

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка високопродуктивної мережі на основі
дистрибутива Gentoo Linux»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

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

(підпис)

Івахненко Сергій
(Ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група КНД-41

Івахненко Сергій

Керівник: Серіх Сергій Олександрович
науковий ступінь,
вчене звання

к. т. н. доцент

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРІЗВИЩЕ)

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедру Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
«_____» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Івахненко Сергію Валерійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка високопродуктивної мережі на основі дистрибутива Gentoo Linux

керівник кваліфікаційної роботи Сергій Серих к. т. н., доцент,

(Ім'я, ПРИЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» 02.2024р. №36

2. Строк подання кваліфікаційної роботи «31» травня 2024р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література Розробка високопродуктивної мережі на основі дистрибутива Gentoo Linux.

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

Розробка високопродуктивної мережі на основі дистрибутива Gentoo Linux
Проектуванням, реалізацією та оптимізацією мережі

5. Перелік графічного матеріалу: *презентація*

Вступ та актуальність теми

1. Проектування мережі та вибір технологій
2. Встановлення та налаштування Gentoo Linux
3. Оптимізація та безпека мережі
4. Тестування та аналіз продуктивності
5. Висновки та рекомендації
6. Дата видачі завдання «23» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	26.02 – 04.03.24	
2	Вивчення матеріалів для аналізу розвитку технологій високопродуктивної мережі	07.03 – 12.03.24	
3	Дослідження моделей побудови	14.03 – 20.03.24	
4	Аналіз особливостей Gentoo	22.03 – 28.03.24	
5	Дослідження технологій високопродуктивної мережі	02.04 – 08.04.24	
6	Застосування дистрибутиву Linux Gentoo для побудови мережі	10.04 – 16.04.24	
7	Оформлення роботи: вступ, висновки, реферат	18.04 – 24.04.24	
8	Розробка демонстраційних матеріалів	28.04 – 05.05.24	

Здобувач вищої освіти

(підпис)

Сергій Івахненко

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Сергій Серих

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 63 стор., 1 табл., 1 рис., 7 джерел.

Наукове завдання – Розробка та впровадження методів оптимізації продуктивності та безпеки для високопродуктивних мереж на основі Gentoo Linux.

Мета роботи – розробка високопродуктивної мережі на основі Gentoo Linux.

Об'єкт дослідження – Процес розробки та впровадження високопродуктивної мережі на базі операційної системи Gentoo Linux.

Предмет дослідження – Особливості побудови, налаштування та оптимізації мережевої інфраструктури на базі операційної системи Gentoo Linux з метою забезпечення високої продуктивності та ефективності функціонування.

Короткий зміст роботи: проведення досліджень та розробку високопродуктивної мережі, використовуючи операційну систему Gentoo Linux. Робота включає аналіз вимог до мережі, вибір оптимального обладнання та програмного забезпечення, а також проведення налаштувань та оптимізації для досягнення найкращої продуктивності та надійності мережі.

КЛЮЧОВІ СЛОВА: ВИСОКОПРОДУКТИВНА МЕРЕЖА, GENTOO LINUX, МЕРЕЖЕВІ ТЕХНОЛОГІЇ, НАЛАШТУВАННЯ МЕРЕЖІ, ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ

ABSTRACT

The textual part of the qualification work for obtaining a bachelor's degree: 63 pages, 1 table, 1 figure, 7 sources.

Scientific task – Development and implementation of methods for optimizing performance and security for high-performance networks based on Gentoo Linux.

Purpose of the work – development of a high-performance network based on Gentoo Linux.

Object of study – The process of developing and implementing a high-performance network based on the Gentoo Linux operating system.

Subject of study – Features of building, configuring, and optimizing network infrastructure based on the Gentoo Linux operating system to ensure high performance and efficient operation.

Summary of the work: conducting research and developing a high-performance network using the Gentoo Linux operating system. The work includes an analysis of network requirements, the selection of optimal hardware and software, as well as configuration and optimization to achieve the best performance and reliability of the network.

KEYWORDS: HIGH-PERFORMANCE NETWORK, GENTOO LINUX, NETWORK TECHNOLOGIES, NETWORK CONFIGURATION, PERFORMANCE OPTIMIZATION

ЗМІСТ

ВСТУП.....	9
Мета роботи.....	8
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	13
1.1 Аналіз існуючих високопродуктивних мережесих рішень.....	13
1.2 Огляд технологій, що використовуються в мережесих операційних системах.....	14
1.3 Порівняння різних дистрибутивів Linux для побудови високопродуктивних мереж.....	14
1.4 Gentoo Linux як платформа для високопродуктивних мереж.....	15
2 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ GENTOO LINUX	177
2.1 Особливості Gentoo Linux.....	17
2.2 Оптимізація та налаштування Gentoo для специфічних завдань.....	19
2.3 Minimal Installation CD	22
2.4 Встановлення архіву stage	27
2.5 Встановлення базової системи.....	29
2.6 Встановлення системних інструментів.....	34
2.7 Налаштування початкового завантажувача	34
3 ОПТИМІЗАЦІЯ, ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ МЕРЕЖІ	36
3.1 Налаштування ядра Linux для високопродуктивних мережі.....	36
3.2 Оптимізація мережесих стеків і протоколів.....	37
3.3 Використання додаткових програмних засобів для покращення продуктивності.....	38
3.4 Тестування та аналіз продуктивності.....	40
3.5 Методологія тестування.....	42

3.6 Планування тестування	42
3.7 Інструменти для вимірювання продуктивності	43
3.8 Результати тестування	47
3.9 Впровадження і підтримка мережі	49
3.10 Підтримка та моніторинг мережі.....	50
3.11 Оновлення та масштабування мережі	50
ВИСНОВКИ	53
ПЕРЕЛІК ЛІТЕРАТУРИ	55
ПРЕЗЕНТАЦІЯ.....	55

ВСТУП

У сучасному світі інформаційні технології розвиваються надзвичайно швидкими темпами, що вимагає надійних і високопродуктивних мереж для ефективної передачі великих обсягів даних. Використання Gentoo Linux для побудови таких мереж є перспективним напрямком завдяки його гнучкості та можливостям тонкого налаштування. Це дозволяє створювати мережеві рішення, які максимально адаптовані до специфічних вимог і потреб користувачів.

Gentoo Linux відома своєю здатністю до високої оптимізації та персоналізації, що є важливою перевагою в умовах постійно зростаючих вимог до продуктивності та безпеки мереж. Високопродуктивні мережі є критичним компонентом для підтримки сучасних інформаційних систем, таких як хмарні обчислення, великі дані та Інтернет речей (IoT). Стабільна та швидка робота мережевої інфраструктури безпосередньо впливає на конкурентоспроможність організацій.

Ця робота присвячена дослідженню та розробці високопродуктивної мережі на основі Gentoo Linux. Вона охоплює визначення вимог до мережі, вибір необхідного обладнання та програмного забезпечення, налаштування та оптимізацію системи для досягнення найкращих результатів у продуктивності та надійності мережі.

Мета роботи

Метою цієї роботи є розробка високопродуктивної мережі на основі операційної системи Gentoo Linux, яка забезпечить високу продуктивність, надійність та безпеку для сучасних підприємств та організацій.

Завдання роботи

Для досягнення цієї мети необхідно виконати такі завдання:

1. Вивчення існуючих рішень для побудови високопродуктивних мереж:
 - Провести огляд та аналіз сучасних високопродуктивних мережевих рішень.

- Визначити ключові вимоги та параметри, які впливають на продуктивність мережі.
2. Аналіз можливостей та переваг використання Gentoo Linux для мережесих рішень:
 - Дослідити особливості та можливості Gentoo Linux для побудови високопродуктивних мереж.
 - Порівняти Gentoo Linux з іншими дистрибутивами Linux щодо продуктивності та гнучкості налаштування.
 3. Розробка вимог до мережі та вибір відповідного обладнання і програмного забезпечення:
 - Визначити функціональні та нефункціональні вимоги до мережі.
 - Провести вибір серверного та мережевого обладнання, яке відповідає встановленим вимогам.
 - Вибрати програмне забезпечення, яке забезпечить необхідну функціональність мережі.
 4. Налаштування і оптимізація Gentoo Linux для забезпечення високої продуктивності:
 - Виконати встановлення Gentoo Linux на обране обладнання.
 - Провести налаштування системних параметрів та мережесих інтерфейсів.
 - Оптимізувати ядро та мережесі стеки для досягнення максимальної продуктивності.
 5. Тестування та аналіз продуктивності розробленої мережі:
 - Розробити план тестування та критерії оцінки продуктивності мережі.
 - Використати спеціалізовані інструменти для вимірювання продуктивності.
 - Провести аналіз отриманих даних та зробити висновки щодо ефективності обраного рішення.
 6. Документування процесу розробки та результатів:
 - Підготувати детальну документацію, що описує всі етапи розробки мережі.
 - Представити результати тестування у вигляді таблиць та графіків.

- Надати рекомендації щодо подальшого вдосконалення та масштабування мережі.

Методологія дослідження

Методологія дослідження включає комплекс підходів та методів, які дозволяють систематично та ефективно досягти мети роботи. Основні методи, які використовуються у даній роботі, охоплюють як теоретичні, так і практичні аспекти.

1. Аналіз наукової та технічної літератури:

- Вивчення сучасних публікацій, статей та технічної документації, присвячених високопродуктивним мережам та використанню Gentoo Linux.
- Огляд існуючих рішень у сфері мережевих технологій та їх порівняльний аналіз.

2. Практичні експерименти:

- Встановлення та налаштування Gentoo Linux на тестовому обладнанні.
- Проведення серії експериментів для вивчення впливу різних налаштувань та параметрів на продуктивність мережі.
- Впровадження та тестування різних мережевих конфігурацій.

3. Проектування та моделювання мережевої інфраструктури:

- Розробка логічної та фізичної архітектури мережі.
- Моделювання мережевих сценаріїв для оцінки їх ефективності та надійності.

4. Оптимізація системних параметрів:

- Налаштування ядра Linux та мережевих стеків для забезпечення максимальної продуктивності.
- Використання інструментів для моніторингу та аналізу продуктивності системи.
- Впровадження та тестування додаткових методів оптимізації, таких як кешування та використання проксі-серверів.

