

УДК 004

В.Маламуж, магістр гр. КН-24М,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРИНЦИПИ ПОБУДОВИ СИСТЕМИ УПРАВЛІННЯ У РОЗПОДІЛЕНИХ CLOUD-СИСТЕМАХ GENI

У статті розроблено програмне забезпечення, яке призначено для системи управління у розподілених Cloud-системах GENI. Метою розробки є дослідження та принципи побудови системи управління у розподілених Cloud-системах GENI. Об'єктом дослідження є процес управління у розподілених Cloud-системах GENI. Предметом дослідження є методи управління у розподілених Cloud-системах GENI. Методи дослідження базуються на методах хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи управління у розподілених Cloud-системах GENI. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

методи управління, розподілені Cloud-системи, GENI

Постановка проблеми. GENI використовується для досліджень у сфері майбутніх інтернет-архітектур, програмно-визначених мереж, нових протокольних наборів, які можуть базуватися на IP, а можуть і не базуватися, бездротових мереж 4G та хмарних обчислень. GENI добре підходить для експериментів, які потребують:

– Масштабна експериментальна інфраструктура. GENI надає вам доступ до сотень широко розподілених ресурсів, включаючи обчислювальні ресурси, такі як віртуальні машини та «голі машини», а також мережеві ресурси, такі як з'єднання, програмовані комутатори та бездротові базові станції 4G.

– Не-IP-з'єднання між ресурсами. GENI дозволяє налаштовувати з'єднання 2-го рівня між комп'ютерними ресурсами та запускати власні протоколи 3-го рівня та вище.

– Глибокі можливості програмування. За допомогою GENI ви можете програмувати не лише кінцеві хости вашої експериментальної мережі, але й комутатори в ядрі вашої мережі. Це дозволяє експериментувати з новими протоколами мережевого рівня або з новими алгоритмами IP-маршрутизації.

– Інструменти та засоби вимірювання. Системи приладів та вимірювань GENI дозволяють вам інструментально аналізувати ваші експерименти, візуалізувати вимірювання та архівувати їх.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-30] було виявлено певні прогалини у забезпеченні системи управління у розподілених cloud-системах GENI.

Мета й завдання дослідження. Метою роботи є дослідження та принципи побудови системи управління у розподілених Cloud-системах GENI.

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

- Огляд існуючих систем управління у розподілених Cloud-системах GENI.
 - Дослідження системи управління у розподілених Cloud-системах GENI.
 - Програмна реалізація системи управління у розподілених Cloud-системах GENI.
- Об'єктом дослідження є процес управління у розподілених Cloud-системах GENI. Предметом дослідження є методи управління у розподілених Cloud-системах GENI.

Методи дослідження базуються на методах хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Проблеми управління розподіленими системами

Керування розподіленими системами створює кілька проблем через їхню притаманну складність та розподілену природу. Керування розподіленими системами створює кілька проблем через їхню притаманну складність та розподілену природу.

– Складність:

○ Впровадження розподілених систем вимагає великої кількості самодостатніх компонентів, які є взаємопов'язаними та мають власну конфігурацію, залежності та протоколи зв'язку.

○ В цифрову епоху, коли правильне виконання цих частин та ефективне управління ними може бути болісним завданням.

– Консистенція:

○ З іншого боку, найскладнішим питанням є досягнення узгодженості даних між розподіленими вузлами, особливо для обробки одночасних оновлень та збоїв.

○ Це вимагає вдосконалених методів координації та синхронізації.

– Безпека:

○ Гарантія того, що розподілені системи не будуть зламані, а неавторизовані особи не отримають до них доступу за допомогою надійних інструментів безпеки, є важливим заходом і потребує шифрування, автентифікації та контролю доступу для всіх зацікавлених сторін.

Централізоване проти децентралізованого управління

1. Централізоване управління

У централізованому управлінні всі повноваження щодо контролю та прийняття рішень зосереджені в межах однієї сутності або центрального органу. Ця сутність зазвичай має повноваження приймати рішення, розподіляти ресурси та забезпечувати дотримання політик у всій системі.

– Повноваження щодо прийняття рішень належать центральному органу влади або керівному органу.

