

УДК 004.75

д-р філософії Бабарика А. О. ORCID: 0000-0001-8534-7764 (НАДПСУ)  
канд. техн. наук Басараб О. К. ORCID: 0000-0002-2852-9534 (НАДПСУ)  
Площик А. С. ORCID: 0000-0002-7034-1930 (НАДПСУ)  
Табенський С. М. ORCID: 0000-0001-8771-5671 (НАДПСУ)

## ВИБІР АРХІТЕКТУРИ ПОБУДОВИ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ ДЕРЖАВНОЇ ПРИКОРДОННОЇ СЛУЖБИ УКРАЇНИ НА ОСНОВІ ХМАРНИХ ТА ТУМАННИХ ТЕХНОЛОГІЙ

Сучасні системи відеоспостереження еволюціонували від замкнених систем телебачення до рівня складних систем, які функціонують у складі складних комплексних систем та вирішують задачі не лише фіксації подій в секторах огляду камер відеоспостереження, але і проводять аналіз отриманої відеоінформації. Збільшення обсягів інформації, що циркулює в сучасних системах, вимагає створення нових концепцій. В останні роки увагу науковців привернули технології розподіленої обробки інформації. Концепція хмарних обчислень (Cloud computing) стрімко розвивається, і основні ідеї, які були закладені при її побудові, вже не можуть вирішувати проблемні питання, які постають перед технологією. Це призвело до появи так званих постхмарних архітектур, які розширюють та доповнюють можливості хмарних обчислень. До таких архітектур відносять Mist, Edge, Fog тощо.

Метою статті є дослідження систем відеоспостереження, побудованих за концепцією хмарних і туманних обчислень, з метою подальшої реалізації у складі інформаційно-комунікаційних систем Державної прикордонної служби України. Під час проведення дослідження використано методи аналізу та узагальнення, моделювання, експеримент. Зазначений підхід дав можливість провести порівняльне дослідження систем відеоспостереження, побудованих за двома архітектурами. Отримані результати вказують на те, що технологія туманних обчислень має перевагу у зменшенні часу затримки, мінімізуючи потребу в повторних запитах до хмари, виконуючи обчислення на межі хмари. В моделі, побудованій за концепцією хмарних обчислень, застосовуються сервіси, які використовують ресурси хмари та призводять до збільшення навантаження на мережу. Разом з тим технологія туманних обчислень дозволяє розвантажити мережеве навантаження, виконуючи частину обчислень вузлами туману.

Результати експериментального дослідження показують переваги туманних обчислень для мереж, які чутливі до затримок. Проте, якщо розглянути систему відеоспостереження з основним завданням фіксації подій в секторах огляду камер відеоспостереження та можливості перегляду відеоінформації в режимі реального часу, то очевидно, що вузли туману не забезпечуватимуть тривалого зберігання відеоінформації, а затримки будуть некритичними. У випадку використання системи відеоспостереження з функціями відеоаналітики, вузли туману зможуть виконувати частину алгоритмів відеоаналітики, розвантажуючи водночас хмару. Тому актуальною задачею є дослідження ефективності побудови систем відеоспостереження з функціями відеоаналітики на основі туманної архітектури.

**Ключові слова:** хмарні обчислення, туманні обчислення, відеоспостереження, моделювання, критерії, ефективність, iFogSim.

**A. Babaryka, O. Basarab, A. Ploshchik, S. Tabenskyi Choice of architecture of video surveillance systems of State Border Guard of Ukraine based on cloud and fog technologies.**

Modern video surveillance systems have evolved from closed-circuit television systems to the level of complex systems that operate as part of complex integrated systems and solve the tasks of not only recording events in the sectors of view of video surveillance cameras but also analysing the received video information. The increase in the amount of information circulating in modern systems requires the creation of new concepts. In recent years, the attention of scientists has been drawn to distributed information processing technologies. The concept of cloud computing is developing rapidly, and the basic ideas that were laid down in its construction can no longer solve the problems faced by the technology. This has led to the emergence of so-called post-cloud architectures that extend and complement the capabilities of cloud computing. These architectures include Mist, Edge, Fog, etc.

The purpose of the article is to study video surveillance systems based on the concept of cloud and fog computing and criteria for assessing their effectiveness. The research was conducted using the methods of analysis and generalisation, modelling, and experimentation. This approach made it possible to conduct a comparative study of video surveillance systems built on two architectures. The results obtained indicate that fog computing technology has the advantage of reducing latency, minimising the need for repeated requests to the cloud by performing calculations at the cloud edge. In a model based on the cloud computing concept, services that use cloud resources lead to an increase in

network load. At the same time, fog computing technology allows you to relieve the network load by performing part of the computation by fog nodes.