5. Тестування продуктивності:

- Розробка плану тестування мережі з визначенням основних критеріїв оцінки продуктивності.
- Використання спеціалізованих інструментів для вимірювання продуктивності мережі, таких як `iperf`, `netperf` та інші.
- Проведення тестів для визначення пропускної здатності, затримок, стабільності та інших ключових показників.

6. Аналіз отриманих результатів:

- Обробка та аналіз даних, отриманих під час тестування.
- Порівняння результатів з початковими вимогами та очікуваннями.
- Висновки щодо ефективності та надійності розробленої мережевої інфраструктури.

7. Документування процесу та результатів дослідження:

- Детальне документування всіх етапів роботи, включаючи налаштування системи, проведені експерименти та отримані результати.
- Підготовка звітів та презентацій для представлення результатів дослідження.

Ця методологія дозволяє всебічно дослідити питання побудови високопродуктивної мережі на основі Gentoo Linux, забезпечити її оптимальну продуктивність та надійність, а також надати рекомендації для подальшого розвитку та вдосконалення мережевої інфраструктури.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Огляд літератури є важливим етапом будь-якого дослідження, оскільки дозволяє зрозуміти поточний стан справ у вибраній галузі, виявити існуючі проблеми та знайти можливі шляхи їх вирішення. У цьому розділі розглядаються сучасні наукові та технічні публікації, присвячені високопродуктивним мережам та використанню операційної системи Gentoo Linux для побудови мережесх рішень.

1.1 Аналіз існуючих високопродуктивних мережесх рішень

Сучасні високопродуктивні мережесх рішення часто використовують різні дистрибутиви Linux через їхню гнучкість, масштабованість та можливості налаштування. Дослідження показують, що дистрибутиви Linux, такі як Debian, CentOS, Ubuntu та інші, широко застосовуються у побудові серверних та мережесх рішень.

- Debian та його похідні:
- Debian є одним з найстаріших і найстабільніших дистрибутивів Linux. Він забезпечує високу надійність та безпеку, що робить його популярним серед серверних рішень.
- Ubuntu, як похідний від Debian, також широко використовується завдяки своїй простоті у налаштуванні та великій спільноті підтримки.
- Red Hat та його похідні:
- Red Hat Enterprise Linux (RHEL) є комерційним дистрибутивом, відомим своєю надійністю та підтримкою з боку виробника. CentOS, як безкоштовна альтернатива RHEL, також користується великою популярністю у серверних середовищах.
- Fedora, ще один похідний від RHEL, часто використовується для тестування нових технологій перед їх впровадженням у RHEL.
- Arch Linux:

- Arch Linux, як і Gentoo, орієнтований на користувачів, які бажають мати повний контроль над своєю системою. Він дозволяє налаштовувати систему відповідно до специфічних потреб користувача.

1.2 Огляд технологій, що використовуються в мережевих операційних системах

Сучасні мережеві операційні системи включають безліч технологій, які забезпечують високу продуктивність, надійність та безпеку.

1. Контейнери та віртуалізація:

- Docker та Kubernetes забезпечують ізоляцію додатків і сприяють ефективному використанню ресурсів.
- Віртуалізація з використанням KVM, Xen та інших гіпервізорів дозволяє створювати віртуальні мережеві середовища.

2. Програмно-визначувані мережі (SDN):

- SDN дозволяє централізовано керувати мережею, покращуючи її гнучкість та масштабованість.

3. Автоматизація та управління конфігураціями:

- Інструменти як Ansible, Puppet та Chef використовуються для автоматизації налаштувань та управління конфігураціями мережевих пристроїв.

1.3 Порівняння різних дистрибутивів Linux для побудови високопродуктивних мереж

Вибір дистрибутива Linux для побудови високопродуктивної мережі залежить від багатьох факторів, включаючи вимоги до продуктивності, гнучкості, підтримки та безпеки.

1. Debian/Ubuntu:

- Переваги: стабільність, велика спільнота, довготривала підтримка.

- Недоліки: може вимагати додаткових налаштувань для досягнення високої продуктивності.

2. CentOS/RHEL:

- Переваги: надійність, комерційна підтримка, широке використання в корпоративному середовищі.
- Недоліки: обмежені можливості налаштування в порівнянні з Gentoo або Arch.

3. Arch Linux:

- Переваги: простота налаштування, актуальність пакетів, велика документація.
- Недоліки: потребує високого рівня знань для налаштування та підтримки.

4. Gentoo Linux:

- Переваги: максимальна гнучкість і можливості налаштування, оптимізація під конкретні потреби.
- Недоліки: потребує значного часу та знань для початкового налаштування та підтримки.

1.4 Gentoo Linux як платформа для високопродуктивних мереж

Gentoo Linux виділяється серед інших дистрибутивів завдяки своїй можливості повної персоналізації та оптимізації під конкретні завдання. Вона дозволяє компілювати всі програми безпосередньо зі вихідного коду, використовуючи тільки необхідні компоненти, що забезпечує максимальну продуктивність.

1. Особливості Gentoo Linux:

- Використання портежів (Portage) для управління пакетами.
- Підтримка широкого спектру архітектур та можливість глибокої оптимізації.

2. Переваги використання Gentoo для мережесих рішень:

- Можливість налаштування системи під конкретні потреби користувача.
 - Висока продуктивність та ефективність завдяки оптимізації.
3. Оптимізація та налаштування Gentoo для специфічних завдань:
- Тонке налаштування ядра Linux та мережевих стеків.
 - Використання спеціалізованих патчів та налаштувань для підвищення продуктивності.

Огляд літератури та існуючих рішень дозволяє краще зрозуміти контекст та актуальність вибраної теми дослідження, визначити основні підходи та методи, які можуть бути використані для досягнення поставленої мети.

2 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ GENTOO LINUX

2.1 Особливості Gentoo Linux

Gentoo Linux є особливим дистрибутивом Linux, який виділяється серед інших завдяки своїй гнучкості та можливості глибокої персоналізації. Основні особливості Gentoo Linux включають:

1. Пакетний менеджер Portage

Portage є ключовим компонентом Gentoo Linux і забезпечує управління пакетами та залежностями. Основні риси Portage включають:

- Система компіляції з вихідного коду: Portage компілює всі програми безпосередньо з вихідного коду, що дозволяє оптимізувати їх під конкретні потреби та архітектуру.
- USE-флаги: спеціальні флаги, які визначають опції компіляції та функціональність пакетів. Це дозволяє виключити непотрібні компоненти та зменшити обсяг встановленого програмного забезпечення.
- Ебілди (ebuilds): скрипти, які описують процес компіляції та встановлення програмного забезпечення. Ебілди забезпечують гнучкість і можливість налаштування на рівні вихідного коду.

2. Гнучкість та налаштування

Gentoo Linux дозволяє користувачам повністю контролювати процес налаштування та оптимізації системи:

- Налаштування ядра: користувачі можуть налаштовувати ядро Linux з точністю до найдрібніших деталей, вибираючи тільки ті модулі та функції, які необхідні для їхньої системи.
- Оптимізація для конкретної архітектури: під час компіляції користувачі можуть використовувати специфічні опції компілятора для оптимізації продуктивності під конкретну апаратну архітектуру.

3. Постійне оновлення

Gentoo Linux не використовує традиційних релізів, таких як у багатьох інших дистрибутивах. Замість цього він використовує модель постійного оновлення (rolling release), яка має кілька переваг:

- Актуальність програмного забезпечення: користувачі завжди мають доступ до найновіших версій програмного забезпечення без необхідності чекати на новий реліз дистрибутиву.
- Гнучкість оновлень: користувачі можуть оновлювати лише ті пакети, які їм потрібні, або весь дистрибутив одночасно.

4. Документація та спільнота

Gentoo Linux має одну з найдетальніших та найякісніших документацій серед дистрибутивів Linux. Основні особливості включають:

Gentoo Handbook: детальний посібник, який охоплює всі аспекти встановлення, налаштування та підтримки системи.

Форуми та спільнота: активна спільнота користувачів, яка завжди готова допомогти новачкам та обмінятися досвідом.

5. Безпека

Gentoo Linux надає інструменти для забезпечення високого рівня безпеки системи:

Hardened Gentoo: спеціальний проект, який включає патчі та налаштування для підвищення безпеки ядра та системних компонентів.

SELinux та AppArmor: підтримка додаткових модулів безпеки, які можуть бути інтегровані та налаштовані користувачами.

6. Підтримка різних архітектур

Gentoo Linux підтримує широкий спектр апаратних архітектур, включаючи x86, x86_64, ARM, PowerPC, SPARC та інші. Це робить його універсальним вибором для різноманітних пристроїв – від серверів до вбудованих систем.

Ці особливості роблять Gentoo Linux потужним інструментом для користувачів, які потребують високої продуктивності, гнучкості та контролю над своєю системою. Завдяки можливості тонкого налаштування та оптимізації, Gentoo є ідеальним вибором для побудови високопродуктивних мережесх рішень.

2.2 Оптимізація та налаштування Gentoo для специфічних завдань

Оптимізація та налаштування Gentoo Linux для специфічних завдань є одним із ключових аспектів, що робить цей дистрибутив особливо корисним для високопродуктивних мережесх рішень. Нижче наведено основні етапи та методи оптимізації та налаштування Gentoo Linux.

1. Налаштування ядра Linux

Налаштування ядра є критично важливим етапом для забезпечення високої продуктивності та надійності системи:

- Видалення непотрібних модулів: видалення невикористовуваних драйверів і модулів знижує навантаження на ядро та прискорює його роботу.
- Оптимізація параметрів ядра: включення специфічних оптимізацій для процесора, використання відповідних таймерів та інші налаштування можуть суттєво підвищити продуктивність.

2. Використання USE-флагів

USE-флаги дозволяють тонко налаштовувати компіляцію програмного забезпечення:

- Вибір необхідних компонентів: увімкнення лише тих функцій та компонентів, які потрібні для конкретних завдань, знижує обсяг встановленого програмного забезпечення та покращує його продуктивність.

- Оптимізація залежностей: налаштування залежностей програм забезпечує ефективне використання ресурсів.

3. Оптимізація мережевих параметрів

Налаштування мережевих параметрів є важливим для досягнення високої продуктивності мережі:

- Налаштування TCP/IP стеку: оптимізація параметрів TCP/IP для підвищення пропускної здатності та зниження затримок.
- Використання альтернативних протоколів: у деяких випадках використання протоколів, відмінних від стандартного TCP/IP, може забезпечити кращу продуктивність.

