УДК 004.4:004.7:004.5
DOI: https://doi.org/10.53920/ITS-2025-1;2-11

Олександр Іванович ГОЛУБЕНКО,
кандидат технічних наук, доцент,
Заклад вищої освіти «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0000-0002-1776-5160
Андрій Вікторович ЛЕМЕШКО,
доктор філософії, доцент,
Державний торговельно-економічний університет
ORCID: 0000-0001-8003-3168
Артем Васильович АНТОНЕНКО,
кандидат технічних наук, доцент,
Національний університет біоресурсів і природокористування України
ORCID ID: 0000-0001-9397-1209
Микола Анатолійович ЛЯШУК,
викладач,
Заклад вищої освіти «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0009-0007-2910-4160

СУЧАСНІ МЕТОДИ ОПТИМІЗАЦІЇ МЕРЕЖНИХ ОПЕРАЦІЙНИХ СИСТЕМ

У статті представлено всебічний аналіз сучасних мережних операційних систем, що застосовуються у серверній інфраструктурі, хмарних середовищах та корпоративних мережах. Досліджено еволюцію Unix-/Linux-подібних платформ, Windows Server та спеціалізованих дистрибутивів, охарактеризовано їх архітектурні принципи, модель безпеки, підходи до масштабування та особливості інтеграції в різні типи IT-інфраструктур. Окрему увагу приділено порівнянню функціональних ролей серверних ОС, їх придатності для виконання веб-, файлових, мережевих, контейнерних і віртуалізаційних сервісів, а також аналізу факторів, що впливають на вибір системними адміністраторами та DevOps-фахівцями конкретної платформи.

Проаналізовано сучасні програмні, апаратні та експлуатаційні вимоги до мережних ОС, включно з впливом апаратної архітектури, дискових систем, мережевих стеків, інструментів керування та механізмів безпеки. Узагальнено ефективні підходи до оптимізації, які охоплюють модернізацію апаратного забезпечення, програмну конфігурацію, застосування контейнеризації

й оркестрації (Docker, Kubernetes), використання гіпервізорів, систем автоматизації та інтелектуальних технологій моніторингу, прогнозування збоїв і адаптивного масштабування.

Визначено ключові обмеження сучасних серверних ОС, пов'язані з високими вимогами до ресурсів, складністю конфігурації, різноманіттям апаратних платформ і швидким розвитком хмарних технологій. Окреслено перспективи подальших досліджень, спрямованих на створення універсальних методик оптимізації, стандартизованих підходів до тестування продуктивності та підвищення відмовостійкості у гібридних інфраструктурах. Отримані результати можуть бути використані для вдосконалення процесів адміністрування, побудови масштабованих серверних систем та підвищення ефективності корпоративних IT-середовищ.

Ключові слова: *мережні операційні системи, серверні ОС, Linux, Windows Server, Unix-подібні системи, віртуалізація, контейнеризація, оптимізація, продуктивність, адміністрування.*

Oleksandr GOLUBENKO,
candidate of technical sciences, associate professor
Higher Education Institution
«Academician Yuriy Bugay InternationalScientific and Technical University»
Andriy LEMESHKO,
PhD, associate professor
State University of Trade and Economics
Artem ANTONENKO,
Candidate of technical sciences, associate professor
National University of Life and Environmental Sciences of Ukraine
Mykola LIASHUK,
teacher,
Higher Education Institution
«Academician Yuriy Bugay InternationalScientific and Technical University»

MODERN METHODS OF OPTIMIZATION OF NETWORK OPERATING SYSTEMS

This article presents a comprehensive analysis of modern network operating systems used in server infrastructures, cloud environments, and corporate networks. The study examines the evolution of Unix/Linux-based platforms, Windows Server, and specialized distributions, detailing their architectural principles, security models, scalability approaches, and integration features within various types of IT infrastructures. Particular attention is given to comparing the functional roles of server operating systems, their suitability for hosting web,

file, network, containerized, and virtualization services, as well as the factors influencing platform selection by system administrators and DevOps engineers.

The work analyzes contemporary software, hardware, and operational requirements for network operating systems, including the impact of processor architecture, storage subsystems, networking stacks, management tools, and security mechanisms. Effective optimization approaches are summarized, encompassing hardware modernization, software configuration tuning, the use of containerization and orchestration technologies (Docker, Kubernetes), hypervisor-based virtualization, automation frameworks, and intelligent monitoring systems for failure prediction and adaptive resource scaling.