The results of the experimental study show the advantages of fog computing for networks that are sensitive to delays. However, if we consider a video surveillance system with the main task of recording events in the sectors of view of CCTV cameras and the ability to view video information in real time, it is obvious that fog nodes will not provide long-term storage of video information, and delays will not be critical. In the case of a video surveillance system with video analytics functions, fog nodes will be able to perform part of the video analytics algorithms, thus unloading the cloud. Therefore, an urgent task is to study the effectiveness of building video surveillance systems with video analytics functions based on fog architecture.

**Keywords:** cloud computing, fog computing, video surveillance, modelling, criterion, efficiency, iFogSim.

**Постановка проблеми у загальному вигляді.** Сучасні виклики та загрози вимагають нових підходів щодо їх протидії. У сфері забезпечення безпеки Державного кордону України особливої ваги набирають сучасні системи відеоспостереження, що розгортаються як на протяжних ділянках кордону, так і на військових об'єктах та територіях з обмеженим доступом.

Територіальна розподіленість систем відеоспостереження, а також можливість використовувати у їхньому складі мобільні складові на основі роботизованих систем, наприклад, БПЛА, вимагає від дослідників шукати нових підходів під час побудови таких систем. При цьому, за своєю функціональністю можуть використовуватись технології Інтернету речей (*англ.* IoT, Internet of Things).

В останні роки, зважаючи на зростання обсягів даних, які потребують обробки, увагу науковців привертають технології хмарних обчислень. Проте, притаманні традиційній клієнт-серверній архітектури недоліки, що пов'язані з проблемами масштабованості та надійності, можуть призвести до перевантаження серверів. Ці недоліки певною мірою вирішують постхмарні технології, такі як Mist, Edge, Fog тощо.

Для розробки моделей систем на основі хмарних та постхмарних концепцій, застосовуються різноманітні симулятори. Для оцінки ефективності функціонування таких систем використовують ряд метрик. Проте вказані метрики не завжди можуть адекватно відображати якісні характеристики систем, оскільки не враховують особливостей їхньої побудови.

**Аналіз останніх публікацій.** У роботі [1] проведено розширене дослідження критеріїв оцінки ефективності використання систем, побудованих на основі Cloud, Mist, Edge та Fog-архітектур. Також у вищевказаному дослідженні описано симулятори, що використовуються для моделювання систем за архітектурами Cloud, Mist, Edge та Fog.

Концепція туманних обчислень (Fog) за принципами побудови достатньо часто використовується під час побудови складних систем, таких як «Розумне місто».

Для вивчення систем, побудованих за Fog-архітектурою, дослідниками було розроблено ряд програмних симуляторів, таких як iFogSim [2–4], Edge-Fog [5], FogNetSim++ [6], YAFS [7], MobFogSim [8] тощо.

Описані авторами Harshit Gupta, Amir Vahid Dastjerdi, Soumya Kanti Ghosh та Rajkumar Vuуа [4] можливості симулятора iFogSim, дозволяють провести порівняльне дослідження критеріїв оцінки ефективності функціонування систем відеоспостереження, побудованих за концепцією туманних та хмарних обчислень (Fog та Cloud).

**Метою статті** є дослідження систем відеоспостереження побудованих за концепцією хмарних і туманних обчислень, з метою подальшої реалізації у складі інформаційно-комунікаційних систем Державної прикордонної служби України.

**Виклад основного матеріалу дослідження.**

Основним завданням пристроїв в парадигмі IoT є періодичний збір даних з реального світу. [9; 10]. Пристроєм IoT може бути БПЛА з камерою відеоспостереження на борту.

Далі інформація з камери відеоспостереження може передаватися до місця відображення та/або зберігання.

У разі застосування технологій відеоаналітики, інформація може оброблятися обчислювальним блоком відразу на борту БпЛА, при цьому в мережу буде передаватися як відеопотік, так й інформація з результатами відеоаналітики. В іншому разі, при варіанті відеоаналітики на серверному обладнанні, інформація передається до місця зберігання, після чого автоматично обробляється, та результати роботи алгоритмів відеоаналітики відображаються користувачам.

Технології хмарних обчислень надають такі моделі послуг [11]:

- хмарне програмне забезпечення як послуга (Cloud Software as a Service, SaaS);
- хмарна платформа як послуга (Cloud Platform as a Service, PaaS);
- хмарна платформа як послуга (Cloud Infrastructure as a Service, IaaS).

SaaS – це готове програмне забезпечення, призначене для користувачів хмари, яке усуває потребу у використанні клієнтських програм, зберіганні даних, підтримці та адмініструванні програмних продуктів.