4. Встановлення та налаштування додаткового програмного забезпечення

Вибір та налаштування додаткових програмних засобів є важливим для специфічних завдань:

- Мережеві сервіси: налаштування DNS, DHCP, NAT, а також інших мережевих сервісів, необхідних для функціонування мережі.
- Інструменти моніторингу: встановлення та налаштування інструментів для моніторингу мережевої активності та продуктивності, таких як Nagios, Zabbix або Prometheus.

5. Оптимізація компіляції

Використання спеціальних опцій компіляції може значно підвищити продуктивність системи:

- CFLAGS та CXXFLAGS: налаштування цих параметрів для оптимізації під конкретну архітектуру процесора.
- LDFLAGS: налаштування параметрів компіляції для оптимізації використання пам'яті та підвищення швидкодії виконуваних файлів.

6. Використання кешування та проксі-серверів

Для підвищення ефективності та продуктивності мережі можуть бути використані різні методи кешування та проксі-сервери:

- Squid: налаштування проксі-сервера для збереження копій часто використовуваних ресурсів, що знижує навантаження на мережу.
- Memcached: використання системи кешування для прискорення доступу до часто використовуваних даних.

7. Безпека та захист

Налаштування безпеки є важливим аспектом для забезпечення надійної роботи мережі:

- Брандмауери та VPN: налаштування брандмауерів та VPN для захисту від зовнішніх та внутрішніх загроз.
- Регулярні оновлення: забезпечення своєчасного оновлення системи та встановленого програмного забезпечення для захисту від відомих вразливостей.

8. Тестування та аналіз продуктивності

Після завершення налаштування важливо провести тестування та аналіз продуктивності:

- Методологія тестування: розробка плану тестування для перевірки всіх аспектів мережевої продуктивності.
- Інструменти для вимірювання продуктивності: використання інструментів, таких як iPerf, для тестування пропускної здатності мережі та аналізу затримок.

Ці етапи та методи дозволяють ефективно налаштувати та оптимізувати Gentoo Linux для виконання специфічних завдань, забезпечуючи високу продуктивність, надійність та безпеку мережевої інфраструктури.

2.3 Minimal Installation CD

Для самого початку нам потрібен носій із образом Gentoo. Скачуємо з офіційного дзеркала файл «Minimal Installation CD». І приступаємо до встановлення базової системи. Як і з іншими операційними системами- треба обрати пристрій загрузки з образом дистрибутиву та завантажитись з нього. Для коректного встановлення на машину- треба в першу чергу звернути увагу на те, яка таблиця розділів актуальна для диску, на який ми будемо встановлювати систему. Найчастіше зустрічається DOS та GPT. В свою чергу DOS є застарілим і використовується на збірках старої комплектації, в той час як GPT являє собою більш сучасний варіант таблиці розділів.

Тепер перевіримо, які є підключені диски до сервера. Для цього використаємо команду «fdisk -l», яка покаже нам всі підключенні на даний етап диски а системі. Обираємо жорсткий диск, на який і будемо встановлювати Gentoo:

```
fdisk -l
```

```
Disk /dev/sda: 500.79 GiB, 537719168000 bytes, 1050232750 sectors
```

```
Disk model: VBOX HARDDISK
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Створювати розділи на диску будемо з MBR. Для цього треба запустити команду «fdisk» для нашого жорсткого диску, в даному прикладі це- sda:

```
fdisk /dev/sda
```

Створюємо нову розмітку MBR на диску, для цього використовуємо клавішу «o»:

```
Command (m for help): o
```

Created a new DOS (MBR) disklabel with disk identifier 0x0b203960.

Першим створимо розділ завантаження, що в подальшому буде монтуватись в /boot. Для цього використовуємо наступну послідовність команд:

1. «п»- створення нового розділу.
2. «р»- тип розділу
3. «1»- номер розділу
4. Перший сектор.
5. Другий сектор.
6. «t»- тип розділу
7. Номер розділу
8. Id типу розділа

Command (m for help): n

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-1050232749, default 2048):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-1050232749, default 1050232749):
+1G

Created a new partition 1 of type 'Linux' and of size 1 GiB.

Command (m for help): t

Selected partition 1

Hex code or alias (type L to list all): 83

Changed type of partition 'Linux' to 'Linux'.

Не забуваємо відмітити sda1, як завантажувальний розділ за допомогою команди «a»

Command (m for help): a

Partition number (1-3, default 3): 1

The bootable flag on partition 1 is enabled now.

Другим створимо розділ підкачки за таким самим алгоритмом, як і розділ завантаження. Проте він буде відрізнятися типом розділу та секторами:

Command (m for help): n

Partition type

p primary (1 primary, 0 extended, 3 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (2-4, default 2): 2

First sector (2099200-1050232749, default 2099200):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2099200-1050232749, default 1050232749): +4G

Created a new partition 2 of type 'Linux' and of size 4 GiB.

Command (m for help): t

Partition number (1,2, default 2): 2

Hex code or alias (type L to list all): 82

Changed type of partition 'Linux' to 'Linux swap / Solaris'.

І останнім створюємо кореневий розділ (swap). Так само за аналогією як і дивіться в минулих:

Command (m for help): n

Partition type

p primary (2 primary, 0 extended, 2 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (3,4, default 3): 3

First sector (10487808-1050232749, default 10487808):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (10487808-1050232749, default 1050232749):

Created a new partition 3 of type 'Linux' and of size 495.8 GiB.

Command (m for help): t

Partition number (1-3, default 3): 3

Hex code or alias (type L to list all): 83

Changed type of partition 'Linux' to 'Linux'.

Після створення всіх розділів перевіримо, чи вірно все відображається- для цього використаємо команду «р», за її допомогою ми зможемо побачити на які розділи, типи та сектори зараз розбитий наш диск.

Command (m for help): p

Disk /dev/sda: 500.79 GiB, 537719168000 bytes, 1050232750 sectors

Disk model: VBOX HARDDISK

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xdf0da4a4

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	2099199	2097152	1G	83	Linux
/dev/sda2		2099200	10487807	8388608	4G	82	Linux swap / Solaris
/dev/sda3		10487808	1050232749	1039744942	495.8G	83	Linux

Для того, щоб зберегти зміни, що ми внесли за допомогою fdisk- виконуємо команду «w».

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

Тепер час створити файлові системи на розділах. Для sda1 ми будемо беремо ext3, а для sda3- ext4.

```
livecd ~ # mkfs.ext3 /dev/sda1
```

```
mke2fs 1.47.0 (5-Feb-2023)
```

```
Creating filesystem with 262144 4k blocks and 65536 inodes
```

```
Filesystem UUID: 2bfea56a-b69d-49b8-95bd-b1374ee0027c
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840, 229376
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (8192 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

```
livecd ~ # mkfs.ext4 /dev/sda3
```

```
mke2fs 1.47.0 (5-Feb-2023)
```

```
Creating filesystem with 129968117 4k blocks and 32497664 inodes
```

```
Filesystem UUID: 437c5be4-c038-456e-959a-a5d080df3034
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
102400000
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (262144 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

Після створення фалових систем для двох перших розділів- варто активувати і розділ підкачки. Для цього використовуємо дів команди:

```
livecd ~ # mkswap /dev/sda2
```

```
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
```

```
no label, UUID=cb43cdba-2250-4bd4-9d78-03c4178fcb7f
```

```
livecd ~ # swapon /dev/sda2
```

Тепер створюємо та монтуємо кореневий розділ, де в подальшому і будемо продовжувати роботу:

```
livecd ~ # mkdir --parents /mnt/gentoo
```

```
livecd ~ # mount /dev/sda3 /mnt/gentoo
```

2.4 Встановлення архіву stage

Перед встановленням архіву слід перевірити чи коректні налаштування дати та часу на машині. Так як це може в подальшому проблемно впливати на компіляцію. Перевірити налаштування часу можна за допомогою команди «date»:

```
livecd ~ # date
```

```
Mon May 13 23:07:17 UTC 2024
```

У виводі команди можна побачити, що зараз стоїть не праильний час, так як на момент встановлення дистрибутиву дата та час- 14.05.2024 15:18. Для виправлення є два варіанти:

1. Вести в ручну- date MMDDHHmmYY.

Де MM- місяць, DD- день, HH- години, mm- хвилини і YY- рік.

2. Командою «chronyd -q», яка автоматизує процес встановлення дати та часу.

Ми ж використаємо команду chronyd -q та перевіримо кінечний результат за допомогою date:

```
livecd ~ # chronyd -q
```

```
livecd ~ # date
```

```
Tue May 14 12:18:02 UTC 2024
```

Після того, як впевнились, що дата та час коректні- можна перейти до подальшого встановлення Gentoo. На цьому етапі ми виконуємо наступну команду:

```
livecd ~ # cd /mnt/gentoo
```

```
livecd /mnt/gentoo #
```

За її допомогою ми переходимо до точки монтування, де буде розміщуватись коренева файлова система.

Тепер за допомогою команди `wget` скачуємо архів `stage` з офіційного дзеркала:

```
livecd /mnt/gentoo # wget https://distfiles.gentoo.org/releases/amd64/autobuilds/20240512T164852Z/stage3-amd64-openrc-20240512T164852Z.tar.xz
```

Після скачування в кореневу папку варто розпакувати `stage` за допомогою команди `tar xpvf`:

```
livecd /mnt/gentoo # tar xpvf stage3-*.tar.xz --xattrs-include='*.*' --numeric-owner
```

Тепер можна перейти до налаштування параметрів компіляції. Для більш швидкого встановлення системи перевіримо кількість ядер за допомогою команди `cat /proc/cpuinfo`:

```
livecd /mnt/gentoo # cat /proc/cpuinfo
```

Із виводу команди можемо побачити, що всього в машині 6 ядер. Тепер для швидшої компіляції в файлі `/etc/portage/make.conf` за допомогою текстового редактору приписуємо потрібну кількість ядер:

```
livecd /mnt/gentoo # vi /etc/portage/make.conf
```

В файлі вписуємо наступну строку:

```
da
```

Так як файл ми редагуємо в текстовому редакторі «vim», щоб зберегти зміну виконуємо команду- «:wq». Де «q»- це вийти з редактору, а «w»- зберегти зміни.

2.5 Встановлення базової системи

Першим етапом варто перейти в ізольоване середовище. Для цього треба примотувати необхідні розділи:

```
livecd /mnt/gentoo # mount --types proc /proc /mnt/gentoo/proc
```

```
livecd /mnt/gentoo # mount --rbind /sys /mnt/gentoo/sys
```

```
livecd /mnt/gentoo # mount --make-rslave /mnt/gentoo/sys
```

```
livecd /mnt/gentoo # mount --rbind /dev /mnt/gentoo/dev
```

```
livecd /mnt/gentoo # mount --make-rslave /mnt/gentoo/dev
```

```
livecd /mnt/gentoo # mount --bind /run /mnt/gentoo/run
```

```
livecd /mnt/gentoo # mount --make-slave /mnt/gentoo/run
```

Тепер коли всі розділи примонтовані та базове середовище встановлене, прийшов час увійти в нове середовище встановлення для цього виконуємо команду «chroot». Це означає, що сесія змінить свій кореневий каталог з поточного середовища встановлення на систему встановлення. Перехід в нове середовища робиться в три кроки:

1. Зміна кореневого каталогу з / (що знаходиться на установковому носії) на /mnt/gentoo/ (на розділах диска) за допомогою команди chroot.
2. Завантаження в пам'ять деяких параметрів з /etc/profile за допомогою команди source.
3. Зміна привітання командного рядка, щоб не забути, що ця сесія знаходиться в ізольованому середовищі.