Key limitations of modern server operating systems are identified, including high resource demands, configuration complexity, heterogeneity of hardware platforms, and the rapid evolution of cloud technologies. The article outlines future development prospects aimed at creating universal optimization methodologies, standardized performance testing approaches, and enhanced fault tolerance in hybrid infrastructures. The results of the study can be applied to improve administration processes, build scalable server environments, and increase the overall efficiency of corporate IT systems.

Keywords: *network operating systems, server OS, Linux, Windows Server, Unix-based systems, virtualization, containerization, optimization, performance, administration.*

Постановка проблеми. Зростання обсягів даних у сучасних інформаційних системах створює серйозні виклики для мережних операційних систем, адже вони повинні забезпечувати обробку терабайтів інформації в режимі реального часу. У випадку банківських систем це означає необхідність безперервної роботи платіжних шлюзів, де навіть кількахвилинний збій може призвести до блокування транзакцій та фінансових втрат [1].

Ускладнення архітектури мереж, зокрема перехід до багаторівневих і розподілених моделей, вимагає від ОС здатності координувати взаємодію між численними серверами та вузлами. У корпоративних середовищах це проявляється у потребі підтримувати стабільну роботу сотень віртуальних машин, які обслуговують бізнес-додатки, CRM-системи та внутрішні сервіси.

Перехід до розподілених обчислень створює нові виклики для забезпечення синхронізації процесів та балансування навантаження. У хмарних платформах відмова одного вузла може спричинити каскадні збої, що впливають на доступність сервісів для тисяч користувачів одночасно.

Проблема своєчасного оновлення стає критичною, адже затримка у встановленні патчів безпеки може призвести до масштабних атак. Відомі

випадки, коли відсутність оновлень у корпоративних ОС дозволяла зловмисникам використовувати вразливості для викрадення даних клієнтів або блокування роботи цілих компаній [3–5].

Кібербезпека є одним із найважливіших аспектів, оскільки сучасні атаки стають дедалі складнішими. У сфері охорони здоров'я збої в серверних ОС можуть призвести до недоступності електронних медичних записів, що ставить під загрозу життя пацієнтів [2, 7].

Продуктивність мережних операційних систем визначає здатність інфраструктури витримувати пікові навантаження. У сфері електронної комерції це означає необхідність обробки тисяч замовлень за секунду, і будь-яке падіння продуктивності може спричинити втрату клієнтів та доходів [6].

Безперервність роботи серверів є ключовою вимогою для державних структур та критичної інфраструктури. Наприклад, у транспортних системах збій у серверних ОС може призвести до порушення роботи систем управління рухом, що створює реальну загрозу безпеці людей.

Таким чином, постає комплексна науково-технічна проблема, яка полягає у забезпеченні стабільності, надійності та оптимізації роботи мережних операційних систем у хмарних і корпоративних середовищах. Її вирішення має безпосередній зв'язок із фінансовою безпекою, захистом персональних даних, стабільністю критичних сервісів та довірою користувачів до цифрової інфраструктури.

Аналіз останніх досліджень і публікацій. Аналіз останніх досліджень і публікацій свідчить, що сучасна наукова думка приділяє значну увагу еволюції UNIX та Linux-дистрибутивів, які стали основою розвитку хмарних технологій та контейнерних платформ. У працях підкреслюється, що відкритість коду, гнучкість налаштувань і висока стійкість до навантажень зробили Linux домінуючим вибором для більшості дата-центрів та провайдерів хмарних сервісів. Поряд із цим дослідження корпоративних середовищ демонструють, що Windows Server зберігає ключові позиції завдяки інтеграції з Active Directory, зручності адміністрування та сумісності з бізнес-додатками, що робить його незамінним у багатьох організаціях, орієнтованих на централізоване управління [8].

Окремий пласт досліджень стосується FreeBSD та Solaris, які залишаються актуальними у високонавантажених середовищах завдяки стабільності, ефективному мережевому стеку та можливостям масштабування. Автори відзначають, що ці системи часто використовуються у спеціалізованих інфраструктурах, де критичною є безперервність роботи та передбачуваність продуктивності.