IaaS – відповідає за розміщення даних, конфігурацію мережі та обчислювальних потужностей. IaaS – це послуга хмарної системи, яка надається у формі платформи у віртуальному середовищі. Клієнтам не потрібно купувати сервери, центри обробки даних, мережеве обладнання або простір. Прикладом є Amazon EC2.

PaaS – відповідає за розвиток платформи з пристроями та компонентами для створення, тестування та запуску застосунків користувачів хмари [11]. PaaS є послугою хмарної системи, яка підтримує повний життєвий цикл програмного забезпечення та дозволяє користувачам розробляти хмарні додатки та сервіси. Програмісти та розробники не повинні купувати власне обладнання; замість цього вони використовують посередницьке обладнання та надсилають розроблені додатки клієнтам через інтернет. Google App Engine, Azure services platform від Google, реляційні служби баз даних Amazon (RDS) є прикладами послуги PaaS.

В останні роки дослідники розглядають також засновану на віртуалізації контейнерів модель послуг SaaS (Container as a Service). SaaS виник як хмарна модель для вирішення проблем розробки додатків у середовищі PaaS. Модель хмарного обчислення SaaS спрямована на розширення можливостей додатків, роблячи їх незалежними від специфікацій середовища PaaS. Прикладами моделі є SaaS Amazon EC2 Container Service (ECS) та Google Container Engine.

Хмарні архітектури можна класифікувати за рівнем належності та розміщенням елементів:

1. Публічна хмара;
2. Приватна хмара;
3. Хмара спільноти;
4. Гібридна хмара.

Публічна архітектура є найпоширенішою. Її використовують великі компанії, адміністративні установи й інші організації. Приватні хмари призначені для внутрішнього використання організаціями; водночас обладнання може бути фізично розташоване як у власника, так і бути під контролем третьої сторони. Громадські хмари призначені виключно для обчислень групами користувачів, які спільно виконують загальні завдання. Гібридні хмари складаються з двох або більше хмарних інфраструктур, які є окремими сутностями, але переносимість даних між ними забезпечується за допомогою стандартизованих технологій.

Централізована архітектура хмарних обчислень викликає проблеми у безпеці та затримках між пристроями та хмарою. Значною мірою нова парадигма туманних обчислень вирішує ці проблеми, доповнюючи хмарні послуги. Туманні обчислення розширюють хмарні

обчислення, включаючи кілька граничних вузлів, безпосередньо підключених до фізичних пристроїв.

Існує ряд переваг, пов'язаних із туманними обчисленнями, які забезпечують їхній успіх. Першою перевагою є зменшення мережевого трафіку, оскільки неконтрольоване збільшення мережевого трафіку може призвести до перевантажень і, як наслідок, до збільшення затримок. Туманні обчислення забезпечують платформу для фільтрації та аналізу даних, що генеруються датчиками, використовуючи ресурси периферійних пристроїв. Це значно зменшує трафік, що надсилається до хмари.

Значне зменшення затримки передачі даних є наступною важливою перевагою використання парадигми туманних обчислень, особливо для критично важливих додатків, які вимагають обробки даних у реальному часі. Одними з найкращих прикладів таких додатків є організація систем відеоспостереження з використанням мобільних роботизованих датчиків (БпЛА з камерами відеоспостереження). Саме тут на допомогу приходять туманні обчислення, які виконують обробку даних системи управління дуже близько до БпЛА, що робить можливим реагування в реальному часі.

Система на основі туманної архітектури являє собою невеликі обчислювальні вузли (Fog nodes), розташовані на граничному сегменті мережі. Головною метою технології є підтримка ресурсоємних інтернет-застосунків, які не потребують спеціальних вимог до часу затримки.

Основна різниця між туманними та хмарними обчисленнями полягає в тому, що хмара – це централізована система, а туман – розподілена децентралізована інфраструктура.

Туманні обчислення виступають посередником між обладнанням і віддаленими серверами. У туманному середовищі визначається, яка інформація буде відправлена на сервер, а яку можна редагувати локально. Так, туман можна розглядати як інтелектуальний шлюз, який розвантажує хмару, забезпечуючи більш ефективну роботу та аналіз даних.

Важливо зазначити, що туманне середовище не має окремої архітектури і не замінює хмарні обчислення, але доповнює їх, максимально наближаючись до джерела інформації [11]. Модель архітектури туманних обчислень показано на рисунку 1.

