

УДК 004

Ю.Молчанов, магістр гр. КІ-24Мз,*Центральноукраїнський національний технічний університет*

ДОСЛІДЖЕННЯ ТА ПРИНЦИПИ ПОБУДОВИ СИСТЕМИ SOFTWARE DEFINED STORAGE

У статті розроблено програмне забезпечення, яке призначено для системи Software Defined Storage. Метою розробки є дослідження та принципи побудови системи Software Defined Storage. Об'єктом дослідження є процес Software Defined Storage. Предметом дослідження є методи Software Defined Storage. Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи Software Defined Storage. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Software Defined Storage

Постановка проблеми. Дані не чекають на бюджетні цикли. Вони зростають саме тоді, коли ваша система зберігання вирішує нагадати вам, хто насправді володіє вашим часом безвідмовної роботи. NAS та SAN все ще мають своє застосування, але їх масштабування схоже на плату за захист постачальникам обладнання – дороге, негнучке та розраховане на вчорашні робочі навантаження.

У 2025 році, коли конвеєри штучного інтелекту, аналітичні завдання та дані користувачів множаться швидше, ніж ваші сповіщення моніторингу, вам потрібне сховище, яке гнучко адаптується до потреб користувачів. Саме тут на допомогу приходять програмно-визначені сховища (Software Defined Storage, SDS): жодних оновлень навантажувачів, жодної прив'язки до постачальника, лише програмний рівень, який дозволяє масштабуватися в горизонтальному напрямку.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-30] було виявлено певні прогалини у забезпеченні системи Software Defined Storage.

Мета й завдання дослідження. Метою роботи є дослідження та принципи побудови системи Software Defined Storage.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем Software Defined Storage.
- Дослідження системи Software Defined Storage.
- Програмна реалізація системи Software Defined Storage.

Об'єктом дослідження є процес Software Defined Storage.

Предметом дослідження є методи Software Defined Storage.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. По суті, SDS – це програмний рівень, який відокремлює фізичні ресурси зберігання від базових пристроїв зберігання даних. Замість того, щоб прив'язувати дані безпосередньо до власного обладнання, SDS використовує віртуалізацію сховища для створення гнучкої та економічно ефективною інфраструктури зберігання даних.

На відміну від традиційних систем зберігання даних, таких як NAS та SAN, де зміни ємності або продуктивності часто вимагають заміни всіх фізичних ресурсів зберігання, система SDS робить масштабування майже безперешкодним. Ви можете додавати більше

дисків, вузлів або навіть хмарну ємність без переписування програм або зміни робочих процесів. Однак віртуалізація має недоліки, тому деякі організації неохоче впроваджують технологію SDS.

Коротше кажучи, програмно-визначені рішення для зберігання даних дозволяють керувати складними операціями зберігання даних з єдиного інтерфейсу, підвищуючи гнучкість та знижуючи витрати. Хіба це не та перевага, яка робить SDS кращим підходом до зберігання даних у 2025 році?

Які переваги та недоліки сучасної системи SDS?

Впровадження програмно-визначених сховищ різко зросло з однієї причини: це вирішує проблеми, які не дають системним адміністраторам та DevOps спати вночі. Вигоди переважають головний біль, особливо якщо ви перестали платити данину завищеним постачальникам обладнання.

Ось що насправді дає вам SDS:

1. Гнучкість у виборі обладнання

За допомогою SDS ви можете запускати сховище даних на виділених серверах або пристроях, які легше оновити або замінити. Це зменшує залежність від масивів певних постачальників і робить фізичні ресурси сховища більш адаптивними.

2. Масштабованість у всіх напрямках

Завдяки віртуалізації сховища, SDS масштабується як вертикально (додаючи більше потужності до існуючих вузлів), так і горизонтально (додаючи більше вузлів або дисків), що робить керування ресурсами сховища приємним процесом.

3. Економічно ефективність

Оскільки SDS абстрагує програмний рівень від фізичних ресурсів зберігання, вам не потрібно інвестувати в дорогі традиційні рішення для зберігання даних. Натомість ви можете перепрофілювати стандартне обладнання та розширювати його лише за потреби.

4. Централізоване управління сховищами

Єдина панель інструментів на базі програмного забезпечення для керування та автоматизації дозволяє адміністраторам керувати реплікацією, резервним копіюванням та моніторингом.