– Політики, правила та процедури встановлюються та застосовуються централізовано.

– Розподіл ресурсів, координація та контроль здійснюються з однієї точки.

– Комунікація та потік інформації зазвичай спрямовуються через центральний орган влади.

2. Децентралізоване управління

У децентралізованому управлінні повноваження щодо прийняття рішень, контроль та координація розподілені між кількома об'єктами або вузлами в системі. Кожен об'єкт або вузол працює автономно та має право самостійно приймати рішення в межах своєї сфери відповідальності.

– Повноваження щодо прийняття рішень розподілені між кількома сутностями або вузлами.

– Сутності або вузли мають автономію приймати рішення та діяти самостійно.

– Координація та комунікація відбуваються безпосередньо між сутностями або вузлами без централізованого контролю.

Моніторинг розподілених систем

Спостереження за розподіленими системами є ключовою функцією; воно має забезпечувати відстеження відхилень, оптимізацію роботи та усунення помилок. Ключові показники для моніторингу включають:

– **Пропускна здатність:** Вимірює час відгуку окремо як показник загальних можливостей системи упакувати та поміщати транзакції в чергу за раз.

– **Затримка:** З іншого боку, вона позначає період часу, необхідний для виконання запиту, що може бути ознакою того, наскільки швидко працює система.

– **Використання ресурсів:** Відстежує використання процесора, а також пам'яті, сховища та пропускної здатності мережі. Таким чином, немає можливості затримки, а ресурси розподіляються ефективно.

– **Коефіцієнти помилок:** Він також досліджує, як часто виникають помилки та збої, пропонуючи вам уявлення про регулярність та надійність систем.

Реєстрація та трасування в розподілених системах

Ведення журналу та трасування є важливими компонентами моніторингу та налагодження розподілених систем, допомагаючи розробникам та адміністраторам розуміти поведінку системи, діагностувати проблеми та оптимізувати продуктивність.

1. Вхід у розподілених системах

Ведення журналу включає запис подій, повідомлень та інформації про системні операції, помилки та дії на постійне сховище (наприклад, файли, бази даних або системи керування журналами). Ведення журналу забезпечує історичний запис поведінки системи, допомагаючи розробникам та адміністраторам відстежувати потік виконання, виявляти помилки або аномалії та усувати неполадки.

– Повідомлення журналу зазвичай містять позначки часу, рівні серйозності (наприклад, INFO, DEBUG, WARN, ERROR), контекстну інформацію (наприклад, назву компонента, ідентифікатор запиту) та описові деталі (наприклад, трасування стека помилок, параметри запиту).

– Розробники використовують ведення журналу для моніторингу стану програм, відстеження потоку програм, налагодження проблем, аудиту дій користувачів та дотримання нормативних вимог.

2. Трасування в розподілених системах

Трасування включає фіксацію та кореляцію розподілених транзакцій або запитів, коли вони поширюються через різні компоненти чи служби в розподіленій системі. Трасування допомагає розробникам та адміністраторам зрозуміти наскрізний потік запитів між розподіленими компонентами, виявити вузькі місця в продуктивності та проаналізувати затримку та залежності.

– Трасування складаються з серії корельованих інтервалів, кожен з яких представляє певну операцію або дію в межах компонента чи сервісу. Інтервали містять метадані, такі як назви операцій, позначки часу початку та завершення, а також контекстну інформацію (наприклад, назву сервісу, ідентифікатор запиту).

– Розробники використовують трасування для візуалізації потоків запитів, вимірювання показників продуктивності на рівні сервісів (наприклад, часу відгуку, пропускної здатності), аналізу залежностей між сервісами та оптимізації продуктивності розподіленої системи.

Управління конфігурацією в розподілених системах

Управління конфігурацією охоплює процеси, інструменти та методи для визначення, розгортання, оновлення та моніторингу параметрів конфігурації та ресурсів розподілених систем. Деякі з цілей управління конфігурацією включають:

– **Забезпечення узгодженості:** управління конфігурацією гарантує, що всі вузли або компоненти в розподіленій системі мають узгоджені налаштування конфігурації, запобігаючи дрейфу конфігурації та невідповідностям.