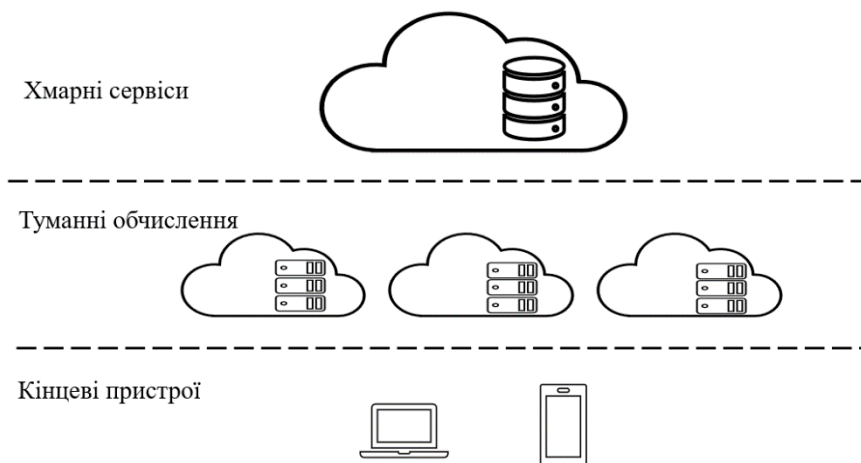


Рис. 1. Архітектура хмарних обчислень

Для дослідження різних методів управління ресурсами та планування, а також зв'язування, міграції та консолідації завдань моделей систем на основі хмарних архітектур необхідно мати інструментарій з можливістю проведення аналітики в режимі реального часу. Хоча реальне середовище Інтернету речей є бажаним тестовим майданчиком, у багатьох випадках воно є занадто дорогим та не забезпечує повторюваність і контрольованість

середовища. Для усунення цього недоліку дослідниками запропоновано симулятор під назвою iFogSim, який дозволяє моделювати управління ресурсами та політику планування додатків на периферійних і хмарних ресурсах за різних сценаріїв і умов.

Авторами Harshit Gupta, Amir Vahid Dastjerdi, Soumya Kanti Ghosh та Rajkumar Buyya в роботі [4], для реалізації функцій туманної архітектури використано базові функції симуляції подій з функціоналу існуючого симулятора CloudSim [13].

Основні класи iFogSim показано на рисунку 2 [4].

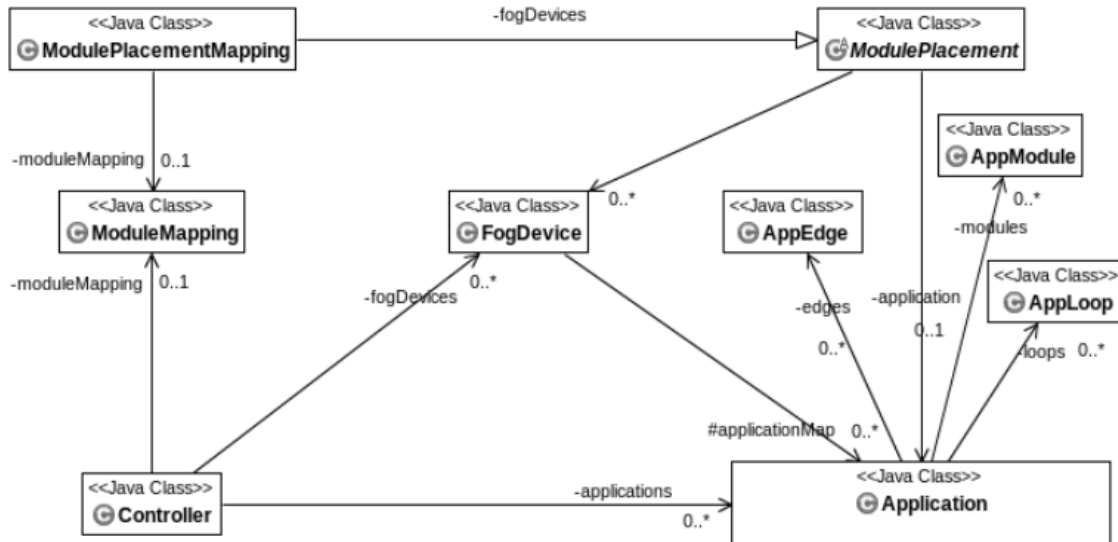


Рис. 2. Основні класи iFogSim

Реалізація iFogSim складається з імітованих сутностей та сервісів. Коротко опишемо основні поняття iFogSim, реалізовані у фізичній топології класів (рис. 3).

FogDevice – клас, який визначає апаратні характеристики Fog-пристроїв, їх підключення до інших Fog-пристроїв, датчиків та виконавчих механізмів. Основними атрибутами класу FogDevice є доступна пам'ять, процесор, розмір сховища, пропускна здатність лінії зв'язку. Методи цього класу визначають, як ресурси Fog-пристрою розподіляються між запущеними на ньому прикладними модулями, а також як модулі розгортаються і виводяться з експлуатації. Перевизначення цих методів дозволяє розробникам підключати власні політики для вищезазначених функцій.