5. Висока доступність та аварійне відновлення

SDS дозволяє реплікувати дані між кількома вузлами або навіть географічними регіонами. У разі збою обладнання або форс-мажору відновлення відбувається швидше та надійніше.

6. Підтримка багатотипного зберігання даних

Добре розроблене програмно-визначене рішення для зберігання даних може поєднувати блочне, файлове та об'єктне сховище в одній уніфікованій системі, оптимізуючи інфраструктуру зберігання даних у різних відділах.

7. Виділений сервер

Виділений хостинг для тих, кому потрібна більша потужність, контроль та справжня стабільність.

Але користувачі технології програмно-визначеного сховища вказують на деякі недоліки:

Оскільки SDS повністю залежить від програмного рівня, продуктивність та надійність залежать від правильного налаштування та моніторингу. Неправильна конфігурація або недостатній моніторинг призводять до погіршення продуктивності.

Розгортання та обслуговування системи SDS вимагає навчання з віртуалізації та автоматизації сховищ. Брак досвіду може уповільнити впровадження.

Абстрагуючи сховище від апаратного забезпечення, SDS вводить нові міркування безпеки. Неправильно налаштоване програмне забезпечення для керування та автоматизації або неконтрольовані пристрої зберігання даних можуть стати векторами атаки.

Як керувати та оптимізувати програмно-визначену систему зберігання даних?

Розгортання програмно-визначеного сховища даних – це не фінішна пряма, а саме початок справжньої роботи. Ставлення до нього як до рішення за принципом «налаштував і забув» – це найшвидший спосіб виявити, що ваш рівень сховища даних перетворився на найслабшу ланку вашої інфраструктури.

Перша проблема – це планування ємності сховища. Занадто висока оцінка – і ви витрачаєте бюджет на обладнання, яке простоює. Занадто низька оцінка – і ви будете боротися з вузькими місцями у найневідповідніший момент. Баланс досягається шляхом моніторингу реальних робочих навантажень, прогнозування зростання за допомогою чітких показників та масштабування навмисно, а не реактивно.

Програмне забезпечення для управління та автоматизації є основою SDS, але воно надійне настільки, наскільки надійне його налаштування. Панелі інструментів та сповіщення повинні зменшувати кількість людських помилок, а не замінювати оперативну обізнаність. Відстеження IOPS, затримки та пропускну здатності в режимі реального часу – це спосіб виявити проблеми, перш ніж вони переростуть у збої.

Стратегії резервного копіювання та відновлення також змінюються в рамках SDS. Знімки та реплікація між вузлами або регіонами значно спрощують відновлення після збоїв, але лише за умови дотримання та тестування політик. Занадто багато команд вважають, що реплікація відбувається тому, що вона була налаштована один раз; на практиці кластер без моніторингу може швидко вийти з синхронізації. Практики безпеки для захисту даних також повинні залишатися суворими.

Зрештою, **підтримка постачальників або спільноти** – це тиха, але критично важлива частина оптимізації SDS. Стек, який сьогодні виглядає надійним, може стати крихким, якщо оновлення перестануть надходити або спільнота проекту втратить імпульс. Перевірка стану екосистеми так само важлива, як і перевірка стану власного кластера.

Програмно-визначене сховище зазвичай буває двох видів: конвергентна інфраструктура та гіперконвергентна інфраструктура.

Конвергентна інфраструктура (CI):

- Структура: Обчислення, мережа та сховище даних – це окремі апаратні блоки, об'єднані між собою спеціальною інтерфейсною структурою.
- Продуктивність: Висока пропускну здатність і передбачувана затримка – добре, якщо ваші робочі навантаження пов'язані з великим обсягом операцій вводу/виводу.
- Відповідність варіантам використання: застарілі програми, транзакційні бази даних або середовища, де спеціалізація обладнання все ще має значення.
- Компроміси: масштабування незграбне, залежність від постачальника залишається, а гнучкість обмежена.

Гіперконвергентна інфраструктура (HCI):

- Структура: Обчислення, мережа та сховище даних об'єднані в один рівень, керований програмним забезпеченням.
- Масштабованість: додавання вузлів, масштабування. Просто. Без модернізації навантажувача.
- Відповідність варіантам використання: хмарні стеки, оркестрація контейнерів, конвеєри з інтенсивним використанням DevOps та робочі навантаження штучного інтелекту/машинного навчання.
- Компроміси: значною мірою залежить від експертизи програмного забезпечення – неправильно налаштуйте його, і ви швидко підвищите свій кластер до аномалії класу Keter.