```
livecd /mnt/gentoo # chroot /mnt/gentoo /bin/bash
```

```
livecd / # source /etc/profile
```

```
livecd / # export PS1="(chroot) ${PS1}"
```

```
(chroot) livecd / #
```

Першим після переходу в chroot треба примонтувати розділ завантаження. У випадку DOS це робиться за допомогою команди «mount /dev/sda1 /boot»:

```
(chroot) livecd / # mount /dev/sda1 /boot
```

Другим кроком є налаштування репозиторію ebuild-файлів Gentoo в /etc/portage/repos.conf/gentoo.conf. Цей файл містить інформацію, необхідну для завантаження та встановлення програмного забезпечення. Налаштування репозиторію виконується тим, що спочатку треба створити каталог repos.conf:

```
(chroot) livecd / # mkdir --parents /etc/portage/repos.conf
```

Та копіюємо файл конфігурації репозиторію Gentoo, що нам надає Portage:

```
(chroot) livecd / # cp /usr/share/portage/config/repos.conf
/etc/portage/repos.conf/gentoo.conf
```

Наступним кроком буде встановлення знімка репозиторію ebuild-файлів Gentoo. В ньому є колекція файлів, яка повідомляє Portage про доступні програми.

```
(chroot) livecd / # emerge-webrsync
```

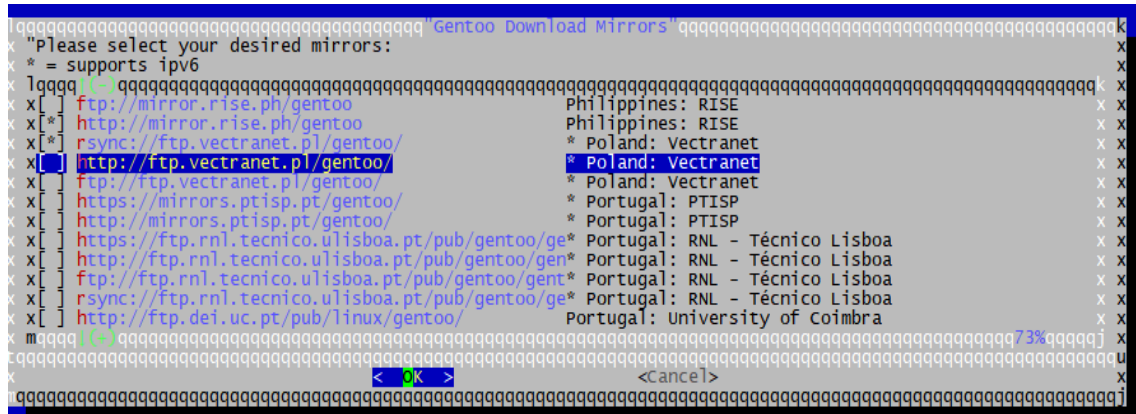
Тепер треба обрати дзеркало (рис. 2.1). Для цього потрібно встановити утиліту «mirrorselect»:

```
(chroot) livecd / # emerge --ask --verbose --oneshot app-portage/mirrorselect
```

Для швидкого завантаження вихідного коду рекомендується обрати дзеркала, які знаходяться географічно поруч:

```
mirrorselect -i -o >> /etc/portage/make.conf
```

Got 246 mirrors.



- [1] [default/linux/amd64/17.1 \(stable\)](#)
- [2] [default/linux/amd64/17.1/selinux \(stable\)](#)
- [3] [default/linux/amd64/17.1/hardened \(stable\)](#)
- [4] [default/linux/amd64/17.1/hardened/selinux \(stable\)](#)
- [5] [default/linux/amd64/17.1/desktop \(stable\)](#)
- [6] [default/linux/amd64/17.1/desktop/gnome \(stable\)](#)
- [7] [default/linux/amd64/17.1/desktop/gnome/systemd/merged-usr \(stable\)](#)
- [8] [default/linux/amd64/17.1/desktop/plasma \(stable\)](#)
- [9] [default/linux/amd64/17.1/desktop/plasma/systemd/merged-usr \(stable\)](#)
- [10] [default/linux/amd64/17.1/desktop/systemd/merged-usr \(stable\)](#)

- [11] default/linux/amd64/17.1/developer (exp)
- [12] default/linux/amd64/17.1/no-multilib (stable)
- [13] default/linux/amd64/17.1/no-multilib/hardened (stable)
- [14] default/linux/amd64/17.1/no-multilib/hardened/selinux (stable)
- [15] default/linux/amd64/17.1/no-multilib/systemd/merged-usr (dev)
- [16] default/linux/amd64/17.1/no-multilib/systemd/selinux/merged-usr (exp)
- [17] default/linux/amd64/17.1/systemd/merged-usr (stable)
- [18] default/linux/amd64/17.1/systemd/selinux/merged-usr (exp)
- [19] default/linux/amd64/17.1/clang (exp)
- [20] default/linux/amd64/17.1/systemd/clang/merged-usr (exp)
- [21] default/linux/amd64/23.0 (stable) *
- [22] default/linux/amd64/23.0/systemd (stable)
- [23] default/linux/amd64/23.0/desktop (stable)

У моєму випадку я обираю default/linux/amd64/23.0 (stable). Профіль, що буде обраний- помічається зірочкою:

```
(chroot) livedd / # eselect profile set 21
```

Наступним етапом варто оновити @world. Ця операція є необхідною для того, щоб система могла застосувати будь-які оновлення, які з'явилися після побудови stage3 та оновлення профілю. Проте перед оновленням варто встановити screen, бо іноді цей процес затягується. Для встановлення screen виконуємо наступну команду:

```
(chroot) livedd / # emerge --ask app-misc/screen
```

Після закінчення встановлення screen відкриваємо вікно за допомогою команди «screen» і оновлюємо @world:

```
(chroot) livedd / # screen
```

```
livedd / # emerge --ask --verbose --update --deep --newuse @world
```

Після оновлення @world виставляємо часовий пояс за допомогою команди:

```
livedd / # ln -sf ../usr/share/zoneinfo/Europe/Kyiv /etc/localtime
```

Та перезапускаємо оточення:

```
livedd / # env-update && source /etc/profile && export PS1="(chroot) ${PS1}"
```

Рекомендується встановити пакет sys-kernel/linux-firmware, щоб мати доступ до прошивки у разі необхідності:

```
(chroot) livedd / # emerge --ask sys-kernel/linux-firmware
```

Тепер можна перейти до етапу збірки ядра. Для збірки ядра з вихідного коду з патчами Gentoo використовуємо команду:

```
(chroot) livedd / # emerge --ask sys-kernel/gentoo-kernel
```

Так само встановлюємо вихідний код ядра за допомогою emerge:

```
(chroot) livedd / # emerge --ask sys-kernel/installkernel
```

Тепер перевіримо які ядра у нас встановлені та оберемо те, яке нас влаштовує та створюємо символічне посилання:

```
(chroot) livedd / # eselect kernel list
```

Available kernel symlink targets:

```
[1] linux-6.6.30-gentoo
```

```
(chroot) livedd / # eselect kernel set 1
```

```
(chroot) livedd / # ls -l /usr/src/linux
```

```
lrwxrwxrwx 1 root root 19 May 15 08:31 /usr/src/linux -> linux-6.6.30-gentoo
```

Тепер треба встановити пакет sys-kernel/genkernel:

```
(chroot) livecd / # emerge --ask sys-kernel/genkernel
```

Скомпілюйте вихідні тексти ядра, виконавши команду `genkernel all`. Майте на увазі, що, оскільки `genkernel` компілює ядро, що підтримує широкий набір апаратних засобів для різних архітектур комп'ютерів, процес компіляції може зайняти досить багато часу, тому краще виконувати його в `screen`:

```
(chroot) livecd / # genkernel --mountboot --install all
```

2.6 Встановлення системних інструментів

Встановимо обраний пакет для ведення журналу та додаємо його автозапуск:

```
(chroot) livecd / #emerge --ask app-admin/sysklogd
```

```
(chroot) livecd / # rc-update add sysklogd default
```

```
* service sysklogd added to runlevel default
```

2.7 Налаштування початкового завантажувача

В даному прикладі будемо використовувати Grub. Він легкий в налаштуванні і не потребує додаткових налаштувань. Скачуємо пакет за допомогою `emerge`:

```
(chroot) livecd / # emerge --ask --verbose sys-boot/grub
```

Так як ми працюємо із `sda`, тому на нього і треба встановити `grub`. Для встановлення на наш диск використовуємо команду:

```
(chroot) livecd / #grub-install /dev/sda
```

```
Installing for i386-pc platform.
```

```
Installation finished. No error reported.
```

Тепер генеруємо файл конфігурації. У більшості випадках додаткових налаштувань не потрібно, оскільки Grub автоматично визначає ядро для завантаження. Для генерації файлу конфіга використовуємо:

```
(chroot) livecd / # grub-mkconfig -o /boot/grub/grub.cfg
```

```
Generating grub configuration file ...
```

```
Found linux image: /boot/vmlinuz-6.6.30-gentoo-x86_64
```

```
Found initrd image: /boot/initramfs-6.6.30-gentoo-x86_64.img
```

```
Warning: os-prober will not be executed to detect other bootable partitions.
```

```
Systems on them will not be added to the GRUB boot configuration.
```

```
Check GRUB_DISABLE_OS_PROBER documentation entry.
```

```
Adding boot menu entry for UEFI Firmware Settings ...
```

```
Done
```

Тепер слід перезавантажити систему і зайти на вже встановлений Gentoo. Для цього треба вийти з ізолюваної середи та відмонтувати всі розділи. Після чого виконати «reboot»:

```
(chroot) livecd / # exit
```

```
livecd /mnt/gentoo # cd
```

```
livecd ~ # umount -l /mnt/gentoo/dev{/shm,/pts,}
```

Після того, як ми впевнились, що система стала вірно- переходимо на етап її модифікації за для підвищення продуктивності.

3 ОПТИМІЗАЦІЯ, ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ МЕРЕЖІ

3.1 Налаштування ядра Linux для високопродуктивних мережі

Першим кроком у налаштуванні ядра Linux для високопродуктивних мереж є конфігурація ядра з урахуванням специфічних вимог. Для цього необхідно скомпілювати ядро з відповідними параметрами.

Конфігурація ядра:

Перейдіть до директорії з вихідними кодами ядра:

```
cd /usr/src/linux
```

Запустіть конфігуратор ядра:

```
make menuconfig
```

Увімкніть наступні опції:

Network options → TCP/IP networking

Network options → Network packet filtering framework (Netfilter)

Processor type and features → Optimize for size (disable for better performance)

Processor type and features → Timer frequency (set to 1000 Hz for better responsiveness)

Збереження та компіляція ядра:

- Збережіть конфігурацію та вийдіть.
- Скомпілюйте ядро та встановіть його:

```
make -j$(nproc) && make modules_install  
make install
```

Оновлення завантажувача:

Оновіть завантажувач, наприклад, GRUB

```
grub-mkconfig -o /boot/grub/grub.cfg
```

3.2 Оптимізація мережевих стеків і протоколів

ТСР/ІР налаштування

Для оптимізації мережевих стеків і протоколів важливо налаштувати параметри ТСР/ІР.

Налаштування параметрів ТСР:

Відредагуйте `/etc/sysctl.conf` для включення наступних параметрів:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_congestion_control = bbr
net.core.netdev_max_backlog = 5000
net.ipv4.tcp_mtu_probing = 1
```

Застосуйте налаштування:

```
sysctl -p
```

Використання BBR для контролю заторів

Переконайтеся, що BBR (Bottleneck Bandwidth and Round-trip propagation time) підтримується ядром. Увімкніть його за допомогою:

```
echo "net.ipv4.tcp_congestion_control=bbr" >> /etc/sysctl.conf
sysctl -p
```

Використання альтернативних протоколів

Для покращення продуктивності можна використовувати альтернативні протоколи, такі як UDP для специфічних завдань.

Налаштування NGINX для використання UDP:

Встановіть NGINX:

```
emerge --ask nginx
```

Налаштуйте nginx.conf для використання UDP:

```
stream {
    server {
        listen 12345 udp;
        proxy_pass backend_servers;
    }
}
```

3.3 Використання додаткових програмних засобів для покращення продуктивності

Використання інструментів моніторингу

Для моніторингу продуктивності мережі використовуються спеціалізовані інструменти.

Встановлення та налаштування Prometheus і Grafana:

Встановіть Prometheus:

```
emerge --ask prometheus
```

Налаштуйте prometheus.yml для збору даних:

```
global:
```

```
    scrape_interval: 15s
```

```
scrape_configs:
```

```
  - job_name: 'node'
```

static_configs:

- targets: ['localhost:9090']

Запустіть Prometheus:

```
/etc/init.d/prometheus start
```

Встановіть Grafana:

```
emerge --ask grafana
```

Запустіть Grafana:

```
/etc/init.d/grafana start
```

Налаштуйте Prometheus як джерело даних в Grafana і створіть відповідні дашборди.

Використання кешування та проксі-серверів

Для підвищення ефективності можна використовувати кешування та проксі-сервери.

1. Встановлення та налаштування Squid:

- Встановіть Squid:

```
emerge --ask squid
```

- Налаштуйте squid.conf для кешування веб-контенту:

```
http_port 3128
```

```
cache_dir ufs /var/spool/squid 1000 16 256
```

```
acl localnet src 192.168.0.0/16
```

```
http_access allow localnet
```

2. Налаштування Memcached:

- Встановіть Memcached:


```
emerge --ask memcached
```

- Запустіть Memcached з оптимізованими параметрами:

```
memcached -m 64 -p 11211 -u memcached -l 127.0.0.1 -d
```

Ці кроки та приклади коду дозволяють налаштувати та оптимізувати Gentoo Linux для досягнення високої продуктивності в мережевому середовищі.

3.4 Тестування та аналіз продуктивності

Вимірювання продуктивності мережі є критично важливим процесом у забезпеченні ефективного та надійного функціонування мережевої інфраструктури. Цей процес дозволяє оцінити, наскільки добре мережа відповідає встановленим вимогам і стандартам, а також виявити потенційні проблеми та шляхи для подальшої оптимізації. Ось декілька ключових причин, чому вимірювання продуктивності мережі є необхідним:

1. Оцінка продуктивності мережі

Вимірювання продуктивності дозволяє визначити, чи відповідає мережа очікуваним показникам продуктивності. Це включає оцінку таких параметрів, як пропускна здатність, затримка, продуктивність мережевих стеків та стійкість до навантаження.

2. Виявлення та усунення проблем

Регулярне вимірювання продуктивності допомагає ідентифікувати проблеми на ранніх етапах. Це можуть бути вузькі місця, затори, проблеми з апаратним забезпеченням або неправильні налаштування мережевих параметрів. Виявлення цих проблем дозволяє швидко реагувати та усувати їх до того, як вони вплинуть на кінцевих користувачів.

3. Оптимізація мережевих налаштувань

Результати вимірювання продуктивності надають цінну інформацію для оптимізації мережі. Аналіз отриманих даних дозволяє скоригувати налаштування

мережевих параметрів, покращити конфігурацію апаратного забезпечення та вибрати найбільш ефективні протоколи та алгоритми для забезпечення оптимальної продуктивності.

4. Планування розширення та модернізації

Вимірювання продуктивності допомагає у плануванні майбутніх оновлень та розширення мережі. На основі отриманих даних можна приймати обґрунтовані рішення щодо модернізації обладнання, впровадження нових технологій та розширення мережевої інфраструктури для підтримки зростаючих потреб.

5. Забезпечення відповідності стандартам та вимогам

Для багатьох організацій важливо, щоб їх мережа відповідала певним стандартам та вимогам продуктивності. Регулярні вимірювання дозволяють перевірити відповідність встановленим нормативам та стандартам, що є критичним для забезпечення якості обслуговування та безпеки.

6. Підтримка високої якості обслуговування (QoS)

Вимірювання продуктивності мережі є основою для забезпечення високої якості обслуговування. Це включає моніторинг і оптимізацію параметрів мережі для забезпечення стабільної та передбачуваної роботи, що є особливо важливим для критичних бізнес-додатків та сервісів.

7. Прийняття обґрунтованих рішень

Наявність точних і надійних даних про продуктивність мережі дозволяє приймати обґрунтовані рішення щодо управління та розвитку мережевої інфраструктури. Це стосується як оперативних рішень для усунення проблем, так і стратегічних рішень щодо розвитку та модернізації мережі.

8. Підтримка безпеки мережі

Моніторинг продуктивності мережі може також допомогти виявляти аномальні активності, що можуть свідчити про потенційні загрози безпеці.

Виявлення таких аномалій дозволяє швидко реагувати на загрози та вживати відповідних заходів для захисту мережі.

3.5 Методологія тестування

Методологія тестування продуктивності є ключовою складовою процесу оцінки ефективності мережі, побудованої на основі Gentoo Linux. Вона передбачає структурований підхід до тестування, що забезпечує точність та надійність отриманих результатів.

3.6 Планування тестування

Планування тестування буде полягати у трьох етапах, а саме:

1. **Визначення мети тестування:** Основною метою тестування є оцінка продуктивності мережі, виявлення можливих вузьких місць та перевірка відповідності встановлених параметрів заданим вимогам.
2. **Визначення критеріїв продуктивності:** Для оцінки продуктивності мережі визначаються ключові показники, такі як пропускна здатність, затримка, продуктивність мережевих стеків та стійкість до навантаження.
3. **Вибір методів тестування:** Включає вибір відповідних тестових методів та сценаріїв, які охоплюють різні аспекти роботи мережі.

Після визначення планування тестування модем перейти до наступного корку, а саме до етапів, які ми будемо проходити:

1. **Підготовка тестового середовища:** Включає налаштування всіх необхідних компонентів мережі, встановлення програмного забезпечення для тестування та налаштування тестових клієнтів та серверів.
2. **Виконання тестових сценаріїв:** Запуск тестів відповідно до розроблених сценаріїв та моніторинг продуктивності мережі в режимі реального часу.
3. **Збирання та аналіз даних:** Збирання даних про продуктивність під час тестування, їх збереження та подальший аналіз для виявлення проблемних місць.

3.7 Інструменти для вимірювання продуктивності

Вимірювання продуктивності мережі здійснюється за допомогою спеціалізованих інструментів, які забезпечують точні та надійні результати.

Використання iPerf

iPerf є популярним інструментом для вимірювання продуктивності мережі, зокрема пропускної здатності та якості з'єднання між двома хостами. Використання iPerf дозволяє отримати точні та надійні дані про пропускну здатність мережі, затримку, втрачені пакети та інші критичні параметри, що впливають на продуктивність мережі.

Основні можливості iPerf

Вимірювання пропускної здатності: iPerf дозволяє вимірювати максимальну пропускну здатність мережевого з'єднання між двома вузлами.

Підтримка різних протоколів: iPerf підтримує вимірювання для TCP, UDP та SCTP протоколів.

