частини

Дані надходять на точку доступу де безпосередньо або в віртуальній машині відбувається розпаралелювання задачі
Проміжні дані зберігаються на дисковому під-масиві самих вузлів і на масиві дисків файл сервера

Результат будування кластеру



1. User. Через локальну мережу або з правами доступу

2. Access point.

3. Computing unit.

4. Local network.

5. File server.



Подальші плани:

- встановлення і налаштування додаткових двох вузлів кластеру
- встановлення і налаштування додаткового дискового масиву виключно для потреб кластеру
- встановлення і налаштування окремого дискового масиву з високим ступенем захисту від збоїв для резервного копіювання
- встановлення і налаштування оптоволоконної мережі для зв'язку між вузлами (SFP+)

Програмне
забезпечення для
роботи кластеру і веб

сайту
Дизайн сайту:

Figma

Таблиці стилів та мова
програмування:

CSS та PHP

Мова розмітки веб
сторінки та додатки:

HTML та JavaScript

Збереження та робота з
даними:

RAID 1 для HDD та SSD для короткотривалого зберігання. RAID 5 для загального зберігання

ПО для віртуалізації:

VirtualBox

Протокол зв'язку між
вузлами:

Message Passing Interface (MPI)

Операційна система для
організації кластеру:

Linux

Наповнення веб сайту

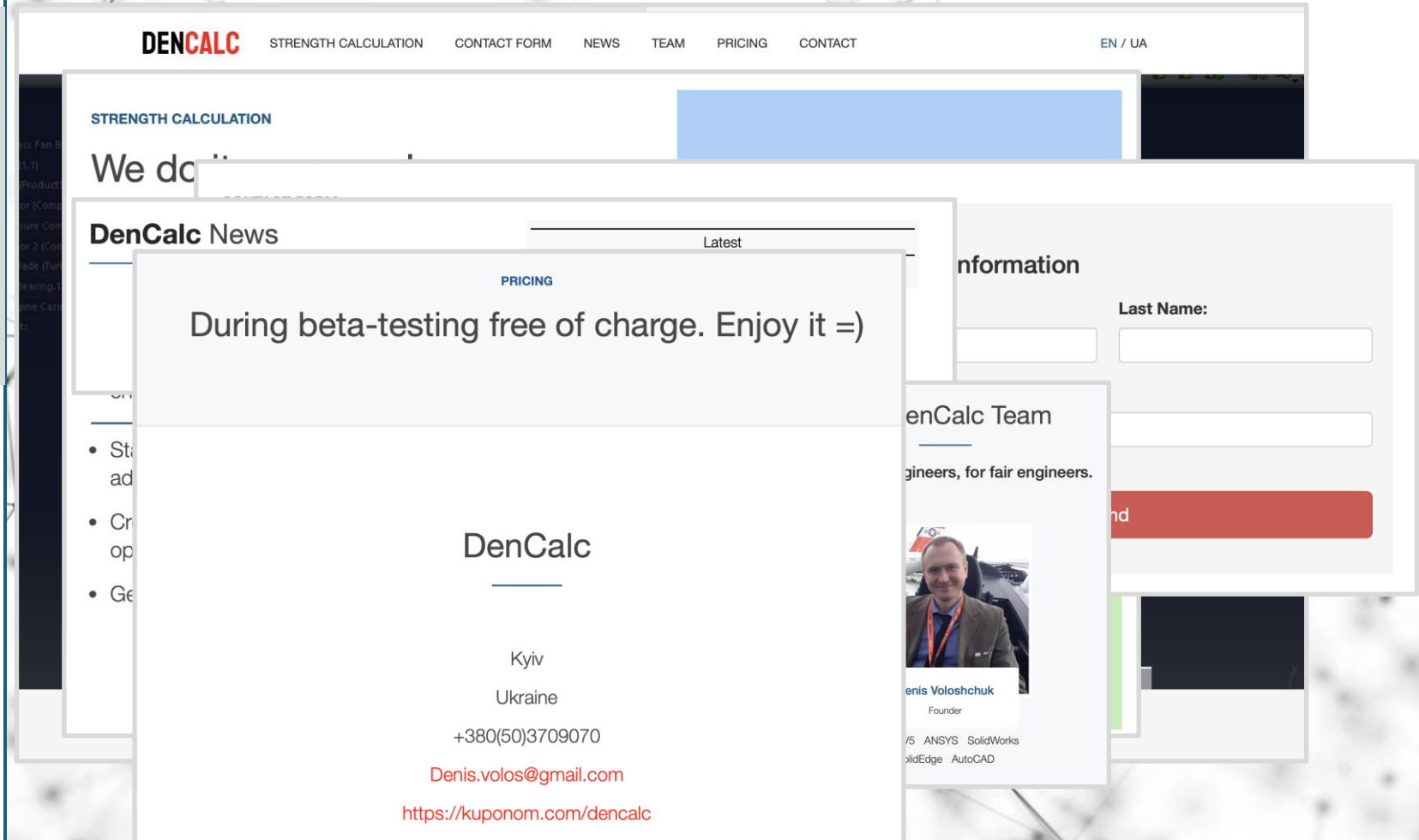
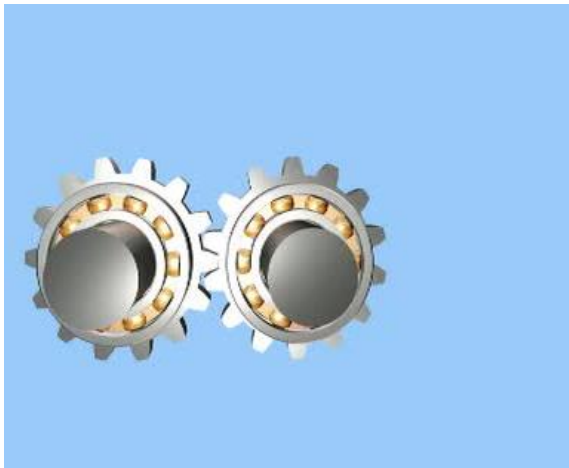
Головна сторінка:

Розділ розрахунків:

Форма для зв'язку:

Новини та команда:

Ціни та контактна інформація:



<https://kuponom.com/dencalc/>

Подальші плани:

- додати модуль для оформлення платежів за користування кластером
- додати опції по оренді дискового простору
- сторінка статистики роботи кластеру та графік навантаження
- плагін статистики використаної електроенергії

Висновки:

В роботі проведено глибокий аналіз видів і вимог до обчислювальних кластерів. Це дозволило зробити висновок що організація HPC Cluster в повній мірі відповідає поставленим задачам і дозволить надалі масштабувати систему.

Було досліджено ОС для роботи головного і додаткових вузлів кластеру. Визначено вид організації HDD/SSD накопичувачів для роботи як самих вузлів кластерів, так і файл сервера. Розглянуто безпосередній монтаж обладнання, налагодження високошвидкісного інтерфейсу комунікації між вузлами кластеру, описані мови програмування та інструментарій для запуску веб-сайту для доступу до кластеру.

Представлено та описано види кластерів по задачам на які вони розраховані.

Апробація:

- II Всеукраїнська Науково-Технічна Конференція Технологічні горизонти: Дослідження та застосування інформаційних технологій для технологічного прогресу України і світу.
- Науково-практична конференція «Проблеми комп'ютерної інженерії».

Дякую за увагу!