Важливим напрямом сучасних досліджень є оптимізація мережних операційних систем, яка охоплює контейнеризацію, віртуалізацію, апаратну модернізацію та використання командних оболонок. У працях наголошується, що Docker і Kubernetes стали стандартом для розгортання мікросервісних архітектур, забезпечуючи гнучкість та масштабованість. Віртуалізаційні технології VMware, Hyper-V та KVM дозволяють ефективно розподіляти ресурси та підвищувати стійкість систем до відмов, хоча водночас створюють додаткове навантаження на апаратні платформи [9–11].

Дослідники також акцентують увагу на апаратній оптимізації, яка включає використання багатоядерних процесорів, швидкісних SSD та NVMe-накопичувачів, а також RAID-масивів для забезпечення надійності та швидкодії. Паралельно розвивається напрям автоматизації адміністрування за допомогою Bash і PowerShell, що дозволяє зменшити людський фактор і підвищити ефективність управління інфраструктурою.

Останні роботи демонструють перспективність застосування інтелектуальних систем моніторингу, які здатні прогнозувати помилки, автоматично масштабувати ресурси та забезпечувати адаптивне керування навантаженням. Проте автори підкреслюють, що бракує уніфікованих методик оцінки ефективності оптимізації та стандартизованих підходів до тестування, що ускладнює порівняння результатів різних досліджень і впровадження їх у практику. Таким чином, сучасна наукова література окреслює широкий спектр рішень, але водночас наголошує на необхідності створення єдиних критеріїв та стандартів для комплексної оцінки мережних операційних систем у реальних умовах експлуатації [13, 14].

Метою статті є розроблення та обґрунтування ефективних підходів до аналізу, оцінювання та оптимізації мережних операційних систем, а також визначення їхніх переваг, недоліків і обмежень у сучасних умовах розвитку серверних інфраструктур. У межах цього дослідження передбачається систематизувати етапи еволюції мережних ОС та охарактеризувати ключові технологічні чинники, що впливають на їх розвиток; здійснити порівняльний аналіз архітектур Linux-, Windows- та Unix-подібних систем із урахуванням їх застосування у корпоративних, хмарних і високонавантажених середовищах; визначити програмні, апаратні та експлуатаційні вимоги, які забезпечують ефективність, масштабованість і стійкість серверних платформ; дослідити функціональні ролі мережних ОС та їхню придатність для різних інфраструктурних сценаріїв, включаючи веб-хостинг, корпоративні сервіси, контейнери та віртуалізацію. Також стаття спрямована на узагальнення сучасних методів оптимізації, що охоплюють апарат-

ні й програмні вдосконалення, оптимізацію мережевих стеків, застосування контейнеризації, автоматизації й інтелектуальних систем аналізу для прогнозування збоїв та адаптивного масштабування. Підсумком роботи є формування практичних рекомендацій щодо підвищення продуктивності, надійності та відмовостійкості мережних операційних систем, а також окреслення перспективних напрямів подальших досліджень у сфері впровадження інноваційних технологій і самовідновлюваних інфраструктур. Вирішення цих проблем має безпосередній зв'язок із важливими науковими та практичними завданнями, зокрема:

- проаналізувати сучасні мережні ОС та їх архітектурні особливості;
- визначити ключові вимоги до серверних ОС;
- дослідити методи оптимізації продуктивності та надійності;
- узагальнити перспективи розвитку та напрями подальших досліджень.

Виклад основного матеріалу дослідження. У межах дослідження наявних рішень щодо підвищення ефективності мережних операційних систем детально розглянуто основні підходи, що застосовуються в сучасних серверних інфраструктурах, а також проаналізовано їх сильні сторони та обмеження. Першочергово було охарактеризовано апаратні методи оптимізації, які традиційно вважаються найбільш прямим способом збільшення продуктивності. Заміна або модернізація процесорів, збільшення обсягу оперативної пам'яті, використання сучасних твердотілих накопичувачів та RAID-масивів дозволяють істотно скоротити час доступу до даних, підвищити пропускну здатність системи та забезпечити вищу стійкість до збоїв. Однак такі рішення пов'язані з високими фінансовими витратами, часто потребують повної перебудови інфраструктури та залежать від апаратних обмежень окремих серверних платформ. Крім того, ефективність апаратних удосконалень може бути зменшена у випадках, коли програмна архітектура ОС або сама природа обчислювальних задач не здатні повною мірою використати нові апаратні ресурси [12].