– **Покращення масштабованості:** автоматизуючи завдання управління конфігурацією, розподілені системи можуть масштабуватися ефективніше, що дозволяє швидко розгортати та надавати нові вузли або ресурси.

– **Підвищення надійності:** Підтримуючи стандартизовану конфігурацію та впроваджуючи найкращі практики, управління конфігурацією допомагає підвищити надійність і стабільність розподілених систем.

– Спрощення управління змінами: управління конфігурацією дозволяє систематично відстежувати, версіювати та аудит змін конфігурації, що спрощує керування та відкат змін за потреби.

Розподіл ресурсів у розподілених системах

Розподіл ресурсів передбачає визначення того, як розподілити доступні ресурси між конкуруючими завданнями або роботами в розподіленій системі для оптимізації продуктивності, використання та справедливості.

– Оптимізація продуктивності: розподіл ресурсів таким чином, щоб максимізувати пропускну здатність системи, мінімізувати час відгуку та відповідати вимогам якості обслуговування (QoS).

– Забезпечення справедливості: розподіл ресурсів справедливо між конкуруючими завданнями або користувачами, щоб запобігти їх дефіциту та сприяти рівному доступу до ресурсів.

Деякі з підходів до розподілу ресурсів у розподілених системах включають:

– Статичне розподілення:

○ Попередньо розподіляйте ресурси між завданнями або роботами на основі попередньо визначених політик, пріоритетів або квот.

○ Цей підхід підходить для передбачуваних робочих навантажень з фіксованими вимогами до ресурсів.

– Динамічне розподілення:

○ Динамічно налаштовуйте розподіл ресурсів на основі характеристик робочого навантаження, системних умов та показників продуктивності.

○ Такі методи, як балансування навантаження, автоматичне масштабування та адаптивне виділення ресурсів, використовуються для коригування розподілу ресурсів у режимі реального часу.

– Розподіл кількох ресурсів:

○ Розподіляючи ресурси між завданнями або роботами, враховуйте одночасно кілька ресурсів (наприклад, процесор, пам'ять і диск), враховуючи взаємозалежність та обмеження ресурсів.

Планування в розподілених системах

Планування включає визначення часу і місця виконання завдань на розподілених вузлах для досягнення цільових показників продуктивності, обмежень ресурсів та системних вимог.

– Мінімізація затримки: Плануйте завдання або роботи, щоб мінімізувати час очікування, час відгуку та затримки обробки, покращуючи швидкість реагування системи та взаємодію з користувачем.

– Максимізація пропускну здатності: Планування завдань або проектів для максимізації пропускну здатності системи та обчислювальної потужності, забезпечення ефективного використання доступних ресурсів.

Деякі з підходів до розподілу ресурсів у розподілених системах включають:

– Планування на рівні завдань:

○ Плануйте окремі завдання або роботи на основі пріоритету, термінів, залежностей та потреб у ресурсах.

○ Такі методи, як планування на основі пріоритетів, планування за термінами та планування з урахуванням залежностей, використовуються для оптимізації виконання завдань.

– Пакетне планування:

○ Плануйте групи пов'язаних завдань або проектів (наприклад, завдання пакетної обробки, завдання MapReduce), щоб оптимізувати використання ресурсів та мінімізувати час виконання завдань.

- Глобальне планування:
 - Координуйте рішення щодо планування між кількома вузлами або кластерами для оптимізації продуктивності всієї системи та розподілу ресурсів.
 - Такі методи, як глобальне балансування навантаження, розподілені алгоритми планування та централізовані планувальники, використовуються для координації рішень щодо планування в розподілених середовищах.

Виявлення та відновлення несправностей у розподілених системах

Інструмент виявлення дефектів – це механізм виявлення несправностей, який допомагає визначити стан розподілених компонентів і, таким чином, швидко виявляти збої та реагувати на них. Вбудовані механізми відновлення, які можуть обробляти реплікацію, резервування та відновлення після відмови в системі, забезпечують безперервність та стійкість обслуговування з метою мінімізації простоїв або перебоїв у роботі.