Що обрати у 2025 році?

- Оберіть неперервну інтеграцію (CI), якщо ви використовуєте застарілі системи або вам потрібні гарантії продуктивності без будь-яких пошкоджень.
- Оберіть HCI, якщо бажаєте використовувати його за замовчуванням для сучасних програмно-визначених середовищ зберігання даних. Якщо відповідність вимогам або

застарілі обмеження не змушують вас використовувати CI, HCI позбавить вас головного болю.

Коли варто використовувати програмно-визначену систему зберігання даних

Не кожній організації потрібне програмно-визначене сховище з першого дня. Перевірте ці випадки, щоб визначитися.

– SDS спрощує налаштування та керування віртуальними машинами, робочими столами та програмами в гібридних або багатохмарних системах.

– Якщо ви аналізуєте поведінку клієнтів, ринкові тенденції або бізнес-рушії, SDS забезпечує єдиний доступ до великих обсягів сховища даних. Його централізований програмний рівень робить аналітику швидшою та легшою для автоматизації.

– Реплікація на кількох пристроях зберігання даних та в різних регіонах забезпечує швидке відновлення під час збоїв, що робить SDS кращим вибором для організацій, де простої дорівнюють фінансовим втратам.

– Деякі підприємства використовують програмно-визначені рішення для зберігання даних на периферії для локальної обробки даних, зберігаючи при цьому синхронізацію з централізованим обладнанням для зберігання даних. Це забезпечує продуктивність та стійкість навіть у розподілених середовищах.

Як захистити програмно-визначену систему зберігання даних?

Безпека – це те, де SDS або витримує тиск, або стає дуже дорогим звітом про інциденти. Оскільки SDS переносить контроль на рівень програмного забезпечення, неправильна конфігурація не просто можлива – вона неминуча без дисципліни.

1. Обмежте права root/адміністратора до найменшої можливої групи. Кожна дія має бути зареєстрована та відстежена, а не закопана на сервері журналів, який ніхто не перевіряє до завершення збою.

2. Дані в стані спокою, під час передачі, між вузлами – шифруйте все. Будь-що менше – це відкрите запрошення до витоку даних. Базовий рівень: TLS для трафіку, AES для зберігання.

3. Автоматизуйте сканування на вразливості та не покладайтеся на електронні листи постачальників, щоб дізнатися про випадки зараження.

4. Перевіряйте кожен запит, ізолюйте робочі навантаження та сегментуйте ресурси сховища. Якщо один вузол скомпрометовано, радіус вибуху слід вимірювати в гігабайтах, а не петабайтах.

5. Відкритий код SDS забезпечує прозорість, але також робить вразливості публічними. Відстежуйте частоту виправлень спільноти.

Яка різниця між SDS, NAS та SAN для зберігання даних?

Коли люди оцінюють сховище даних, вони завжди використовують одні й ті ж три аббревіатури. Ось їх простий розбив:

Програмно-визначене сховище (SDS)

SDS абстрагує сховище на програмно-керований рівень на стандартному обладнанні. Підтримує блоки, файли та об'єкти в одній системі.

– Сильні сторони: Висока масштабованість, незалежність від апаратного забезпечення, автоматизація. Ідеально підходить для гібридної хмари, конвеєрів штучного інтелекту/машинного навчання або робочих процесів DevOps.

– Слабкі сторони: Складніший у розгортанні, вимагає фактичного досвіду та додає поверхню для атаки у разі неправильного налаштування.

– Перевірка реальності: якщо ви хочете гнучкості та втомилися від прив'язки до постачальника, то саме в цьому напрямку рухається галузь – це не «новітній розвиток», це стандартний процес.

Мережеве сховище даних (NAS)

Мережеве сховище даних (NAS) – це архітектура сховища даних на основі файлів, яка дозволяє користувачам отримувати доступ до даних через мережу, зазвичай за допомогою протоколів NFS або SMB.