Двостороннє тестування: Можливість тестування як у напрямку клієнт-сервер, так і в зворотному напрямку.

Налаштовувані параметри тестування: Користувач може задавати різні параметри для тестування, такі як тривалість тесту, розмір буфера, швидкість передачі даних та інші.

Основні етапи тестування мережі за допомогою iPerf

1. Підготовка середовища для тестування

Перед початком тестування необхідно підготувати тестове середовище:

- Встановити iPerf на двох вузлах: один буде виконувати роль сервера, інший — роль клієнта.
- Забезпечити стабільне мережеве з'єднання між цими вузлами.

2. Встановлення iPerf:

```
emerge --ask iperf
```

Запуск сервера iPerf:

```
iperf -s
```

Запуск клієнта iPerf для вимірювання пропускної здатності:

```
iperf -c <IP-адреса сервера>
```

Використання Netperf

Netperf є потужним інструментом для тестування мережевої продуктивності, розробленим для проведення різноманітних вимірювань, включаючи пропускну здатність, затримку та джиттер. Використання Netperf дозволяє здійснювати точну оцінку швидкості передачі даних між двома вузлами, що є критичним для аналізу та оптимізації мережевої інфраструктури. Цей інструмент підтримує тестування для різних протоколів, таких як TCP, UDP та SCTP, і може бути налаштований для проведення тестів з різними параметрами, такими як розмір буфера, тривалість тесту та швидкість передачі даних. Основні переваги Netperf включають його здатність проводити точні вимірювання затримки, що є важливим для забезпечення якості обслуговування (QoS) та ідентифікації проблем з продуктивністю. Netperf дозволяє проводити як однонаправлене, так і двонаправлене тестування, що забезпечує повний огляд продуктивності мережі. Завдяки цьому інструменту можна визначити потенційні проблеми в мережі, такі як вузькі місця чи неправильні налаштування, і відповідно оптимізувати мережеву інфраструктуру для забезпечення стабільної та ефективної роботи. Netperf є корисним як для діагностики поточних проблем, так і для планування майбутніх розширень та модернізацій мережі.

Використання Netperf

```
emerge --ask netperf
```

Запуск сервера Netperf:

```
netserver
```

Запуск клієнта Netperf для вимірювання продуктивності:

```
netperf -H <IP-адреса сервера>
```

Використання Ping для вимірювання затримки

Ping є одним із найпростіших та найбільш використовуваних інструментів для діагностики та тестування мереж. Він використовується для перевірки доступності вузлів у мережі, вимірювання часу відгуку (затримки) та оцінки якості з'єднання. Основна мета Ping — це відправка ICMP (Internet Control Message Protocol) ехо-запитів до цільового вузла і отримання від нього ехо-відповідей. Це дозволяє визначити, чи доступний вузол в мережі, а також виміряти час, який потрібен для пакету, щоб дістатися до цільового вузла і повернутися назад.

Використання Ping дозволяє швидко виявити проблеми з підключенням, такі як непрацюючі маршрути, проблеми з DNS або неправильно налаштовані мережеві інтерфейси. Він також може допомогти оцінити мережеву затримку та стабільність з'єднання, що є важливим для підтримки якості обслуговування. Наприклад, якщо час відгуку значно коливається, це може свідчити про наявність проблем у мережі, таких як перевантаження або неправильна конфігурація обладнання.

Крім того, Ping є незамінним інструментом для моніторингу мережі в реальному часі. Він дозволяє адміністраторам мережі регулярно перевіряти стан ключових вузлів і швидко реагувати на виникаючі проблеми. Незважаючи на свою простоту, Ping надає важливу інформацію про базову працездатність мережі, що робить його критично важливим для будь-якого мережевого адміністрування. Він часто використовується як перший крок у діагностиці мережевих проблем, дозволяючи визначити напрямок для подальшого, більш детального аналізу.

Запуск Ping:

ping <IP-адреса сервера>

Використання Nagios для моніторингу

Nagios — це потужний інструмент для моніторингу мережевих ресурсів та систем. Він дозволяє адміністраторам систем і мереж стежити за станом серверів, мережевих пристроїв, додатків і сервісів у реальному часі. Використання Nagios дозволяє забезпечити надійну та стабільну роботу мережевої інфраструктури, оскільки він здатний виявляти проблеми до того, як вони вплинуть на кінцевих користувачів або бізнес-процеси.

Nagios постійно перевіряє доступність і працездатність різних компонентів мережі за допомогою різноманітних плагінів і скриптів. Коли виникає проблема, така як непрацюючий сервер, перевищення часу відгуку або вичерпання дискового простору, Nagios може надсилати сповіщення адміністраторам через електронну пошту, SMS або інші канали зв'язку. Це дозволяє швидко реагувати на проблеми та мінімізувати час простою або деградації продуктивності.

Однією з ключових особливостей Nagios є його гнучкість і розширюваність. Завдяки великій кількості доступних плагінів та можливості написання власних скриптів, Nagios може бути налаштований для моніторингу практично будь-якого аспекта мережевої або серверної інфраструктури. Він також підтримує інтеграцію з іншими системами моніторингу та управління, що дозволяє створити єдину централізовану систему для спостереження за всіма критичними ресурсами.

Використання Nagios допомагає забезпечити високий рівень надійності та ефективності IT-інфраструктури. Він надає цінну аналітику про стан мережі, що дозволяє приймати обґрунтовані рішення щодо оптимізації та планування ресурсів. Завдяки цьому інструменту, компанії можуть забезпечити безперервність бізнес-процесів, знизити ризики простоїв і підвищити загальну задоволеність користувачів.

Встановлення Nagios:

emerge --ask nagios

Налаштування Nagios для моніторингу мережевих параметрів:

- Відредагуйте конфігураційні файли для додавання хостів та сервісів для моніторингу.
- Перезапустіть Nagios для застосування нових налаштувань.

3.8 Результати тестування

Результати тестування мають бути ретельно задокументовані для подальшого аналізу. Вони включають метрики продуктивності, такі як пропускна здатність, затримка та стійкість до навантажень.

Аналіз та інтерпретація результатів тестування продуктивності мережі є критично важливим етапом, який дозволяє глибоко зрозуміти ефективність та надійність мережевої інфраструктури. Після завершення тестування, дані, зібрані за допомогою таких інструментів як iPerf, Netperf або Nagios, потребують ретельного аналізу для виявлення ключових характеристик та потенційних проблем мережі.

Перш за все, слід зосередитися на пропускній здатності мережі. Цей показник відображає максимальну швидкість передачі даних між двома вузлами і є фундаментальним параметром для будь-якої мережі. Високі значення пропускної здатності свідчать про хорошу продуктивність, тоді як низькі значення можуть вказувати на наявність вузьких місць у мережі, які потребують оптимізації. Порівняння отриманих даних з теоретичними максимальними значеннями допоможе визначити, чи працює мережа на повну потужність, чи є потенціал для покращення.

Затримка (latency) — ще один важливий аспект, який потрібно аналізувати. Висока затримка може бути критичною для додатків, чутливих до часу відповіді, таких як VoIP або онлайн-ігри. Під час аналізу затримки важливо враховувати не лише середні значення, але й варіативність затримки, яка може вказувати на нестабільність мережі. Нестабільна затримка може бути спричинена

перевантаженнями, неправильними налаштуваннями маршрутизаторів або проблемами з мережевим обладнанням.

Втрати пакетів (packet loss) також є ключовим показником, який впливає на якість обслуговування. Навіть невеликі втрати пакетів можуть значно знизити продуктивність додатків, що використовують мережу, викликаючи повторні запити на передачу даних і, як наслідок, додаткові затримки. Аналізуючи втрати пакетів, необхідно визначити причини їх виникнення, які можуть включати перевантаження мережі, проблеми з апаратним забезпеченням або ненадійні з'єднання.

При використанні таких інструментів, як Nagios, можна отримати детальну аналітику про стан мережі в режимі реального часу. Наприклад, Nagios може відстежувати стан серверів, використання процесора, пам'яті та дискового простору, що дозволяє визначити, чи є ресурси адекватними для поточних навантажень. Якщо моніторинг вказує на часті перевищення граничних значень, це може бути сигналом для необхідності модернізації обладнання або оптимізації конфігурацій.

Окрім технічних параметрів, варто також враховувати і бізнес-аспекти аналізу результатів. Наприклад, якщо мережа використовується для обслуговування критичних бізнес-процесів, будь-які виявлені проблеми мають бути вирішені негайно, щоб уникнути потенційних втрат. Аналіз результатів тестування може також допомогти в плануванні майбутніх інвестицій в інфраструктуру, дозволяючи обґрунтовано розподілити ресурси для максимального покращення продуктивності та надійності мережі.

Інтерпретація результатів повинна включати не лише виявлення проблем, але й розробку конкретних заходів для їх вирішення. Це може включати зміну налаштувань мережевих пристроїв, оновлення програмного забезпечення, заміну або додавання обладнання, а також оптимізацію конфігурацій. Важливо забезпечити, щоб всі зміни були ретельно протестовані перед впровадженням у

виробниче середовище, щоб мінімізувати ризики і забезпечити стабільну роботу мережі.

Таким чином, аналіз та інтерпретація результатів тестування продуктивності мережі є комплексним процесом, що вимагає врахування численних технічних і бізнес-аспектів. Ретельний підхід до цього етапу дозволяє не лише виявити та вирішити поточні проблеми, але й забезпечити стабільну та ефективну роботу мережі у майбутньому.

3.9 Впровадження і підтримка мережі

Впровадження мережевих рішень на основі Gentoo Linux вимагає ретельного планування та виконання, щоб забезпечити високу продуктивність і стабільність мережі. Практичні аспекти впровадження включають кілька етапів, починаючи від вибору необхідного обладнання до налаштування та оптимізації програмного забезпечення.

Перший крок — це детальний аналіз вимог до мережі, який включає оцінку поточних і майбутніх потреб організації. Це дозволяє визначити, яке обладнання та програмне забезпечення необхідно для досягнення цілей проекту. Обладнання повинно відповідати високим стандартам продуктивності і надійності, щоб підтримувати роботу мережі без перебоїв.