1. Підходи до виявлення несправностей

– Моніторинг на основі серцебиття: Вузли періодично надсилають повідомлення про серцебиття, щоб вказати свою доступність та стан. Вузли моніторингу виявляють несправності, аналізуючи повідомлення про серцебиття та виявляючи відхилення від очікуваних шаблонів.

– Детектори відмов: Розподілені алгоритми та протоколи використовують детектори відмов для виявлення відмов вузлів або збоїв на основі спостережуваної поведінки, таймаутів повідомлень або збоїв зв'язку.

– Виявлення аномалій: Машинне навчання та статистичні методи використовуються для виявлення аномалій або незвичайних закономірностей у системних метриках, даних про продуктивність або поведінці зв'язку, що вказує на потенційні несправності або збої.

2. Підходи до відновлення після несправностей

– Резервування та реплікація: Використовуйте методи резервування та реплікації для реплікації критично важливих даних або послуг на кількох вузлах або в центрах обробки даних. У разі збою резервні копії можуть бути активовані для підтримки доступності послуг.

– Відкат та контрольні точки: Механізми відкату та методи контрольних точок дозволяють системі повернутися до попереднього відомого справного стану до виникнення помилки. Транзакції або процеси можна відкатити до узгодженого стану, а контрольні точки можна використовувати для відновлення обробки з відомої точки.

– Реконфігурація та самовідновлення: механізми самовідновлення автоматично переконфігурують систему, перерозподіляють робоче навантаження або замінюють компоненти, що вийшли з ладу, для відновлення функціональності системи. Для автоматизації процесів відновлення використовуються такі методи, як автоматичне масштабування, динамічна реконфігурація та автоматичне перемикання на резервний ПК.

Безпека та контроль доступу в розподілених системах

Заходи безпеки, включаючи використання шифрування, автентифікації та авторизації, значною мірою допомагають гарантувати безпеку розподіленої системи від кібератак та несанкціонованого доступу. Заходи контролю доступу – це засоби забезпечення дотримання політики щодо заборони доступу до деяких конфіденційних ресурсів і даних. Це гарантує конфіденційність даних та дотримання чинних правил.

Масштабування та балансування навантаження в розподілених системах

Масштабування означає або динамічний приплив, або зменшення ресурсів, що збігається зі збільшенням або зменшенням використання цих ресурсів.

– Балансування навантаження служить для рівномірного розподілу вхідних запитів до кожного вузла розподіленої системи, що сприяє ефективному використанню ресурсів та зменшує перевантаження, тим самим забезпечуючи масштабованість, високу доступність та продуктивність у розподілених системах.

– Такі методи, як горизонтальне та вертикальне масштабування, а також автоматичне масштабування, є одними зі способів, за допомогою яких розподілені системи

масштабуються. Ці методи використовуються для розподілу робочого навантаження на основі доступності ресурсів та моделей попиту.

Ключові характеристики розподілених систем

1. Масштабованість: Розподілені системи можуть масштабуватися горизонтально (додаючи більше машин), а не вертикально (збільшуючи потужність однієї машини), що робить їх ідеальними для застосувань, які потребують швидкого зростання.

2. Паралелізм: Багато розподілених систем обробляють тисячі або навіть мільйони запитів одночасно. Це призводить до потреби в ефективних протоколах зв'язку та обробці спільних ресурсів.

3. Відмовостійкість: оскільки розподілені системи працюють на кількох машинах, вони розроблені таким чином, щоб переносити збої окремих вузлів, не впливаючи на доступність усієї системи.

4. Гетерогенність: Розподілені системи можуть складатися з різних типів машин, операційних систем або мереж, що означає, що вони повинні безперешкодно обробляти цю різноманітність.

5. Прозорість: Розподілені системи намагаються приховати складність кількох машин від кінцевого користувача, забезпечуючи єдиний інтерфейс, ніби система працює на одному комп'ютері.

Типи розподілених систем

1. Клієнт-серверні системи

У цій моделі сервер надає послуги, а клієнти споживають ці послуги, надсилаючи запити до сервера. Прикладами є веб-сервери та сервери баз даних. Приклад: Браузер (клієнт) надсилає HTTP-запит на веб-сервер, а сервер відповідає запитуваними даними (веб-сторінкою, відповіддю API тощо).