Sensor – це сутності, які діють як датчики IoT. Клас містить атрибути, що являють собою характеристики датчика, починаючи від його підключення і закінчуючи вихідними атрибутами. Клас містить атрибут посилання на шлюзовий пристрій Fog, до якого підключено датчик, та затримку з'єднання між ними.

Tuple – це кортежі, які є основною одиницею зв'язку між сутностями в туманній архітектурі. Характеризується типом, модулями джерела та призначення. Атрибути класу визначають вимоги до обробки та довжину даних, інкапсульованих у кортежі.

Actuator – це клас, що моделює пристрій, визначаючи властивості його мережевого з'єднання. Атрибут у класі посилається на шлюз, до якого підключено Actuator, та затримку цього з'єднання. У класі визначено метод, що виконує дію при надходженні кортежу (Tuple) з модуля додатку (Application).

Application – додаток, який моделюється у вигляді орієнтованого графа, вершини якого є модулями, що виконують обробку вхідних даних, а ребра позначають залежності даних між модулями. Ці сутності реалізуються за допомогою класів AppModule, AppEdge, AppLoop.

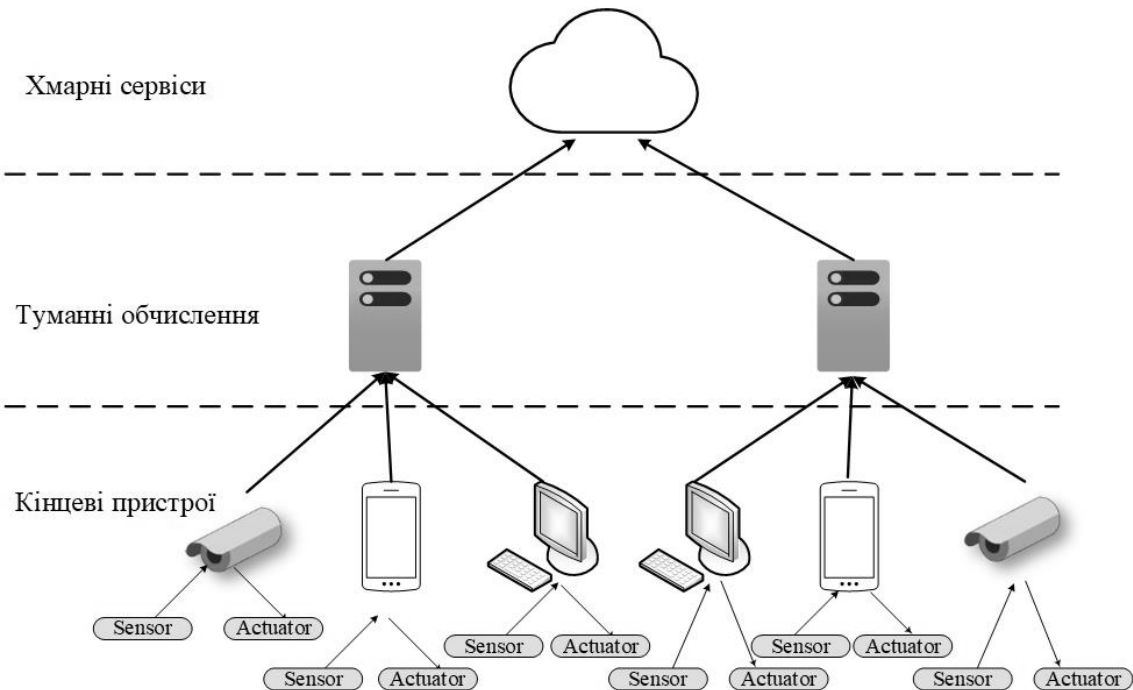


Рис. 3. Фізична топологія iFogSim

Після моделювання постає актуальне питання оцінки ефективності змодельованої системи. На основі проведеного в роботі [1] аналізу, авторами роботи виокремлено ряд критеріїв ефективності функціонування систем, побудованих за хмарними та постхмарними архітектурами (у т. ч. і за туманною). У своїй роботі автори наводять класифікацію на основі концепції MARE-K, яка була запропонована компанією ІВМ [13; 14]. Відповідно до вказаної концепції критерії категорійовано за класами: моніторинг, аналіз, планування та виконання. Враховуючи особливості туманної архітектури, визначено критерії, за якими доцільно оцінювати ефективність систем, побудованих на основі туманної архітектури:

- використання ресурсів;
- навантаження на ресурси;
- термін використання ресурсу;
- максимальне навантаження на контейнер;
- час відгуку;
- час затримки/очікування;
- резервованість контейнерів;
- енергоспоживання;
- температура;
- надійність;
- безпека.