Програмні методи оптимізації охоплюють більш широкий спектр інструментів та підходів, що дозволяють адаптувати систему без необхідності заміни апаратних компонентів. Це включає оптимізацію системних служб, раціональне використання планувальників задач, налаштування мережевого стеку, застосування контейнеризації для ізоляції середовищ виконання та оркестрації для централізованого керування ними. Використання Docker, Kubernetes або аналогічних платформ забезпечує гнучкість, масштабованість та високий рівень портативності сервісів. Проте складність конфігурації таких інструментів, потреба у високій кваліфікації адмі-

ністраторів, значна кількість залежностей і суттєві вимоги до підтримки системи роблять ці методи менш доступними для невеликих або недостатньо досвідчених IT-команд. Без чітких стандартів і оптимізованих робочих процесів програмні оптимізації можуть не лише не покращити, але й погіршити продуктивність та надійність системи [15].

Віртуалізація — ще один потужний підхід, що суттєво вплинув на розвиток сучасних IT-інфраструктур. Технології VMware, Hyper-V, KVM та інші гіпервізори дозволяють ефективно розподіляти ресурси між віртуальними машинами, забезпечувати ізоляцію сервісів, створювати тестові середовища, реконфігурувати інфраструктуру без зупинки роботи та підвищувати рівень відмовостійкості завдяки кластерам та механізмам міграції. Проте віртуалізація створює додаткове навантаження на апаратну частину, збільшує споживання пам'яті та процесорного часу, вимагає суворої сумісності драйверів, специфічного налаштування мережевих інтерфейсів та забезпечення безпеки між віртуальними середовищами. У деяких випадках гіпервізор стає критичною точкою відмови, а складність адміністрування кластерів може значно зрости разом зі зростанням кількості віртуальних машин.

Окремо було розглянуто інтелектуальні системи моніторингу, які на основі алгоритмів аналізу даних здатні виявляти аномалії, прогнозувати можливі збої, автоматично регулювати розподіл ресурсів та ініціювати масштабування. Такі системи відіграють ключову роль у великих хмарних середовищах та на підприємствах, де безперервність роботи сервісів має вирішальне значення. Водночас їх застосування вимагає значних обсягів історичних і телеметричних даних, ретельної побудови моделей, акуратної інтеграції зі службами ОС та глибокої експертизи з боку персоналу. Ускладнюється також інтеграція таких рішень із застарілими системами, де стандартизовані інтерфейси відсутні або обмежені. Висока вартість впровадження, потреба у постійному навчанні моделей та складність адаптації алгоритмів до динамічних умов можуть стати вагомими стримувальними факторами [16].

Таким чином, аналіз наявних рішень демонструє, що ефективна оптимізація мережних операційних систем повинна базуватися на комбінуванні апаратних, програмних та інтелектуальних підходів, а також враховувати специфіку конкретного середовища, рівень підготовки фахівців та ресурси організації. Жоден окремий метод не може забезпечити універсального результату, натомість максимальна ефективність досягається за умови комплексного планування та гнучкої інтеграції різних технологій.

Модель системи контролю стану мережної операційної системи

Запропонована модель системи контролю стану мережної операційної системи передбачає багаторівневий підхід до моніторингу, оцінювання та автоматичного реагування на відхилення, що впливають на стабільність роботи серверного середовища. На першому рівні здійснюється безперервне збирання телеметричних даних ядра та системних компонентів: показників використання процесора, оперативної пам'яті, дискових операцій, статусу драйверів та ядрових модулів, активності системних викликів і мережевих потоків. Зібрані дані дозволяють відтворити поточне навантаження на систему та визначити базові тенденції у роботі ОС.

Другий рівень моделі охоплює аналіз стану служб та демони операційної системи, включаючи веб-сервери, DNS/DHCP-служби, каталоги Active Directory або LDAP, засоби контейнеризації, модулі віртуалізації та засоби резервного копіювання. Кожна служба оцінюється за параметрами доступності, часу відповіді, коректності конфігурації та стабільності виконання. Окрема увага приділяється виявленню службових конфліктів, несанкціонованих змін конфігурацій та взаємоблокувань.