2. Однорангові системи

На відміну від клієнт-серверних систем, однорангові системи не мають централізованого сервера. Натомість кожен вузол (або одноранговий вузол) може виступати як клієнтом, так і сервером. Приклад: Системи обміну файлами, такі як BitTorrent, де кожен вузол завантажує та вивантажує файли з інших вузлів.

3. Кластерні обчислення

Кластер складається з групи комп'ютерів (часто званих вузлами), які працюють разом як єдина система для виконання великомасштабних обчислень. Вузли в кластері зазвичай розташовані в одному фізичному місці та з'єднані через високошвидкісну локальну мережу. Приклад: кластери Hadoop, що використовуються для обробки великих даних.

4. Грід-обчислення

Грід-обчислення схожі на кластерні обчислення, але в більшому масштабі. Вузли в грід-обчисленнях часто географічно розподілені по різних місцях і з'єднані через Інтернет. Приклад: SETI@home, проект з мережевих обчислень, який здійснює пошук позаземного розуму шляхом аналізу радіосигналів з космосу.

5. Хмарні обчислення

У хмарних обчисленнях ресурси (обчислювальні, сховища, мережеві) надаються як послуга через Інтернет. Хмарні системи – це розподілені системи, що забезпечують гнучкість, масштабованість та моделі ціноутворення «оплата за використання». Приклад: Amazon Web Services (AWS), Microsoft Azure та Google Cloud.

Ключові проблеми розподілених систем

1. Затримка

Коли вузли в розподіленій системі розташовані в різних географічних місцях, мережеві затримки (латентність) стають серйозною проблемою. Латентність впливає на швидкість зв'язку між вузлами, що може вплинути на продуктивність.

2. Послідовність

Узгодженість означає, що всі вузли мають однаковий вигляд даних у будь-який момент часу. У розподілених системах підтримка узгодженості може бути складною, особливо коли вузли реплікуються в різних регіонах.

– Теорема CAP: Теорема CAP стверджує, що розподілена система не може одночасно забезпечити всі три наступні гарантії:

– Узгодженість: кожне зчитування отримує останню записану інформацію або помилку.

– Доступність: На кожен запит надходить відповідь без гарантії, що вона містить найновіший запис.

– Допуск розділу: Система продовжує працювати, незважаючи на випадкову втрату повідомлень або збій частини системи.

– Розподіленим системам часто доводиться йти на компроміси між цими властивостями.

3. Відмовостійкість

Вузли в розподіленій системі можуть вийти з ладу з різних причин (збій обладнання, проблеми з мережею тощо). Система повинна мати можливість виявляти такі збої та відновлюватися після них без втрати даних або послуг.

4. Безпека

Розподілені системи стикаються з проблемами безпеки, оскільки дані передаються через мережі, які можуть бути вразливими до таких атак, як перехоплення даних, атаки типу «людина посередині» або несанкціонований доступ.

5. Координація та синхронізація

У розподілених системах координація між вузлами є критично важливою. Вузлам часто потрібно узгоджувати стан системи, обробляти одночасні операції та синхронізувати спільні ресурси.

Проектування розподіленої системи

Під час проектування розподіленої системи слід дотримуватися певних принципів та найкращих практик:

1. Реплікація: Для покращення відмовостійкості та доступності дані та послуги слід реплікувати на кількох вузлах.

2. Балансування навантаження: балансувальник навантаження повинен розподіляти вхідні запити між кількома вузлами, щоб запобігти перевантаженню будь-якого вузла.

3. Шардінг: У великих системах дані можна розділити на менші частини (шарди), які розподіляються по різних вузлах. Це покращує продуктивність, дозволяючи паралельну обробку.

4. Кешування: Дані, до яких часто звертаються, слід кешувати, щоб зменшити затримку та покращити продуктивність системи.

5. Моніторинг та спостережуваність: Впровадження надійних інструментів моніторингу забезпечує справність системи та допомагає виявляти збої на ранній стадії.