Використання ресурсів (*англ.* Resource Utilization) – визначається як відсоток ресурсів, який споживається вхідним навантаженням. Іншими словами, цей критерій показує наскільки завантажений процесор. Ресурсами можуть бути процесор, оперативна пам'ять, пам'ять, пропускна здатність тощо.

Ресурсне навантаження (*англ.* Resource Load) – визначається як відсоток навантаження на фізичний хост, віртуальну машину, контейнер тощо. Це вимір кількості завдань, що очікують на виконання в черзі на процесор, а також тих, що виконуються у цей момент.

Термін використання ресурсу (*англ.* Resource lifetime) – це критерій для обчислення часу обслуговування віртуальною машиною у хвиликах, годинах тощо. Тобто час, протягом якого ресурс віртуальної машини був орендований обчислювальним процесом [15].

Час відгуку (*англ.* Response time) – вказує на час від надходження завдання на вхід навантаження до повернення відповіді користувачеві. Час відгуку також називають часом виконання – це час, необхідний для повного виконання робочого обчислювального процесу. Він також відомий як час завершення – термін, який необхідний для конкретних хмар або завдань для завершення роботи [14]. Цей критерій є більш ефективним для оцінки продуктивності додатків та графічно орієнтованих робочих навантажень [16].

Час очікування/затримки (*англ.* Delay/latency) – це проміжок часу, протягом якого завдання очікує на виконання. Наприклад, якщо очікуваний час виконання завдання становить 2 секунди, а фактичний – 3 секунди, то час затримки для цього завдання становить 1 секунду. Цей критерій використовується для ситуації, коли фактичний час відгуку для такої задачі вже вимірний [17].

Енергоспоживання (*англ.* Energy Consumption) – визначається як кількість енергії, що споживається ресурсом для завершення виконання робочого обчислювального процесу. Цей критерій є ефективним для оцінки робочих навантажень при тестуванні продуктивності систем [17].

Температура (*англ.* Temperature) – визначається як кількість тепла, що генерується в обчислювальному середовищі базовою інфраструктурою, такою як центр обробки даних, при виконанні завдань. Ідеться про тепловиділення в центрі обробки даних, коли завдання виконуються на базовій інфраструктурі, що здебільшого стосується хмарних інфраструктур, які мають потужні обчислювальні ресурси, а не обмежені в енергоспоживанні пристрої Інтернету речей або периферійні вузли. Детальний опис критеріїв для хмарних обчислень з урахуванням температури можна знайти в [18].

Надійність (*англ.* Reliability) – визначається як здатність системи зберігати у часі в установлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування. Надійність вузлів у віртуальному середовищі змінюється адаптивно. Вузол є надійним лише тоді, коли він може працювати в невизначених ситуаціях, таких як збій, інакше вузол не є надійним [17]. Цей критерій є найбільш ефективним для оцінки вебсайтів, сервісів зберігання та резервного копіювання, а також мобільних обчислювальних сервісів.

Змоделюємо систему відеоспостереження з використанням iFogSim за фізичною топологією на основі концепції Fog (рис. 4) та Cloud (рис. 5).