На третьому рівні модель передбачає оцінювання стану мережевих підсистем. Це включає аналіз пропускної здатності інтерфейсів, рівня втрат пакетів, стабільності з'єднань, коректності роботи мережевих стеків IPv4/IPv6, ефективності маршрутизації та взаємодії з балансувальниками навантаження. На цьому рівні система здатна виявляти вузькі місця мережевих конфігурацій, нестабільну поведінку firewall-політик чи надмірне навантаження на окремі порти.

Четвертий рівень моделі охоплює безпековий контроль: відстеження підозрілих процесів, аналіз авторизаційних подій, виявлення несанкціонованих змін у критичних файлах, аномалій трафіку, потенційних атак та спроб ескалації привілеїв. Система інтегрує журналювання подій (Syslog, Event Log), інструменти контролю цілісності та механізми виявлення вторгнень, забезпечуючи раннє виявлення ризиків і реакцію на загрози.

П'ятим рівнем є модуль прогнозування та автоматичного реагування. На основі історичних даних, моделей машинного навчання та евристичних правил система визначає ймовірність появи критичних станів, таких як деградація дискових підсистем, витоки пам'яті, відмова мережевих служб або надмірне навантаження на CPU. У разі виявлення відхилень модель ініціює автоматичні дії: перезапуск служб, корекцію конфігурацій, міграцію віртуальних машин, масштабування контейнерів або активацію резервних ресурсів.

Таким чином, запропонована модель системи контролю стану мережної операційної системи забезпечує комплексне бачення її роботи в реаль-

ному часі та створює основу для підвищення надійності, відмовостійкості й ефективності серверних інфраструктур. Інтеграція моніторингу, аналітики та автоматичного реагування дозволяє мінімізувати простої, забезпечити передбачувану продуктивність і підтримувати високу доступність сервісів у динамічних умовах експлуатації.

Метод раннього виявлення зависань та некоректних режимів роботи мережних операційних систем

Метод раннього виявлення зависань та некоректних режимів роботи мережних операційних систем ґрунтується на комбінованому використанні моніторингу поведінкових показників, аналізу системних журналів, контролю працездатності служб і механізмів автоматичного діагностування ядра. Основна ідея методу полягає у створенні багатоканальної моделі спостереження, яка дозволяє фіксувати відхилення в роботі ОС ще на початкових етапах, до моменту критичного збою або повного блокування системи.

На першому етапі здійснюється моніторинг ключових метрик продуктивності, таких як рівень використання CPU, обсяг доступної та зайнятої оперативної пам'яті, інтенсивність дискових операцій, затримки вводу-виводу та стабільність системних викликів. Особливу увагу приділено аномаліям, що свідчать про початок деградації, зокрема різкому зростанню черги процесів, неконтрольованим витокам пам'яті, зависанням ядрових потоків чи значним затримкам при обробці мережевого трафіку.

Другим етапом є аналіз поведінки служб та демонів, що забезпечують роботу мережної інфраструктури: DNS-, DHCP-, веб-, каталогових та контейнерних сервісів. Для кожної служби формується профіль нормальної поведінки — середній час відповіді, частота рестартів, інтенсивність логування, обсяг споживаних ресурсів. Відхилення від цього профілю, навіть незначні, можуть сигналізувати про початок некоректного режиму роботи, конфлікти між модулями, помилки оновлення або процеси, що перебувають у стані часткового зависання.

Третій етап методу включає роботу з журналами подій. Раннє виявлення аномалій здійснюється через кореляцію повідомлень системного журналу, попереджень ядра, записів про помилки драйверів, повідомлень про тайм-аути, відмови мережевих стеків або нетипові спроби виконання системних викликів. Логічний аналіз дозволяє виявити ланцюги подій, характерні для початку деградації сервісів, наприклад поступові збої у файловій системі, зменшення часу між повторюваними помилками, поступове зростання кількості невдалих мережевих транзакцій.

На четвертому етапі застосовується модель проактивного аналізу поведінкових шаблонів. Використовуються евристичні алгоритми або ML-мо-

делі, які навчаються на історичних даних і визначають відхилення у динаміці системних показників. Це дозволяє виявляти неочевидні причини зависань — наприклад, комбінації пікових навантажень, рідкісні конфлікти модулів, періодичні витоки пам'яті або зациклені мережеві операції. У результаті система здатна передбачати початок переходу ОС у нестабільний режим приблизно за кілька хвилин або навіть годин до фактичного збою.