Розподілені системи є основою сучасних застосунків, що дозволяє їм масштабуватися, обробляти збої та задовольняти потреби мільйонів користувачів. Вони стали важливими для підприємств, яким потрібна висока доступність, відмовостійкість та масштабованість. Однак проектування та управління розподіленими системами пов'язане з власним набором проблем, таких як затримка, узгодженість та відмовостійкість.

Як інженери-програмісти, розуміння принципів та проблем розподілених систем може значно покращити вашу здатність проектувати надійні, масштабовані та ефективні системи. Незалежно від того, чи створюєте ви мікросервіси, хмарні платформи чи великомасштабні конвеєри обробки даних, розподілені системи відіграватимуть вирішальну роль у вашій архітектурі.

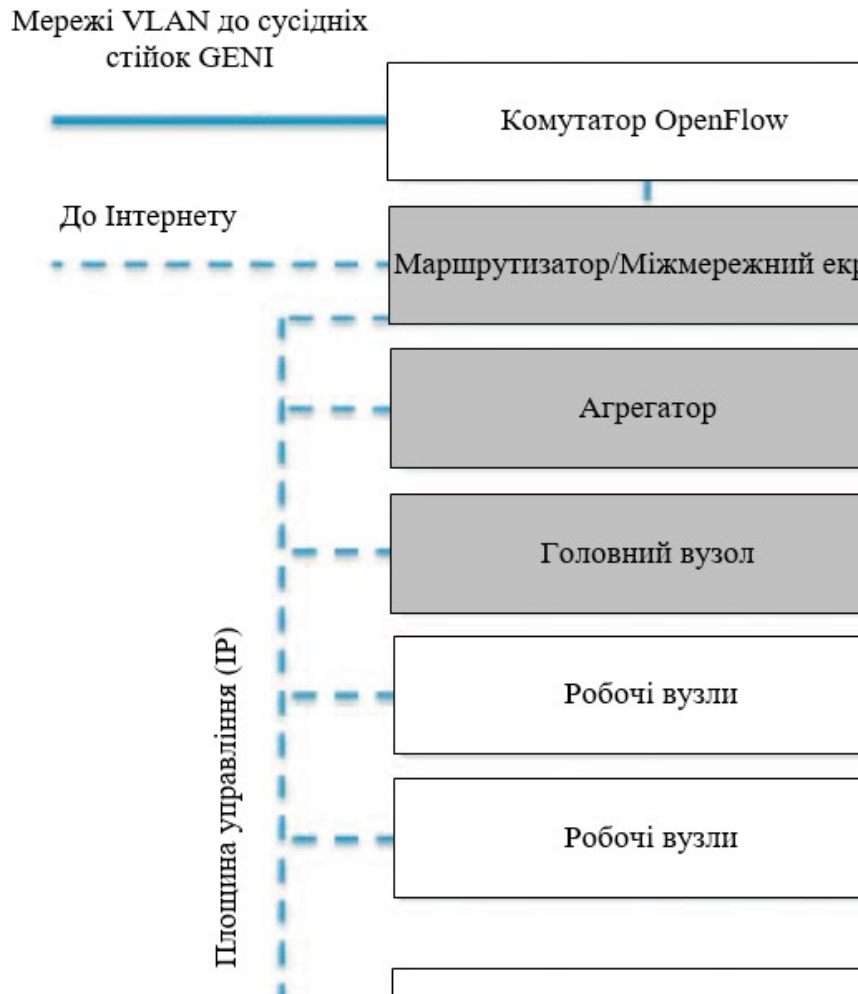


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів управління у розподілених Cloud-системах GENI. Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем управління у розподілених Cloud-системах GENI.
- Досліджена система управління у розподілених Cloud-системах GENI.
- На основі отриманих результатів досліджень створена програмна реалізація системи управління у розподілених Cloud-системах GENI.

Розроблені алгоритми дозволяють успішно вирішувати завдання управління у розподілених Cloud-системах GENI. 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. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
2. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchov, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
3. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

4. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
12. 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.
13. Smirnov O., Kuznetsov A., Kiian 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.
14. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiiienko 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, 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.
22. 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.
23. 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.

24. 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.
25. 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.
26. 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.
27. 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.
28. 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
29. 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.
30. 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.
31. 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.
32. 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.
33. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнотрапінський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.