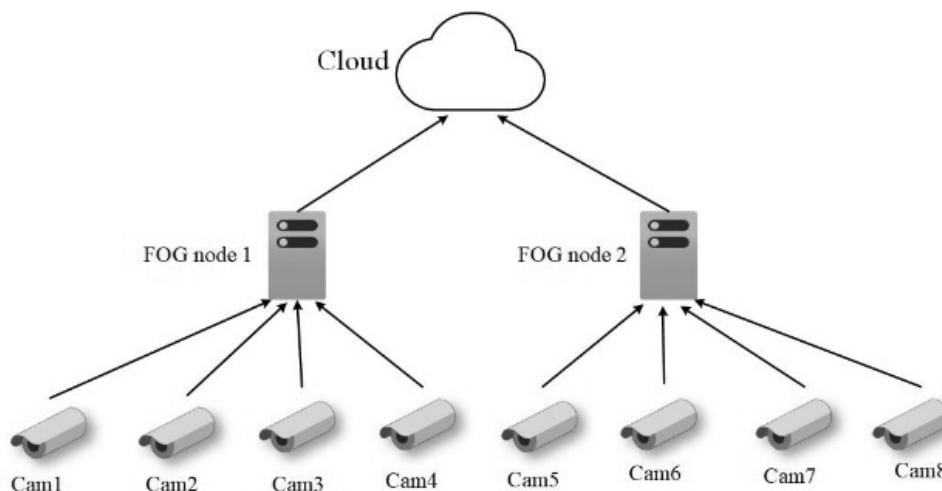


Рис. 4. Модель системи відеоспостереження за концепцією Fog

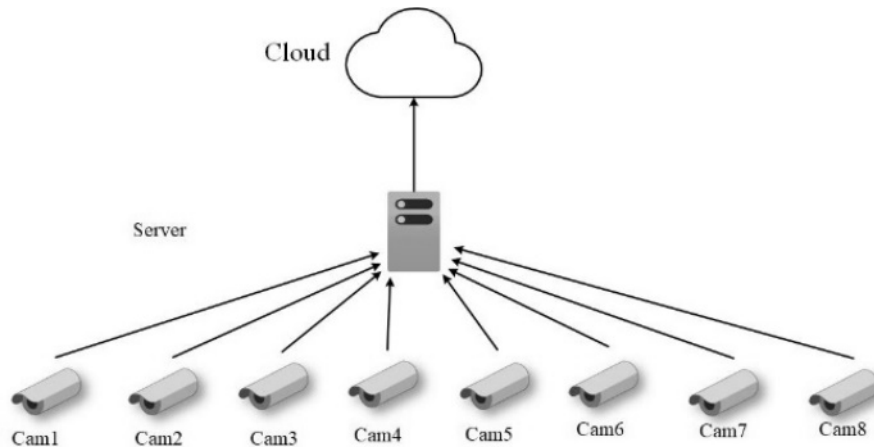


Рис. 5. Модель системи відеоспостереження за концепцією Cloud

Характеристики структурних елементів для моделі Cloud показано в таблиці 1.

Таблиця 1

Параметр	Cloud	Server	Cam X
CPU length (MIPS)	50000	4000	1000
RAM(MB)	40000	8000	1000
Downlink Bandwidth	10000	10000	
Uplink Bandwidth	100	10000	10000

Характеристики структурних елементів для моделі Fog показано в таблиці 2.

Таблиця 2

Параметр	Cloud	Fog node X	Cam X
CPU length (MIPS)	50000	4000	1000
RAM(MB)	40000	8000	1000
Downlink Bandwidth	10000	10000	
Uplink Bandwidth	100	10000	10000

Використавши при моделюванні параметри, визначені в таблиці 1 та таблиці 2, скомпілюємо створений Java-файл, використовуючи різні змінні `static int numOfCamerasPerArea`: 4, 8, 16, 32, 64.

В результаті проведеного експерименту отримано дані, на основі яких сформовано таблицю 3 та таблицю 4. У вказаних таблицях результати експериментального дослідження наведено за критеріями *Latency* та *Network usage*.

Таблиця 3

Кількість камер	Latency	Network usage
4	4.27	796.0
8	5.62	3518.08
16	8.31	3925.0
32	9.92	6425.20
64	12.6	12736.0



Таблиця 4

## Результати експериментального дослідження для моделі Cloud

Кількість камер	Latency	Network usage
4	4.31	1512.0
8	6.11	3518.08
16	10.12	29125.2
32	302.54	63258.2
64	3221.45	73258.1

Критерій *Latency* характеризується часом, який потрібний для передачі даних або виконання обчислень між кінцевим пристроєм та вузлом туману Fog Node, який розташований ближче до кінцевого пристрою, ніж обчислювальні ресурси хмари. Це важливий критерій ефективності, що впливає на загальну реакцію системи. Зменшення *Latency* (1) в туманних обчисленнях допомагає покращити реакцію системи на події в реальному часі та забезпечити швидку і ефективну обробку даних:

$$Latency = \alpha + \mu + \varphi, \quad (1)$$

де  $\alpha$  – затримка обчислювального пристрою Cam під час отримання та обробки зображення;  
 $\mu$  – час завантаження зображень на вузол туману та у хмару для обробки і зберігання;  
 $\varphi$  – час, необхідний для відображення обробленої інформації на дисплеї монітору оператора системи відеоспостереження.

Критерій *Network usage* характеризує навантаження на мережу, яке виникає внаслідок збільшення трафіку.

**Висновки і перспективи подальших досліджень.**

Аналіз отриманих результатів (табл. 3, табл. 4) вказує на те, що в системах, побудованих за архітектурами Cloud та Fog, час виконання обчислювального процесу та навантаження на мережу зростає зі збільшенням кількості камер відеоспостереження. Проте, кількісні показники вказують на те, що технологія Fog має перевагу у зменшенні часу затримки, мінімізуючи потребу в повторних запитах до хмари, виконуючи обчислення на межі хмари. В моделі, побудованій за концепцією Cloud, порівняно з Fog, сервіси, які використовують ресурси хмари, призводять до збільшення навантаження на мережу. Проте технологія Fog дозволяє розвантажити мережеве середовище, виконуючи частину обчислень вузлами Fog nodes.