Фінальний етап — автоматичне реагування. У разі фіксації перших ознак некоректної роботи система може ініціювати перезапуск окремих служб, ізоляцію проблемного процесу, очищення кешів, відновлення конфігурацій, переміщення контейнерів на інші вузли або сповіщення адміністратора. У критичних випадках можливе виконання контрольованого м'якого перезавантаження ядра або перенаправлення трафіку на резервні вузли.

Таким чином, запропонований метод раннього виявлення залежностей та некоректних режимів роботи ОС забезпечує підвищення стабільності, мінімізацію часу простою та зниження ймовірності критичних збоїв, завдяки глибокому аналізу системної поведінки та інтеграції механізмів автоматичного реагування. Метод є універсальним і може застосовуватись у Linux-, Windows- та Unix-подібних середовищах, адаптуючись до конкретних інфраструктурних вимог.

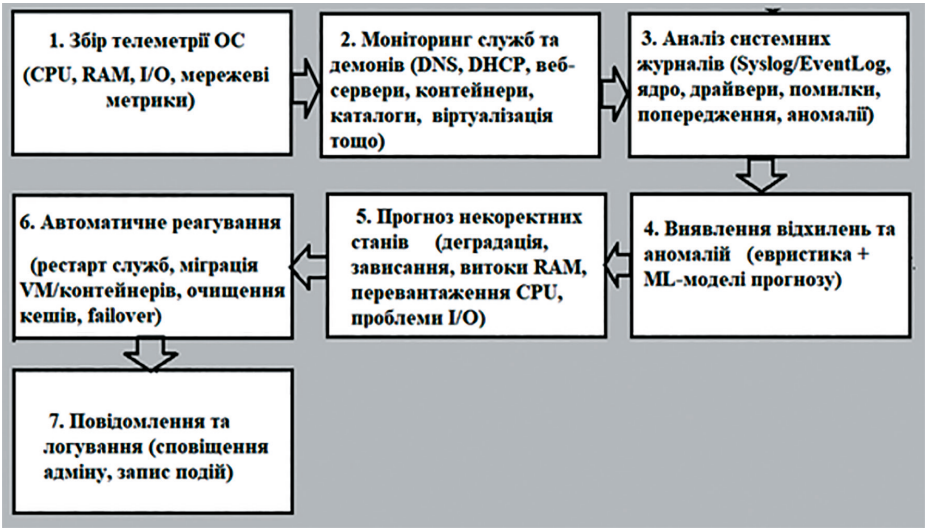

Рис. 1. Структурна схема роботи методу

Отже, перевагами запропонованого методу є такі:

- раннє виявлення деградації та зависань ще до настання критичного збою або втрати працездатності серверної операційної системи.
- зниження часу простою завдяки своєчасному усуненню проблем, автоматичному перезапуску служб або ізоляції некоректних процесів.
- комплексний контроль системних компонентів, що охоплює ядро, служби, мережеві підсистеми, драйвери, журнали подій і поведінкові моделі.
- можливість адаптації до різних платформ, включаючи Linux, Windows Server та Unix-подібні системи, без необхідності суттєвих змін у інфраструктурі.
- підтримка масштабованості, що дозволяє використовувати метод як у невеликих локальних серверах, так і у великих хмарних кластерах.
- сумісність із сучасними системами автоматизації та оркестрації, такими як Ansible, Kubernetes, Systemd, що спрощує інтеграцію у DevOps-процеси.
- підвищена точність діагностики, забезпечена аналізом журналів, мережевих потоків, системних викликів та ML-моделями прогнозування.
- зменшення навантаження на адміністраторів, оскільки метод дозволяє автоматизувати реакцію на ранні ознаки нестабільності.
- покращення відмовостійкості та надійності інфраструктури за рахунок превентивного виявлення проблем, що раніше проявлялися лише у критичних фазах роботи.

Висновки та пропозиції. У ході проведеного дослідження встановлено, що сучасні мережні операційні системи потребують комплексних підходів до контролю працездатності та раннього виявлення нестабільних режимів, оскільки традиційні механізми моніторингу не завжди здатні своєчасно ідентифікувати приховані форми деградації, часткові зависання служб або накопичення системних відхилень. Запропонована модель контролю стану та метод раннього виявлення аномалій забезпечують підвищення надійності серверних інфраструктур, скорочення часу простою та покращення відмовостійкості шляхом автоматизованого аналізу поведінкових показників, кореляції системних журналів та прогнозування критичних станів.