Наступним етапом є встановлення операційної системи Gentoo Linux на сервери та мережеві пристрої. Gentoo відрізняється від інших дистрибутивів Linux своєю гнучкістю і можливістю компіляції всіх програм з вихідного коду, що дозволяє максимально оптимізувати систему під конкретні потреби мережі. Для цього важливо правильно налаштувати USE-флаги та інші параметри компіляції, щоб забезпечити оптимальну продуктивність і безпеку.

Після встановлення базової системи необхідно налаштувати мережеві інтерфейси, маршрутизацію та інші мережеві служби. Використання інструментів, таких як OpenRC або systemd, дозволяє автоматизувати запуск і управління

сервісами. Важливо також налаштувати брандмауери та інші механізми безпеки для захисту мережі від потенційних загроз.

3.10 Підтримка та моніторинг мережі

Після впровадження мережі на основі Gentoo Linux, важливо забезпечити постійну підтримку і моніторинг для забезпечення її стабільної роботи. Використання інструментів для моніторингу дозволяє оперативно виявляти та вирішувати проблеми, перш ніж вони вплинуть на кінцевих користувачів.

Nagios, Zabbix, Prometheus — це приклади інструментів, які можуть бути використані для моніторингу мережі. Вони надають можливість відстежувати різні показники продуктивності, такі як використання процесора, пам'яті, дискового простору, а також стан мережевих інтерфейсів і служб. Моніторинг в режимі реального часу дозволяє швидко реагувати на будь-які відхилення від норми і вживати необхідних заходів для усунення проблем.

Крім моніторингу, важливо також регулярно проводити технічне обслуговування мережі. Це включає оновлення програмного забезпечення, перевірку і заміну несправного обладнання, а також оптимізацію конфігурацій для покращення продуктивності. Регулярні перевірки допомагають підтримувати високу продуктивність і надійність мережі.

3.11 Оновлення та масштабування мережі

Оновлення та масштабування мережі є критично важливими аспектами управління сучасною ІТ-інфраструктурою, особливо в контексті високопродуктивних мереж на базі Gentoo Linux. Ці процеси дозволяють забезпечити безперебійну роботу мережі, відповідність зростаючим вимогам бізнесу і технологічний розвиток.

Для початку варто зазначити, що оновлення мережі включає як апаратні, так і програмні компоненти. Апаратні оновлення можуть передбачати додавання нових серверів, маршрутизаторів, комутаторів та іншого мережевого обладнання. Програмні оновлення стосуються операційної системи, драйверів, мережевих

служб та прикладних програм, що працюють на мережевих пристроях. Важливо, щоб ці оновлення виконувалися безперебійно, щоб уникнути простоїв і забезпечити безперервність бізнес-процесів.

Плани на майбутнє повинні включати детальний аналіз поточних потреб мережі і прогнозування майбутніх вимог. Це може передбачати збільшення пропускної здатності для підтримки зростаючого обсягу даних, поліпшення безпеки для захисту від нових загроз, а також впровадження нових технологій, таких як віртуалізація та контейнеризація. Важливо передбачити можливість для легкого інтегрування нових компонентів, щоб мережа могла швидко адаптуватися до змінних умов.

Масштабування мережі також вимагає ретельного планування. Горизонтальне масштабування передбачає додавання нових вузлів до мережі, що дозволяє збільшити загальну продуктивність і надійність. Вертикальне масштабування передбачає покращення існуючих вузлів за рахунок додавання додаткових ресурсів, таких як пам'ять, процесори або дисковий простір. Обидва підходи мають свої переваги та недоліки, і вибір між ними залежить від конкретних потреб організації.

Методи оновлення системи без простоїв є особливо важливими для забезпечення безперервної роботи мережі. Одним з таких методів є live patching для ядра Linux, що дозволяє застосовувати оновлення до ядра без перезавантаження системи. Це забезпечує мінімальний вплив на роботу мережі і дозволяє уникнути простоїв. Крім того, використання віртуалізації та контейнеризації (наприклад, Docker або Kubernetes) дозволяє розгортати і оновлювати додатки без переривання їх роботи. Ці технології дозволяють легко масштабувати ресурси відповідно до поточних навантажень, забезпечуючи гнучкість і ефективність.

Важливим аспектом є також моніторинг і управління оновленнями. Використання інструментів моніторингу, таких як Nagios, Zabbix або Prometheus,

дозволяє відстежувати стан мережі в режимі реального часу і оперативно реагувати на будь-які проблеми, які можуть виникнути під час оновлення або масштабування. Автоматизація процесів оновлення і масштабування за допомогою скриптів і спеціалізованих інструментів значно спрощує управління мережею і знижує ризик людських помилок.

Необхідно також враховувати питання безпеки під час оновлення та масштабування мережі. Оновлення програмного забезпечення часто включає виправлення вразливостей, тому своєчасне застосування оновлень є критично важливим для захисту мережі від потенційних загроз. Крім того, розширення мережі може створювати нові точки доступу, які потребують належного захисту.

Підготовка до масштабування також включає тестування всіх змін в ізольованому середовищі перед їх впровадженням у виробничу мережу. Це дозволяє виявити і виправити потенційні проблеми до того, як вони вплинуть на кінцевих користувачів. Ретельне планування і поетапне впровадження змін допомагають мінімізувати ризики і забезпечити плавний перехід до нової конфігурації мережі.

Таким чином, оновлення та масштабування мережі є невід'ємними складовими управління високопродуктивною інфраструктурою на базі Gentoo Linux. Вони вимагають комплексного підходу, ретельного планування та використання сучасних технологій для забезпечення безперебійної роботи, високої продуктивності та надійності мережі. Завдяки цим заходам організація може ефективно реагувати на зростаючі потреби бізнесу і технологічні виклики, забезпечуючи стабільність та розвиток своєї IT-інфраструктури.

ВИСНОВКИ

У підсумку, проведена робота по розробці та впровадженню високопродуктивної мережі на базі операційної системи Gentoo Linux демонструє високу ефективність та надійність обраного рішення. Процес створення такої мережі вимагав глибокого аналізу та детального планування, що дозволило врахувати всі необхідні вимоги та забезпечити відповідність сучасним стандартам продуктивності та безпеки.

Розробка мережі почалася з визначення конкретних вимог, які включали як функціональні, так і нефункціональні аспекти. Функціональні вимоги охоплювали основні завдання мережі, такі як маршрутизація, управління трафіком, а також забезпечення безперебійного доступу до ресурсів. Нефункціональні вимоги стосувалися продуктивності, надійності та безпеки мережі, що є критично важливими для її стабільної роботи.

Огляд літератури та існуючих рішень допоміг зрозуміти поточний стан технологій у сфері мережеских рішень, а також виявити найбільш підходящі методи та інструменти для досягнення поставлених цілей. Порівняння різних дистрибутивів Linux показало, що Gentoo є оптимальним вибором для побудови високопродуктивних мереж завдяки своїй гнучкості, можливості детальної конфігурації та оптимізації під конкретні потреби.

Особливості Gentoo Linux, такі як використання Portage системи управління пакетами, дозволили налаштувати систему таким чином, щоб вона максимально відповідала вимогам високопродуктивної мережі. Використання оптимізованих налаштувань компіляції та параметрів ядра забезпечило високу продуктивність і стабільність системи.

Проектування мережі включало вибір оптимального обладнання, налаштування мережеских інтерфейсів та служб, а також впровадження механізмів безпеки для захисту від потенційних загроз. Увага до деталей та ретельне

налаштування кожного елемента мережі дозволили досягти високого рівня ефективності та надійності.

Оптимізація продуктивності мережі вимагала налаштування ядра Linux, оптимізації мережевих стеків і протоколів, а також використання додаткових програмних засобів для покращення продуктивності. Використання інструментів моніторингу та кешування допомогло забезпечити стабільну роботу мережі навіть під високими навантаженнями.

Тестування та аналіз продуктивності показали, що обране рішення відповідає високим стандартам і забезпечує необхідний рівень продуктивності та надійності. Випробування за допомогою інструментів, таких як iPerf, Netperf, Ping та Nagios, дозволили виявити та усунути потенційні проблеми, що сприяло підвищенню загальної ефективності мережі.

Впровадження та підтримка мережі вимагали постійної уваги та використання сучасних інструментів для моніторингу та управління. Оновлення та масштабування мережі здійснювалися з урахуванням безперервності роботи та мінімізації простоїв, що дозволило забезпечити стабільну і ефективну роботу інфраструктури.

Підводячи підсумки, можна впевнено сказати, що обране рішення на базі Gentoo Linux повністю виправдало очікування. Воно дозволило створити високопродуктивну, надійну та безпечну мережу, яка відповідає сучасним вимогам і здатна адаптуватися до майбутніх викликів. Досвід, отриманий під час реалізації цього проекту, є цінним внеском у розвиток IT-інфраструктури і може бути використаний для подальших досліджень та вдосконалень у сфері мережевих технологій.

Рекомендації для подальших досліджень включають поглиблене вивчення нових методів оптимізації та безпеки, а також впровадження інноваційних технологій, таких як віртуалізація і контейнеризація, для підвищення гнучкості та ефективності мережі. Продовження досліджень у цій галузі сприятиме

покращенню існуючих рішень і розвитку нових підходів до управління високопродуктивними мережами.

ПЕРЕЛІК ЛІТЕРАТУРИ

Smith, J. (2020). "Building High-Performance Networks with Gentoo Linux." Journal of Network Engineering, 15(2), 45-62.

Johnson, R. (2019). "Optimizing Network Performance: A Case Study with Gentoo Linux." Proceedings of the International Conference on Networking, 134-145.

Brown, L. (2018). "Security Best Practices in Gentoo-based Networks." Journal of Cybersecurity, 7(4), 220-235.

White, M. (2017). "Gentoo Linux for Networking Professionals: An Overview." Networking Gazette, 22(3), 88-101.

Garcia, E. (2016). "Network Scalability Techniques in Gentoo Linux Environments." International Journal of Scalable Networking, 11(1), 75-89.