Результати експериментального дослідження показують переваги архітектури Fog для мереж, які чутливі до затримок. Але якщо розглянути систему відеоспостереження з основним завданням фіксації подій в секторах огляду камер відеоспостереження та можливості перегляду відеоінформації в режимі реального часу, то очевидно, що Fog nodes не забезпечуватимуть тривалого зберігання відеоінформації, а затримки будуть некритичними.

У разі використання системи відеоспостереження з функціями відеоаналітики, що є особливо актуальним у прикордонних підрозділах, вузли Fog nodes зможуть виконувати частину алгоритмів відеоаналітики, розвантажуючи при цьому хмару.

Тому, актуальною задачею є **подальші дослідження** ефективності побудови систем відеоспостереження з функціями відеоаналітики на основі туманної архітектури.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Performance evaluation metrics for cloud, Fog and edge computing: A review, taxonomy, benchmarks and standards for future research. *Internet Things* / Aslanpour Mohammad Sadegh et al. 2020. Vol. 12. P. 20–31.
2. Rajkumar B., Satish Narayana S. Modeling and Simulation of Fog and Edge Computing Environments Using iFogSim Toolkit. *Fog and Edge Computing: Principles and Paradigms*. 2019. P. 433–465.
3. iThermoFog: IoT-Fog based Automatic Thermal Profile Creation for Cloud Data Centers using Artificial Intelligence Techniques. *Internet Technology Letters* / Tuli Shreshth et al. 2020. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.198> (date of access: 20.02.2024).
4. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments Gupta. *Software: Practice and Experience* / Harshit Gupta, Amir Vahid Dastjerdi, Soumya Kanti Ghosh and Rajkumar Buyya. 2016. Vol. 47. P. 1275–1296.
5. Mohan N., Kangasharju J. Edge-Fog cloud: A distributed cloud for Internet of Things computations. *Cloudification of the Internet of Things (CIoT)*. Paris, France. 2016. P. 1–6.
6. FogNetSim++: A Toolkit for Modeling and Simulation of Distributed Fog Environment. *IEEE Access* / Qayyum T. et al. 2018. Vol. 6. P. 63570–63583.
7. Lera I., Guerrero C., Juiz C. YAFS: A simulator for IoT scenarios in fog computing. *IEEE Access*. 2019. Vol. 7. P. 91745–91758.
8. Tuli S., Mahmud R., Buyya R. Fogbus: A blockchain-based lightweight framework for edge and fog computing. *J. Syst. Softw.* 2019. Vol. 154. P. 22–36.
9. Evans D. The internet of things: How the next evolution of the internet is changing everything. 2011. 11 p. URL: [https://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf) (date of access: 20.02.2024).
10. Mattern F., Floerkemeier C. From the Internet of Computers to the Internet of Things. *From Active Data Management to Event-Based Systems and More*. 2010. P. 242–259.
11. Заковоротний О., Орлова Т. Порівняльний аналіз хмарних та туманних середовищ інтернету речей. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2023. Т. 3 (73). С. 152–154.
12. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* / Calheiros R. et al. 2011. Vol. 41 (1). P. 23–50.
13. LARPA: A learning automata-based resource provisioning approach for massively multiplayer online games in cloud environments. *International Journal of Communication Systems* / Aslanpour M. S. et al. 2019. URL: <https://onlinelibrary.wiley.com/doi/10.1002/dac.4090> (date of access: 20.02.2024).
14. CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing* / Gill Sukhpal Singh et al. 2018. P. 1–39.
15. Calheiros R. N., Ranjan R., Buyya R. Virtual machine provisioning based on analytical performance and QoS in cloud computing environments. *Parallel processing (ICPP)*. 2011. P. 295–304.
16. Gill S. S., Garraghan P., Buyya R. ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices. *Journal of Systems and Software*. 2019. Vol. 154. P. 125–138.
17. Madni S. H. H., Latiff M. S. A., Coulibaly Y. Recent advancements in resource allocation techniques for cloud computing environment: a systematic review. *Cluster Computing*. 2017. Vol. 20. No. 3. P. 2489–2533.
18. ThermoSim: Deep learning-based framework for modeling and simulation of thermal-aware resource management for cloud computing environments. *Journal of Systems and Software* / Gill S. S. et al. 2020.