Комплексне поєднання моніторингу ядра, служб, мережевих підсистем і модулів безпеки із використанням адаптивних алгоритмів та ме-

ханізмів автоматичного реагування дозволяє суттєво зменшити навантаження на адміністраторів і забезпечити стабільну роботу ОС у різних інфраструктурних середовищах — від локальних серверів до розподілених хмарних платформ. Отримані результати підтверджують ефективність запропонованого підходу для Linux-, Windows- та Unix-подібних операційних систем.

Пропозиції:

- застосовувати комбіновану багаторівневу систему контролю, що інтегрує моніторинг телеметрії, аналіз системних журналів, перевірку ключових служб та поведінкові моделі прогнозування.
- використовувати адаптивні механізми раннього виявлення аномалій, що поєднують евристичний аналіз, статистичні методи та ML-моделі.
- інтегрувати автоматичні self-healing механізми, такі як перезапуск служб, ізоляція проблемних процесів, динамічна міграція контейнерів або віртуальних машин.
- впроваджувати стандартизовані підходи до логування, моніторингу та діагностики, що спрощують порівняння ефективності методів між різними ОС.
- використовувати сучасні інструменти автоматизації (Ansible, PowerShell DSC, Kubernetes Operators) для уніфікації реакцій на нестабільні режими та мінімізації людського фактору.
- розробити набір тестів продуктивності та fault-injection сценаріїв для оцінки стійкості ОС до різних типів відмов та аномалій.

ЛІТЕРАТУРА/ REFERENCES

1. Silberschatz, A., Galvin, P., & Gagne, G. (2022). Operating System Concepts. 10th ed. Wiley.

2. Tanenbaum, A., & Bos, H. (2021). Modern Operating Systems. 4th ed. Pearson.

3. Love, R. (2019). Linux Kernel Development. Addison-Wesley.

4. Bovet, D., & Cesati, M. (2020). Understanding the Linux Kernel. 3rd ed. O'Reilly.

5. Nemeth, E., Snyder, G., Hein, T., et al. (2018). UNIX and Linux System Administration Handbook. 5th ed. Pearson.

6. Minasi, M., Lowe, D. (2020). Mastering Windows Server 2019. Wiley.

7. Krzyzanowski, P. (2023). Systems Programming and Operating Systems. Rutgers University Press.

8. Hwang, K., & Dongarra, J. (2022). Distributed and Cloud Computing. Morgan Kaufmann.

9. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2021). Kubernetes: The Definitive Guide. O'Reilly.

10. Morabito, R. (2020). Virtualization and containerization of application infrastructures: A performance comparison. *Journal of Cloud Computing*, 9(1). DOI: 10.1186/s13677-020-00179-3.

11. Ahmed, E., Saeed, S., & Mahmoud, Q. (2023). Performance optimization techniques for Linux-based server systems. *Future Internet*, 15(4). DOI: 10.3390/fi15040114.

12. Zaitsev Ievgen, Bondarenko Oleh, Golubenko Oleksandr, Antonenko Artem, Savchenko Andrii. Securing Applied Information Systems With SAST Integration Into the Gulp Pipeline. In: Applied Information Systems and Technologies in the Digital Society : proceedings of the 9th International Scientific and Practical Conference "Applied Information Systems and Technologies in the Digital Society" AISTDS'2025, Kyiv, Ukraine, October 1, 2025 / ed. Vitaliy Snytyuk [et al.].

14. Khan, R., & Ghani, M. (2021). Performance evaluation of network subsystems in modern operating systems. *Computer Networks*, 190. DOI: 10.1016/j.comnet.2021.107962.

15. Gupta, S. (2022). Automated monitoring and self-healing mechanisms for enterprise OS environments. International Journal of Computer Applications, 183(25).

16. Soni, A. (2020). Windows Server 2022 Administration Fundamentals. Packt Publishing.

17. Thakar, U., & Patil, S. (2021). Optimization methods for high-performance computing operating systems. *Procedia Computer Science*, 190, 573–582.

СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 16.10.2025