- Сильні сторони: Просто, дешево, працює одразу після розпакування.
- Слабкі сторони: залежність від одного обладнання для зберігання даних, незграбне масштабування та обмежена продуктивність.
- Перевірка реальності: Чудово підходить для невеликої команди або лабораторного середовища. Завантажте його на виробниче навантаження, і ви зрозумієте, чому системні адміністратори ненавидять фразу «просто поставте його на NAS».

Мережа зберігання даних (SAN)

SAN, або мережа зберігання даних, – це блочна архітектура сховища, яка з'єднує сервери зі сховищем через виділену мережу за допомогою протоколів Fibre Channel або iSCSI.

- Сильні сторони: Низька затримка, висока пропускна здатність, надійна робота з критично важливими для продуктивності корпоративними робочими навантаженнями.
- Слабкі сторони: Власницька розробка, дорога, масштабування – це джерело доходу постачальника, а не ваш вибір.
- Перевірка реальності: SAN – це «безпечний варіант», який досі просувають фінансові та традиційні IT-фахівці. Він працює – доки вам не знадобиться масштабування або оплата рахунків.

Який вам слід обрати?

SDS: найкраще підходить для організацій, що переходять на гібридну хмару, робочі навантаження зі штучним інтелектом/машинним навчанням або автоматизовані середовища зберігання даних.

NAS: підходить для невеликих команд, яким потрібен простий обмін файлами з мінімальними накладними витратами.

SAN: все ще цінна для критично важливих для продуктивності робочих навантажень, але менш адаптивна, ніж сучасні програмно-визначені рішення для зберігання даних.

Програмно-визначене сховище даних (SDS) – це базовий варіант для тих, хто хоче масштабовану інфраструктуру без необхідності платити постачальникам обладнання за цю перевагу. Абстрагуючи програмний рівень від фізичних пристроїв, SDS надає вам гнучкість для зростання, відновлення та адаптації до ваших робочих навантажень.

Впровадження SDS означає планування потужності, моніторинг IOPS/затримки, забезпечення належних політик резервного копіювання та реплікації, виконання регулярних оновлень та перевірку справності стеку. Натомість ви отримуєте гнучкість обладнання, передбачувану масштабованість та зменшену залежність від пристроїв різних постачальників.

Коротко кажучи:

- Що робить SDS: він абстрагує сховище за допомогою програмного рівня, усуваючи залежність від конкретного обладнання та знижуючи сукупну вартість володіння (TCO) завдяки стандартним серверам та еластичному зростанню.
- Що змінюється на практиці: стало легше масштабувати горизонтально та вертикально, швидше відновлюватися після збоїв та стандартизувати управління в гібридних та багатохмарних середовищах.
- Хто отримує найбільшу вигоду: команди, чиї робочі навантаження не зростають за календарним графіком – конвеєри штучного інтелекту/машинного навчання, аналітика, веб-проекти з високим трафіком та ландшафти DevOps.

NAS та SAN все ще мають свої ніші, але якщо ви створюєте системи, які повинні витримувати реальний трафік та реальні збої, SDS – це практичний вибір.



Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів Software Defined Storage. Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем Software Defined Storage.
- Досліджена система Software Defined Storage.
- На основі отриманих результатів досліджень створена програмна реалізація системи Software Defined Storage.

Розроблені алгоритми дозволяють успішно вирішувати завдання Software Defined Storage. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.
2. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». Проблеми інформатизації та управління, № 2(70). 2022. С. 28-37.
3. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98.
4. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
5. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
6. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.
7. Smirnov O., Kuznetsov A., Kiiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

8. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.
9. Smirnov O., Kuznetsov A., Kiiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
10. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.
11. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 125-136.
12. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 366-379.
13. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings Volume 2608*, 2020, Pages 633-645.
14. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.
15. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
16. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019. P.517-522.
17. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyzy, M., «QoE optimization technique for media delivery in 5G networks». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.
18. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.
19. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
20. Smirnov, O., Kuznetsov, A., Kiiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.
21. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.
22. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353*, *CEUR Workshop Proceedings 2019*, Pages 618-629.
23. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.
24. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95
25. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.
26. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.
27. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. *Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф.* 2020. – 294 с.
28. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у *Кібербезпека та інформаційні технології: монографія*. – Х.: ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
29. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного

- трафіку на основі ланцюга Маркова». Центральнотрафіку український науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
30. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.
 31. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
 32. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139
 33. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральнотрафіку український науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.