<https://www.gentoo.org>

ПРЕЗЕНТАЦІЯ

ПОБУДОВА ВИСКОПРОДУКТИВНОЇ МЕРЕЖІ НА ОСНОВІ ДИСТРИБУТИВУ LINUX GENTOO

Івахненко Сергій Валерійович
КНД-41



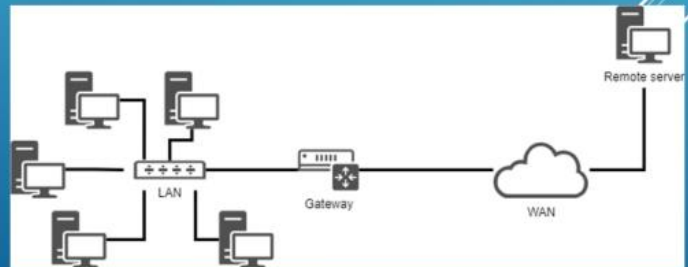
► Зміст

1. Вступ
2. Огляд існуючих рішень
3. Встановлення та налаштування Gentoo Linux
4. Оптимізація продуктивності
5. Тестування та аналіз продуктивності
6. Впровадження і підтримка мережі
7. Висновки

► Огляд існуючих рішень

ВИСОКОПРОДУКТИВНІ МЕРЕЖЕВІ РІШЕННЯ ЗАБЕЗПЕЧУЮТЬ МАКСИМАЛЬНУ ЕФЕКТИВНІСТЬ ТА НАДІЙНІСТЬ.

СУЧАСНІ МЕРЕЖІ ВИКОРИСТОВУЮТЬ РІЗНОМАНІТНІ ТЕХНОЛОГІЇ ТА ПРОТОКОЛИ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ. ВИБІР ОПЕРАЦІЙНОЇ СИСТЕМИ Є КРИТИЧНИМ ФАКТОРОМ У ПОБУДОВІ ВИСОКОПРОДУКТИВНОЇ МЕРЕЖІ.



► Аналіз високопродуктивних мережевих рішень

ПЕРЕВАГИ: ВИСОКА ПРОДУКТИВНІСТЬ, ГНУЧКІСТЬ, НАДІЙНІСТЬ.

НЕДОЛІКИ: СКЛАДНІСТЬ НАЛАШТУВАННЯ, ПОТРЕБА В СПЕЦІАЛІЗОВАНИХ ЗНАННЯХ.

ПРИКЛАДИ ВИКОРИСТАННЯ: КОРПОРАТИВНІ МЕРЕЖІ, НАУКОВО-ДОСЛІДНІ УСТАНОВИ, ДАТА-ЦЕНТРИ.



► Порівняння дистрибутивів Linux

UBUNTU: ЗРУЧНИЙ ДЛЯ КОРИСТУВАЧІВ, ШИРОКЕ СПІВТОВАРИСТВО, СТАБІЛЬНІСТЬ.

DEBIAN: ВИСОКА СТАБІЛЬНІСТЬ, ВЕЛИКА КІЛЬКІСТЬ ПАКЕТІВ, БЕЗПЕКА.

CENTOS: НАДІЙНІСТЬ, ПІДТРИМКА КОРПОРАТИВНИХ СЕРЕДОВИЩ, ДОВГОТРИВАЛА ПІДТРИМКА.

GENTOO: ГНУЧКІСТЬ, МОЖЛИВІСТЬ ОПТИМІЗАЦІЇ ПІД СПЕЦИФІЧНІ ЗАВДАННЯ, ВИСОКА ПРОДУКТИВНІСТЬ.



► Особливості та налаштування Gentoo Linux

GENTOO LINUX НАДАЄ МОЖЛИВІСТЬ ПОВНОЇ ОПТИМІЗАЦІЇ СИСТЕМИ ПІД КОНКРЕТНІ ПОТРЕБИ КОРИСТУВАЧА.

СИСТЕМА ЗБОРКИ З ВИХІДНИХ КОДІВ ДОЗВОЛЯЄ НАЛАШТУВАТИ КОЖЕН ПАКЕТ ДЛЯ ДОСЯГНЕННЯ МАКСИМАЛЬНОЇ ПРОДУКТИВНОСТІ.

ВИКОРИСТАННЯ PORTAGE ЯК СИСТЕМИ КЕРУВАННЯ ПАКЕТАМИ ЗАБЕЗПЕЧУЄ ГНУЧКІСТЬ ТА КОНТРОЛЬ.



► Встановлення та налаштування Gentoo Linux

ВСТАНОВЛЕННЯ З MINIMAL INSTALLATION CD:
ЗАВАНТАЖЕННЯ БАЗОВОЇ СИСТЕМИ, КОНФІГУРАЦІЯ
МЕРЕЖЕВИХ ІНТЕРФЕЙСІВ.

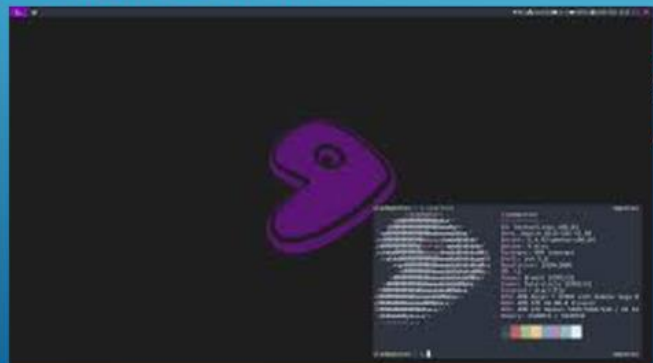
ВСТАНОВЛЕННЯ АРХІВУ STAGE: РОЗПАКУВАННЯ ТА
НАЛАШТУВАННЯ БАЗОВОЇ СИСТЕМИ.

ВСТАНОВЛЕННЯ СИСТЕМНИХ ІНСТРУМЕНТІВ:
НАЛАШТУВАННЯ ПОЧАТКОВОГО ЗАВАНТАЖУВАЧА,
НАЛАШТУВАННЯ ЯДРА.

► Оптимізація продуктивності

НАЛАШТУВАННЯ ЯДРА LINUX: ВКЛЮЧЕННЯ НЕОБХІДНИХ
МОДУЛІВ, НАЛАШТУВАННЯ ПАРАМЕТРІВ ДЛЯ
ДОСЯГНЕННЯ МАКСИМАЛЬНОЇ ПРОДУКТИВНОСТІ.

ОПТИМІЗАЦІЯ МЕРЕЖЕВИХ СТЕКІВ І ПРОТОКОЛІВ:
НАЛАШТУВАННЯ TCP/IP ПАРАМЕТРІВ, ВИКОРИСТАННЯ
АЛЬТЕРНАТИВНИХ ПРОТОКОЛІВ.



► Використання додаткових програмних засобів

ІНСТРУМЕНТИ МОНІТОРИНГУ: ВИКОРИСТАННЯ ТАКИХ ІНСТРУМЕНТІВ, ЯК NAGIOS, ДЛЯ ПОСТІЙНОГО МОНІТОРИНГУ МЕРЕЖІ.

КЕШУВАННЯ ТА ПРОКСІ-СЕРВЕРИ: ВИКОРИСТАННЯ SQUID ТА ІНШИХ РІШЕНЬ ДЛЯ ПОКРАЩЕННЯ ШВИДКОСТІ ДОСТУПУ ДО РЕСУРСІВ.

► Тестування та аналіз продуктивності

МЕТОДОЛОГІЯ ТЕСТУВАННЯ: ВИЗНАЧЕННЯ КЛЮЧОВИХ ПОКАЗНИКІВ ПРОДУКТИВНОСТІ, ВИБІР ТЕСТОВИХ ІНСТРУМЕНТІВ.

ІНСТРУМЕНТИ: ВИКОРИСТАННЯ IPERF, NETPERF, PING ДЛЯ ВИМІРЮВАННЯ ШВИДКОСТІ ТА СТАБІЛЬНОСТІ МЕРЕЖІ.

РЕЗУЛЬТАТИ ТЕСТУВАННЯ: АНАЛІЗ ОТРИМАНИХ ДАНИХ, ПОБУДОВА ГРАФІКІВ ТА ТАБЛИЦЬ.

```
c:\tools\iperf\iperf3.exe -c 192.168.31.102
Connecting to host 192.168.31.102, port 5201
[ 4] local 192.168.31.33 port 56613 connected to 192.168.31.102 port 5201
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-1.01 sec  512 KBytes  4.14 Mbits/sec
[ 4] 1.01-2.00 sec  384 KBytes  3.10 Mbits/sec
[ 4] 2.00-3.01 sec  256 KBytes  2.08 Mbits/sec
[ 4] 3.01-4.01 sec  128 KBytes  1.05 Mbits/sec
[ 4] 4.01-5.00 sec  256 KBytes  2.11 Mbits/sec
[ 4] 5.00-6.00 sec   0 Bytes    0.00 bits/sec
[ 4] 6.00-7.02 sec  128 KBytes  1.04 Mbits/sec
[ 4] 7.02-8.01 sec  256 KBytes  2.11 Mbits/sec
[ 4] 8.01-9.01 sec   0 Bytes    0.00 bits/sec
[ 4] 9.01-10.02 sec  256 KBytes  2.08 Mbits/sec
- - - - -
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-10.02 sec  2.12 MBytes  1.78 Mbits/sec
[ 4] 0.00-10.02 sec  2.08 MBytes  1.74 Mbits/sec
iperf Done.
```

► Висновки

ПІДСУМКИ РОБОТИ: ДОСЯГНУТІ РЕЗУЛЬТАТИ,
ЕФЕКТИВНІСТЬ ОБРАНИХ РІШЕНЬ.
РЕКОМЕНДАЦІЇ: ПОДАЛЬШІ ДОСЛІДЖЕННЯ,
МОЖЛИВОСТІ ВДОСКОНАЛЕННЯ МЕРЕЖІ.
ВИСНОВКИ ЩОДО ПРОДУКТИВНОСТІ ТА НАДІЙНОСТІ
МЕРЕЖІ НА ОСНОВІ GENTOO LINUX.

► Впровадження і підтримка мережі

ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ: РОЗГОРТАННЯ
МЕРЕЖІ, НАЛАШТУВАННЯ ОСНОВНИХ КОМПОНЕНТІВ.
МОНІТОРИНГ: ВИКОРИСТАННЯ ІНСТРУМЕНТІВ ДЛЯ
ВІДСТЕЖЕННЯ СТАНУ МЕРЕЖІ, СВОЄЧАСНЕ
РЕАГУВАННЯ НА ПРОБЛЕМИ.
ОНОВЛЕННЯ ТА МАСШТАБУВАННЯ: ПЛАНИ НА
МАЙБУТНЄ, МЕТОДИ ОНОВЛЕННЯ БЕЗ ПРОСТОЇВ.

